Table of Contents:

Lecture 1 9
Introduction 9
1.1 Charles Babbage (1791-1871) 9
1.2 The Analytical Engine........ 9
1.3 Ada, Countess of Lovelace(1815-52) 9
1.4 Course Contents & Structure 10
Lecture 2 13
Evolution of Computing 13
2.1 Turing Machine – 1936........ 13
2.2 The “Turing test” 13
2.3 Vacuum Tube – 1904:........ 13
2.4 ABC – 1939 14
2.5 Harvard Mark 1 – 1943:........ 14
2.6 ENIAC – 1946:........ 14
2.7 Transistor – 1947........ 14
2.8 Floppy Disk – 1950........ 14
2.9 UNIVAC 1 – 1951 14
2.10 Compiler - 1952........ 15
2.11 ARPANET – 1969 15
2.12 Intel 4004 – 1971 15
2.13 Altair 8800 – 1975........ 15
2.14 Cray 1 – 1 976 16
2.15 IBM PC – 1981 16
2.16 Apple Macintosh – 1984........ 16
2.17 World Wide Web -1989 16
2.18 Quantum Computing with Molecules........ 17
Lecture 3 18
3.1 Browser 18
3.2 URL........ 18
3.3 What is a Web site?........ 18
3.4 What is Home Page of a web site? 19
3.5 Who invented the Web & Why? 19
3.6 Future of the Web: Semantic Web........ 20
3.7 Useful Web page 20
Lecture 4 21
4.1 Computer Types According to Capability........ 21
4.2 Supercomputers 21
4.3 Mainframe Computers........ 21
4.4 Servers / Minicomputers 21
4.5 Desktops........ 21
4.6 Portables........ 21
4.7 Ranking w.r.t. installed number........ 22
4.8 All computers have the following essential hardware components:........ 22
4.9 Input Devices........ 23
4.10 What is Port? 24
4.11Many Types of Ports 24
4.12 Processor........ 24
4.13 Memory/Storage 24
4.14 Classifying Memory/Storage........ 25
4.15 Output Devices........ 25
4.16 Modem........ 25

Lecture 5 27
- 5.1 PC Parts 27
- 5.2 Inside of the CPU 27
- 5.3 The Processor Module 27

Lecture 6 28
- 6.1 To develop your personal Web page 28

Lecture 7 31
- 7.1 Microprocessor 31
- 7.2 Integrated Circuits 31
- 7.3 Devices 31
- 7.4 Microprocessor system 33
- 7.5 Micro-controllers 33
- 7.6 The Main Memory Bottleneck 33
- 7.7 Cache 33
- 7.8 Microprocessors Building Blocks 34

Lecture 8 39

Binary Numbers & Logic Operations 39
- 8.1 Why binary 42
- 8.2 Boolean Logic Operations 43
- 8.3 Truth Table for the XOR Operation 45
- 8.4 STRATEGY: Divide & Conquer 45

Lecture 9 47

HTML Lists & Tables (Web Development Lecture 3) 47
- 9.1 Single Tags 47
- 9.2 Types of Lists 52
- 9.3 Ordered List Types 52
- 9.4 Useful URL 57

Lecture 10 59

Computer Software 59
- 10.1 Machine Language 59
- 10.2 Language Translators 59
- 10.3 Software Development 59
- 10.4 Major Types of SW 60
- 10.5 System SW are programs that … 60
- 10.6 Operating System 60
- 10.7 Utilities: 61
- 10.8 Language Translators 61
- 10.9 Device Drivers 61
- 10.10 Application SW 61
- 10.11 Another way of classifying SW 62
- 10.12 Who Owns Software? 62
- 10.13 Main types of SW licensees 62
- 10.14 Proprietary SW License 62
- 10.15 Freeware SW License 63
- 10.16 Open-Source SW License 63
- 10.17 Shareware SW License 63
- 10.18 Trialware 63

Lecture 11 65

Operating Systems 65
- 11.1 Why Have OSes? 65
- 11.2 Core Tasks of an OS 65
- 11.3 OS Components 66

11.4 Kernel 67
11.5Types of OS’es 67
11.6 Another Way of Classifying 67
11.7 How many different OS’es are there? 67
11.8 Comparing Popular OS’es 68
Lecture 12 69
Interactive Forms (Web Development Lecture 4) 69
12.1 Server-Side Scripts 71
12.2 Checkbox Input Element 76
12.3 Radio Button Input Element 77
12.4 Select from a (Drop Down) List 78
12.5 File Upload Input Element 79
Lecture 13 81
Application Software 81
13.1 Two Major Types of Software 81
13.2 Application Software 81
13.3 Classification According to the Mode 81
13.4 Classification According to Application Area 81
13.5 Scientific/Engineering/Graphics Apps 81
13.6 Scientific SW 82
13.7 Engineering SW 82
13.8 Graphics & Animation SW (1) 82
13.9 Business Applications 82
13.10 E-Commerce Software 82
13.11 ERP (Enterprise Resource Planning) SW 83
13.12 DSS (Decision Support Systems) SW 83
13.13 Productivity SW 83
13.14 Word Processors 83
13.15 Web Page Development SW 83
13.16 Spreadsheet SW (1) 83
13.17 Spreadsheet SW (2) 83
13.18 Presentation Development SW 83
13.19 Small-Scale Databases SW (1) 84
13.20 Small-Scale Databases SW (2) 84
13.21 Productivity SW Suites 84
13.22 Document-Centered Computing (DCC) - 1 84
13.23 Document-Centered Computing (DCC) - 2 84
13.24 Entertainment SW 84
13.25 Music & Video Players 84
13.26 Music Generation & Movie Editing SW 84
13.27 Games 84
13.28 Educational SW 85
13.29 Electronic Encyclopedias 85
13.30 On-Line Learning 85
13.31 Interactive CD’s 85
13.32 Attributes of Good Application Software 85
13.33 Most Popular Application Software Categories 85
Lecture 14 87
Word Processing 87
14.1 Word Processor 87
14.2 Types: WYSIWYG-based & Markup-based 88
14.3 Desktop Publishing (DTP) 88

14.4 Word Processors for the Web 88
14.5 Let's try to use MS Word for creating a CV 90
Lecture 15 91
More on Interactive Forms (Web Development Lecture 5) 91
15.1 Single-Line Text Input Field 91
15.2 Password Input Field 91
15.3 Hidden Input 91
15.4 Checkbox Input Element 91
15.5 Radio Button Input Element 91
15.6 File Upload Input Element 91
15.7 Reset Button Input Element 92
15.8 Submit Button Input 92
15.9 Multi-Line Text Input Area 92
15.10 Select from a (Drop Down) List 92
15.11 Client-Side Scripting is a viable alternate 94
15.12 Server-Side Scripts: Review 94
15.13 Why JavaScript? 96
Lecture 16 99
Algorithms 99
16.1 Algorithm for Decimal-to-Binary Conversion 100
16.2 Algorithm (Better Definition) 100
16.3 Why Algorithms are Useful? 101
16.4 Analysis of Algorithms 101
16.5 Al-Khwarzmi 101
16.6 Greedy Algorithm 102
16.7 Deterministic Algorithm (1) 102
16.8 Randomized Algorithm (1) 102
16.9 Randomized Algorithm (2) 102
16.10 Deterministic Algorithm (2) 102
16.11 Heuristic 102
16.12 The Brute Force Strategy (1) 103
16.13 The Brute Force Strategy (2) 103
16.14 A Selection of Algorithmic Application Areas 103
16.15 Flowchart 104
Lecture 17 106
Algorithms II 106
17.1 Algorithm Building Blocks 106
17.2 Solution in Pseudo Code 108
17.3 Tips on Writing Good Pseudo Code 108
17.4 Pros and Cons of Flowcharts (1) 117
17.5 Pros and Cons of Flowcharts (2) 117
17.6 Pros and Cons of Pseudo Code (1) 117
17.7 Pros and Cons of Pseudo Code (2) 117
Lecture 18 118
Objects, Properties, Methods (Web Development Lecture 6) 118
18.1 New Concept: Client-Side Scripts 119
18.2 Advantages of Client-Side Scripting 119
18.3 Disadvantages 119
18.4 JavaScript 119
18.5 Client-Side JavaScript 120
18.6 Properties 121
18.7 Event Handlers 128

Lecture 19 129
Programming Languages 129
19.1 Batch Programs 129
19.2 Event-Driven Programs 129
19.3 Types of Prog. Languages 130
19.4 Programming SW Development 131
19.5 Object Oriented Design 131
19.6 Structured Design 131
19.7 Object-Oriented Languages 132
Lecture 20 133
SW Development Methodology 133
Lecture 21 142
Data Types & Operators (Web Development Lecture 7) 142
21.1 JavaScript Data Types 143
21.2 Declaring Variables 144
21.3 JavaScript Operators 148
21.4 Comparison Operators 148
21.5 Logical Operators 148
21.6 Elements of JavaScript Statements 149
Lecture 22 151
Spreadsheets 151
22.1 Business Plan for a New Software Development Company 151
22.2 The Structure of A Spreadsheet 152
22.3 Goal Seek 154
Lecture 23 158
Flow Control & Loops (Web Development Lecture 8) 158
JavaScript Variables are Dynamically Typed 158
Lecture 24 166
Design Heuristics 166
24.1 Heuristic 166
24.2 System 166
24.3 System Architecture 166
24.4 Heuristics for system architecting 166
Lecture 25 170
Web Design for Usability 170
25.2 SPEED: 170
25.3 Elements of Website Design: 171
25.4 Website Navigation: 171
25.5 A Few Navigation Design Heuristics: 171
25.6 Navigation Design Heuristics (contd.): 171
25.7 Good designs assist the user in recovering from errors 173
25.8 Assisting the User Recover from Errors: 173
25.9 A few constructive recommendations 173
25.10 Making Display Elements Legible: 175
25.11 Ensuring Text is Readable: 175
25.12 Using Pictures & Illustrations: 176
25.13 Using Motion 176
Arrays (Web Development Lecture 9) 177
26.1 Arrays in JavaScript 178
26.2 Array Identifiers 180
26.3 The 'length' Property of Arrays 181
26.4 Array Methods: sort( ) 26.5 Sorts the elements in alphabetical order . 181

26.6 Array Methods: reverse( ) 26.7 Reverses the order of the elements 181
26.7 Pseudo Code 182
Lecture 27 185
Computer Networks 185
27.1 Private Networks 186
27.2 Public Networks 186
27.3 VPN: Virtual Private Network (1) 187
27.4 Network Topologies 188
27.5 Networking Protocols 190
27.6 Types of Communication Channels 191
27.7 Network Security 191
Lecture 28 193
Introduction to the Internet 193
28.1 Internet: Network of Networks 196
28.2 Internet Networking Protocols 196
Lecture 29 199
Functions & Variable Scope (Web Development Lecture 10) 199
29.1 Function 199
29.2 Advantages of Functions 200
29.3 Function Identifiers 201
29.4 Arguments of a Function 201
29.5 Event Handlers 203
29.6 Scope of Variable 204
Lecture 30 209
Internet Services 209
30.1 Internet Addressing 210
30.2 DNS: Domain Name System 210
30.3 Internet Services 210
30.3 How does an eMail system work? 213
30.4 Using Instant Messaging 215
30.5 VoIP: Voice over IP 220
Lecture 31 221
Developing Presentations 221
31.1 Presentations: 221
31.2 The Structure of A Presentation: 224
31.3 Presentation Development SW: 224
Lecture 32 226
Event Handling (Web Development Lecture 11) 226
32.1 What is Event Handling? 228
32.2 In-Line JavaScript Event Handling : 229
Lecture 33 234
Graphics & Animation 234
33.1 Computer Graphics: 235
33.2 Displaying Images: 235
33.3 Pixel Colors : 235
33.4 Color Mapping : 235
33.5 Dithering: 236
33.6 Aliasing: 236
33.7 Anti-Aliasing: 236
33.8 Graphics File Formats: 237
33.9 Vector or Object-Oriented Graphics: 237
33.10 Bit-Mapped or Raster Graphics: 237

33.11 File Formats Popular on the Web (1): 237
33.12 Image Processing: 237
33.13-D Graphics: 238
33.14 Animation: 238
Lecture 34 240
Intelligent Systems 240
34.1 (Artificial) Intelligent Systems: 241
34.2 Fuzzy Logic: 242
34.3 Robotics: 244
Lecture 35 245
Mathematical Methods (Web Development Lecture 12) 245
35.1 Problems & Solutions: 246
35.2 Mathematical Functions in JavaScript: 248
Lecture 36 251
Data Management 251
36.1 BholiBooks.com : 252
36.2 Issues in Data Management: 252
36.3 DBMS : 253
36.4 OS Independence: 254
36.5 The Trouble with Flat-File Databases: 257
Lecture 37 259
Database Software 259
37.1 RDBMS 262
37.2 Some Terminology 263
Lecture 38 264
String Manipulations (Web Development Lecture 13) 264
38.1 String Manipulation in JavaScript 267
Lecture 39 274
Cyber Crime 274
39.1 07 February 2000 275
39.2 DoS Attack: A Cyber Crime 276
39.3 More cybercrimes … 276
39.4 Viruses 277
39.5 Other Virus-Like Programs 278
Lecture 40 279
Social Implications of Computing 279
40.1 Introduction 280
40.2 Powerful Global Corporations 280
40.3 The Network Organization 280
40.4 Working from Home 281
40.5 From Mass- to Personalized-Marketing 281
40.6 The Political Process 282
Lecture 41 284
Images & Animation (Web Development Lecture 14) 284
41.1 Images in JavaScript 286
41.2 Flash Animation 293
Lecture 42 294
The Computing Profession 294
42.1 IT: Information Technology 295
42.2 Organization: A Collection of Teams 296
Lecture 43 302
The Future of Computing 302

Lecture 44 308
Programming Methodology (Web Development Lecture 15) 308
    44.1 Design Guidelines 309
    44.2 Coding Guidelines 309
    44.3 Guidelines for Developing Short Programs 310
    44.4 Design & Code Reviews 311
    44.5 Testing & Debugging 312
    44.6 Helpful Editors 314
Lecture 45 316
Review & Wrap-Up 316
    Course Objectives 323

Lecture 1
Introduction

1.1. Charles Babbage (1791-1871)

Creator of the Analytical Engine - the first general-purpose digital computer (1833)
The Analytical Engine was not built until 1943 (in the form of the Harvard Mark I)

1.2. The Analytical Engine

A programmable, mechanical, digital machine
Could carryout any calculation
Could make decisions based upon the results of the previous calculation
Components: input; memory; processor; output

1.3. Ada, Countess of Lovelace(1815-52)

Babbage: the father of computing
Ada: the mother?
Wrote a program for computing the Bernoulli's sequence on the Analytical Engine - world's 1st computer program
Ada: A programming language specifically designed by the US Dept of Defense for developing military applications was named Ada to honor her contributions towards computing

A lesson that we all can learn from Babbage's Life

Charles Babbage had huge difficulties raising money to fund his research
As a last resort, he designed a clever mathematical scheme along with Ada, the Countess of Lovelace
It was designed to increase their odds while gambling. They bet money on horse races to raise enough money to support their research experiments
Guess what happened at the end? The lost every penny that they had.
Fast
Bored
Storage
Here is a fact:

It could analyze up to 300 billion chess moves in three minutes

In 1997 Deep Blue, a supercomputer designed by IBM, beat Gary Kasparov, the World Chess Champion
That computer was exceptionally fast, did not get tired or bored. It just kept on **analyzing** the situation and kept on **searching** until it found the perfect move from its list of possible moves …

Goals for Today:

To develop an appreciation about the capabilities of computing
To find about the structure & policies of this course

CS101 Introduction to Computing

1.4 Course Contents & Structure

Course Objectives

To build an appreciation for the fundamental concepts in computing
To achieve a beginners proficiency in Web page development
To become familiar with popular PC productivity software

Week	Lecture 1	Lecture 2	Lecture 3 Web Dev	Readings		Assignment
				UC	JS	
1			░			
2			░		░	░
3			░		░	
4			░		░	░
5		▓	░	▓	░	▓
6			░		░	
7			░		░	░
8	Midterm Exam	▓	░	▓	░	
9			░			▓
10			░			░
11		▓	░		░	▓
12			░		░	░
13		▓	░	▓	░	▓
14			░		░	░
15			░		░	
	Finals Week					

Fundamental concepts

Intro to computing
Evolution of computing
Computer organization
Building a PC
Microprocessors
Binary numbers & logic
Computer software
Operating systems
Application software
Algorithms
Flowcharts
Programming languages

Development methodology
Design heuristics
Web design for usability
Computer networks
Intro to the Internet
Internet services
Graphics & animation
Intelligent systems
Data management
Cyber crime
Social implications
The computing profession
The future of computing

Web page development

Web Development

The World Wide Web
Making a Web page
Lists & tables
Interactive forms
Objective & methods
Data types & operators
Flow control & loops Arrays
Built-in functions
User-defined functions
Events handling
String manipulation
Images & graphics
Programming methodology

Productivity Applications

Word processor
Spreadsheet
Presentation software
Database

Instructor:

Altaf Khan
cs101@vu.edu.pk

Course Web Page:

http://vulms.vu.edu.pk/

Textbooks:

UC - Understanding Computers (2000 ed.)
JS - Learn JavaScript in a Weekend

Reading Assignments

Please make sure to read the assigned material for each week before the commencement of the corresponding week

Reading that material beforehand will help you greatly in absorbing with ease the matter discussed during the lecture
Check your e-mail often for announcements related to this and other VU courses

Marks
distribution …

Assignments (15%)

Almost one every week, 13 in all
No credit for late submissions
The lowest 2 assignment grades will be dropped

Midterm Exam (35%)

During the 8th week
Duration: One hour
Will cover all material covered during the first seven weeks

Final Exam (50%)

During the 16th week
Will cover the whole of the course with a slight emphasis on the material covered after the midterm exam
Duration: 2 hours

First Assignment

Send an email message to me at altaf@vu.edu.pk with the subject "Assignment 1" giving me some information (in around 50 words) about what you see yourself doing ten years from now

Go to the CS101 message board and post a message (consisting of approx. 50 words) about how we could make the contents of this course more suitable for your individual needs. The subject for this message should be "Assignment 1"

Consult the CS101 syllabus for the submission deadline

A suggestion about unfamiliar terms

We try not to use any new terms without explaining them first

However, it is not possible to do that all the time

If you encounter any unfamiliar terms during the lectures, please note them down and consult the GLOSSARY provided at the end of the "Understanding Computers" text book for their meaning

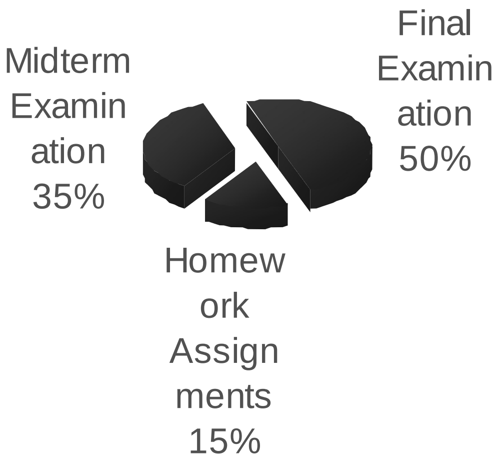
Let's summarize the things that we have covered today?

A few things about:

the very first digital computer & its inventor

the capability of modern computers

the structure and contents of CS101



In the Next Lecture …

We'll continue the story of the evolution of digital computers form the Analytical Engine onwards.

We'll discuss many of the key inventions and developments that he lead to the shape of the current field of computing.

Lecture 2

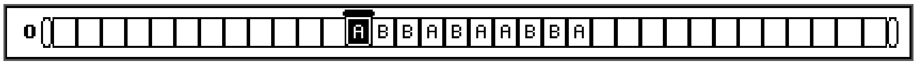
Evolution of Computing

Today's Goal

To learn about the evolution of computing
To recount the important and key events
To identify some of the milestones in computer development
Babbage's Analytical Engine - 1833
Mechanical, digital, general-purpose
Was crank-driven
Could store instructions
Could perform mathematical calculations
Had the ability to print
Could punched cards as permanent memory
Invented by Joseph-Marie Jacquard

2.1 Turing Machine – 1936

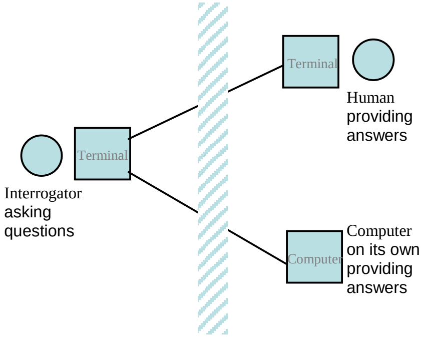
Introduced by Alan Turing in 1936, Turing machines are one of the key abstractions used in modern computability theory, the study of what computers can and cannot do. A Turing machine is a particularly simple kind of computer, one whose operations are limited to reading and writing symbols on a tape, or moving along the tape to the left or right. The tape is marked off into squares, each of which can be filled with at most one symbol. At any given point in its operation, the Turing machine can only read or write on one of these squares, the square located directly below its "read/write" head.



2.2 The "Turing test"

A test proposed to determine if a computer has the ability to think. In 1950, Alan Turing (Turing, 1950) proposed a method for determining if machines can think. This method is known as The Turing Test.

The test is conducted with two people and a machine. One person plays the role of an interrogator and is in a separate room from the machine and the other person. The interrogator only knows the person and machine as A and B. The interrogator does not know which the person is and which the machine is.
Using a teletype, the interrogator, can ask A and B any question he/she wishes. The aim of the interrogator is to determine which the person is and which the machine is. The aim of the machine is to fool the interrogator into thinking that it is a person. If the machine succeeds then we can conclude that machines can think.


Terminal
Human providing answers
Terminal
Interrogator asking questions
Computer
Computer on its own providing answers

2.3 Vacuum Tube – 1904:

A vacuum tube is just that: a glass tube surrounding a vacuum (an area from which all gases has been removed). What makes it interesting is that when electrical contacts are put on the ends, you can get a current to flow though that vacuum.
A British scientist named John A. Fleming made a vacuum tube known today as a diode. Then the diode was known as a "valve,"

2.4 ABC – 1939

The Atanasoff-Berry Computer was the world's first electronic digital computer. It was built by John Vincent Atanasoff and Clifford Berry at Iowa State University during 1937-42. It incorporated several major innovations in computing including the use of binary arithmetic, regenerative memory, parallel processing, and separation of memory and computing functions.

2.5 Harvard Mark 1 – 1943:

Howard Aiken and Grace Hopper designed the MARK series of computers at Harvard University. The MARK series of computers began with the Mark I in 1944. Imagine a giant roomful of noisy, clicking metal parts, 55 feet long and 8 feet high. The 5-ton device contained almost 760,000 separate pieces. Used by the US Navy for gunnery and ballistic calculations, the Mark I was in operation until 1959.
The computer, controlled by pre-punched paper tape, could carry out addition, subtraction, multiplication, division and reference to previous results. It had special subroutines for logarithms and trigonometric functions and used 23 decimal place numbers. Data was stored and counted mechanically using 3000 decimal storage wheels, 1400 rotary dial switches, and 500 miles of wire. Its electromagnetic relays classified the machine as a relay computer. All output was displayed on an electric typewriter. By today's standards, the Mark I was slow, requiring 3-5 seconds for a multiplication operation

2.6 ENIAC – 1946:

ENIAC I (**E**lectrical **N**umerical **I**ntegrator **A**nd **C**alculator). The U.S. military sponsored their research; they needed a calculating device for writing artillery-firing tables (the settings used for different weapons under varied conditions for target accuracy).
John Mauchly was the chief consultant and J Presper Eckert was the chief engineer. Eckert was a graduate student studying at the Moore School when he met John Mauchly in 1943. It took the team about one year to design the ENIAC and 18 months and 500,000 tax dollars to build it.
The ENIAC contained 17,468 vacuum tubes, along with 70,000 resistors and 10,000 capacitors.

2.7 Transistor – 1947

The first transistor was invented at Bell Laboratories on December 16, 1947 by William Shockley. This was perhaps the most important electronics event of the 20th century, as it later made possible the integrated circuit and microprocessor that are the basis of modern electronics. Prior to the transistor the only alternative to its current regulation and switching functions (TRANSfer resISTOR) was the vacuum tubes, which could only be miniaturized to a certain extent, and wasted a lot of energy in the form of heat.
Compared to vacuum tubes, it offered:
smaller size
better reliability
lower power consumption
lower cost

2.8 Floppy Disk – 1950

Invented at the Imperial University in Tokyo by Yoshiro Nakamats

2.9 UNIVAC 1 – 1951

UNIVAC-1. The first commercially successful electronic computer, UNIVAC I, was also the first general purpose computer - designed to handle both numeric and textual information. It was designed by J. Presper Eckert and John Mauchly. The implementation of this machine marked the real beginning of the computer era.

Remington Rand delivered the first UNIVAC machine to the U.S. Bureau of Census in 1951. This machine used magnetic tape for input.
first successful commercial computer
design was derived from the ENIAC (same developers)
first client = U.S. Bureau of the Census
$1 million
48 systems built

2.10 <u>Compiler</u> - 1952

Grace Murray Hopper an employee of Remington-Rand worked on the NUIVAC. She took up the concept of reusable software in her 1952 paper entitled "The Education of a Computer" and developed the first software that could translate symbols of higher computer languages into machine language. (Compiler)

2.11 <u>ARPANET – 1969</u>

The Advanced Research Projects Agency was formed with an emphasis towards research, and thus was not oriented only to a military product. The formation of this agency was part of the U.S. reaction to the then Soviet Union's launch of Sputnik in 1957. (ARPA draft, III-6). ARPA was assigned to research how to utilize their investment in computers via Command and Control Research (CCR). Dr. J.C.R. Licklider was chosen to head this effort.
Developed for the US DoD Advanced Research Projects Agency
60,000 computers connected for communication among research organizations and universities

2.12 <u>Intel 4004 – 1971</u>

The 4004 was the world's first universal microprocessor. In the late 1960s, many scientists had discussed the possibility of a computer on a chip, but nearly everyone felt that integrated circuit technology was not yet ready to support such a chip. Intel's Ted Hoff felt differently; he was the first person to recognize that the new silicon-gated MOS technology might make a single-chip CPU (central processing unit) possible.
Hoff and the Intel team developed such architecture with just over 2,300 transistors in an area of only 3 by 4 millimeters. With its 4-bit CPU, command register, decoder, decoding control, control monitoring of machine commands and interim register, the 4004 was one heck of a little invention. Today's 64-bit microprocessors are still based on similar designs, and the microprocessor is still the most complex mass-produced product ever with more than 5.5 million transistors performing hundreds of millions of calculations each second - numbers that are sure to be outdated fast.

2.13 Altair <u>8800</u> – 1975

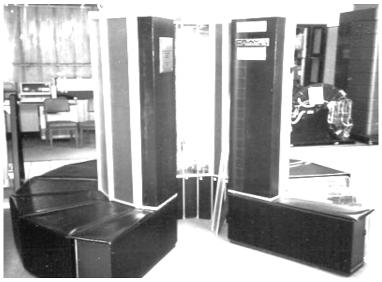
By 1975 the market for the personal computer was demanding a product that did not require an electrical engineering background and thus the first mass produced and marketed personal computer (available both as a kit or assembled) was welcomed with open arms. Developers Edward Roberts, William Yates and Jim Bybee spent 1973-1974 to develop the MITS (Micro Instruments Telemetry Systems) Altair 8800. The price was $375, contained 256 bytes of memory (not 256k),but had no keyboard, no display, and no auxiliary storage device. Later, Bill Gates and Paul Allen wrote their first product for the Altair -- a BASIC compiler (named after a planet on a *Star Trek* episode).

2.14 Cray 1 – 1 976

It looked like no other computer before, or for that matter, since. The Cray 1 was the world's first "supercomputer," a machine that leapfrogged existing technology when it was introduced in 1971.

And back then, you couldn't just order up fast processors from Intel. "There weren't any microprocessors," says Gwen Bell of The Computer Museum History Center. "These individual integrated circuits that are on the board performed different functions."

Each Cray 1, like this one at The Computer Museum History Center, took months to build. The hundreds of boards and thousands of wires had to fit just right. "It was really a hand-crafted machine," adds Bell. "You think of all these wires as a kind of mess, but each one has a precise length."



2.15 IBM PC – 1981

On August 12, 1981, IBM released their new computer, re-named the IBM PC. The "PC" stood for "personal computer" making IBM responsible for popularizing the term "PC".

The first IBM PC ran on a 4.77 MHz Intel 8088 microprocessor. The PC came equipped with 16 kilobytes of memory, expandable to 256k. The PC came with one or two 160k Floppy Disks Drives and an optional color monitor. The price tag started at $1,565, which would be nearly $4,000 today.

2.16 Apple Macintosh – 1984

Apple introduced the Macintosh to the nation on January 22, 1984. The original Macintosh had 128 kilobytes of RAM, although this first model was simply called "Macintosh" until the 512K model came out in September 1984. The Macintosh retailed for $2495. It wasn't until the Macintosh that the general population really became aware of the mouse-driven graphical user interface.

2.17 <u>World Wide Web -1989</u>

"CERN is a meeting place for physicists from all over the world, who collaborate on complex physics, engineering and information handling projects. Thus, the need for the WWW system arose "from the geographical dispersion of large collaborations, and the fast turnover of fellows, students, and visiting scientists," who had to get "up to speed on projects and leave a lasting contribution before leaving."

CERN possessed both the financial and computing resources necessary to start the project. In the original proposal, Berners-Lee outlined two phases of the project:

First, CERN would "make use of existing software and hardware as well as implementing simple browsers for the user's workstations, based on an analysis of the requirements for information access needs by experiments."

Second, they would "extend the application area by also allowing the users to add new material."

Berners-Lee expected each phase to take three months "with the full manpower complement": he was asking for four software engineers and a programmer. The proposal talked about "a simple scheme to incorporate several different servers of machine-stored information already available at CERN."

Set off in 1989, the WWW quickly gained great popularity among Internet users. For instance, at 11:22 am of April 12, 1995, the WWW server at the SEAS of the University of Pennsylvania "responded to 128 requests in one minute. Between 10:00 and 11:00

2.18 Quantum Computing with Molecules

by Neil Gershenfeld and Isaac L. Chuang

Factoring a number with 400 digits--a numerical feat needed to break some security codes--would take even the fastest supercomputer in existence billions of years. But a newly conceived type of computer, one that exploits quantum-mechanical interactions, might complete the task in a year or so, thereby defeating many of the most sophisticated encryption schemes in use. Sensitive data are safe for the time being, because no one has been able to build a practical quantum computer. But researchers have now demonstrated the feasibility of this approach. Such a computer would look nothing like the machine that sits on your desk; surprisingly, it might resemble the cup of coffee at its side.

Several research groups believe quantum computers based on the molecules in a liquid might one day overcome many of the limits facing conventional computers. Roadblocks to improving conventional computers will ultimately arise from the fundamental physical bounds to miniaturization (for example, because transistors and electrical wiring cannot be made slimmer than the width of an atom). Or they may come about for practical reasons--most likely because the facilities for fabricating still more powerful microchips will become prohibitively expensive. Yet the magic of quantum mechanics might solve both these problems.

Lecture 3

Today's Goal is to …

Become familiar with the World Wide Web
Become familiar with the Web's structure and how the Web works
Learn about its genesis, its evolution, and its future
About its impact on computing, society, commerce

3.1 Browser

A browser is an application program that provides a way to look at and interact with all the information on the World Wide Web. The word "browser" seems to have originated prior to the Web as a generic term for user interfaces that let you browse (navigate through and read) text files online. By the time the first Web browser with a graphical user interface was generally available (Mosaic, in 1993), the term seemed to apply to Web content, too. Technically, a Web browser is a client program that uses the Hypertext Transfer Protocol (HTTP) to make requests of Web servers throughout the Internet on behalf of the browser user.

3.2 URL

URL (Uniform Resource Locator, previously Universal Resource Locator) - pronounced YU-AHR-EHL or, in some quarters, UHRL - is the address of a file (resource) accessible on the Internet. The type of file or resource depends on the Internet application protocol. Using the World Wide Web's protocol, the Hypertext Transfer Protocol (HTTP), the resource can be an HTML page (like the one you're reading), an image file, or any other file supported by HTTP. The URL contains the name of the protocol required to access the resource, a domain name that identifies a specific computer on the Internet, and a pathname (hierarchical description of a file location) on the computer.

On the Web (which uses the Hypertext Transfer Protocol), an example of a URL is:

http://www.ietf.org/rfc/rfc2396.txt

Which describes a Web page to be accessed with an HTTP (Web browser) application that is located on a computer named www.ietf.org. The pathname for the specific file in that computer is /rfc/rfc2396.txt.

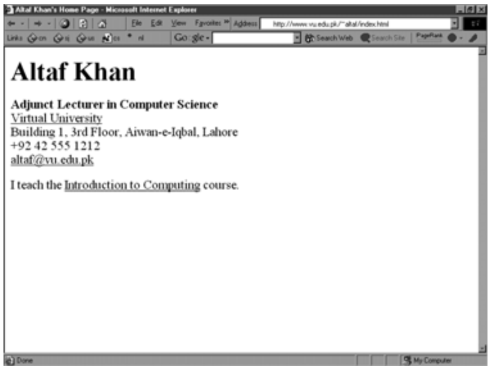
An HTTP URL can be for any Web page, not just a home page, or any individual file.

Examples:
http://dawn.com
http://www.vu.edu.pk
http://www.smeda.org.pk

3.3 What is a Web site?

A Web site is a related collection of World Wide Web (WWW) files that includes a beginning file called a home page. A company or an individual tells you how to get to their Web site by giving you the address of their home page. From the home page, you can get to all the other pages on their site. For example, the Web site for IBM has the home page address of http://www.ibm.com. IBM's home page address leads to thousands of pages but a web site can also be just of few pages.

http://www.vu.edu.pk/



Altaf Khan's Home Page - Microsoft Internet Explorer

http://www.vu.edu.pk/~altaf/index.html

Altaf Khan

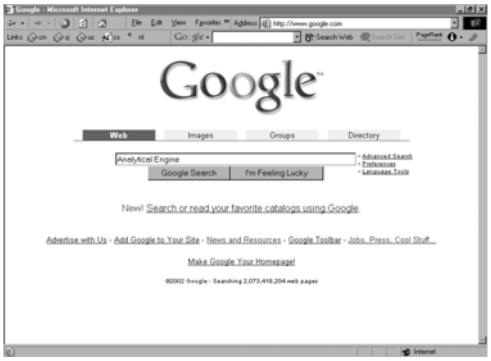
Adjunct Lecturer in Computer Science
Virtual University
Building 1, 3rd Floor, Aiwan-e-Iqbal, Lahore
+92 42 555 1212
altaf@vu.edu.pk

I teach the Introduction to Computing course.

3.4 **What is Home Page of a web site?**

1) For a Web user, the home page is the first Web page that is displayed after starting a Web browser like Netscape's Navigator or Microsoft's Internet Explorer. The browser is usually preset so that the home page is the first page of the browser manufacturer. However, you can set it to open to any Web site. For example, you can specify that "http://www.yahoo.com" or "http://whatis.com" be your home page. You can also specify that there be no home page (a blank space will be displayed) in which case you choose the first page from your bookmark list or enter a Web address.

2) For a Web site developer, a home page is the first page presented when a user selects a site or presence on the World Wide Web. The usual address for a Web site is the home page address, although you can enter the address (Uniform Resource Locator) of any page and have that page sent to you.



Google - Microsoft Internet Explorer

http://www.google.com

Analytical Engine

New! Search or read your favorite catalogs using Google.

Advertise with Us - Add Google to Your Site - News and Resources - Google Toolbar - Jobs, Press, Cool Stuff...

Make Google Your Homepage!

©2002 Google - Searching 2,073,418,204 web pages

<u>3.5 Who invented the Web & Why?</u>

"CERN is a meeting place for physicists from all over the world, who collaborate on complex physics, engineering and information handling projects. Thus, the need for the WWW system arose "from the geographical dispersion of large collaborations, and the fast turnover of fellows, students, and visiting scientists," who had to get "up to speed on projects and leave a lasting contribution before leaving."

CERN possessed both the financial and computing resources necessary to start the project. In the original proposal, Berners-Lee outlined two phases of the project:

First, CERN would "make use of existing software and hardware as well as implementing simple browsers for the user's workstations, based on an analysis of the requirements for information access needs by experiments."

Second, they would "extend the application area by also allowing the users to add new material."

Berners-Lee expected each phase to take three months "with the full manpower complement": he was asking for four software engineers and a programmer. The proposal talked about "a simple scheme to incorporate several different servers of machine-stored information already available at CERN."

Set off in 1989, the WWW quickly gained great popularity among Internet users. For instance, at 11:22 am of April 12, 1995, the WWW server at the SEAS of the University of Pennsylvania "responded to 128 requests in one minute. Between 10:00 and 11:00

3.6 Future of the Web: Semantic Web

The Semantic Web is an idea of World Wide Web inventor Tim Berners-Lee that the Web as a whole can be made more intelligent and perhaps even intuitive about how to serve a user's needs. Berners-Lee observes that although search engines index much of the Web's content, they have little ability to select the pages that a user really wants or needs. He foresees a number of ways in which developers and authors, singly or in collaborations, can use self-descriptions and other techniques so that context-understanding programs can selectively find what users want.

3.7 Useful Web page

Web page for our “Understanding Computers” text book
http://www.hbcollege.com/infosys/parker2000

What have we learnt today?

What is the World Wide Web?
How does it work?
About its expected evolution into the Semantic Web
The impact of the Web on computing, society, and commerce

Lecture 4

Today's Goal

To learn to classify computers according to their capability and targeted applications
To find out about the essential building blocks that make up a modern computer
Computer Types According to Capability

4.1 Computer Types According to Capability

4.2 Supercomputers

A supercomputer is a computer that performs at or near the currently highest operational rate for computers. A supercomputer is typically used for scientific and engineering applications that must handle very large databases or do a great amount of computation (or both). At any given time, there are usually a few well-publicized supercomputers that operate at the very latest and always incredible speeds.

Perhaps the best-known builder of supercomputers has been Cray Research, now a part of Silicon Graphics. Some supercomputers are at "supercomputer center," usually university research centers, some of which, in the United States, are interconnected on an Internet backbone (A backbone is a larger transmission line that carries data gathered from smaller lines that interconnect with it) known as vBNS or NSFNet.

At the high end of supercomputing are computers like IBM's "Blue Pacific," announced on October 29, 1998. Built in partnership with Lawrence Livermore National Laboratory in California, Blue Pacific is reported to operate at 3.9 teraflop (trillion floating point operations per second), 15,000 times faster than the average personal computer. It consists of 5,800 processors containing a total of 2.6 trillion bytes of memory and interconnected with five miles of cable.

4.3 Mainframe Computers

A very large and expensive computer capable of supporting hundreds, or even thousands, of users simultaneously. In the hierarchy that starts with a simple microprocessor (in watches, for example) at the bottom and moves to supercomputers at the top, mainframes are just below supercomputers. In some ways, mainframes are more powerful than supercomputers because they support more simultaneous programs. But supercomputers can execute a single program faster than a mainframe. The distinction between small mainframes and minicomputers is vague (not clearly expressed), depending really on how the manufacturer wants to market its machines.

4.4 Servers / Minicomputers

A midsized computer. In size and power, minicomputers lie between *workstations* and *mainframes*. In the past decade, the distinction between large minicomputers and small mainframes has blurred, however, as has the distinction between small minicomputers and workstations. But in general, a minicomputer is a multiprocessing system capable of supporting from 4 to about 200 users simultaneously.

4.5 Desktops

These are also called microcomputers. Low-end desktops are called PC's and high-end ones "Workstations". These are generally consisting of a single processor only, some times 2, along with MB's of memory, and GB's of storage. PC's are used for running productivity applications, Web surfing, messaging. Workstations are used for more demanding tasks like low-end 3-D simulations and other engineering & scientific apps. These are not as reliable and fault-tolerant as servers. Workstations cost a few thousand dollars; PC around a $1000.

4.6 Portables

Portable computer is a personal computer that is designed to be easily transported and relocated, but is larger and less convenient to transport than a notebook computer. The earliest PCs designed for easy transport were called portables. As the size and weight of most portables decreased, they became known as laptop computer and later as notebook computer. Today, larger transportable computers continue to be called *portable computers*. Most of these are special-purpose computers - for example, those for use in industrial environments where they need to be moved about frequently.

PDA (personal digital assistant) is a term for any small mobile hand-held device that provides computing and information storage and retrieval capabilities for personal or business use, often for keeping schedule calendars and address book information handy. The term handheld is a synonym. Many people use the name of one of the popular PDA products as a generic term. These include Hewlett-Packard's Palmtop and 3Com's PalmPilot.

Most PDAs have a small keyboard. Some PDAs have an electronically sensitive pad on which handwriting can be received. Apple's Newton, which has been withdrawn from the market, was the first widely-sold PDA that accepted handwriting. Typical uses include schedule and address book storage and retrieval and note-entering. However, many applications have been written for PDAs. Increasingly, PDAs are combined with telephones and paging systems.

Some PDAs offer a variation of the Microsoft Windows operating system called Windows CE. Other products have their own or another operating system.

4.7 Ranking w.r.t. installed number

PC's

PDA's

Workstations

Servers

Wearable (picture is provided)

Mainframes

Supercomputers

At the highest level, two things are required for computing

Hardware

Computer equipment such as a CPU, disk drives, CRT, or printer

Software

A computer program, which provides the instructions which enable the computer hardware to work

4.8 All computers have the following essential hardware components:

Input

The devices used to give the computer data or commands are called Input devices. Includes keyboard, mouse, scanner, etc

Processor

A processor is the logic circuitry that responds to and processes the basic instructions that drive a computer.

The term processor has generally replaced the term central processing unit (CPU). The processor in a personal computer or embedded in small devices is often called a microprocessor.

Short for microprocessor, the central processing unit in a computer. The processor is the logic of a computer and functions comparably to a human central nervous system, directing signals from one component to another and enabling everything to happen

Memory

Memory is the electronic holding place for instructions and data that your computer's microprocessor can reach quickly. When your computer is in normal operation, its memory usually contains the main parts of the operating system and some or all of the application programs and related data that are being used. Memory is often used as a shorter synonym for random access memory (RAM). This kind of memory is located on one or more microchips that are physically close to the microprocessor in your computer. Most desktop and notebook computers sold today include at least 16 megabytes of RAM, and are upgradeable to include more. The more RAM you have, the less frequently the computer has to access instructions and data from the more slowly accessed hard disk form of storage.

Memory is also called primary or main memory.

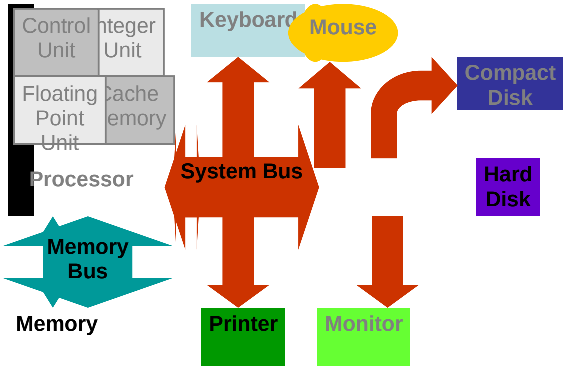
Storage

Computer storage is the holding of data in an electromagnetic form for access by a computer processor. It is also called secondary storage. In secondary storage data resides on hard disks, tapes, and other external devices.

Primary storage is much faster to access than secondary storage because of the proximity of the storage to the processor or because of the nature of the storage devices. On the other hand, secondary storage can hold much more data than primary storage.

Output

The devices to which the computer writes data are called Output devices. Often converts the data into a human readable form. Monitor and printer are output devices.



4.9 Input Devices

Mouse

A mouse is a small device that a computer user pushes across a desk surface in order to point to a place on a display screen and to select one or more actions to take from that position. The mouse first became a widely-used computer tool when Apple Computer made it a standard part of the Apple Macintosh. Today, the mouse is an integral part of the graphical user interface (GUI) of any personal computer. The mouse apparently got its name by being about the same size and color as a toy mouse.

Keyboard

On most computers, a keyboard is the primary text input device. A keyboard on a computer is almost identical to a keyboard on a typewriter. Computer keyboards will typically have extra keys, however. Some of these keys (common examples include Control, Alt, and Meta) are meant to be used in conjunction with other keys just like shift on a regular typewriter. Other keys (common examples include Insert, Delete, Home, End, Help, function keys, etc.) are meant to be used independently and often perform editing tasks.

Joystick

In computers, a joystick is a cursor control device used in computer games. The joystick, which got its name from the control stick used by a pilot to control the ailerons and elevators of an airplane, is a hand-held lever that pivots on one end and transmits its coordinates to a computer. It often has one or more push-buttons, called switches, whose position can also be read by the computer.

Digital Camera

A digital camera records and stores photographic images in digital form that can be fed to a computer as the impressions are recorded or stored in the camera for later loading into a computer or printer. Currently, Kodak, Canon, and several other companies make digital cameras.

Microphone

A device that converts sound waves into audio signals. These could be used for sound recording as well as voice chatting through internet.

Scanner

A scanner is a device that captures images from photographic prints, posters, magazine pages, and similar sources for computer editing and display. Scanners come in hand-held, feed-in, and flatbed types and for scanning black-and-white only, or color. Very high resolution scanners are used for scanning for high-resolution printing, but lower resolution scanners are adequate for capturing images for computer display. Scanners usually come with software, such as Adobe's Photoshop product, that lets you resize and otherwise modify a captured image

4.10 What is Port?

On computer and telecommunication devices, a *port* (noun) is generally a specific place for being physically connected to some other device, usually with a socket and plug of some kind. Typically, a personal computer is provided with one or more serial ports and usually one parallel port.

4.11Many Types of Ports

Parallel

An interface on a computer that supports transmission of multiple bits at the same time; almost exclusively used for connecting a printer. On IBM or compatible computers, the parallel port uses a 25-pin connector.

Serial

It is a general-purpose personal computer communications port in which 1 bit of information is transferred at a time. In the past, most digital cameras were connected to a computer's serial port in order to transfer images to the computer. Recently, however, the serial port is being replaced by the much faster USB port on digital cameras as well as computers.

SCSI

A port that's faster than the serial and parallel ports but slower and harder to configure than the newer USB port. Also know as the Small Computer System Interface.

A high-speed connection that enables devices, such as hard-disk drives and network adapters, to be attached to a computer.

USB

USB (Universal Serial Bus) is a plug-and-play hardware interface for peripherals such as the keyboard, mouse, joystick, scanner, printer and modem. USB has a maximum bandwidth of 12 Mbits/sec and up to 127 devices can be attached. With USB, a new device can be added to your computer without having to add an adapter card. It typically is located at the back of the PC

Firewire

FireWire is simply a really fast port that lets you connect computer peripherals and consumer electronics to your computer without the need to restart. It is a simple common plug-in serial connector on the back of your computer.

It has the ability to chain devices together in a number of different ways without terminators for example, simply join 2 computers with a FireWire cable for instant high-speed networking.

4.12 Processor

Pentium
Celeron
Athlon
PowerPC
StrongARM (PDA)
Crusoe (Laptops)
SPARC (Workstations)

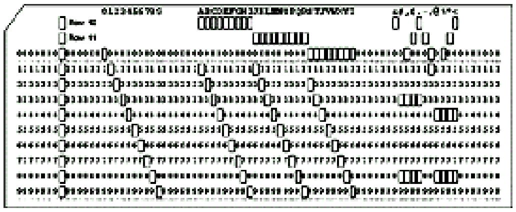
4.13 Memory/Storage

RAM

RAM (random access memory) is the place in a computer where the operating system, application programs, and data in current use are kept so that they can be quickly reached by the computer's processor. RAM is much faster to read from and write to than the other kinds of storage in a computer, the hard disk, floppy disk, and CD-ROM. However, the data in RAM stays there only as long as your computer is running. When you turn the computer off, RAM loses its data. When you turn your computer on again, your operating system and other files are once again loaded into RAM, usually from your hard disk.

Punch cards

A card on which data can be recorded in the form of punched holes.



ROM

ROM is "built-in" computer memory containing data that normally can only be read, not written to. ROM contains the programming that allows your computer to be "booted up" or regenerated each time you turn it on. Unlike a computer's random access memory (RAM), the data in ROM is not lost when the computer power is turned off.

The ROM is sustained by a small long-life battery in your computer.

Hard disk

Hard disk is a computer storage device which saves and retrieves the data when required. Its capacity is much greater than the computer memory (RAM, ROM). Data on hard disk is stored and retrieved from electromagnetically charged surface.

Today we can save huge amount of data on a single hard disk. Now hard disks can contain several billion bytes.

Floppy disk

A diskette is a random access, removable data storage medium that can be used with personal computers. The term usually refers to the magnetic medium housed in a rigid plastic cartridge measuring 3.5 inches square and about 2 millimeters thick. Also called a "3.5-inch diskette," it can store up to 1.44 megabytes (MB) of data.

Tape

In computers, tape is an external storage medium, usually both readable and writable, can store data in the form of electromagnetic charges that can be read and also erased. A tape drive is the device that positions, writes from, and reads to the tape.

CD

A compact disc [sometimes spelled *disk*] (CD) is a small, portable, round medium for electronically recording, storing, and playing back audio, video, text, and other information in digital form.

DVD

DVD (digital versatile disc) is an optical disc technology that is expected to rapidly replace the CD-ROM disc (as well as the audio compact disc) over the next few years. The digital versatile disc (DVD) holds 4.7 gigabyte of information on one of its two sides, or enough for a 133-minute movie.

4.14 Classifying Memory/Storage

Electronic (RAM, ROM), magnetic (HD, FD, Tape), optical (CD, DVD)
Volatile (RAM), non-volatile (HD)
Direct access (RAM, HD), serial access (Tape)
Read/write (HD, RAM), read-only (CD)

4.15 Output Devices

Printer
Plotter
Speakers
Monitor

4.16 Modem

Modem is output as well as input device at the same time. It receives the data (analog signal) coming through telephone line, converts them to digital signals and sends them to computer to which it is attached. It also receives the data from computer and changes it to analog signals.

What have we learnt today?

What are the various types of computers with respect to their size, capability, applications (FIVE TYPES)

The five essential components of any computer are input devices, processor, memory, storage and output devices

Lecture 5

Today's goal is quite simple …

To learn about the various components of the popular personnel computer.
How those things are put together to form a PC

5.1 PC Parts

Monitor
Keyboard
Mouse
Speaker/headphone
Microphone
CPU
Front buttons
Backside ports, fan, slots, cables

5.2 Inside of the CPU

Power supply/fan & connectors
Motherboard
Bus
Edge connectors
Ports
Video card
Modem
Network card
Sound card
ROM
RAM
Slots
DIMM's

5.3 The Processor Module

The slot on the motherboard
The housing
Fan
Heat sink
Pins (256?), Transistors (10 million?)

Lecture 6

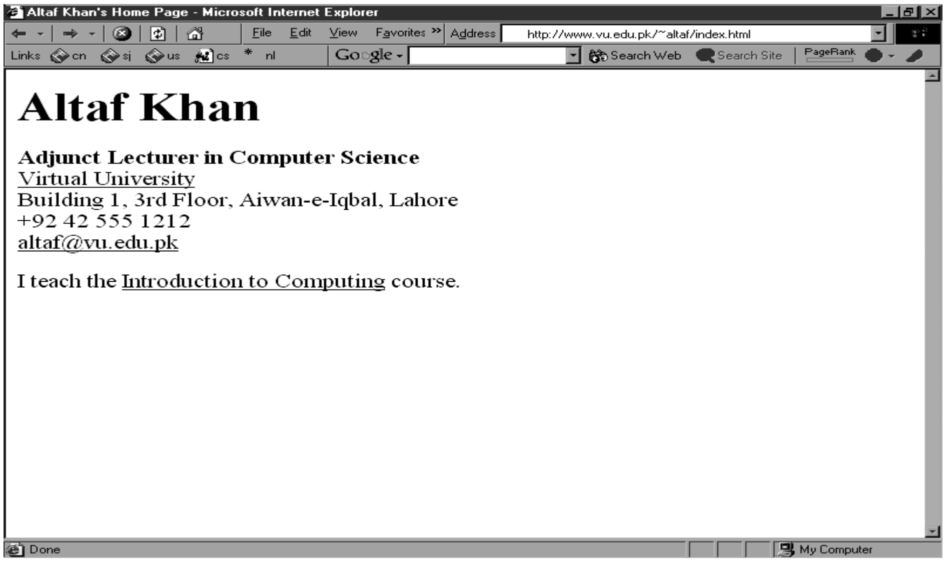
Learning Goals for Today

6.1 **To develop your personal Web page**

To upload your Web page to VU's Web server so that it becomes visible on the Internet as http://www.vu.edu.pk/~xxxxxxx/

where xxxxxxx is your user ID

http://www.vu.edu.pk/~altaf


Altaf Khan's Home Page - Microsoft Internet Explorer
File Edit View Favorites
Address http://www.vu.edu.pk/~altaf/index.html
Links cn sj us cs nl
Google Search Web Search Site PageRank
Done
My Computer

Altaf Khan

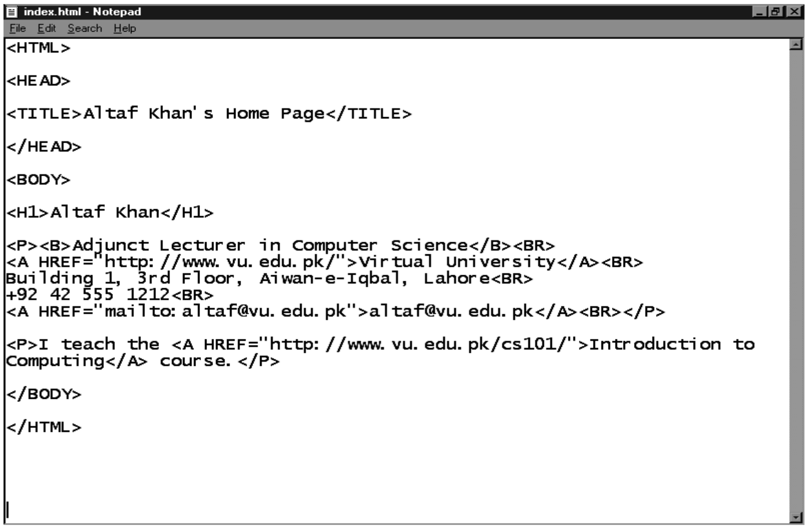
Adjunct Lecturer in Computer Science
Virtual University
Building 1, 3rd Floor, Aiwan-e-Iqbal, Lahore
+92 42 555 1212
altaf@vu.edu.pk

I teach the Introduction to Computing course.

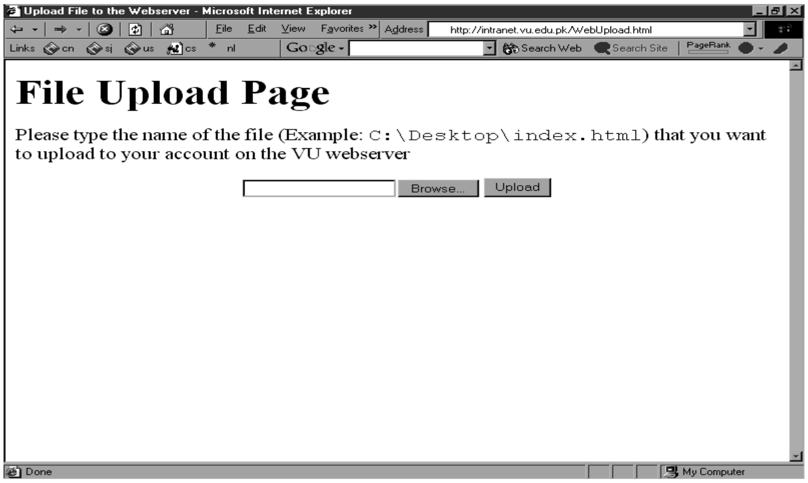
HTML

Hyper Text Markup Language

```
<HTML>
<HEAD>
<TITLE>Altaf Khan's Home Page</TITLE>
</HEAD>
<BODY>
<H1>Altaf Khan</H1>
<P><B>Adjunct Lecturer in Computer Science</B><BR>
<A HREF="http://www.vu.edu.pk/">Virtual University</A><BR>
Building 1, 3rd Floor, Aiwan-e-Iqbal, Lahore<BR>
+92 42 555 1212<BR>
<A HREF="mailto:altaf@vu.edu.pk">altaf@vu.edu.pk</A><BR></P>
<P>I teach the <A HREF="http://www.vu.edu.pk/cs101/">Introduction to Computing</A> course. </P>
</BODY>
</HTML>
```



```
<HTML>

<HEAD>

<TITLE>Altaf Khan's Home Page</TITLE>

</HEAD>

<BODY>

<H1>Altaf Khan</H1>

<P><B>Adjunct Lecturer in Computer Science</B><BR>
<A HREF="http://www.vu.edu.pk/">Virtual University</A><BR>
Building 1, 3rd Floor, Aiwan-e-Iqbal, Lahore<BR>
+92 42 555 1212<BR>
<A HREF="mailto:altaf@vu.edu.pk">altaf@vu.edu.pk</A><BR></P>

<P>I teach the <A HREF="http://www.vu.edu.pk/cs101/">Introduction to
Computing</A> course.</P>

</BODY>

</HTML>
```


Upload File to the Webserver - Microsoft Internet Explorer
File Edit View Favorites
Address http://intranet.vu.edu.pk/WebUpload.html
Links cn sj us cs nl
Google
Search Web Search Site PageRank
Browse...
Upload
Done
My Computer

File Upload Page

Please type the name of the file (Example: `C:\Desktop\index.html`) that you want to upload to your account on the VU webserver

http://www.vu.edu.pk/~altaf/index.html
http://www.vu.edu.pk/~altaf
http://www.vu.edu.pk/~xxxxxxx
where xxxxxxx is your user ID
<HTML>
…
…
</HTML>
<HEAD>
…
…
</HEAD>
<TITLE> … </TITLE>
<BODY>
…
…
</BODY>
<P> … </P>
Paragraph
<BR>
Line break

<B> … </B>
Bold text

<A HREF = *"action"* > *label* </A>
http://
example: "http://www.vu.edu.pk"
mailto:
example: "mailto:altaf@vu.edu.pk"

HTML Code

I am at the
<A HREF="http://www.vu.edu.pk">Virtual University</A>. You can send me an
e-mail by clicking
<A HREF="mailto:bhola@vu.edu.pk">here</A>.

Browser Display

I am at the Virtual University. You can send me an e-mail by clicking here.

What have we learned today?

We now know how Web pages are built using HTML
We also know how to make our personal Web pages available to *everyone* on the Internet

Useful URL's

HTML for the Conceptually Challenged
http://www.arachnoid.com/lutusp/html_tutor.html

NCSA's Beginner's Guide to HTML

http://archive.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimerAll.html

Homework Assignment

Develop your own home page. It should be accessible as http://www.vu.edu.pk/~xxxxxxx (xxxxxxxx is your user ID)

Among other things, it should contain
At least one **link** to http://www.vu.edu.pk/~altaf
Your (clickable) **email** address
A **paragraph** (50-100 words) on what you see yourself doing 10 years from now.
Consult your syllabus for the submission deadline for this assignment

Lecture 7
Goals for Today

Today we want to learn about the microprocessor, the key component, the brain, of a computer
We'll learn about the function of a microprocessor
And its various sub-systems
Bus interface unit
Data & instruction cache memory
Instruction decoder
Arithmetic-Logic unit
Floating-point unit
Control unit

7.1 Microprocessor

A microprocessor (abbreviated as **µP** or **uP**) is a computer processor on a microchip. It's sometimes called a *logic chip*. A microprocessor is designed to perform arithmetic and logic operations that make use of small number-holding areas called *registers*. Typical microprocessor operations include adding, subtracting, comparing two numbers, and fetching numbers from one area to another. These operations are the result of a set of instructions that are part of the microprocessor design. When the computer is turned on, the microprocessor is designed to get the first instruction from the basic input/output system (BIOS) that comes with the computer as part of its memory. After that, either the BIOS, or the operating system that BIOS loads into computer memory, or an application program is "driving" the microprocessor, giving it instructions to perform. The number of transistors available has a huge effect on the performance of a processor. As seen earlier, a typical instruction in a processor like an 8088 took 15 clock cycles to execute. Because of the design of the multiplier, it took approximately 80 cycles just to do one 16-bit multiplication on the 8088. With more transistors, much more powerful multipliers capable of single-cycle speeds become possible.
A microprocessor is made from miniaturized transistors and other circuit elements on a single semiconductor integrated circuit (IC) . These are made up of semiconductor and silicon.

7.2 Integrated Circuits

A **chip** is also called an **(**integrated circuit (IC) (aka *microchip* or just *chip*). It is a microelectronic semiconductor device consisting of many interconnected transistors and other components.Generally it is a small, thin piece of silicon onto which the transistors making up the microprocessor have been etched.
A chip might be as large as an inch on a side and can contain tens of millions of transistors. Simpler processors might consist of a few thousand transistors etched onto a chip just a few millimeters square. Integrated circuits can be classified into analog, digital and mixed signal (both analog and digital on the same chip). Digital integrated circuits can contain anything from one to millions of logic gates, flip-flops, multiplexers**,** etc. in a few square millimeters. The small size of these circuits allows high speed, low power dissipation, and reduced manufacturing cost compared with board-level integration.
The growth of complexity of integrated circuits follows a trend called "Moore's Law", it states that the number of transistors in an integrated circuit doubles every two years.

7.3 Devices

7.3.1 Transistors

The **transistor** is a solid state semiconductor device used for amplification and switching, and has three terminals. A small current or voltage applied to one terminal controls the current through the other two, hence the term *trans*istor; a voltage- or current-controlled resistor. It is the key component in all modern electronics. In digital circuits, transistors are used as very fast electrical switches, and arrangements of transistors can function as logic gates, RAM-type memory and other devices. In analog circuits, transistors are essentially used as amplifiers.

7.3.2 Diodes

A **diode** functions as the electronic version of a one-way valve. By restricting the direction of movement of charge carriers, it allows an electric current to flow in one direction, but blocks it in the opposite direction.

A diode's current-voltage, or I-V, characteristic can be approximated by two regions of operation. Below a certain difference in potential between the two leads, the diode can be thought of as an open (non-conductive) circuit. As the potential difference is increased, at some stage the diode will become conductive and allow current to flow, at which point it can be thought of as a connection with zero (or at least very low) resistance. In a typical semiconductor p-n diode, conventional current can flow from the p-doped side to the n-doped side, but not in the opposite direction. When the diode is reverse-biased, the charge carriers are pulled away from the center of the device, creating a depletion region. More specifically, the transfer function is logarithmic, but so sharp that it looks like a corner.

7.3.3 Resistors

A resistor is an electrical component designed to have an electrical resistance that is independent of the current flowing through it. The common type of resistor is also designed to be independent of temperature and other factors. Resistors may be fixed or variable. Variable resistors are also called **potentiometers** or **rheostats**

A few resistor types

Some resistors are long and thin, with the actual resisting material in the centre, and a conducting metal leg on each end. This is called an *axial* package.

Resistors used in computers and other devices are typically much smaller, often in *surface-mount* (Surface-mount technology) packages without leads.

Larger power resistors come in more sturdy packages designed to dissipate heat efficiently, but they are all basically the same structure. Resistors are used as part of electrical networks and incorporated into microelectronic semiconductor devices. The critical measurement of a resistor is its *resistance*, which serves as a ratio of voltage to current and is measured in ohms, an SI unit. Any physical object is a kind of resistor. Most metals are conductors, and have low resistance to the flow of electricity. The human body, a piece of plastic, or even a vacuum has a resistance that can be measured. Materials that have very high resistance are called insulators.

7.3.4 Capacitors

A capacitor (historically known as a "condenser") is a device that stores energy in an electric field, by accumulating an internal imbalance of electric charge. An ideal capacitor can store electronic energy when disconnected from its charging circuit, so it can be used like a fast battery. In AC or signal circuits it induces a phase difference of 90 degrees, current leading potential.

They are connected in parallel with the power circuits of most electronic devices and larger systems (such as factories) to shunt away and conceal current fluctuations from the primary power source to provide a "clean" power supply for signal or control circuits. The effect of such capacitors can be thought of in two different ways. One way of thinking about it is that the capacitors act as a local reserve for the DC power source, to smooth out fluctuations by charging and discharging each cycle. The other way to think about it is that the capacitor and resistance of the power supply circuitry acts as a filter and removes high frequencies, leaving only DC.

Wires

And are made of the following materials

Silicon - semiconductor
Copper - conductor
Silicon Dioxide - insulator

7.4 Microprocessor system

Microprocessors are powerful pieces of hardware, but not much useful on their own. They do not have the sense of their own. Like the human sample it needs some instructions inputs and outputs to process some task. As per instruction given to the microprocessor.

A microprocessor system is microprocessor plus all the components it requires to do a certain task.

Shortly, a microprocessor needs help of some components to make up the task to fulfill. These components are input, output, storage, and memory. All these components and microprocessor make up a microprocessor system.

Personal Computer is an example of microprocessor System. Another example is the microcontroller.

7.5 Micro-controllers

A microcontroller is a microprocessor optimised to be used to control electronic equipment. Microcontrollers represent the vast majority of all computer chips sold, over 50% are "simple" controllers, and another 20% are more specialized decipline processors. While you may have one or two general-purpose microprocessors in your house (you're using one to read this), you likely have somewhere between one and two dozen microcontrollers. They can be found in almost any electrical device, washing machines, microwave ovens, telephones etc.

A microcontroller includes CPU, memory for the program (ROM), memory for data (RAM), I/O lines to communicate with peripherals and complementary resources, all this in a closed chip. A microcontroller differs from a standalone CPU, because the first one generally is quite easy to make into a working computer, with a minimum of external support chips. The idea is that the microcontroller will be placed in the device to control, hooked up to power and any information it needs, and that's that.

7.6 The Main Memory Bottleneck

Modern super-fast microprocessors can process a huge amount of data in a short duration. They need data to be processed at the same speed. Other wise they have to sit idle and wait for the input/data, because speed of input is rather small then processing of data. They require quick access to data to maximize their performance. If they don't receive the data that they require, they literally stop and wait, this results in reduced performance and wasted power.

Current microprocessors can process an instruction in about ns (nanosecond). Time required for fetching data from main memory (RAM) is of the order of 100 ns

Solution to the Bottleneck Problem

In order to eliminate the solution it was suggested to make the main memory faster. But that evolved a problem that the 1-ns memory is extremely expensive as compared the currently popular 100-ns memory.

Finally it was decided that in addition to the relatively slow main memory, put a small amount of ultra-fast RAM right next to the microprocessor on the same chip and make sure that frequently used data and instructions resides in that ultra-fast memory

It increases the performance. It supports better over performance due to fast access to frequently used data and instructions.

7.7 Cache

A **cache** is a collection of duplicate data, where the original data is expensive to fetch or compute (usually in terms of access time) relative to the cache. Future accesses to the data can be made by accessing the cached copy rather than refetching or recomputing the original data, so that the perceived average access time is lower. Caches may mark the cached data as 'stale' when the original data is changed, but this is not always the case.

On-Chip Cache Memory (1)

That small amount of memory located on the same chip as the microprocessor is called On-Chip Cache Memory.

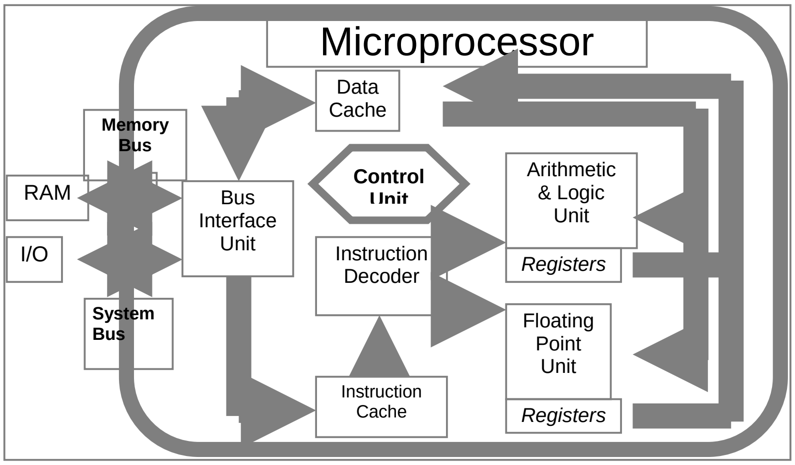
The microprocessor stores a copy of frequently used data and instructions in its cache memory. When the microprocessor desires to look at a piece of data, it checks in the cache first. If it is not there, only then the microprocessor asks for the same from the main memory

On-Chip Cache Memory (2)

L2, cache memory, which is on a separate chip from the microprocessor but faster to access than regular RAM.

It is the small size and proximity to the microprocessor makes access times short, resulting in a boost in performance. Microprocessors predict what data will be required for future calculations and it pre-fetches that data and places it in the cache so that it is available immediately when the need arises.

7.8 Microprocessors Building Blocks



Bus Interface Unit

The bus interface unit is the part of the processor that interfaces with the rest of the PC. Its name comes from the fact that it deals with moving information over the processor data bus, the primary conduit for the transfer of information to and from the CPU. The bus interface unit is responsible for responding to all signals that go to the processor, and generating all signals that go from the processor to other parts of the system.

It receives instructions & data from main memory to be processed and operations. After the operations are processed it then sends back the information (processed data) to the cache. It also receives the processed data to send it to the main memory.

Instruction Decoder

The *instruction decoder* of a processor is a combinatorial circuit sometimes in the form of a read-only memory, sometimes in the form of an ordinary combinatorial circuit. Its purpose is to translate an instruction code into the address in the micro memory where the micro code for the instruction starts.

A **decoder** is a device which is the reverse, undoing the encoding so that the original information can be retrieved. The same method used to encode is usually just reversed in order to decode.This unit receives the programming instructions and decodes them into a form that is understandable by the processing units, i.e. The ALU or FPU Then, it passes on the decoded instruction to the ALU or FPUs as desired.

Arithmetic & Logic Unit (ALU)

An arithmetic and logical unit **(ALU)** also known as **"Integer Unit"** is one of the core components of all central processing units. It is capable of calculating the results of a wide variety of common computations. The most common available operations are the integer arithmetic operations of addition, subtraction, and multiplication, the bitwise logic operations of AND, NOT, OR, and XOR, and various shift operations.

The ALU takes as inputs the data to be operated on and a code from the control unit indicating which operation to perform, and for output provides the result of the computation. In some designs it may also take as input and output a set of condition codes, which can be used to indicate cases such as carry-in or carry-out, overflow, or other statuses.

The new breed of popular microprocessors have not one but two almost identical ALU's that can do calculations simultaneously, doubling the capability

Floating-Point Unit (FPU)

A **floating point unit (FPU)** is a part of a CPU specially designed to carry out operations on floating point numbers. Typical operations are floating point arithmetic (such as addition and multiplication), but some systems may be capable of performing exponential or trigonometric calculations as well (such as square roots or cosines).

Not all CPUs have a dedicated FPU. In the absence of an FPU, the CPU may use a microcode program to emulate an FPUs function using an arithmetic and logical unit (ALU), which saves the added hardware cost of an FPU but is significantly slower.

In some computer architectures, floating point operations are handled completely separate from integer operations, with dedicated floating point registers and independent clocking schemes. Floating point addition and multiplication operations are typically pipelined, but more complicated operations, like division, may not be, and some systems may even have a dedicated floating point divider circuit.

Registers

A **register** is a device for storing data. It is a small amount of very fast computer memory used to speed the execution of computer programs by providing quick access to commonly used values. These registers are the top of the memory hierarchy, and are the fastest way for the system to manipulate data. It is common to measure registers by the number of bits it can hold, for example, an "8-bit register" or "32-bit register". Registers are now usually implemented as an array of SRAMs, but they have also been implemented using individual flip flops, high speed core memory, thin film memory, and other ways in various machines.

There are several other classes of registers:

Data registers are used to store integer numbers.

Address registers hold memory addresses and are used to access memory.

General Purpose registers can store both data and addresses.

Floating Point registers are used to store floating point numbers.

Constant registers hold read-only values (e.g zero or one).

Vector registers hold data for Single Instruction Multiple Data (SIMD) instructions.

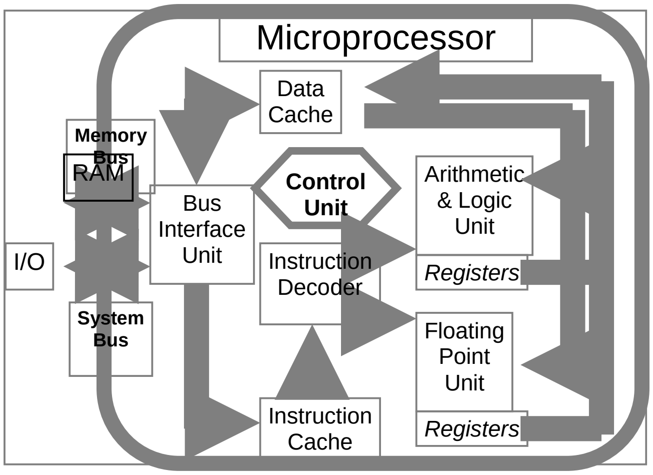
Special Purpose registers which store internal CPU data like the stack pointer or processor status words.

The ALU & FPU store intermediate and final results from their calculations in these registers. Then the processed data goes back to the data cache and then to main memory from these registers.

Control Unit

A control unit is the part of a CPU or other device that directs its operation. The outputs of the unit control the activity of the rest of the device. A control unit can be thought of as a finite state machine. It is called the brain of computer microprcessor. It manages whole process of the microprocessor. For it identifes which data is sent to the ALU or memory etc.

At one time control units for CPUs were ad-hoc logic, and they were difficult to design. Now they are often implemented as a microprogram that is stored in a control store.



That was the structure, now let's talk about the language of a microprocessor

Instruction Set

The set of machine instructions that a microprocessor recognizes and can execute – the only language microprocessor knows

An instruction set includes low-level, a single step-at-a-time instructions, such as add, subtract, multiply, and divide

Each microprocessor family has its unique instruction set

Bigger instruction-sets mean more complex chips (higher costs, reduced efficiency), but shorter programs

An **instruction set**, or instruction set architecture (ISA), is a specification detailing the commands that a computer's CPU should be able to understand and execute, or the set of all commands implemented by a particular CPU design. The term describes the aspects of a computer or microprocessor typically visible to a programmer, including the native datatypes, instructions, registers, memory architecture, interrupt and fault system, and external I/O (if any). "Instruction set architecture" is sometimes used to distinguish this set of characteristics from the Micro-Architecture, which are the elements and techniques used to implement the ISA, e.g. microcode, pipelining, cache systems, etc. Bigger instruction-sets mean more complex chips (higher costs, reduced efficiency), but shorter programs. Each microprocessor family has its unique instruction set. Following are the few ISA;

MIPS
Motorola 6800
ARM
PowerPC
x86 (Pentium)
ALGOL Object Code
SPARC

7.9The 1st microprocessor : Intel 4004

The first microprocessor was the Intel 4004, introduced in 1971. The 4004 was not very powerful all it could do was add and subtract, and it could only do that 4 bits at a time. But it was amazing that everything was on one chip. Prior to the 4004, engineers built computers either from collections of chips or from discrete components (transistors wired one at a time). The 4004 powered one of the first portable electronic calculators. It was as powerful as ENIAC which had 18000 tubes and occupied a large room. It cost less then $100. Its targeted use was of calculation. It consisted of 2250 transistors and 16pins. Speed was 108 kHz, 60,000 ops/sec.

Why Intel came up with the idea?

A Japanese calculator manufacturer, Busicom wanted Intel to develop 16 separate IC's for a line of new calculators. Intel, at that point in time known only as a memory manufacturer, was quite small and did not have the resources to do all 16 chips. Then Ted Hoff came up with the idea of doing all 16 on a single chip. Later, Intel realized that the 4004 could have other uses as well.

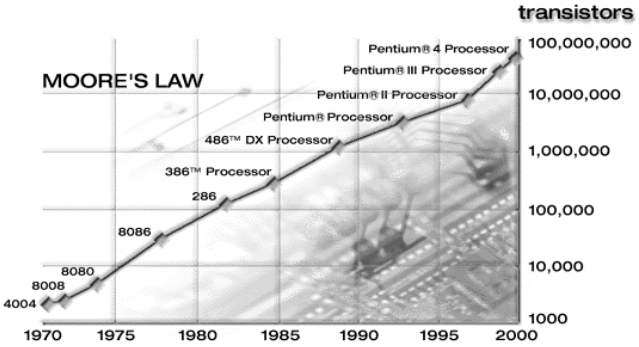
Currently Intel came with – Intel Pentium 4 (2.2GHz).

It was introduced in December 2001. It got 55 million transistors. 32-bit word size. Within the processor it has 2 ALU's each working at 4.4GHz. It costs around $600.

Moore's Law

Moore's law(1965) is an empirical observation stating in effect that at our rate of technological development and advances in the semiconductor industry the complexity of integrated circuits doubles every 18 months. His original empirical observation was that the number of components on semiconductor chips with lowest per-component cost doubles roughly every 12 months, and he conjectured that the trend will stay for at least 10 years. In 1975, Moore revised his estimate for the expected doubling time, arguing that it was slowing down to about two years

Evolution of Intel Microprocessors



4-, 8-, 16-, 32-, 64-bit (Word Length)

The 4004 dealt with data in chunks of 4-bits at a time

Pentium 4 deals with data in chunks (words) of 32-bit length

The new Itanium processor deals with 64-bit chunks (words) at a time

kHz, MHz, GHz (Clock Frequency)

4004 worked at a clock frequency of 108kHz

The latest processors have clock freqs. in GHz

Out of 2 microprocessors having similar designs, one with higher clock frequency will be more powerful

Same is not true for 2 microprocessors of dissimilar designs. Example: Out of PowerPC & Pentium 4 microprocessors working at the same freq, the former performs better due to superior design. Same for the Athlon microprocessor when compared with a Pentium

Enhancing the capability of a microprocessor ?

The computing capability of a microprocessor can be enhanced in many different ways:

By increasing the clock frequency
By increasing the word-width
By having a more effective caching algorithm and the right cache size
By adding more functional units (e.g. ALU's, FPU's, Vector/SIMD units, etc.)
Improving the architecture

What have we learnt today?

Today we learnt about the microprocessor, the key component, the brain, of a computer
We learnt about the function of a microprocessor
And its various sub-systems
Bus interface unit
Data & instruction cache memory
Instruction decoder
ALU
Floating-point unit
Control unit

Lecture 8

Binary Numbers & Logic Operations

The focus of the last lecture was on the microprocessor

During that lecture we learnt about the function of the central component of a computer, the microprocessor

And its various sub-systems

Bus interface unit

Data & instruction cache memory

Instruction decoder

ALU

Floating-point unit

Control unit

Learning Goals for Today

To become familiar with number system used by the microprocessors - binary numbers

To become able to perform decimal-to-binary conversions

To understand the NOT, AND, OR and XOR logic operations – the fundamental operations that are available in all microprocessors

BINARY
(BASE 2)
numbers

DECIMAL
(BASE 10)
numbers

Decimal (base 10) number system consists of 10 symbols or digits

0 1 2 3 4
5 6 7 8 9

Binary (base 2) number system consists of just two

0 1

Other popular number systems

Octal

base = 8

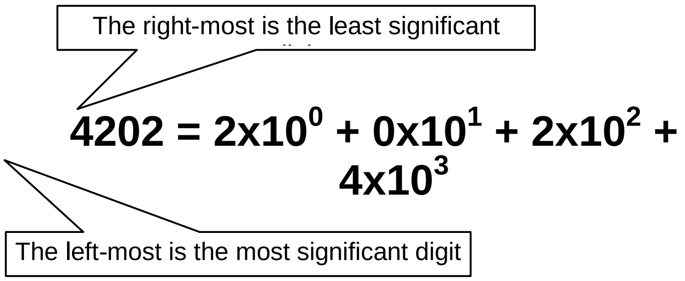
8 symbols (0,1,2,3,4,5,6,7)

Hexadecimal

base = 16

16 symbols (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)

Decimal (base 10) numbers are expressed in the positional notation

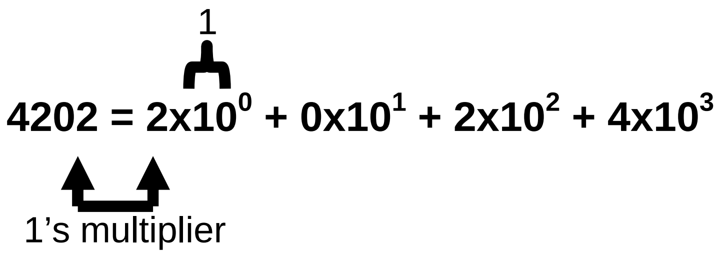


The right-most is the least significant

$$4202 = 2x10^0 + 0x10^1 + 2x10^2 + 4x10^3$$

The left-most is the most significant digit

Decimal (base 10) numbers are expressed in the *positional notation*



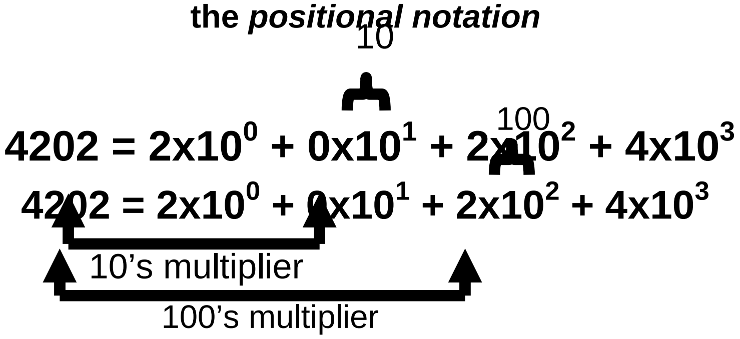
1

$4202 = 2x10^0 + 0x10^1 + 2x10^2 + 4x10^3$

1's multiplier

Decimal (base 10) numbers are expressed in the *positional notation*

Decimal (base 10) numbers are expressed in the *positional notation*



10

100

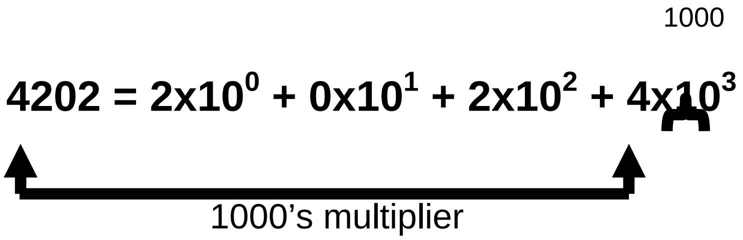
$4202 = 2x10^0 + 0x10^1 + 2x10^2 + 4x10^3$

$4202 = 2x10^0 + 0x10^1 + 2x10^2 + 4x10^3$

10's multiplier

100's multiplier

Decimal (base 10) numbers are expressed in the *positional notation*

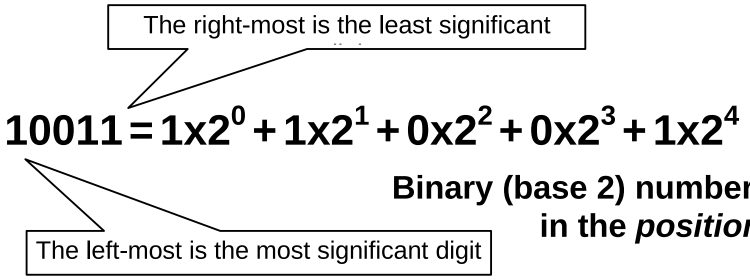


1000

$4202 = 2x10^0 + 0x10^1 + 2x10^2 + 4x10^3$

1000's multiplier

Binary (base 2) numbers are also expressed in the *positional notation*

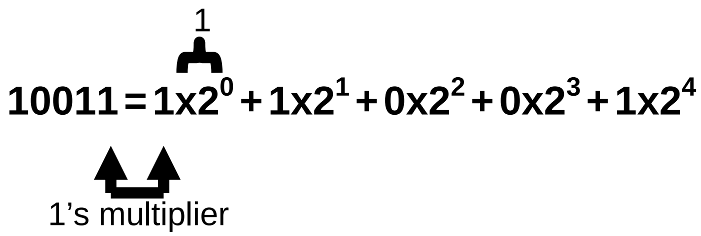


The right-most is the least significant

$$10011 = 1x2^0 + 1x2^1 + 0x2^2 + 0x2^3 + 1x2^4$$

The left-most is the most significant digit

Binary (base 2) numbers are also expressed in the *positional notation*

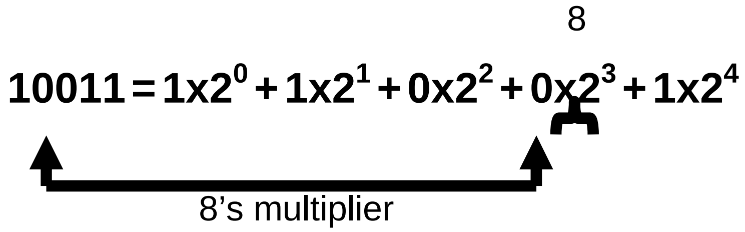


1

$$10011 = 1x2^0 + 1x2^1 + 0x2^2 + 0x2^3 + 1x2^4$$

1’s multiplier

Binary (base 2) numbers are also expressed in the *positional notation*

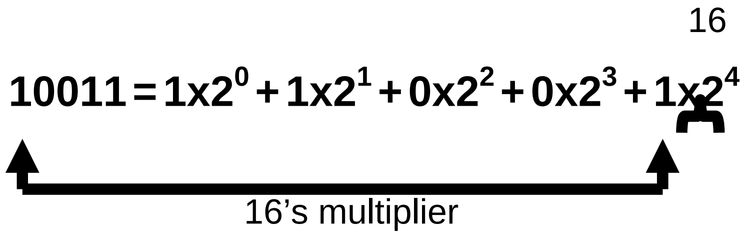


8

$$10011 = 1x2^0 + 1x2^1 + 0x2^2 + 0x2^3 + 1x2^4$$

8’s multiplier

Binary (base 2) numbers are also expressed in the *positional notation*



16

$$10011 = 1x2^0 + 1x2^1 + 0x2^2 + 0x2^3 + 1x2^4$$

16’s multiplier

Counting in Decimal

0	10	20	30
1	11	21	31
2	12	22	32
3	13	23	33
4	14	24	34
5	15	25	35
6	16	26	36
7	17	27	.
8	18	28	.
9	19	29	.

Counting in Binary

0	1010	10100	11110
1	1011	10101	11111
10	1100	10110	100000
11	1101	10111	100001
100	1110	11000	100010
101	1111	11001	100011
110	10000	11010	100100
111	10001	11011	.
1000	10010	11100	.
1001	10011	11101	.

8.1 Why binary

Because this system is natural for digital computers

The fundamental building block of a digital computer – the switch – possesses two natural states, ON & OFF.

It is natural to represent those states in a number system that has only two symbols, 1 and 0, i.e. the binary number system

In some ways, the decimal number system is natural to us humans. Why?

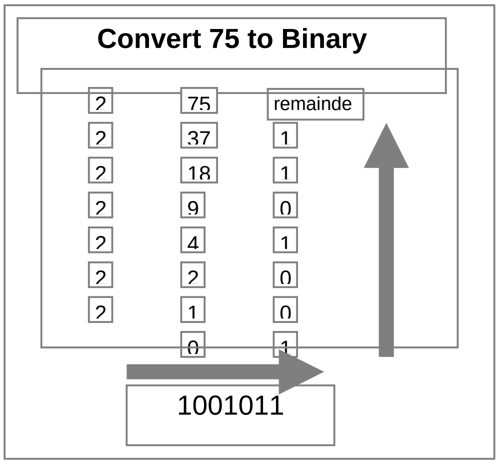
bit

binary digit

Byte = 8 bits

Decimal Binary conversion →

.



Convert 75 to Binary

2	75	remainde
2	37	1
2	18	1
2	9	0
2	4	1
2	2	0
2	1	0
	0	1

1001011

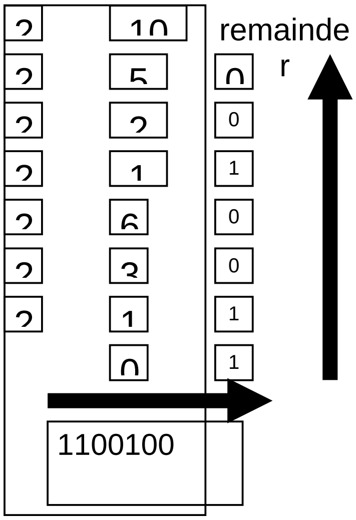
Check

1001011 = $\mathbf{1x2^0 + 1x2^1 + 0x2^2 + 1x2^3 + 0x2^4 + 0x2^5 + 1x2^6}$

$\mathbf{= \quad 1 + 2 + 0 + 8 + 0 + 0 + 64}$

$\mathbf{= \quad 75}$

Convert 100 to Binary



That finishes our first topic - introduction to binary numbers and their conversion to and from decimal numbers

Our next topic is …

8.2 Boolean Logic Operations

Let *x, y, z* be Boolean variables. Boolean variables can only have binary values i.e., they can have values which are either 0 or 1.

For example, if we represent the state of a light switch with a Boolean variable *x*, we will assign a value of 0 to *x* when the switch is OFF, and 1 when it is ON

A few other names for the states of these Boolean variables

0	**1**
Off	**On**
Low	**High**
False	**True**

We define the following logic operations or functions among the Boolean variables

Name	Example	Symbolically
NOT	y = NOT(x)	x'
AND	$z = x$ AND y	$x \cdot y$
OR	$z = x$ OR y	$x + y$
XOR	$z = x$ XOR y	$x \oplus y$

We'll define these operations with the help of truth tables

what is the truth table of a logic function

A truth table defines the output of a logic function for all possible inputs

Truth Table for the NOT Operation
(y true whenever x is false)

X	***y = x´***
0	
1	

Truth Table for the NOT Operation

X	*y = x´*
0	1
1	0

Truth Table for the AND Operation
(z true when both x & y true)

X	*y*	*z = x ·y*
0	0	
0	1	
1	0	
1	1	

Truth Table for the AND Operation

X	*y*	*z = x ·y*
0	0	0
0	1	0
1	0	0
1	1	1

Truth Table for the OR Operation
(z true when x or y or both true)

x	*y*	*z = x + y*
0	0	
0	1	
1	0	
1	1	

Truth Table for the OR Operation

x	y	$z = x + y$
0	0	0
0	1	1
1	0	1
1	1	1

Truth Table for the XOR Operation
(z true when x or y true, but not both)

X	y	$z = x \oplus y$
0	0	
0	1	
1	0	
1	1	

8.3 Truth Table for the XOR Operation

X	y	$z = x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Those 4 were the fundamental logic operations. Here are examples of a few more complex situations

$z = (x + y)'$
$z = y \cdot (x + y)$
$z = (y \cdot x) \oplus w$

8.4 STRATEGY: Divide & Conquer

$z = (x + y)'$

x	y	$x + y$	$z = (x + y)'$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

$z = y \cdot (x + y)$

x	y	$x + y$	$z = y \cdot (x + y)$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	1	1

$z = (y \cdot x) \oplus w$

x	y	W	$y \cdot x$	$z = (y \cdot x) \oplus w$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	0

Number of rows in a truth table?
2^n
n = number of input variables
What have we learnt today?
About the binary number system, and how it differs from the decimal system
Positional notation for representing binary and decimal numbers
A process (or algorithm) which can be used to convert decimal numbers to binary numbers
Basic logic operations for Boolean variables, i.e. **NOT, OR, AND, XOR, NOR, NAND, XNOR**
Construction of truth tables (How many rows?)

<u>Focus of the Next Lecture</u>

Next lecture will be the 3rd on Web dev
The focus of the one after that, the 10th lecture, however, will be on software. During that lecture we will try:
To understand the role of software in computing
To become able to differentiate between system and application software

Lecture 9
HTML Lists & Tables
(Web Development Lecture 3)

Today is our 3rd Web Dev lecture During our 2nd lecture on Web dev …

We learnt to develop our own Web pages in HTML

We learnt about some of the tags used in HTML pages

<BR>, <P>, </P>, <B>, <TITLE>, </TITLE>, <H1>, </H1>

<HTML></HTML>, <HEAD></HEAD>, <BODY></BODY>

<A HREF = *"action"* > *label* </A>, action=http:// or mailto:

We also learnt about how to upload our Web pages to VU's Web server so that it becomes visible on the Internet as http://www.vu.edu.pk/~xxxxxxxx/

where xxxxxxxx is your VU user ID

Today's Lecture

We will extend our Web pages by adding a few more tags

Specifically, we will learn about various types of lists that can be added to a Web page

And also, about tables

But first …

A few comments on the general structure of HTML tags

9.1 Single Tags

<*tagName*>

Example: <BR>

Single Tags with Atributes

<*tagName attributes*>

Example: <HR width="50%">

Paired Tags

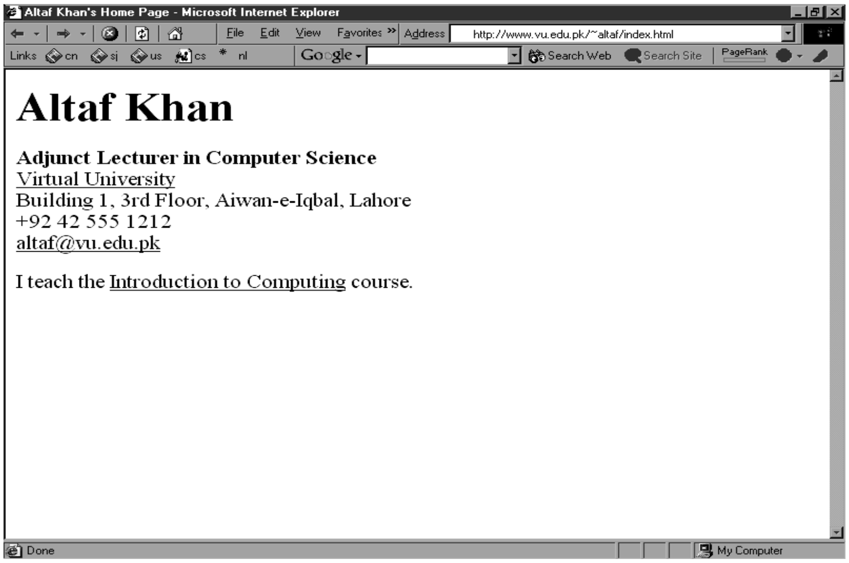
<*tagName*> … </*tagName*>

Example: <H1> … </H1>

Paired Tags with Attributes

<*tagName attributes* > … </*tagName*>

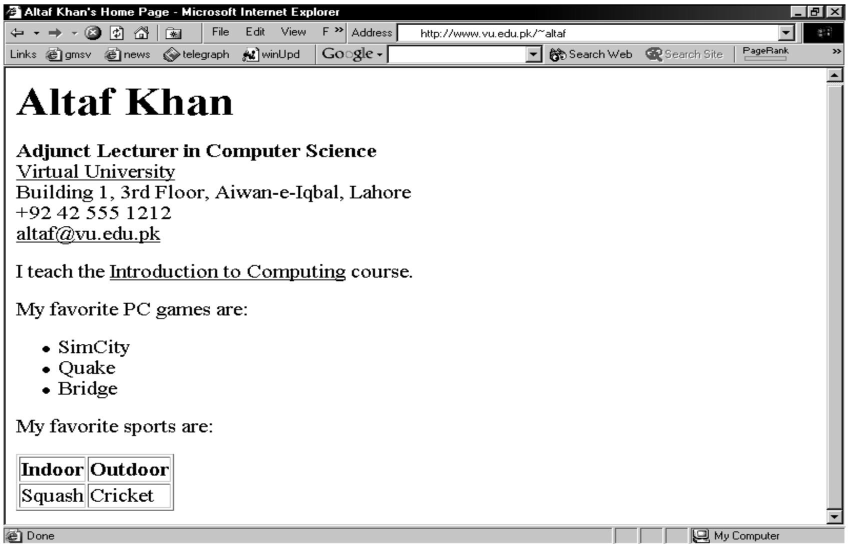
Example: <H1 align="center"> … </H1>


Altaf Khan's Home Page - Microsoft Internet Explorer
File Edit View Favorites
Address http://www.vu.edu.pk/~altaf/index.html
Links cn sj us cs nl
Google
Search Web
Search Site
PageRank
Done
My Computer

Altaf Khan

Adjunct Lecturer in Computer Science
Virtual University
Building 1, 3rd Floor, Aiwan-e-Iqbal, Lahore
+92 42 555 1212
altaf@vu.edu.pk

I teach the Introduction to Computing course.


Altaf Khan's Home Page - Microsoft Internet Explorer
File Edit View F
Address http://www.vu.edu.pk/~altaf
Links gmsv news telegraph winUpd
Google
Search Web
Search Site
PageRank
Done
My Computer

Altaf Khan

Adjunct Lecturer in Computer Science
Virtual University
Building 1, 3rd Floor, Aiwan-e-Iqbal, Lahore
+92 42 555 1212
altaf@vu.edu.pk

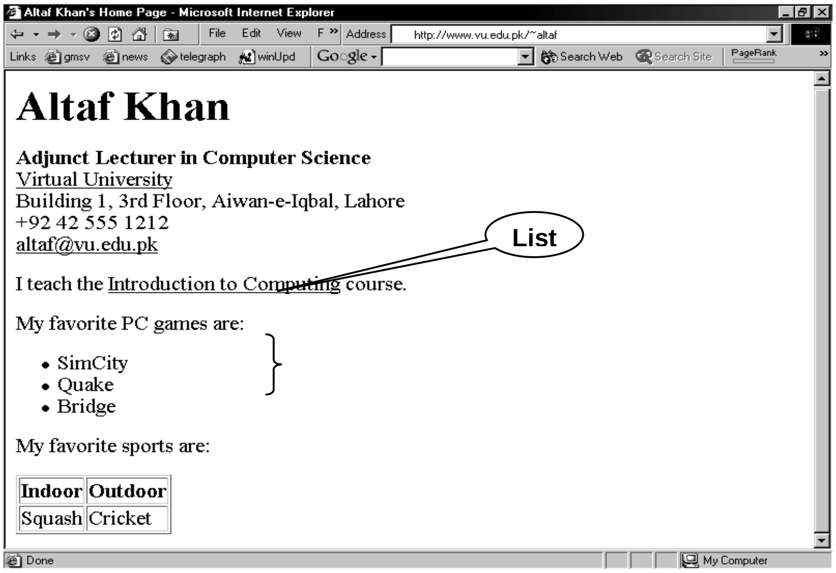
I teach the Introduction to Computing course.

My favorite PC games are:

- SimCity
- Quake
- Bridge

My favorite sports are:

Indoor	Outdoor
Squash	Cricket


Altaf Khan's Home Page - Microsoft Internet Explorer
File Edit View F
Address http://www.vu.edu.pk/~altaf
Links gmsv news telegraph winUpd
Google
Search Web
Search Site
PageRank
List
Done
My Computer

Altaf Khan

Adjunct Lecturer in Computer Science
Virtual University
Building 1, 3rd Floor, Aiwan-e-Iqbal, Lahore
+92 42 555 1212
altaf@vu.edu.pk

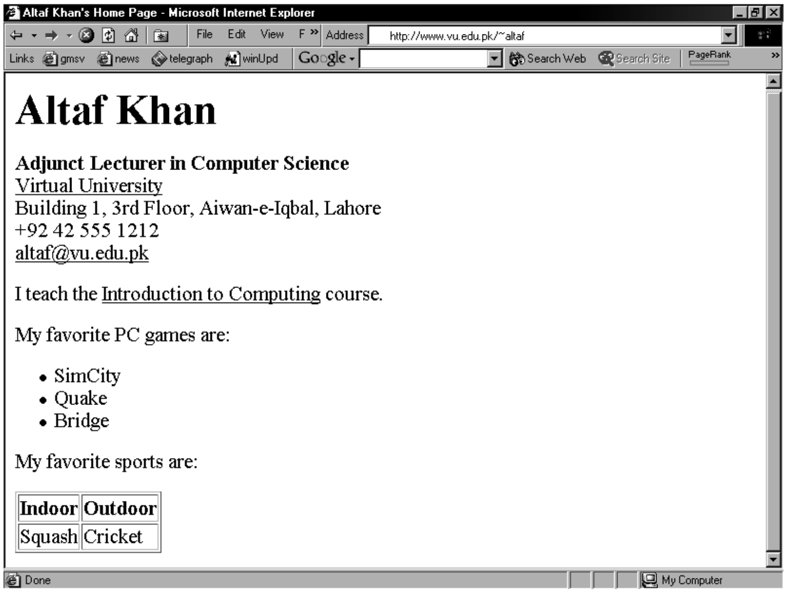
I teach the Introduction to Computing course.

My favorite PC games are:

- SimCity
- Quake
- Bridge

My favorite sports are:

Indoor	Outdoor
Squash	Cricket


Altaf Khan's Home Page - Microsoft Internet Explorer
File Edit View F
Address http://www.vu.edu.pk/~altaf
Links gmsv news telegraph winUpd
Google
Search Web Search Site PageRank
Done
My Computer

Altaf Khan

Adjunct Lecturer in Computer Science
Virtual University
Building 1, 3rd Floor, Aiwan-e-Iqbal, Lahore
+92 42 555 1212
altaf@vu.edu.pk

I teach the Introduction to Computing course.

My favorite PC games are:

- SimCity
- Quake
- Bridge

My favorite sports are:

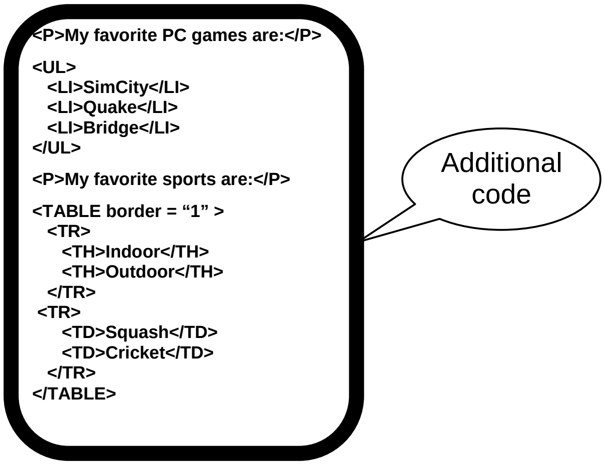
Indoor	Outdoor
Squash	Cricket

```
<HTML>
<HEAD>
<TITLE>Altaf Khan's Home Page</TITLE>
</HEAD>
<BODY>
<H1>Altaf Khan</H1>
<P><B>Adjunct Lecturer in Computer Science</B><BR>
<A HREF="http://www.vu.edu.pk/">Virtual University</A><BR>
Building 1, 3rd Floor, Aiwan-e-Iqbal, Lahore<BR>
+92 42 555 1212<BR>
<A HREF="mailto:altaf@vu.edu.pk">altaf@vu.edu.pk</A><BR></P>
<P>I teach the <A HREF="http://www.vu.edu.pk/cs101/">Introduction to
Computing</A> course. </P>
</BODY>
</HTML>
```

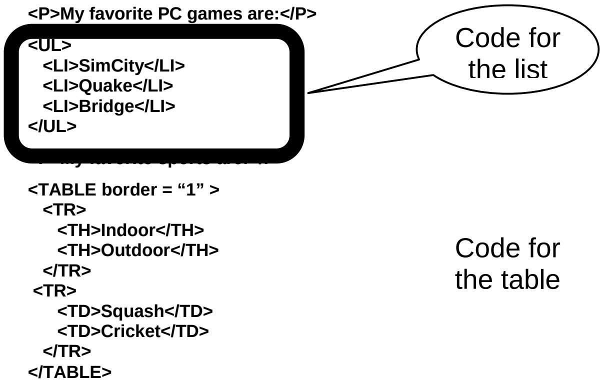
```
<HTML>
<HEAD>
<TITLE>Altaf Khan's Home Page</TITLE>
</HEAD>
<BODY>
<H1>Altaf Khan</H1>
<P><B>Adjunct Lecturer in Computer Science</B><BR>
<A HREF="http://www.vu.edu.pk/">Virtual University</A><BR>
Building 1, 3rd Floor, Aiwan-e-Iqbal, Lahore<BR>
+92 42 555 1212<BR>
<A HREF="mailto:altaf@vu.edu.pk">altaf@vu.edu.pk</A><BR></P>
<P>I teach the <A HREF="http://www.vu.edu.pk/cs101/">Introduction to
Computing</A> course. </P>
```

The additional code for the list and table goes here

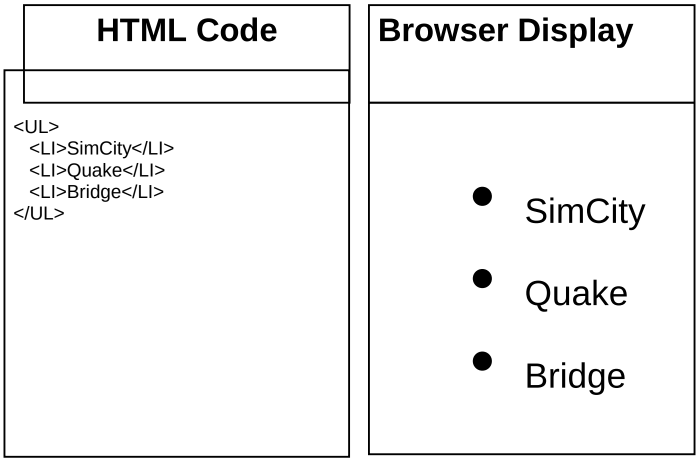
```
</BODY>
</HTML>
```



```
<P>My favorite PC games are:</P>

<UL>
  <LI>SimCity</LI>
  <LI>Quake</LI>
  <LI>Bridge</LI>
</UL>

<P>My favorite sports are:</P>

<TABLE border = “1” >
  <TR>
    <TH>Indoor</TH>
    <TH>Outdoor</TH>
  </TR>
 <TR>
    <TD>Squash</TD>
    <TD>Cricket</TD>
  </TR>
</TABLE>
```



```
<P>My favorite PC games are:</P>

<UL>
  <LI>SimCity</LI>
  <LI>Quake</LI>
  <LI>Bridge</LI>
</UL>

<P>My favorite sports are:</P>

<TABLE border = “1” >
  <TR>
    <TH>Indoor</TH>
    <TH>Outdoor</TH>
  </TR>
 <TR>
    <TD>Squash</TD>
    <TD>Cricket</TD>
  </TR>
</TABLE>
```



HTML Code

```
<UL>
  <LI>SimCity</LI>
  <LI>Quake</LI>
  <LI>Bridge</LI>
</UL>
```

Browser Display

- SimCity
- Quake
- Bridge

<UL>	Un-ordered List

<LI>	Line Item

The default "bullet" for these lists is a "disc"
That, however, can be changed to a "circle" or a "square" with the help of the type attribute

HTML Code

```
<UL type = "circle">
  <LI>SimCity</LI>
  <LI>Quake</LI>
  <LI>Bridge</LI>
</UL>
```

Browser Display

```
●  ●  SimCity
●
●  ●  Quake

   ●  Bridge
```

Q: What happens if I start a new list without closing the original one?

```
<UL>
      <LI>SimCity</LI>
      <LI>Quake II</LI>

      <UL>
      <LI>SimCity 3000</LI>
      <LI>Quake III</LI>
      </UL>

          <LI>Bridge</LI>
</UL>
```

- ➢ Different bullets
- ➢ Additional tab

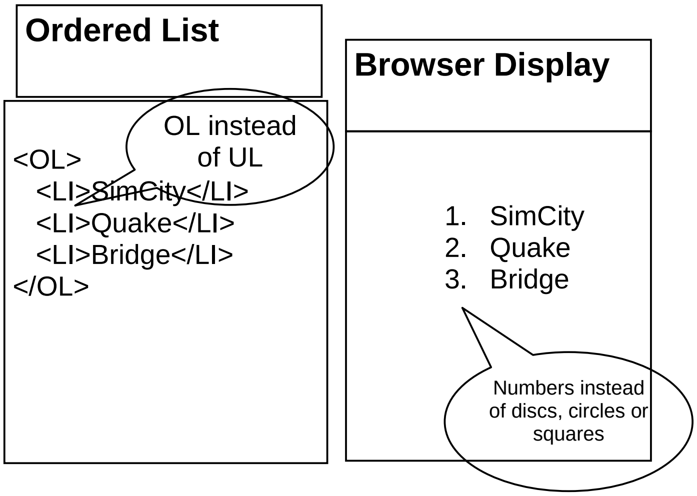
Browser Display

```
   ●  SimCity

   ●  Quake II

          ●  SimCity
o            3000

o         ●  Quake
             III

   ●  Bridge
```

Such structures, i.e., those in which another starts before the first list is finished, are called Nested Lists

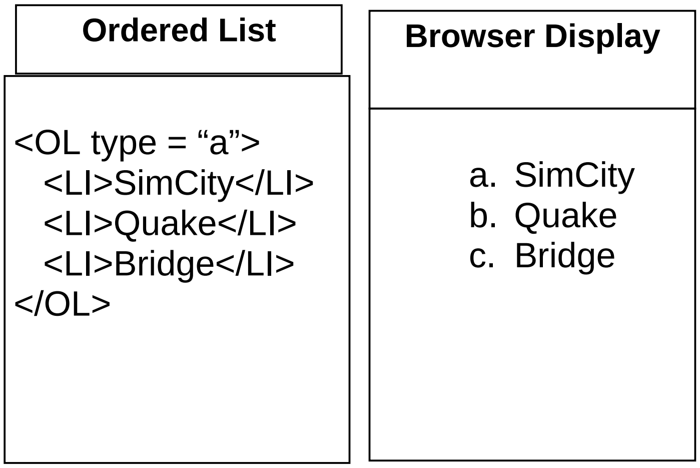
9.2 Types of Lists

In addition to un-ordered lists, HTML supports two other types
Ordered Lists
Definition List



```
<OL>
  <LI>SimCity</LI>
  <LI>Quake</LI>
  <LI>Bridge</LI>
</OL>
```

1. SimCity
2. Quake
3. Bridge



```
<OL type = “a”>
  <LI>SimCity</LI>
  <LI>Quake</LI>
  <LI>Bridge</LI>
</OL>
```

a. SimCity
b. Quake
c. Bridge

9.3 Ordered List Types

Type	Result
“A”	A, B, C, …
“a”	a, b, c, …
“I”	I, II, III, IV, …
“i”	i, ii, iii, iv, …
“1”	1, 2, 3, …

Q: How would one start an ordered list with something other than 1

Browser Display
25. SimCity 26. Quake 27. Bridge

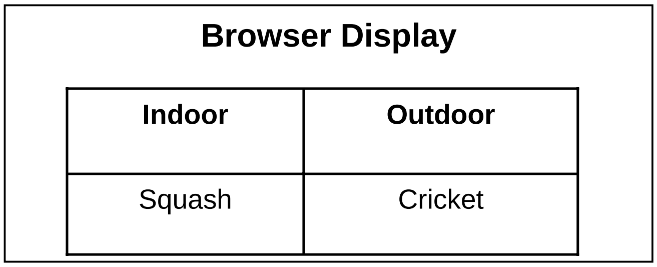
Ordered List	Browser Display

<OL start = “25”>
  <LI>SimCity</LI>
  <LI>Quake</LI>
  <LI>Bridge</LI>
</OL>
	25. SimCity
	26. Quake
	27. Bridge

Definition List	Browser Display
<DL> <DT>SimCity</DT> <DD>A great simulation game in which one build cities </DD> <DT>Quake</DT> <DD> One of the best of the shoot-em-up genre </DD> </DL>	SimCity Ter A great simulation game in which one build cities Definition Quake One of the best of the shoot-em-up genre

<DL>	Definition List
<DT>	Term
<DD>	Definition

Ordered lists as well as definition lists can be nested just like the un-ordered lists
Can any type of list be nested into any other type?
Lists are one way of presenting data in a an ordered or formal fashion
Tables provide another - more customizable - way of displaying ordered information on Web pages

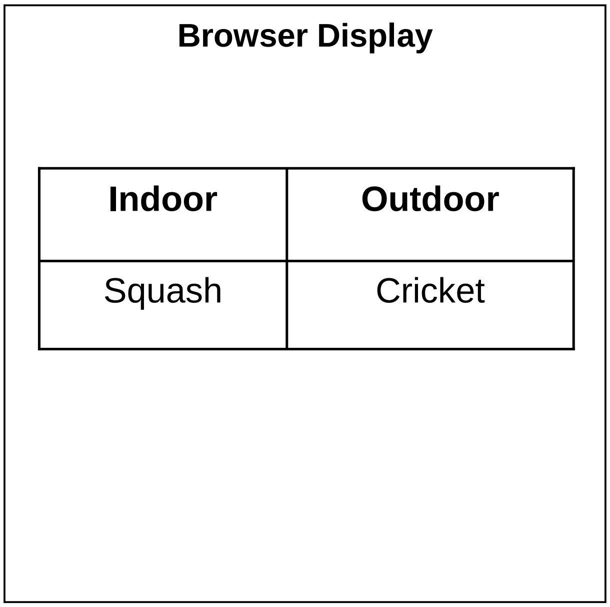


Browser Display

Indoor	Outdoor
Squash	Cricket

HTML Code

```
<TABLE border = "1" >
  <TR>
    <TH>Indoor</TH>
    <TH>Outdoor</TH>
  </TR>
  <TR>
    <TD>Squash</TD>
    <TD>Cricket</TD>
  </TR>
</TABLE>
```



Browser Display

Indoor	Outdoor
Squash	Cricket

<TABLE>	Table (made up of rows)
<TR>	Row (made up of data cells)
<TH>	Heading Data Cell (Can contain paragraphs, images, lists, forms, tables)
<TD>	Data Cell (Can contain paragraphs, images, lists, forms, tables)

<TABLE> Attributes
BORDER

Determines the thickness of the table border
Example: <TABLE BORDER = “2”>
CELLPADING
Determines the distance between the border of a cell and the contents of the cell
Example: <TABLE CELLPADDING = “3”>
CELLSPACING
Determines the empty spacing between the borders of two adjacent cells
Example: <TABLE CELLSPACING = “1”>

HTML Code

```
<TABLE border = "1" >
   <TR>
     <TH>Indoor</TH>
     <TH>Outdoor</TH>
  </TR>
  <TR>
     <TD>Squash</TD>
     <TD>Cricket</TD>
   </TR>
</TABLE>
```

HTML Code

Indoor	**Outdoor**
Squash	Cricket

HTML Code

```
<TABLE>
   <TR>
     <TH>Indoor</TH>
     <TH>Outdoor</TH>
  </TR>
  <TR>
     <TD>Squash</TD>
     <TD>Cricket</TD>
   </TR>
</TABLE>
```

Browse Display

Indoor	Outdoor
Squash	Cricket

<TABLE>,<TR>,<TH>,<TD> Attributes
ALIGN
Possible values: Center, Left, Right
Example: <TH ALIGN = “center”>
BGCOLOR
Example: <TD BGCOLOR = “red”>
WIDTH

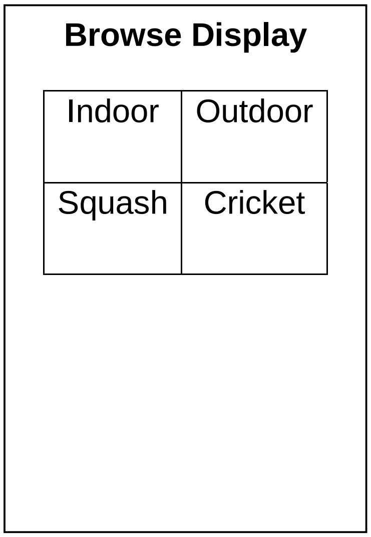
50% of the screen width

Example: <TR WIDTH = “40%”>
HEIGHT
Example: <TABLE HEIGHT = “200”>
<TR> Attributes
VLAIGN
Determines the vertical alignment of the contents of all of the cells in a particular row
Possible values: Top, Middle, Bottom
Example: <TR VALIGN = “bottom”>
<TH> & <TD> Attributes
NOWRAP
Extend the width of a cell, if necessary, to fit the contents of the cell in a single line
Example: <TD NOWRAP>
COLSPAN
No. of rows the current cell should extend itself downward
Example: <TD COLSPAN = “2”>

ROWSPAN
The number of columns the current cell should extend itself
Example: <TD ROWSPAN = “5”>
VALIGN
Same as that for <TR>

HTML CODE

```
<TABLE border=“1” >
  <TR>
    <TH colspan=“2”>
      Indoor Outdoor
    </TH>
  </TR>
  <TR>
    <TD>Squash</TD>
    <TD>Cricket</TD>
  </TR>
</TABLE>
```



Browse Display

Indoor	Outdoor
Squash	Cricket

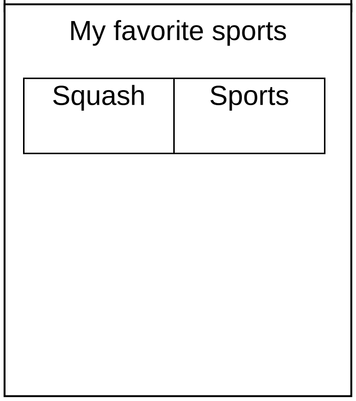
Year	Quarter	Expenses		Income	
		Quetta	Dubai	Quetta	Dubai
2001	1	1,900	8,650	9,000	7,780
	2	2,230	8,650	8,500	8,670
	3	4,000	8,650	9,900	9,870

	4	2,200	8,650	9,800	9,900
2002	1	7,780	8,650	7,780	9,000
	2	8,670	8,650	8,670	8,500
	3	9,870	8,650	9,870	9,900
	4	9,900	8,650	9,900	9,800

HTMAL Code

```
<TABLE border = "1" >
 <CAPTION>
   My favorite sports
 </CAPTION>
   <TR>
     <TD>Squash</TD>
     <TD>Cricket</TD>
  </TR>
</TABLE>
```

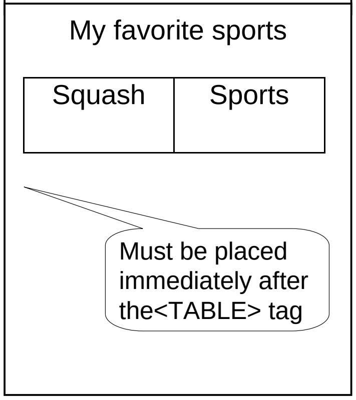
Browser Display



HTMAL Code

```
<TABLE border = "1" >
 <CAPTION>
   My favorite sports
 </CAPTION>
   <TR>
     <TD>Squash</TD>
     <TD>Cricket</TD>
  </TR>
</TABLE>
```

Browser Display



9.4 Useful URL
The Table Sampler

http://hissa.nist.gov/□black/tableQuikRef.html
In Today's Lecture …
We learnt how to extend our Web pages by adding a few more tags

Specifically, we discussed various types of lists that can be added to a Web page – un-ordered, ordered and definition lists
And also, about tables: about various tags used in a table and their associated attributes

Next Web Dev Lecture: Interactive Forms

We will try to understand the utility of forms on Web pages
We will find out about the various components that are used in a form
We will become able to build a simple, interactive form

Lecture 10

Computer Software

Lecture 8 was on the binary number system and logic operations

Learning Goals for Today

To discuss the role of software in computing systems

To learn to differentiate among software belonging to the system and application categories

To learn about software ownership

We mentioned in Lecture 4 that at the highest level, two things are required for computing

Hardware: The physical equipment in a computing environment such as the computer and its peripheral devices (printers, speakers...)

Software: The set of instructions that operates various parts of the hardware. Also termed as "computer program"

Computer Software

The HW needs SW to be useful; the SW needs HW to be useful

When the user needs something done by the computer, he/she gives instructions in the form of SW to computer HW

These instructions need to be written in a language that is readily understood by computer uP

10.1 Machine Language

A system of codes directly understandable by a computer's CPU is termed this CPU's **native** or **machine language**. Although *machine code* may seem similar to *assembly language* they are in fact two different types of languages. Machine code is composed only of the two binary digits 0 and 1.

Every CPU has its own machine language, although there is considerable overlap between some. If CPU *A* understands the full language of CPU *B* it is said that *A* is compatible with *B*. CPU *B* may not be compatible with CPU *A*, as *A* may know a few codes that *B* does not.

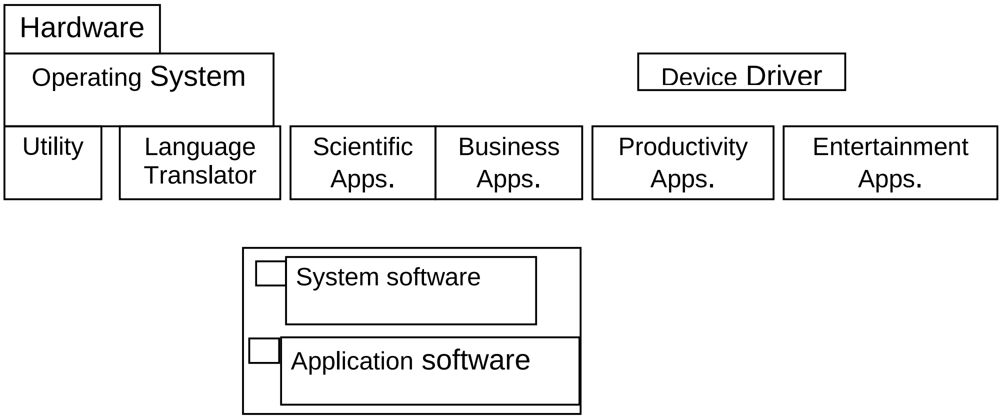
10.2 Language Translators

Human programmers write programs in a language that is easy to understand for them. They use language translators to convert that program into machine language. It converts the human understandable code in uPs understandable code, i.e. a language that is easy to understand for the uPs

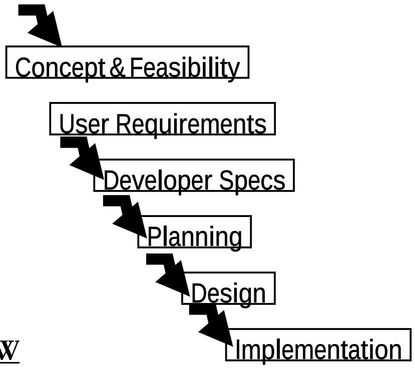
10.3 Software Development

A **software development process** is a process used to develop computer software. It may be an ad hoc process, devised by the team for one project, but the term often refers to a standardised, documented methodology which has been used before on similar projects or one which is used habitually within an organisation.

The SW development process involves many steps, and coding, that is typing the instructions in a high-level language is only a small part of that process – taking-up only around 15% of the effort



The Software Development Process



10.4 Major Types of SW

System SW

System software is responsible for controlling, integrating, and managing the individual hardware components of a computer system.

System software performs tasks like transferring data from memory to disk, or rendering text onto a display specific kinds of system software include loading programs, operating systems, device drivers, compilers, assemblers, linkers, and utilities.

Software libraries that perform generic functions also tend to be regarded as system software. System software stored on non-volatile storage on integrated circuits is usually termed firmware. These generally perform the background tasks in a computer. These programs, many times, talk directly to the HW.

Application SW

Programs that generally interact with the user to perform work that is useful to the user. These programs generally talk to the HW through the assistance of system SW.

10.5 System SW are programs that …

Control the overall operation of the computer

OS

Interact directly with HW

Device drivers

Perform system management & maintenance

Utilities

Are used to develop or maintain other programs

Language translators

10.6 Operating System

It performs its work invisibly to control the internal functions of a computer, e.g. maintaining files on the disk drive, managing the screen, controlling which tasks the uP performs and in what order. It interacts directly with the computer HW. Other SW normally does not directly interact with the HW, e.g.

Windows Mac OS Linux
UNIX Solaris DOS
CP/M VMS Firmware

ROM is a component of OS that permanently stored on a chip. It is a firm ware program. When a computer is powered-on, it is the first program that it always executes. Firmware consists of startup and a few low-level I/O routines that assist the computer in finding out and executing the rest of the OS. On IBM-compatible PC's, it is called **BIOS**

10.7 Utilities:

It is a small program that provides an addition to the capabilities provided by the operating system. In some usages, a utility is a special and nonessential part of the operating system. These are the computer programs that perform a particular function related to computer system management and maintenance

Examples:

1. Anti-virus SW
2. Data compression SW

Disk optimization SW

Disk backup SW

10.8 Language Translators

Programs that take code written in a HLL and translate it into a low-level language that is easily understood by the uP

1. **Compiler translates** the program written in a HLL in one go. The translated code is then used by the uP whenever the program needs to be run

2**. Interpreter translates** the HLL program one statement at time. It reads a single statement, translates it into machine language and passes that machine language code to the uP and then translates the next statement, and so on …

10.9 Device Drivers

A **device driver,** often called a **driver** for short, is a computer program that is intended to allow another program (typically, an operating system) to interact with a hardware device. Think of a **driver** as a manual that gives the operating system (e.g., Windows) instructions on how to use a particular piece of hardware.

A device driver essentially converts the more general input/output instructions of the operating system to messages that the device type can understand.

10.10 Application SW

Application SW are programs that interact directly with the user for the performance of a certain type of work

Scientific/engineering/graphics SW

Mathematica; AutoCad; Corel Draw

Business SW

The billing system for the mobile phone company

Productivity SW

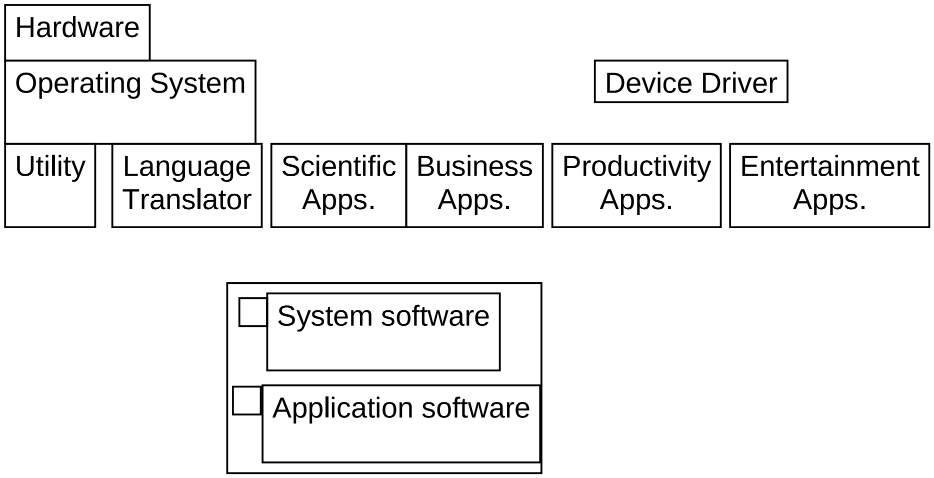
Word processors; Spreadsheets

Entertainment SW

Games

Educational SW

Electronic encyclopedias; The VU Web site



10.11 Another way of classifying SW

Shrink-Wrapped SW

You can just go to a shop and buy it

Custom-built SW

You cannot just go to a shop and buy it; you have to find someone who can develop it for you

Shrink-Wrapped SW

SW built in such a way that it is useful for many different users in many different ways.

Example: MS Word. Individuals use it and so do many large corporations. It is used for writing one-page letters and also to typeset books

Custom-Built SW (1)

These SW are built for a particular organization to fulfill the needs of that particular organization. This type of SW is expensive because the builder has to recoup costs and make a profit from a single sale

Example: A system for predicting the preferences of the Nortwest Airline pilots

Custom-Built SW (2)

This is other type of custom built SW. The delivery time is longer. Customers get more productivity out of it because it is built according to their exact specifications – just like a custom-built shoe fits better, but generally is more expensive, and requires a longer period for delivery

10.12 Who Owns Software?

Generally, although a piece of SW that is being used by millions, it is not owned by any of them! Instead, it is owned by the maker of the SW

The makers let us use their SW but keep the ownership to themselves. When we buy a SW package, we do not really buy it – we just buy a license that allows us to use it, the ownership stays with the maker

However, there are variations on this theme …

10.13 Main types of SW licensees

Proprietary – Most software on a Windows PC or a Macintosh belongs to this category

Freeware – Most software on a Linux PC belongs to that category

Shareware – the category which lies between the above two categories

10.14 Proprietary SW License

Proprietary software, as defined by the Free Software Foundation, means any software that is not free software or is only partially free. The modification, use and redistribution are prohibited, or requires express permissions from the originator. The user needs to pay the maker of the SW for buying a license that allows the user to use the SW

The license, generally, does not transfer the ownership of the SW; it just allows the user to use it. The user is legally barred from making copies of the licensed SW. Generally, the license is for the personal use only. Most SW in use in the world is of this type.
Examples: Windows, Mac OS, MS Word, Adobe Photoshop, Norton Antivirus

Types of Proprietary Licenses

Single-user license
Multi-user license
Concurrent-user license
Site license

10.15 Freeware SW License

It is also known as "Public Domain SW". It allows the user to free use of the SW. The author, however, generally retains ownership. It can usually be downloaded from various Web sites.
Examples: Linux; LaTeX; Netscape Web browser – the Navigator; MS Web browser – the Internet Explorer

10.16 Open-Source SW License

Some authors give away the machine code only, which is extremely difficult to modify, if at all. Others even give away the high-level language source code so that users can make changes according to their own requirements. The later practice is called open-source licensing.

Generally is any computer software whose source code is either in the public domain or, more commonly, is copyrighted by one or more persons/entities and distributed under an open-source license . Such a license may require that the source code be distributed along with the software, and that the source code be freely modifiable, with at most minor restrictions, such as a requirement to preserve the authors' names and copyright statement in the code,
Examples: Linux; Netscape Navigator

10.17 Shareware SW License

Shareware is software that is distributed without payment ahead of time as is common for proprietary software. Typically shareware software is obtained free of charge by downloading, thus allowing one to try out the program ahead of time. A shareware program is accompanied by a request for payment, and often payment is required per the terms of the license past a set period of time. shareware are similar in that they can be obtained and used without monetary cost. Usually shareware differs from open source software in that requests of voluntary "shareware fees" are made, often within the program itself, and in that source code for shareware programs is generally not available in a form that would allow others to extend the program.

A shareware's program source, maintenance and extensibililty can sometimes be negotiated for a licensing fee with the author(s) similar to standard proprietary software.
Examples: WinZip, Download Accelerator

10.18 Trialware

It is similar to shareware but difference is that the SW is usable for a short period only. When the period is expired, it is no more in use until the license is not purchased. The trial period may vary according to its developer. This period may range from a week to a few months.
It can be downloaded from the Internet or alternatively.

What have we learnt today?

We have found out about the role software plays in a computing environment
We also learned to distinguish between software belonging to the system and application categories
We also discussed the different types of software licenses

Topics of some of the future lectures

Operating system

Application SW
Productivity SW
Word processor
Spreadsheets
Presentation making
Databases
Programming Languages
The SW development process
The Web development series of lectures is clearly focused on developing SW

Focus of the Next Lecture

The role of the OS in a computing environment
The various functions that an OS performs
The main components of an OS
Various types of OSes

Lecture 11

Operating Systems

Focus of the last lecture: computer SW

We found out about the role SW plays in a computing environment

We learned to distinguish between SW belonging to the system & application categories

Also discussed the different types of SW licenses:

Proprietary

Free

Open source

Shareware

Trialware

Learning Goals for Today

The role of the operating system in a computing environment

The various functions that an operating system performs

The main components of an operating system

Various types of operating systems

11.1 Why Have OSes?

User/programmer convenience

Greater resource utilization

The Role of An OS

The 1st program that runs when a typical computer is turned ON, and the last one to finish running when the computer is turned OFF.

It manages the HW and SW resources of the computer system, often invisibly. These include the processor, memory, disk drives, etc.

It provides a simple, consistent way for applications to interact with the HW without having to know all the details of the HW

Advantage for App. Developers

Application developers do not need to know much about the HW while they are developing their app

They just develop with a particular OS in mind. If the OS runs on many types of computers having different HW configurations, so will the app – without making any HW-specific modifications in the app SW. The OS hides the HW differences from the app

Are OS'es Essential?

No. If a computer has been designed for limited functionality (e.g. it runs just a single program all the time as in a automatic clothes washing machine), it does not require a traditional OS

In limited-functionality computers, an OS just adds to the overhead unnecessarily, which impedes the computer's performance

In these situations, the required parts of the OS are integrated into the only program that is going to run

In the beginning …

A single user ran a single program ran on a single computer – there was no need for an OS

Then came computer operators who ran multiple programs for multiple users one after the other – still, no need for an OS

Later computers became powerful and able to run multiple programs, simultaneously. That's when the need for OS'es arose for:

Managing the resources of the computers efficiently

Making use of computers convenient for users/programmers

11.2 Core Tasks of an OS

Processor management

Memory management

Device management

Storage management

Application Interface
User Interface

Processor Management

Memory Management

Straight forward for a single-user, single tasking
Each app must have enough private memory in which to execute
App can neither run into the private memory space of another app, nor be run into by another app
Different types of memory (e.g. main, cache) in the system must be used properly, so that each app can run most effectively

Storage Management

The OS manages storage through one of its sub-modules, the File Manager
A file system is a collection of directories, subdirectories, and files organized in a logical order.
File manager maintains an index of the filenames & where they are located on the disk.
File manager make it easy to find the required file in a logical and timely fashion.

Device Management

Applications talk to devices through the OS and OS talks to and manages devices through Device Drivers
Example: When we print to a laser printer, we do not need to know its details. All we do is to tell the printer device driver about what needs to be printed and it takes care of the details

Application Interface

App developers do not need to know much about the HW, especially the uP, while they are developing their app
The OS provides all apps with a straight-forward and consistent interface to the HW

Example: An app uses the OS to store data on the disk drive. For that, the app does not need to know about the exact physical characteristics of that drive; it just tells the OS to do that through the app interface, and the OS takes cares of all the details of the task

User Interface

Users communicate with the computer using a consistent user interface provided by the OS
This UI can be a command-line interface in which a user types in the commands. Example:

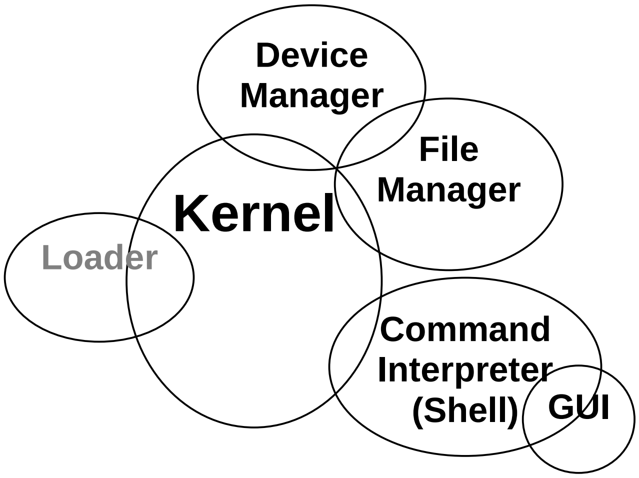
copy a:/file1.html c:/file1.html

Or, it can be a graphical UI, where Windows, Icons, Menus, and a Pointing device (such as a mouse) is used to receive and display information. Example:

With the help of the mouse, drag file1.html from drive a to drive c

11.3 OS Components

Error!



11.4 Kernel

The heart of the OS

Responsible for all the essential operations like basic house keeping, task scheduling, etc.

Also contains low-level HW interfaces

Size important, as it is memory-resident

11.5Types of OS'es

Classification w.r.t. the type of computers they run on and the type of applications they support

Real-Time Operating System (RTOS)

Single-User, Single Task

Single-User, Multi-Tasking

Multi-User

RTOS (1)

Used to run computers embedded in machinery, robots, scientific instruments and industrial systems

Typically, it has little user interaction capability, and no end-user utilities, since the system will be a "sealed box" when delivered for use

Examples: Wind River, QNX, Real-time Linux, Real-time Windows NT

RTOS (2)

An important part of an RTOS is managing the resources of the computer so that a particular operation executes in precisely the same amount of time every time it occurs

In a complex machine, having a part move more quickly just because system resources are available may be just as catastrophic as having it not move at all because the system was busy

Single-User, Single Task

OS'es designed to manage the computer so that one user can effectively do one thing at a time

The Palm OS used in many palmtop computers (PDA's) is an example of a single-user, single-task OS

Single-User, Multi-Tasking

Most popular OS

Used by most all PC's and Laptops

Examples: Windows, Mac OS, Linux

Lets a single user interact with several programs, simultaneously

Multi-User

A multi-user OS allows many users to take advantage of the computer's resources, simultaneously

The OS must make sure that the requirements of the various users are balanced, and that the programs they are using each have sufficient and separate resources so that a problem with one user doesn't affect any of the other users

Examples: Linux, Unix, VMS and mainframe OS'es, such as MVS

11.6 Another Way of Classifying

Uni-processor OS'es

Designed to schedule tasks on a single uP only

Example: DOS

Multi-processor OS'es

Can control computers having multiple uPs, at times 1000's of them

Example: Current versions of Windows, Mac OS, Linux, Solaris

11.7 How many different OS'es are there?

100's

OS'es from the Windows family dominate the desktops and run on millions of PC's

OS'es from the Unix family (Unix, Linux, etc) are quite popular on servers

There are hundreds more. Some designed for mainframes only. Some for embedded applications only.

11.8 Comparing Popular OS'es

OS	*HW*	*Stability*	*Cost*	*Apps.*	*Support*	*Security*	*Popularity*
Windows (GUI)	PC	Poor	$300	Huge no.	OK	Poor	Amazing
Mac OS (Shell/GUI)	Mac	Good	$60	Many	OK	Good	Low
Linux (Shell/GUI)	Many	Good	Low	Many	Variable	Good	Low
Unix (Shell/GUI)	Many	Excellent	High	Many	Expensive	Excellent	Servers

What have we learnt today?

The role of the OS in a computing environment
The various functions that an OS performs
The main components of an OS
Various types of OS'es

Next Lecture: Application SW

We'll learn about application SW, i.e. programs that interact directly with the user for the performance of a certain type of work
We'll try to become familiar with various SW used in the following application areas:
Scientific/engineering/graphics
Business
Productivity
Entertainment
Educational

Lecture 12

Interactive Forms
(Web Development Lecture 4)

Focus of the last lecture was on HTML Lists & Tables

We learnt how to extend our Web pages by adding a few more tags

Specifically, we discussed various types of lists that can be added to a Web page – un-ordered, ordered and definition lists

And also, about tables: about various tags used in a table and their associated attributes

Today's Lecture

We will try to understand the utility of forms on Web pages

We will find out about the various components that are used in a form

We will become able to build a simple, interactive form

Interactive Forms (1)

Without forms, a Web site is "read-only" – it just provides information to the user

Forms enable the user to provide information to the Web site. For example, the user can:

Perform searches on Web site

Give comments

Ask for info that is not available on the Website

Place order for goods and services

Interactive Forms (2)

Can be simple or very complex

Can fill a whole page or just a single line

Can contain a single element or many

Are always placed between the <BODY> and </BODY> tags of a Web page

Interactive Forms (3)

Are GUI-based

May contain:

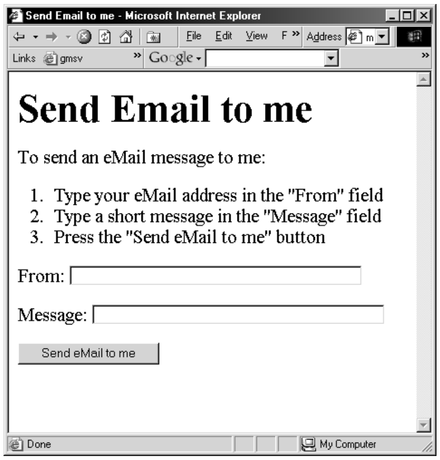
Text fields

Check boxes

Buttons

Scrollable lists

A Simple Example of Interactive Forms



Send Email to me

To send an eMail message to me:

1. Type your eMail address in the "From" field
2. Type a short message in the "Message" field
3. Press the "Send eMail to me" button

From:

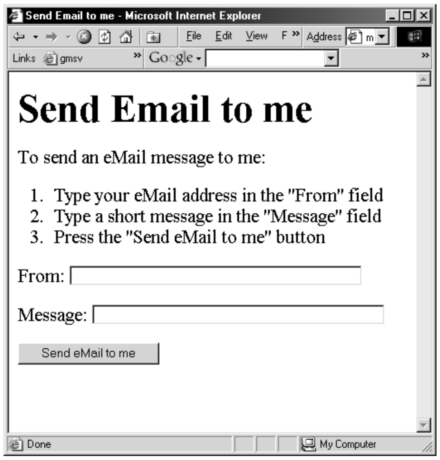
Message:

Code for that Example

```
<HTML>
<HEAD>
	<TITLE>Send Email to me</TITLE>
</HEAD>
```

```
<BODY>

        <H1>Send Email to me</H1>
        Code for the instructions
        Code for the form

</BODY>
</HTML>
```

A Simple Example of Interactive Forms



Send Email to me

To send an eMail message to me:

1. Type your eMail address in the "From" field
2. Type a short message in the "Message" field
3. Press the "Send eMail to me" button

From:

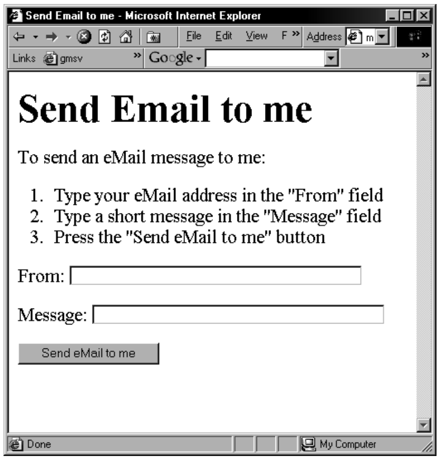
Message:

Send eMail to me

Code for the Instructions

```
<P>To send an eMail message to me:</P>

<OL>
        <LI>Type your eMail address in the &quot;From&quot; field</LI>
        <LI>Type a short message in the   &quot;Message&quot; field</LI>
        <LI>Press the &quot;Send eMail to me&quot; button</LI>
</OL>
```

A Simple Example of Interactive Forms



Send Email to me

To send an eMail message to me:

1. Type your eMail address in the "From" field
2. Type a short message in the "Message" field
3. Press the "Send eMail to me" button

From:

Message:

Send eMail to me

Code for the Form

```
<FORM name="sendEmail" method="post"       action="sendMailScriptURL">
        Elements of the form
</FORM>
```

```
<FORM name="sendEmail" method="post"
      action="sendMailScriptURL">
      Elements of the form
</FORM>
```

name: Name given to the form

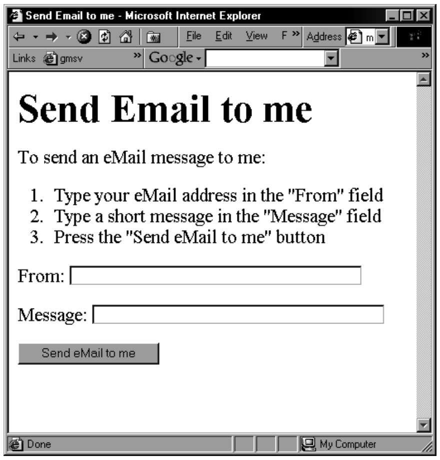
method: Forms can be submitted through two alternate methods – GET & POST

action: Specifies the URL that is accessed when the form is being submitted

12.1 Server-Side Scripts

Are programs that reside on Web servers
Receive info that a user enters in a form
Process that info and take appropriate action
Examples:
CGI scripts on Unix servers
ASP scripts on Windows servers

A Simple Example of Interactive Forms



Send Email to me

To send an eMail message to me:

1. Type your eMail address in the "From" field
2. Type a short message in the "Message" field
3. Press the "Send eMail to me" button

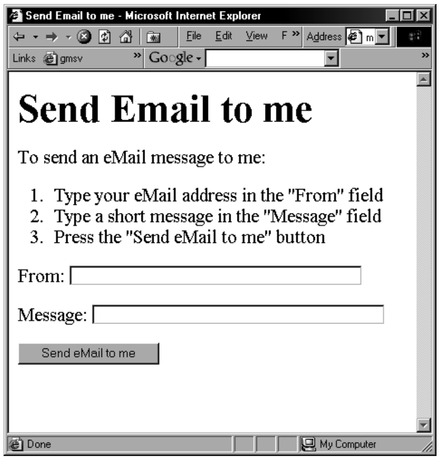
From:

Message:

Send eMail to me

Elements of the Form (1)
<P>From: <INPUT type="text" name=“sender" size="50"></P>
<P>Message: <INPUT type="text" name="message" size="50"></P>

A Simple Example of Interactive Forms



Send Email to me

To send an eMail message to me:

1. Type your eMail address in the "From" field
2. Type a short message in the "Message" field
3. Press the "Send eMail to me" button

From:

Message:

Elements of the Form (2)

```
<P><INPUT type="hidden" name="receiver"    value="altaf@vu.edu.pk"></P>

<P><INPUT type="hidden" name="subject"     value="eMail from the form"></P>

<P><INPUT type="submit" name="sendEmail"  value="Send eMail to me"></P>
```

A Simple Example of Interactive Forms

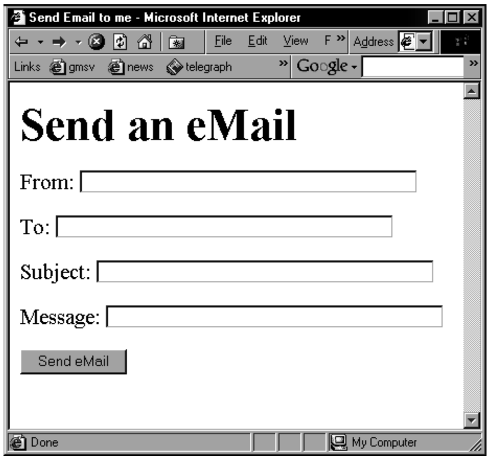
Send Email to me

To send an eMail message to me:

1. Type your eMail address in the "From" field
2. Type a short message in the "Message" field
3. Press the "Send eMail to me" button

From:

Message:


Send Email to me - Microsoft Internet Explorer
File Edit View F
Address
Links gmsv news telegraph
Google
Send eMail
Done
My Computer

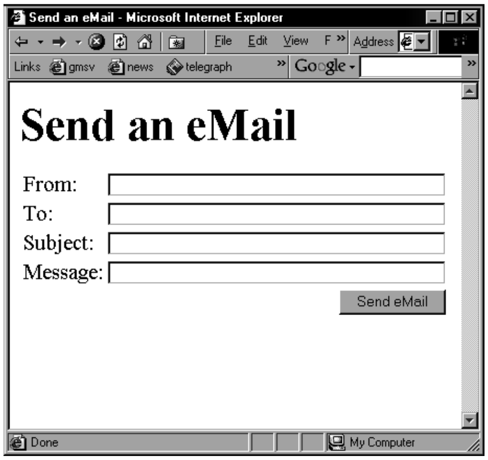
Send an eMail

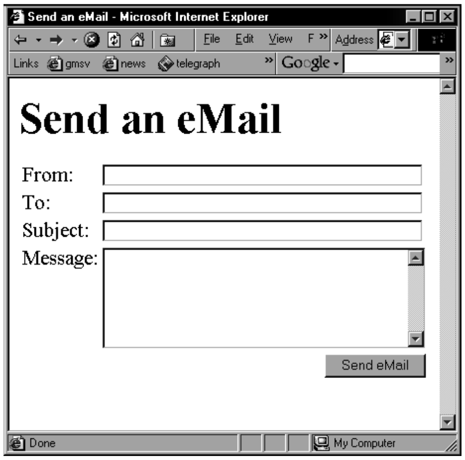
From:

To:

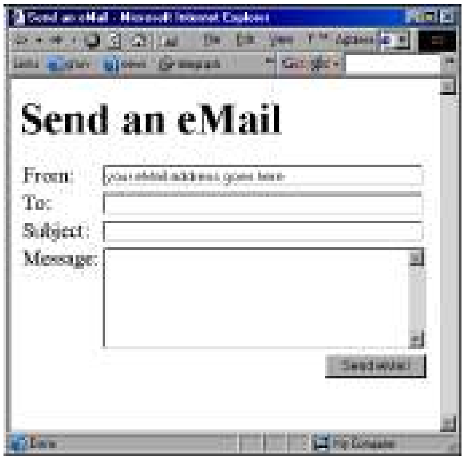
Subject:

Message:





```
<TEXTAREA
       name=“message”
       cols=“38”
       rows=“6”
>
</TEXTAREA>
<FORM name="sendEmail" method="post" action=“sendMailScriptURL">
  <table><tr>
     <td>From: </td>
     <td><INPUT type="text" name="sender" size="50"></td>
    </tr><tr>
     <td>To: </td>
     <td><INPUT type="text" name="receiver" size="50"></td>
    </tr><tr>
     <td>Subject: </td>
     <td><INPUT type="text" name="subject" size="50"></td>
    </tr><tr>
     <td valign="top">Message: </td>
     <td><TEXTAREA name="message" cols="38"rows="6">
       </TEXTAREA></td>
    </tr><tr>
    <td colspan="2" align="right">
     <INPUT type="submit" name="sendEmail" value="Send eMail">
    </td>
  </tr></table>
</FORM>
```


Send an eMail - Microsoft Internet Explorer
Send an eMail
From:
your eMail address goes here
To:
Subject:
Message:
Send eMail
Done
My Computer

```
<INPUT
        type="text"     name="sender"
        size="50"
        value="your eMail address goes here"
```

Review of the Tags Used in Forms

```
<FORM>
        name="nameOfTheForm"
        method="get" or "post"
        action="URL"
```

Elements of the form

```
</FORM>
```

Single-Line Text Input Field

```
<INPUT
        type="text"
        name="fieldName"
        size="widthInCharacters"
        maxlength="limitInCharacters"
        value="initialDefaultValue"
>
```

Hidden Input

```
<INPUT
        type="hidden"
name="fieldName"
        value="value"
>
```

Submit Button Input

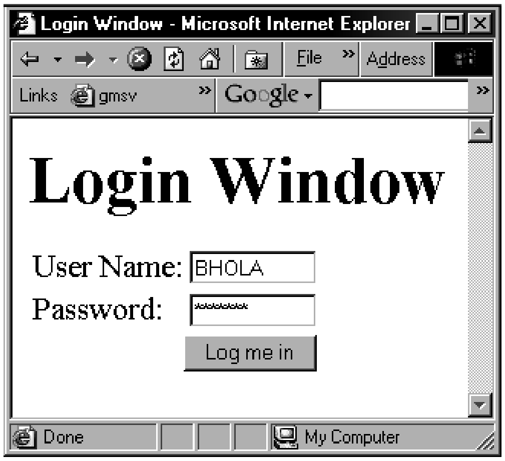
```
<INPUT
        type="submit"
name="buttonName"
        value="displayedText"
>
```

Multi-Line Text Input Area

```
<TEXTAREA
        name=“areaName”
        cols=“widthInCharacters”
        rows=“numberOfLines”
>

        initial default value

</TEXTAREA>
```

This was a review of the new tags (and associated attributes) that we have used in today’s examples. There are many more tags that can be used in a form.

Let us take a look at a few



Login Window

User Name: BHOLA

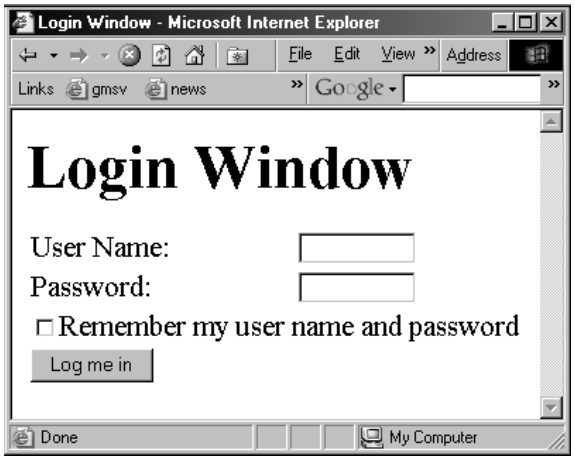
Password: *******

Log me in

```
<form name="login" method="post" action="loginScript">
  <table><tr>
      <td>User Name: </td>
      <td>
        <input type="text" name="userName" size="10">
      </td>
    </tr><tr>
      <td>Password: </td>
      <td>
        <input type="password" name="password" size="10">
      </td>
    </tr><tr>
      <td colspan="2" align="right">
        <input type="submit" name="login" value="Log me in">
      </td>
  </tr></table>
</form>
```

Password Input Field

```
<INPUT
        type=“password”
        name=“fieldName”
        size=“widthInCharacters”
        maxlength=“limitInCharacters”
        value=“initialDefaultValue”
>
```



Login Window

User Name:

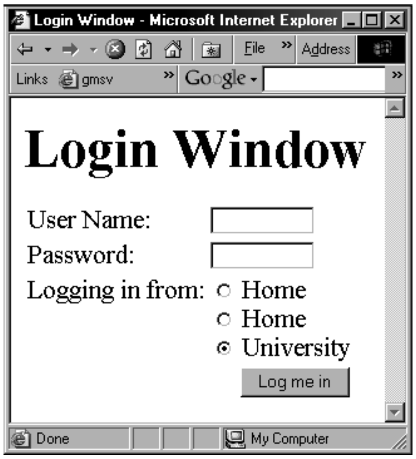
Password:

□ Remember my user name and password

```
<form name="login" method="post" action="loginScript">
 <table><tr>
     <td>User Name: </td>
     <td>
       <input type="text" name="userName" size="10">
     </td>
   </tr><tr>
     <td>Password: </td>
     <td>
       <input type="password" name="password" size="10">
     </td>
   </tr><tr>
     <td colspan="2">
       <input type="checkbox" name="remember" value="remember">
             Remember my   user name and password<br>
     </td>
         </tr><tr>
     <td colspan="2">
       <input type="submit" name="login" value="Log me in">
     </td>
 </tr></table>
</form>
```

12.2 Checkbox Input Element

```
<INPUT
        type="checkbox"
        name="checkboxName"
        checked
        value="checkedValue"
>
```



Login Window

User Name:

Password:

Logging in from: ○ Home
○ Home
● University

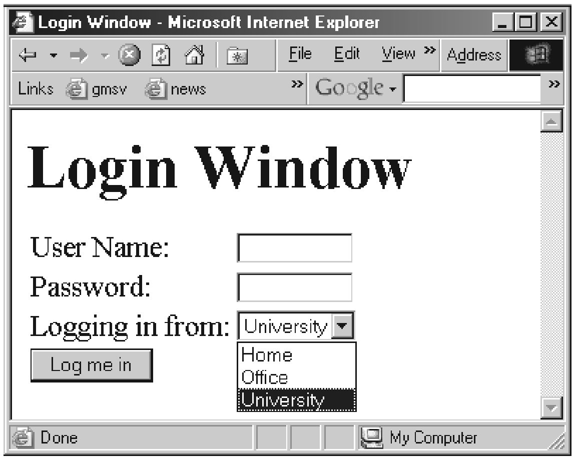
Log me in

```
<form name="login" method="post" action="loginScript">
 <table><tr>
     <td>User Name: </td>
     <td>
       <input type="text" name="userName" size="10">
     </td>
   </tr><tr>
     <td>Password: </td>
     <td>
       <input type="password" name="password" size="10">
     </td>
   </tr><tr>
     <td valign="top">Logging in from:</td>
     <td>
       <input type="radio" name="from" value="home"> Home<br>
       <input type="radio" name="from" value="office"> Home<br>
       <input type="radio" name="from" value="university" checked> University
     </td>
        </tr><tr>
     <td colspan="2" align="right">
       <input type="submit" name="login" value="Log me in">
     </td>
 </tr></table>
</form>
```

12.3 Radio Button Input Element

```
<INPUT
        type=“radio”
        name=“radioButtonName”
        checked
        value=“selectedValue”
>
```

What is the difference between checkboxes and radio buttons?



Login Window

User Name:

Password:

Logging in from: University

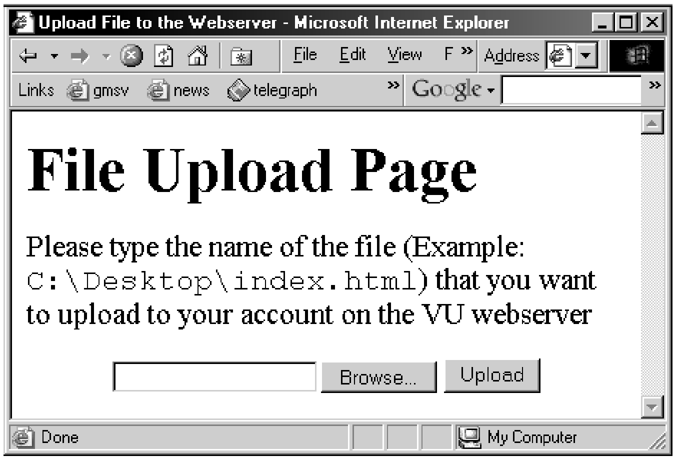
Home
Office
University

Log me in

```
<form name="login" method="post" action="loginScript">
 <table><tr>
   <td>User Name: </td>
   <td><input type="text" name="userName" size="10"></td>
  </tr><tr>
   <td>Password: </td>
   <td>
    <input type="password" name="password" size="10">
   </td>
  </tr><tr>
     <td valign="top">Logging in from:</td>
   <td>
    <select size="2" name="from">
    <option value="home"> Home</option>
    <option value="office"> Office </option>
    <option value="university" selected> University </option>
    </select>
   </td>
     </tr><tr>
   <td colspan="2">
    <input type="submit" name="login" value="Log me in">
   </td>
 </tr></table>
</form>
```

12.4 Select from a (Drop Down) List

```
<SELECT
    name="listName"
    size="numberOfDisplayedChoices"
    multiple
>
<OPTION value="value1"> text1 </OPTION>
<OPTION value="value2" selected> text2 </OPTION>
<OPTION value="value3"> text2 </OPTION>
    …
    …
</SELECT>
```


Upload File to the Webserver - Microsoft Internet Explorer
File Edit View F Address
Links gmsv news telegraph Google
Browse...
Upload
Done
My Computer

File Upload Page

Please type the name of the file (Example: C:\Desktop\index.html) that you want to upload to your account on the VU webserver

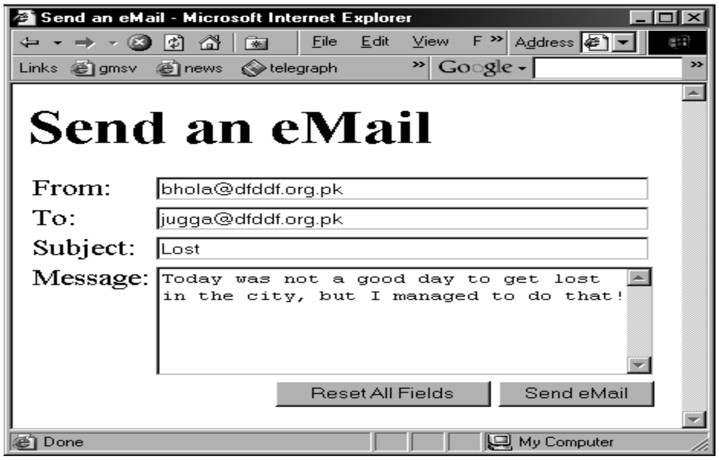
12.5 File Upload Input Element

```
<INPUT
    type="file"
    name="buttonName"
    value="nameOfSelectedFile"
    enctype="fileEncodingType"
>
<form
    name="uploadForm"
    method="post"
    action="uploadScript"
  <input
    type="file"
    name="uploadFile"
    enctype="multipart/form-data"
  >
  <input
    type="submit"
    name="submit"
    value="Upload"
   >
</form>
```

Reset Button Input Element

(Resets the contents of a form to default values)

```
<INPUT
    type="reset"
    value="dispalyedText"
>
```



Send an eMail

From: bhola@dfddf.org.pk

To: jugga@dfddf.org.pk

Subject: Lost

Message: Today was not a good day to get lost in the city, but I managed to do that!

Today's Lecture was the …

We looked at the utility of forms on Web pages
We found out about the various components that are used in a form
We became able to build a simple, interactive form

Lecture 13

Application Software

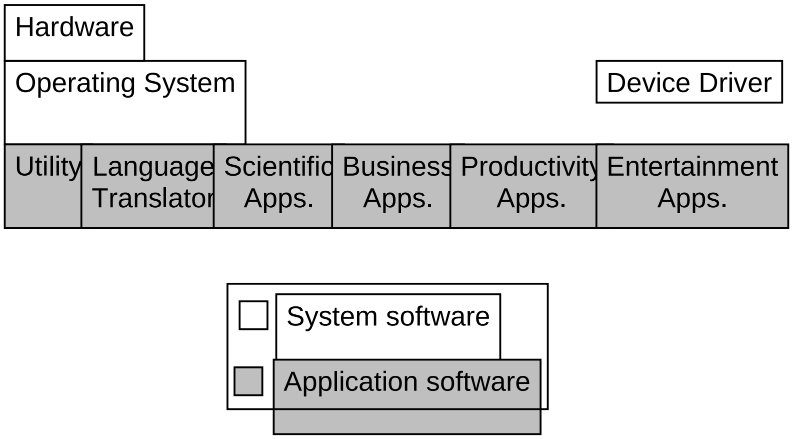
The focus of the last lecture was on Operating Systems

Learning Goals for Today

To learn about application software
To become familiar with various software used in the following application areas:
e.g.
Scientific/engineering/graphics
Business
Productivity
Entertainment
Educational

13.1 Two Major Types of Software

System Software
Application Software



13.2 Application Software

Application software are programs that interact directly with the user
They generally do not talk directly to the hardware

13.3 Classification According to the Mode

Interactive-mode
The user runs the program on the computer and keeps on interacting with the computer while the program runs
Example: Word processor
Batch-mode
The user starts the program and the computer processes the provided data and produces results without any further intervention of from the user
Example: Payroll

13.4 Classification According to Application Area

Scientific/engineering/graphics
Business
Productivity
Entertainment
Educational

13.5 Scientific/Engineering/Graphics Apps

Key feature: Intense floating-point calculations
Scientific/engineering/mathematical apps
Computers initially were designed just to run them

Generally designed for specialists
Rudimentary UI's
Many run in batch mode

13.6 Scientific SW

Simulation of natural systems
Deforestation and effect on green-house gases
Simulation of artificial systems
NeuralWare (Simulator for artificial neural networks)
Thermo-nuclear explosions
Mathematical computation packages
Mathematica (can do hundreds, if not thousands of functions, e.g. solving a differential eq, symbolically)
MathCAD

13.7 Engineering SW

Computer-aided design (CAD)
AutoCAD
SPICE
Virtual wind tunnels
Computer-aided manufacturing (CAM)
Telecommunication system SW
Centrex
Industrial control SW

13.8 Graphics & Animation SW (1)

Moving graphics e.g. cartoons

Two types:

1. Vector graphics

Treats everything that is drawn as an object
Objects retain their identity after they are drawn
These objects can later be easily moved, stretched, duplicated, deleted, etc
Are resolution independent
Relatively small file size
Example: MS Visio, Corel Draw, Flash

Graphics & Animation SW (2)

2. Bit-mapped or raster graphics

Treats everything that is drawn as a bit-map
If an object is drawn on top of another, it is difficult to move just 1 of them while leaving the other untouched
Changing the resolution often requires considerable touch-up work
Relatively large file size
Example: MS Paint, Adobe Photoshop

13.9 Business Applications

Most of the SW being developed today belongs to this category
SW that is required to run most any sort of biz:
Payroll
General ledger
Order entry
Accounts receivable & accounts payable
Inventory control
Let's now discuss a few business SW categories which are not that common, but are becoming more and more popular with time

13.10 E-Commerce Software

Key requirements:
Reliability
Security

Ability to handle 1000's of transactions, simultaneously

13.11 ERP (Enterprise Resource Planning) SW

Very large scale, complex & expensive SW

Ties together all key activities & major systems of an organization into a single SW system

Key benefit: Optimization of the business processes of an organization as a single system instead of many loosely-related stand-alone systems

Example: SAP, Oracle, PeopleSoft, Baan

13.12 DSS (Decision Support Systems) SW

Sometimes also called "expert systems"

Many times are based on a branch of computer science called "artificial intelligence"

This category of SW is designed to help business managers in making effective decisions in complex situations based on the analysis of the relevant data

13.13 Productivity SW

Most popular category in terms of licenses sold

Common features

Ability to simplify, automate everyday business tasks

Highly interactive and user-friendly design

Designed to run on PC's

Most users do not use 90% of the SW features

Popular productivity SW

Word Processing -- Spreadsheets

Presentations -- Databases

13.14 Word Processors

Probably the most popular productivity app

Initially designed as a replacement for the typewriter

Automation

Automatic end-of-line soft carriage return

Style sheets

Table of contents & index

Spelling & grammar checking

Two approaches: WYSIWYG (e.g. Word, WordPerfect, Star) or traditional markup (LaTeX)?

Desktop publishing

13.15 Web Page Development SW

Web pages can be developed using a simple plain-text editor like the "notepad", but more efficient, easy-to-use HTML editors can make the process quicker

Some of them are WYSIWYG (i.e. you don't really need to know any HTML to use them), others are not, while some provide both types of interfaces (DreamWeaver)

Most popular word processors now come with a built-in Web page development facility

13.16 Spreadsheet SW (1)

Electronic replacement for ledgers

Is used for automating engineering, scientific, but in majority of cases, business calculations

A spreadsheet - VisiCalc - was the first popular application on PC's.

It helped in popularizing PC's by making the task of financial-forecasting much simpler, allowing individuals to do forecasts which previously were performed by a whole team of financial wizard

13.17 Spreadsheet SW (2)

Consist of cells arranged in rows and columns

Each cell may contain numeric values, text or formulas

Automation

Recalculations

Charts

13.18 Presentation Development SW

Used to prepare multimedia material for lectures & presentations to display key points, graphics, animation, or video with the help of multimedia projectors
Have replaced acetate films (slides) that were used with over-head projectors
Key advantage over acetate slides:
Easy to modify
Can be sent electronically
Its multimedia nature makes it more interesting for the audience

13.19 Small-Scale Databases SW (1)

Easy to use applications designed for efficient storage and fast and easy retrieval of data
That data may be in the form of numbers, text, or even multimedia, i.e. sounds, graphics, animation, video

13.20 Small-Scale Databases SW (2)

Before the advent of the currently popular "relational" database model, the databasing function was performed using what is called the "flat-file" model
That model is not very efficient for storing and searching in large databases
A database consists of a file or a set of files. Information in these is stored in the form of records, and the records are further subdivided into fields

13.21 Productivity SW Suites

A set of stand-alone productivity applications designed to work easily with each other
Share a common UI
Are available as a bundle along with additional useful utilities
Examples: MS Office, Corel WordPerfect Office, Lotus SmartSuite, Star Office
SW Suites for other app areas are available as well, e.g. the Adobe suite of graphics apps

13.22 Document-Centered Computing (DCC) - 1

The increasing cooperation among the apps included in productivity suites has given rise to a new computing paradigm called DCC
DCC implies that instead of developing parts of a doc in a number of apps, and then cutting-&-pasting them to form the final doc, you stay in a single doc and call-up appropriate apps to insert the required objects

13.23 Document-Centered Computing (DCC) - 2

Let's say that we want to write a letter containing a map, a table and a graph
We can:
Launch the WP and type the text in
Insert a drawing by calling up the drawing toolbar app (without leaving the WP) & draw the map
Insert a table by calling up the spreadsheet app (without leaving the WP) & build the table
Insert a graph based on that table using the same spreadsheet app without leaving the WP

13.24 Entertainment SW

Key feature: Simple, intuitive, many times social UI's
The user is generally assumed to know nothing about computers
Both Microsoft & Apple are pursuing a PC-as-a-personal-entertainment-hub strategy. Probable result: Already popular entertainment SW will become even more popular

13.25 Music & Video Players

Music players (WinAmp)
Video/Music players (Real player, Windows Media player, QuickTime player)
The Web Browsers can also display video, animation, and play music with the help of helper applications like Flash

13.26 Music Generation & Movie Editing SW

A PC can be made the hub of a music making studio with help of appropriate HW & SW
Inexpensive, easy-to-use video editing SW has recently become available for the iMac

13.27 Games

Many types

Educational (especially for toddlers)
Strategy/Simulation
Sports
Shoot'em ups
The saddest aspect: You do not need any opponents or partners to play computer games
The application SW category that provides the toughest challenge for computer HW

13.28 Educational SW

Category with probably the highest growth rate
Current focus on augmenting traditional training and education methods, but it is shifting towards replacing traditional methods

13.29 Electronic Encyclopedias

Great resource of useful information presented in a very interesting format
Superior to the paper-based version because:
Access speed is dramatically higher
Can contain animation and sound
Much lower cost as thousand's of pages in dozens of volumes have been replaced by a couple of CD's

13.30 On-Line Learning

With time, the VU Web site will become more and more focused on interactive online learning
The Website of our textbook "Understanding Computers" is an example of an on-line learning Website
Key features of good online learning SW:
The student can learns at his or her own pace
The student can select his or her own hours

13.31 Interactive CD's

Same as on-line learning, but through a CD instead of a Web site
Key advantage:
Ideal for students with slow Internet access

13.32 Attributes of Good Application Software

Easy to install, un-install
User Interface
Consistent
Intuitive
Configurable
Adapts to the users need
Has a tutorial and a complete help manual
Does not have any critical bugs

13.33 Most Popular Application Software Categories

Web browsers
Email clients
Word processors

What have we learnt today?

Application software are programs that interact directly with the user for the performance of a certain type of work
That work generally falls into one of the following usage areas
Scientific/engineering/graphics
Business
Productivity
Entertainment
Educational

Focus of the Next Lecture

Next lecture will be the first among the four lectures that we plan to have on productivity SW

That first lecture will be on word processing
We'll learn about what we mean by word processing
We'll discuss the usage of various functions provided by common word processors

Lecture 14
Word Processing

Focus of the last lecture was on Application SW

Application SW are programs that interact directly with the user for the performance of a certain type of work
That work generally falls into one of the following usage areas
Scientific/engineering/graphics
Business
Productivity
Entertainment
Educational

Today's Lecture

First among the four lectures that we plan to have on productivity software, a sub-category of application software
This first lecture will be on word processing
We'll learn about what we mean by word processing and also desktop publishing
We'll discuss the usage of various functions provided by common word processors

Word Processing

The art and science of converting written information into a form that looks pleasing when printed
One of the most popular activities on the PC

14.1 Word Processor

The tool used to perform word processing
Long time ago, a word processor was a HW/SW combination used solely for performing the word processing task. It looked like a computer terminal or a PC, but could do only one task – word processing
Today, the term "word processor" generally means the SW used on a computer to perform the task of word processing

Uses of Word Processors

Write a letter
Address labels
Research paper or report
Advertisement
Newsletter
Magazines
Book
And thousands of other tasks

Common Features

Type, cut, copy, paste, move text
Automatic line-breaks
Change font type, face, size, color
Change number of columns
Adjust margins and line, word, letter spacing
Have running headers, footers, page nos.
Insert tables, charts, graphics, drawings

Evolution of WP's

Manual & electric typewriters (1930-1960)
Were page oriented
Type face/size was changed by replacing the typing ball
Typewriters with magnetic storage (1960's)
IBM added storage capability using magnetic tape

Line editors on computers(1960's)
Stand alone word processors (1960's-1970's)
cost: $15,000 to 20,000
Current WP programs on uCs (1980's onwards)

14.2 Types: WYSIWYG-based & Markup-based

All early WP's and some of the modern ones as well are markup-based: similar to HTML
Generally are harder to learn, but may provide better control and smaller file size
Example: LaTeX
Most current PC-based WP's belong to the WYSIWYG category
Easy to get started due to the WIMP interface
Example: MS Word, Corel WordPerfect, Sun Star

14.3 Desktop Publishing (DTP)

A combination of word processing and graphic design. Used to develop elegant documents
In the olden times, DTP was used for designing magazines, newspapers & other professional-looking items
These days, because of the low cost of DTP SW, it is being used for less-demanding and ordinary tasks as well
The original Macintosh PC started the era of DTP or "Personal Publishing" in 1984

DTP –vs– WP

The difference between the two is diminishing with time
Most WP's now include many tools that, not long ago, were found only in DTP SW
Generally, DTP SW is a bit more difficult to use for us common computer users, whereas WP SW is quite user-friendly
DTP SW generally provides finer control over the design/layout of a document

DTP: Requirements

High-end PC with a large-screen monitor
Laser printer
Scanner
DTP SW
Examples:
Adobe PageMaker
QuarkXPress
Corel Ventura
MS Publisher

14.4 Word Processors for the Web

Most common WP's and DTP packages now have the Web development ability
They also include features like auto-recognition of eMail addresses and URL's
However, specialized SW just for developing Web pages and sites is also available
Examples: DreamWeaver, FrontPage

The right font face & size for normal text

If text is too small, it becomes hard to read
Too large, wastage of space is the result. Plus the reader has to turn more pages than necessary
Either way, the reader gets annoyed
For general WP, 10-12 point size works well
Most users, either use the Times New Roman or Arial/Helvetica type face

Bold, Italic, Underlined Text

Bold – **fat**
Italic – *slanted* (Why the name italic?)
Underlined
All used to emphasize a certain segment of text
Plea:

Please do not over-do them
Their over-use makes it very difficult for the reader
And please, use one at a time: Text that is no only bold but also italic & underlined looks +ively ***awful***

Select, Cut, Copy, Drag, Paste

Just select and cut or copy or drag
Can also paste after a cut or a copy
Just think about the pain that people suffered before the advent of the modern WP's
Movement of a single sentence from one page to another would have required re-doing all the pages in between

Spelling & Grammar

Grammar checkers are not very helpful yet, but still useful and are improving with time
Warning: Spell checkers are not all that smart! Use them with care.
Disadvantage: My spelling ability is deteriorating day-by-day because of over-reliance on WP spell-checkers. I am having great difficulty in writing even short-ish hand-written notes without spelling errors

Thesaurus

My favorite tool
Helps you find synonyms and, sometimes, antonyms as well

Tables

Tables are sometimes useful for presenting info in an ordered fashion
Most WP's provide extensive table construction & manipulation features

Graphics & Drawings

You can insert graphics that are made using other apps into a WP document
Several WP's have a built-in drawing tool, which can be used for adding simple diagrams (e.g. a flow chart, a simple street map) into a WP document

The Best Feature: Undo

Allows you to recover from your mistakes
Allows you to experiment without risk

Document View Mode

Most WP's provide several ways of viewing a document
I normally work in and recommend what is known as the "Print Layout" view mode
In this view, the WP works in a true WYSIWYG mode

Print-Preview & Printing

Make sure to preview your document before printing it
Do this to make sure about the"look" of the document before it is printed
Most people these days either use inkjet printers or laser printers
Color inkjet printers cost less but are slower
B&W laser printers cost around twice as much, but are faster and generally have finer resolution
Color laser printers are expensive

Automation

Table of contents
TOC can be automatically generated
Page nos. in the TOC get readjusted automatically
Index
Can be automatically generated
Page nos. in the index get readjusted automatically
Application of predefined styles
Change style; text changes automatically throughout the doc
Headers & Footers
Page numbers
Spelling error auto-highlight

Getting On-Screen Help

All WP's generally have some form of built-in help mechanism

To me, it seems like that many of those help-systems are designed to be "not-very-helpful": they make finding answers to simple questions quite difficult

Nevertheless, do try them when you are searching for answers

14.5 Let's try to use MS Word for creating a CV

(Remember the TOC)

Today's Lecture was the …

First among the four lectures that we plan to have on productivity software, a sub-category of application software

This first lecture was on word processing

We learnt about what we mean by word processing and also desktop publishing

We also discussed the usage of various functions provided by common word processors

Focus of the Next Lecture: Algorithms

To become familiar with the concept of algorithms

What they are?

What is there use?

To become able to write algorithms for simple problems

Lecture 15

More on Interactive Forms
(Web Development Lecture 5)

Focus of the last lecture was on Interactive Forms

We looked at the utility of forms on Web pages

We found out about the various components that are used in a form

We became able to build a simple, interactive form

In Today's Lecture …

We will learn ways of adding more interactivity to forms

We will get our first taste of JavaScript – the object-based language that we will be employing throughout the rest of the Web development part of this course

Last time we mentioned server-side scripts; today we will write (simple) client-side scripts in JavaScript

15.1 Single-Line Text Input Field

```
<INPUT
    type=“text”
    name=“name”
    size=“widthInCharacters”
    maxlength=“limitInCharacters”
    value=“initialDefaultValue”
>
```

15.2 Password Input Field

```
<INPUT
    type=“password”
    name=“name”
    size=“widthInCharacters”
    maxlength=“limitInCharacters”
    value=“initialDefaultValue”
>
```

15.3 Hidden Input

```
<INPUT
    type=“hidden”name=“name”
    value=“value”
>
```

15.4 Checkbox Input Element

```
<INPUT
    type=“checkbox”
    name=“name”
    checked
    value=“checkedValue”
>
```

15.5 Radio Button Input Element

```
<INPUT
    type=“radio”
    name=“name”
    checked
    value=“selectedValue”
>
```

15.6 File Upload Input Element

```
<INPUT
    type=“file”
    name=“name”
    value=“nameOfSelectedFile”
```

```
	enctype=“fileEncodingType”
>
```

15.7 Reset Button Input Element

```
<INPUT
	type=“reset”
	value=“buttonLabel”
>
```

15.8 Submit Button Input

```
<INPUT
	type=“submit”			name=“name”
	value=“buttonLabel”
>
```

8 Possible Values for the “type” Attribute of <INPUT> tag

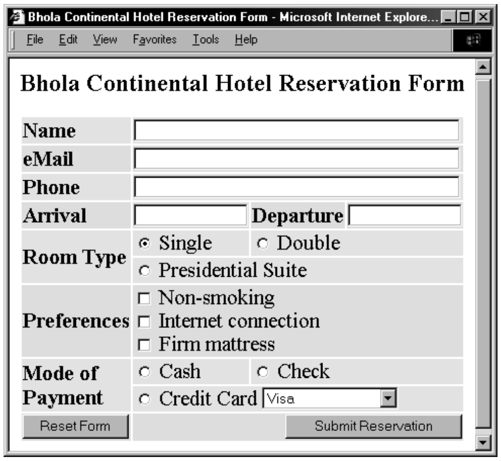
text
password
hidden
checkbox
radio
file
reset
submit

15.9 Multi-Line Text Input Area

```
<TEXTAREA
	name=“areaName”
	cols=“width”
	rows=“lines”
>
	initial default value
</TEXTAREA>
```

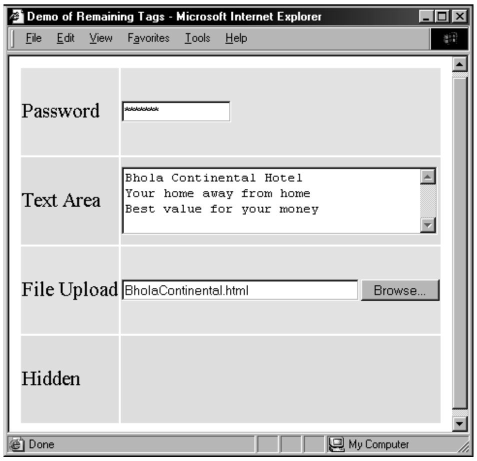
15.10 Select from a (Drop Down) List

```
<SELECT
	name=“name”
	size=“numberOfDisplayedChoices”
	multiple
>
<OPTION value=“value1”> text1
<OPTION value=“value2” selected> text2
<OPTION value=“value3”> text2
	…
	…
</SELECT>
```


Bhola Continental Hotel Reservation Form - Microsoft Internet Explore...
File Edit View Favorites Tools Help
Reset Form
Submit Reservation

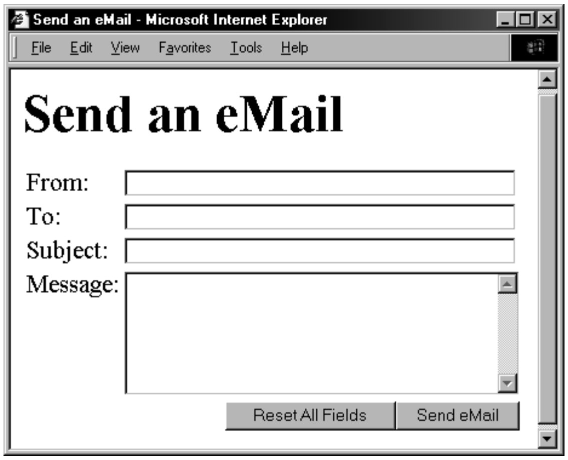
Bhola Continental Hotel Reservation Form

Name			
eMail			
Phone			
Arrival		Departure	
Room Type	● Single	○ Double	
	○ Presidential Suite		
Preferences	□ Non-smoking □ Internet connection □ Firm mattress		
Mode of Payment	○ Cash	○ Check	
	○ Credit Card	Visa	


Demo of Remaining Tags - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Browse...
Done
My Computer

Password	*******
Text Area	Bhola Continental Hotel Your home away from home Best value for your money
File Upload	BholaContinental.html
Hidden	

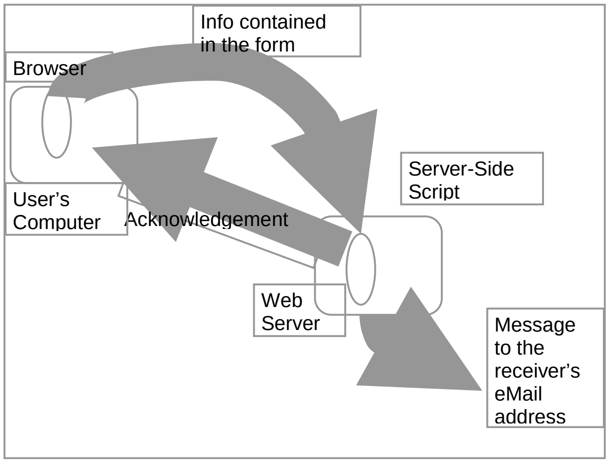
End of the Review of Tags Used in Forms
Now let's take a look at a form that we constructed last time, and see how we can make it better


Send an eMail - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Reset All Fields
Send eMail

Send an eMail

From:
To:
Subject:
Message:

Let's now review what happens when I enter all the required info and press the "Send eMail" button?


Info contained in the form
Browser
User's Computer
Acknowledgement
Server-Side Script
Web Server
Message to the receiver's eMail address

This is what happens when the form is filled correctly. What if the form is filled **incorrectly?**

What if the users leaves one of the essential fields, blank?

What if the user enters an illegal eMail address? Examples:

altaf2vu.edu.pk

bhola@hotmail.con

bhola@yahoo

A Reasonable Scenario

When the "Send eMail" button is clicked, the browser sends the data collected through the form to a script running on the Web server

That server-side script:

receives that data

analyzes it

discovers the missing or incorrect data

sends a message back to the user's browser stating the problem and asks the user to re-send

This process …

That is the process of user:

Filling the incomplete/incorrect data

Sending it to the server

Receiving the response back from the server

Correcting and resending

is quite time-consuming and uses the server's resources to help the user correct his mistakes

It can really bog down the server if a large number of users are using that Web server

15.11 Client-Side Scripting is a viable alternate

In this technique, one uses the user's browser to checking the form data

If data is missing or is incorrect, the browser can prompt the user to take corrective action

This way, the form data is sent to the server-side script only after it has been established that the collected data is complete and correct

15.12 Server-Side Scripts: Review

Are programs that reside on Web servers

Receive info that a user enters in a form

Process that info and take appropriate action

Examples:

CGI scripts on Unix servers

ASP scripts on Windows servers

New Concept: Client-Side Scripts

Small programs that are a part of the Web page and run on the user's (client's) computer

They interact with the user to collect info or to accomplish other tasks

Once it has been collected, they may help pass the collected info on to a server-side script

We'll use JavaScript to do client-side scripting. However, there are many other languages that can be used for that purpose, e.g. VBScript

Advantages of Client-Side Scripting

Reduced server load as it does not have to send messages to the user's browser about missing or incorrect data

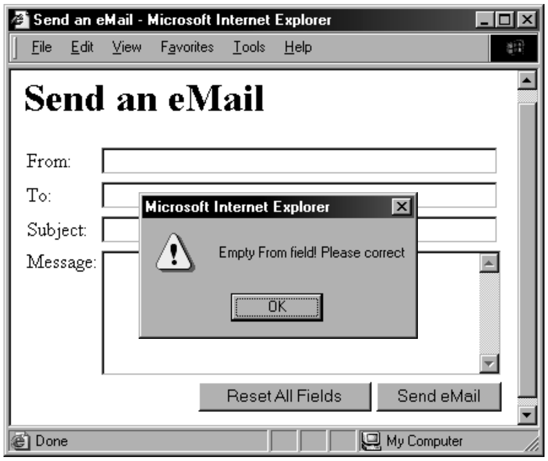
Reduced network traffic as the form's data is sent only once instead of many to's and fro's

Disadvantages

Client-side scripts do not work with all browsers

Some user intentionally turn scripting off on their browsers

This increases the complexity of the Web page, as it now has to support both situations: browsers with scripting capability, and those not having that capability



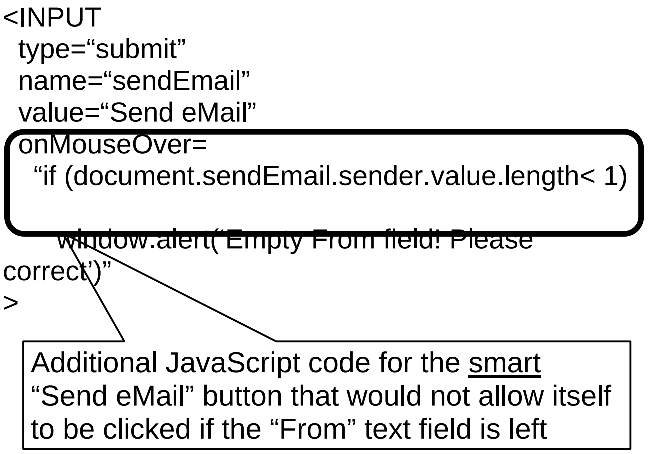
Send an eMail

Microsoft Internet Explorer

Empty From field! Please correct

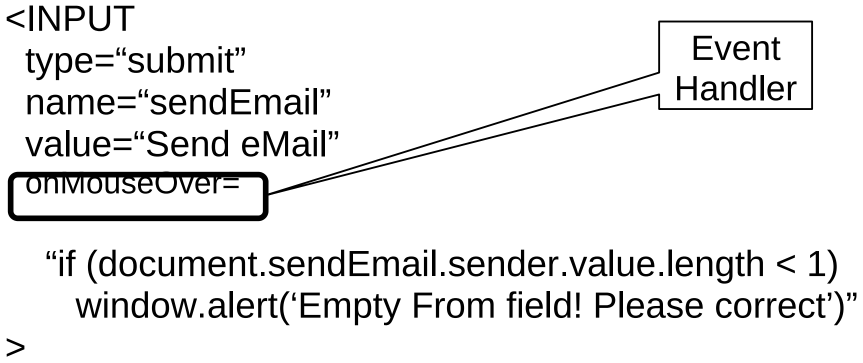
```
<INPUT
  type="submit"
  name="sendEmail"
  value="Send eMail"
>
```

Code for the simple "Send eMail" button as was described during the last Web development lecture



```
<INPUT
  type="submit"
  name="sendEmail"
  value="Send eMail"
  onMouseOver=
    "if (document.sendEmail.sender.value.length< 1)

      window.alert('Empty From field! Please
correct')"
>
```

Additional JavaScript code for the smart "Send eMail" button that would not allow itself to be clicked if the "From" text field is left



```
<INPUT
  type="submit"
  name="sendEmail"
  value="Send eMail"
  onMouseOver=

    "if (document.sendEmail.sender.value.length < 1)
      window.alert('Empty From field! Please correct')"
>
```

This is one way of including JavaScript code in an HTML document – that is, including a short JavaScript segment as part of an HTML tag

There are a few others as well. Let's now find out about another.

But before we do that …

… let's just make clear why we are interested in including JavaScript in our Web pages

15.13 Why JavaScript?

HTML is great for static Web pages; however, supports only rudimentary interactivity through forms and hyperlinks

JavaScript can be used (along with HTML) to develop interactive content for the Web

What is JavaScript?

A programming language specifically designed to work with Web browsers

It is designed to be used for developing small programs – called scripts – that can be embedded in HTML Web pages

JavaScript:

Is an interpreted language

Supports event-driven programming

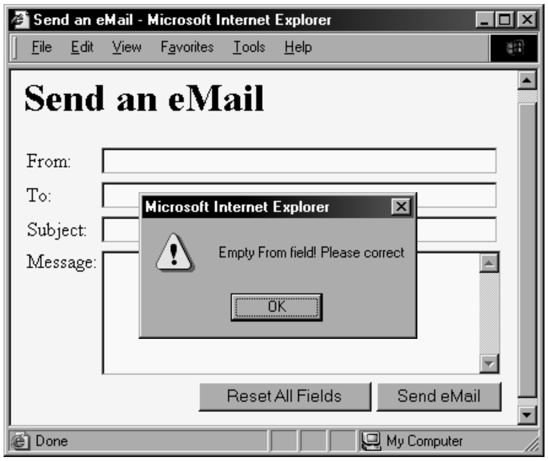
Is object-based

Object Based?

Everything that JavaScript manipulates, it treats as an *object* – e.g. a window or a button

An object has *properties* – e.g. a window has size, position, status, etc.

Properties are modified with *methods* – e.g. a resize a window with *resizeTo(150, 200)*


Send an eMail - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Send an eMail
From:
To:
Subject:
Message:
Microsoft Internet Explorer
Empty From field! Please correct
OK
Reset All Fields
Send eMail
Done
My Computer

```
<INPUT
 type="submit"
 name="sendEmail"
 value="Send eMail"
 onMouseOver=
   "if (document.sendEmail.sender.value.length < 1)
      window.alert('Empty From field! Please correct')"
>
<INPUT
 type="submit"
 name="sendEmail"
 value="Send eMail"
 onMouseOver="checkForm()"
>
<INPUT
 type="submit"
 name="sendEmail"
 value="Send eMail"
 onMouseOver=
   "if (document.sendEmail.sender.value.length < 1)
      window.alert('Empty From field! Please correct')"
>
```

checkForm()

JavaScript understands onMouseOver – it is one of the pre-defined keywords in JavaScript

However, the case for checkForm() is different

A checkForm() function does not exist in JavaScript. Therefore, we will have to define it ourselves

It can either be defined in the HEAD portion or BODY portion. We will do it in the HEAD.

```
<HTML>
<HEAD>
<TITLE>Send an eMail</TITLE>
<SCRIPT>
function checkForm() {
   if ( document.sendEmail.sender.value.length < 1)  {
      window.alert( “Empty From field! Please correct” );
}
}
</SCRIPT>
</HEAD>
<BODY bgcolor=“#FFFFCC”>
   … body content …
</BODY>
</HTML>
```

JavaScript code enclosed in the new <SCRIPT>,</SCRIPT> HTML tags

We have looked at 2 techniques for embedding JavaScript code in a Web page:
1. Put the code in the tag for the “Send eMail” button - as was shown to you earlier
2. Put the checkForm() code in the HEAD portion & put the onMouseOver=“checkForm()” attribute in the tag for the “Send eMail” button
Both perform the required function satisfactorily.
Q: *Which of two techniques is better?*
The “put all code in the tag” technique seems to require less code
For very short scripts, “all code in the tag” works well. However, this technique does not work when one needs to put multiple script statements in the same tag
The “code in the HEAD portion” is more general-purpose, and the right choice for developing larger JavaScript scripts
The main code segment that goes between the <SCRIPT>, </SCRIPT> tags in the HEAD:

```
function checkForm() {
  if ( document.sendEmail.sender.value.length < 1)  {
    window.alert( “Empty From field! Please correct” );
  }
}
```

The JavaScript code included as an attribute of the “Send eMail” button:
onMouseOver=“checkForm()”
Today we checked if the “From” field of the form was empty
How can we modify the JavaScript code for checking if the “To” field is empty as well?
How about checking all four fields?
How about checking if the addresses given in the “From” and “To” fields are legal eMail addresses?
Please try thinking about those possibilities?

In Today’s Lecture …

We learnt ways of constructing forms that were a bit more interactive
We got our first taste of JavaScript – the object-based language that we will be employing throughout the rest of the Web development part of this course
Last time we mentioned server-side scripts; today we wrote (simple) client-side scripts in JavaScript

Next (the 6th) Web Dev Lecture:

JavaScript Object, Properties, Methods

We will have a more formal introduction to JavaScript and client-side scripting
We will become able to appreciate the concept of objects in JavaScript
We will learn about the properties of those objects
We will become able to perform simple tasks through the application of methods

Lecture 16
Algorithms

Focus of the last lecture was on Word Processing

First among the four lectures that we plan to have on productivity software, a sub-category of application software
That first lecture was on WP
We learnt about what we mean by WP and also desktop publishing
We also discussed the usage of various functions provided by common WP's

The Objective of Today's Lecture

To become familiar with the concept of algorithms:
What they are?
What is their use?
What do they consist of?
What are the techniques used for representing them?

Solving Problems (1)

When faced with a problem:
We first clearly define the problem
Think of possible solutions
Select the one that we think is the best under the prevailing circumstances
And then apply that solution
If the solution woks as desired, fine; else we go back to step 2

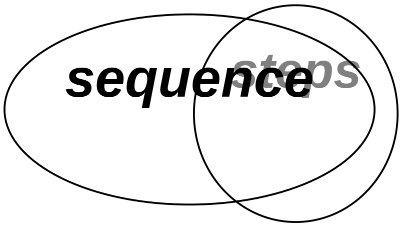
Solving Problems (2)

It is quite common to first solve a problem for a particular case
Then for another
And, possibly another
And watch for patterns and trends that emerge
And to use the knowledge form those patterns and trends in coming up with a general solution

Solving Problems (3)

It helps if we have experienced that problem or similar ones before
Generally, there are many ways of solving a given problem; the best problem-solvers come-up with the most appropriate solution more often than not!
The process that can be used to solve a problem is termed as the "algorithm"

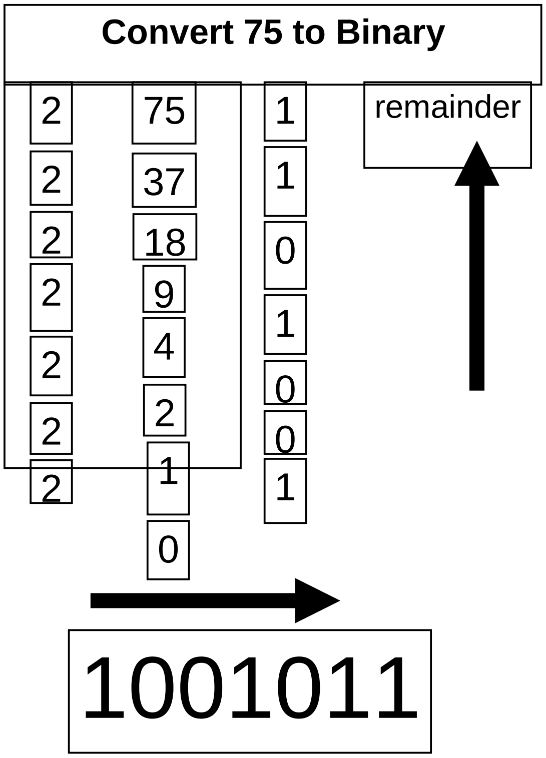
Algorithm:
***Sequence of steps* that can be taken to solve a given problem**



Examples

Addition
Conversion from decimal to binary
The process of boiling an egg
The process of mailing a letter
Sorting
Searching

Let us write down the algorithm for a problem that is familiar to us
Converting a decimal number into binary



16.1 Algorithm for Decimal-to-Binary Conversion

Write the decimal number
Divide by 2; write quotient and remainder
Repeat step 2 on the quotient; keep on repeating until the quotient becomes zero
Write all remainder digits in the reverse order (last remainder first) to form the final result

Points to Note:

The process consists of repeated application of simple steps
All steps are unambiguous (clearly defined)
We are capable of doing all those steps
Only a limited no. of steps needs to be taken
Once all those steps are taken according to the prescribed sequence, the required result will be found
Moreover, the process will stop at that point

16.2 Algorithm (Better Definition)

1st Definition:

Sequence of steps that can be taken to solve a problem

Better Definition:

A *precise sequence* of a *limited number* of *unambiguous, executable steps* that *terminates* in the form of a *solution*

Three Requirements:

Sequence is:
Precise
Consists of a limited number of steps
Each step is:
Unambiguous
Executable
The sequence of steps terminates in the form of a solution

16.3 Why Algorithms are Useful?

Once we find an algorithm for solving a problem, we do not need to re-discover it the next time we are faced with that problem

Once an algorithm is known, the task of solving the problem reduces to following (almost blindly and without thinking) the instructions precisely

All the knowledge required for solving the problem is present in the algorithm

Why Write an Algorithm Down?

For your own use in the future, so that you don't have to spend the time for rethinking it

Written form is easier to modify and improve

Makes it easy when explaining the process to others

16.4 Analysis of Algorithms

Analysis in the context of algorithms is concerned with predicting the resources that re requires:

Computational time

Memory

Bandwidth

Logic functions

However, Time – generally measured in terms of the number of steps required to execute an algorithm - is the resource of most interest

By analyzing several candidate algorithms, the most efficient one(s) can be identified

Selecting Among Algorithms

When choosing among competing, successful solutions to a problem, choose the one which is the least complex

This principle is called the "Ockham's Razor," after William of Ockham - famous 13-th century English philosopher

Early History:

Search for a Generic Algorithm

The study of algorithms began with mathematicians and was a significant area of work in the early years

The goal of those early studies was to find a single, general algorithm that could solve all problems of a single type

Origin of the Term "Algorithm"

The name derives from the title of a Latin book: Algoritmi de numero Indorum

That book was a translation of an Arabic book: Al-Khwarizmi Concerning the Hindu Art of Reckoning

That book was written by the famous 9-th century Muslim mathematician, Muhammad ibn Musa al-Khwarizmi

16.5 Al-Khwarzmi

Al-Khwarizmi lived in Baghdad, where he worked at the Dar al-Hikma

Dar al-Hikma acquired and translated books on science and philosophy, particularly those in Greek, as well as publishing original research

The word Algebra has its origins in the title of another Latin book which was a translation of yet another book written by Al-Khwarzmi:

Kitab al-Mukhtasar fi Hisab al-Jabr wa'l-Muqabala

Al-Khwarizmi's Golden Principle

All complex problems can be and must be solved
using the following simple steps:

Break down the problem into small, simple sub-problems

Arrange the sub-problems in such an order that each of them can be solved without effecting any other

Solve them separately, in the correct order

Combine the solutions of the sub-problems to form the solution of the original problem
That was some info on history.
Now, let us to take a look at several types of algorithms & algorithmic strategies

16.6 Greedy Algorithm

An algorithm that always takes the best immediate, or local solution while finding an answer
Greedy algorithms may find the overall or globally optimal solution for some optimization problems, but may find less-than-optimal solutions for some instances of other problems
KEY ADVANTAGE: Greedy algorithms are usually faster, since they don't consider the details of possible alternatives

Greedy Algorithm: Counter Example

During one of the international cricket tournaments, one of the teams intentionally lost a match, so that they could qualify for the next round
If they had won that particular match, some other team would have qualified
This is an example of a non-greedy algorithm

Greedy Algorithm: Example

A skier skiing downhill on a mountain wants to get to the bottom as quickly as possible
What sort of an algorithm should the skier be using?
The greedy-algorithm approach will be to always have the skies pointed towards the largest downhill slope (*dy*/*dx*), at all times
What is the problem with that approach?
In what situations that will be the best algorithm?
In which situations would it perform poorly?

16.7 Deterministic Algorithm (1)

An algorithm whose behavior can be completely predicted from the inputs
That is, each time a certain set of input is presented, the algorithm gives the same results as any other time the set of input is presented.

16.8 Randomized Algorithm (1)

Any algorithm whose behavior is not only determined by the input, but also values produced by a random number generator
These algorithms are often simpler and more efficient than deterministic algorithms for the same problem
Simpler algorithms have the advantages of being easier to analyze and implement.

16.9 Randomized Algorithm (2)

These algorithm work for all practical purposes but have a theoretical chance of being wrong:
Either in the form of incorrect results
Or in the form of impractically long running time
Example: Monte Carlo algorithms.

16.10 Deterministic Algorithm (2)

There can be degrees of deterministic behavior: an algorithm that also uses a random number generator might not be considered deterministic
However, if the "random numbers" come from a pseudo-random number generator, the behavior may be deterministic
Most computing environments offer a “pseudo random number generators,” therefore, most randomized algorithms, in practice, behave deterministically!

16.11 Heuristic

A procedure that usually, but not always, works or that gives nearly the right answer

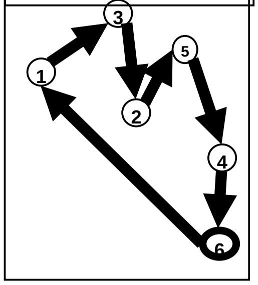
Some problems, such as the traveling salesman problem, take far too long to compute an exact, optimal solution. A few good heuristics have been devised that are fast and find a near-optimal solution more often than not
Is a heuristic, an algorithm? Yes? No? Why?

The Traveling Salesman Problem

A salesman needs to visit each of the *n* cities one after the other and wants to finish the trip where it was started

Determine the sequence of cities such that the traveling distance is minimized

A possible sequence for *n* = 6



A Few Questions

Is that the best possible sequence?
How do you know?
How do I determine the best sequence?

16.12 The Brute Force Strategy (1)

A strategy in which all possible combinations are examined and the best among them is selected
What is the problem with this approach?
A: Doesn't scale well with the size of the problem
How many possible city sequences for n=6? For n=60? For n=600?

16.13 The Brute Force Strategy (2)

However, with the relentless increase in computing power, certain problems that – only a few years ago - were impossible to solve with brute force, are now solvable with this technique

16.14 A Selection of Algorithmic Application Areas

Search
Sort
Cryptography
Parallel
Numeric
Graphical
Quantum computing
Combinatory
We'll now talk about the various ways of representing algorithms. But, before we do that please allow me to say a few words about …

Syntax & Semantics

An algo. is “correct” if its:

- Semantics are correct
- Syntax is correct

Semantics:
The concept embedded in an algorithm (the soul!)

Syntax:
The actual representation of an algorithm (the body!)

WARNINGS:

1. An algo. can be syntactically correct, yet semantically incorrect – very dangerous situation!

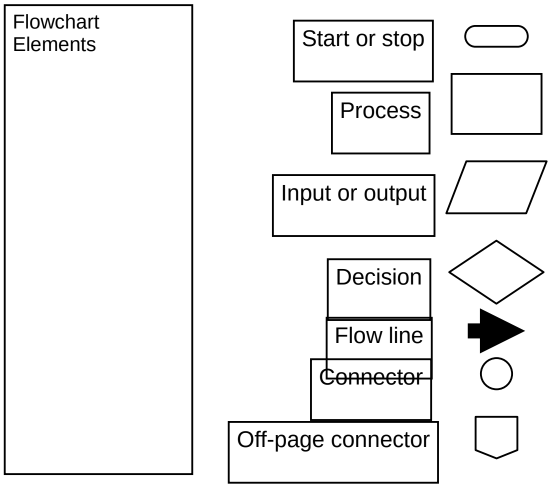
2. Syntactic correctness is easier to check as compared with semantic

Now onto Algorithm Representation

We have said enough about algorithms – their definition, their types, etc.
But, how do we actually represent them?
Generally, SW developers represent them in one of three forms:
Pseudo code
Flowcharts
Actual code
Pseudo Code
Language that is typically used for writing algorithms
Similar to a programming language, but not as rigid
The method of expression most suitable for a given situation is used:
At times, plain English
At others, a programming language like syntax

16.15 Flowchart

A graphical representation of a process (e.g. an algorithm), in which graphic objects are used to indicate the steps & decisions that are taken as the process moves along from start to finish
Individual steps are represented by boxes and other shapes on the flowchart, with arrows between those shapes indicating the order in which the steps are taken



In Today's Lecture, We …

Became familiar with the concept of algorithms:
What they are?
What is their use?
What do they consist of?
What are the techniques used for representing them?

<u>Next Lecture: Algorithms II</u>

We will continue our discussion on algorithms during the next lecture
In particular, we will discuss the pseudo code and flowcharts for particular problems
We will also discuss the pros and cons of these two algorithm representation techniques i.e. pseudo code and flow charts

Lecture 17
Algorithms II

Focus of the last lecture was on Algorithms

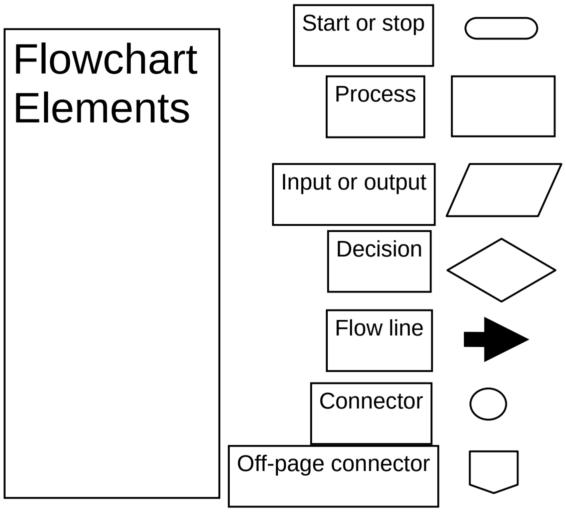
Became familiar with the concept of algorithms:
What they are? (SEQUENCE OF STEPS)
What is their use?
What are their types?
What are the techniques used for representing them?
Pseudo code
Flowcharts
Actual code

Today …

We will continue our discussion on algorithms that we started during the 16th lecture
In particular, we will look at the building blocks that are used in all algorithms
We will also discuss the pseudo code and flowcharts for particular problems
In addition, we will outline the pros and cons of those two techniques

17.1 Algorithm Building Blocks

All problems can be solved by employing any one of the following building blocks or their combinations
Sequences
Conditionals
Loops



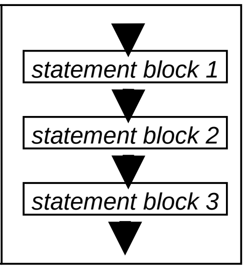
This review was essential because we will be using these building blocks quite often today.

OK. Now on with the three building blocks of algorithms. First ..

Sequences

A sequence of instructions that are executed in the precise order they are written in:

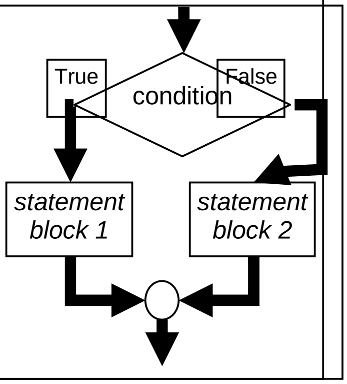
statement block 1
statement block 2
statement block 3



Conditionals

Select between alternate courses of action depending upon the evaluation of a condition

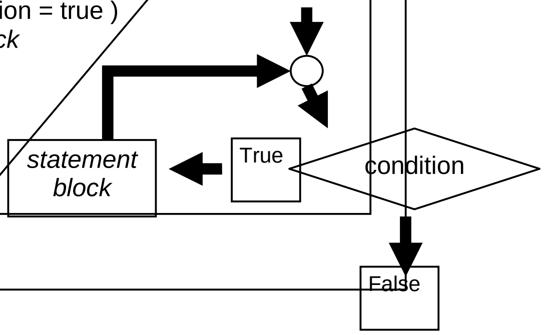
```
If ( condition = true )
        statement block 1
Else
        statement block 2
End if
```



Loops

Loop through a set of statements as long as a condition is true

```
Loop while ( condition = true )
        statement block
End Loop
```

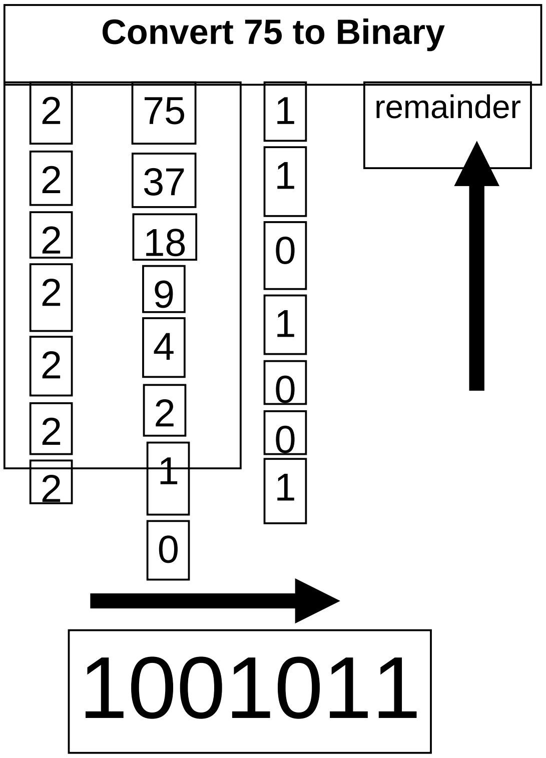


We will now present the algorithm for a problem whose solution is familiar to us
We will first go through the problem statement and then present the algorithm in three different formats:

1. Pseudo code
2. Flowchart
3. Actual code

Problem Statement

Convert a decimal number into binary



We did write down the pseudo code for this problem last time. Lets do it again, and in a slightly more formal way

17.2 Solution in Pseudo Code

```
Let the decimal number be an integer x, x > 0
Let the binary equivalent be an empty string y
Repeat while x > 0 {
        Determine the quotient & remainder of x ÷ 2
        y = CONCATENATE( remainder, y )
        x = quotient
    }
Print y
Stop
```

Q: Is this the only possible algorithm for converting a decimal number into a binary representation?

If not, then is this the best?

In terms of speed?

In terms of memory requirement?

In terms of ease of implementation?

You must ask these questions after writing any algorithm

17.3 Tips on Writing Good Pseudo Code

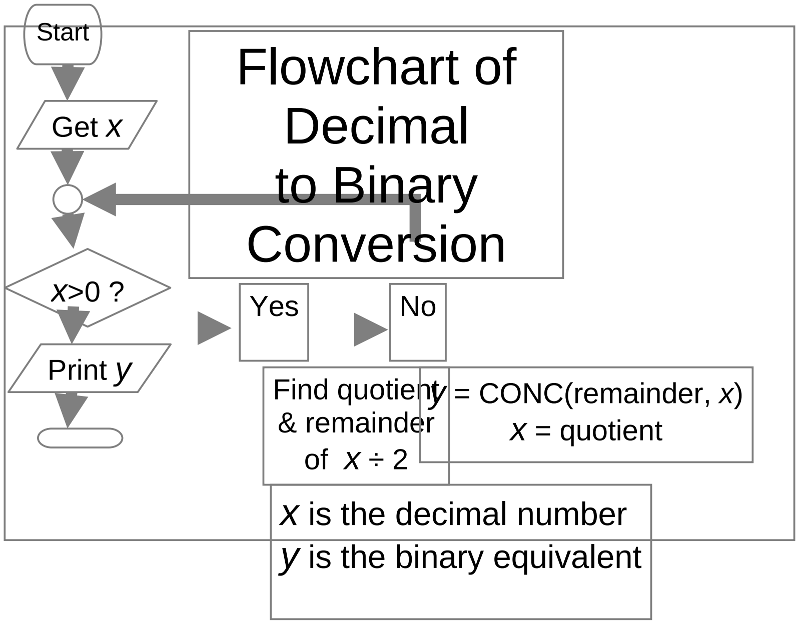
Use indention for improved clarity

Do not put "code" in pseudo code – make your pseudo code language independent

Don't write pseudo code for yourself – write it in an unambiguous fashion so that anyone with a reasonable knowledge can understand and implement it

Be consistent

Prefer formulas over English language descriptions


Start
Flowchart of Decimal to Binary Conversion
Get x
x>0 ?
Yes
No
Print y
Find quotient & remainder of x ÷ 2
y = CONC(remainder, x)
x = quotient
x is the decimal number
y is the binary equivalent

Does the flowchart depict the "correct" algorithm?

What do we mean by "correct", or better yet, what do we check for "correctness"?

One way is to check the algorithm for a variety of inputs

Does it perform satisfactorily for:

x = 0 ?

negative numbers?

numbers with fractional parts?

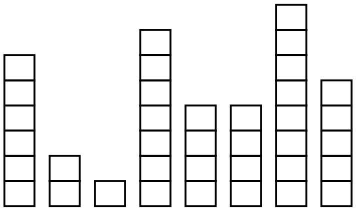
Decimal to Binary Conversion in JavaScript

```
<SCRIPT>
            x = 75;         // x is the decimal number
            y = “”;         // y is the binary equivalent
            while ( x > 0) {
                            remainder = x % 2;
                            quotient = Math.floor( x / 2 );
                            y = remainder + y;
                            x = quotient;
            }
            document.write(“y = ” + y);
</SCRIPT>
```

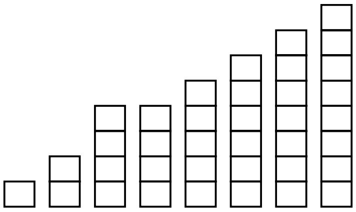
NOTE: Don’t worry if you don’t understand this code for now; you will - later!

Another Example: Sorting

Sort the following objects w.r.t. their heights



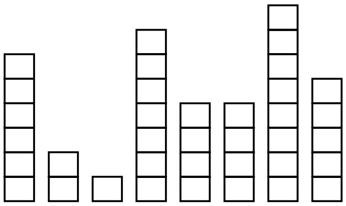
Expected Result



Strategy

There are many strategies for solving this problem. We demonstrate a simple one:
Repeat the following steps while the list is un-sorted:
Start with the first object in the list
Swap it with the one next to it if they are in the wrong order
Repeat the same with the next to the first object
Keep on repeating until you reach the last object in the list

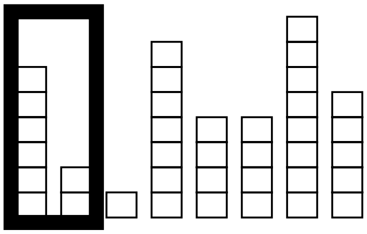
Back to the Objects to be Sorted



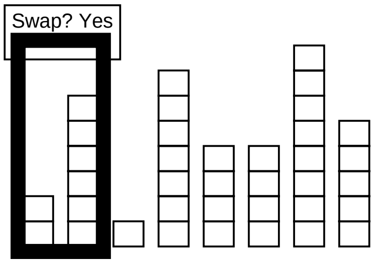
Q: Is the list sorted?

A: No

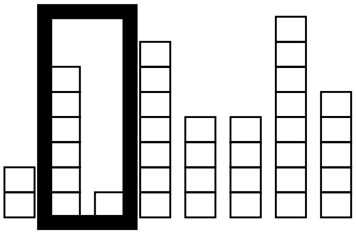
Sorting: Step A1



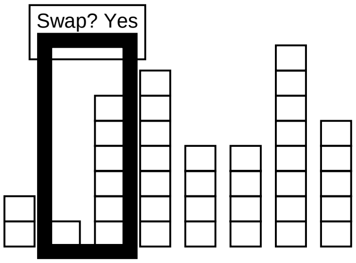
Sorting: Step A1



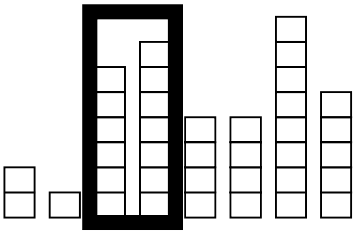
Sorting: Step A2



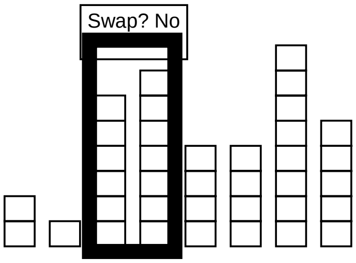
Sorting: Step A2



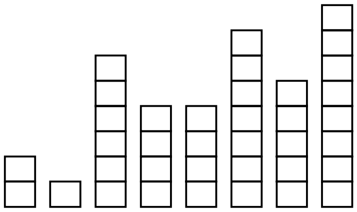
Sorting: Step A3



Sorting: Step A3



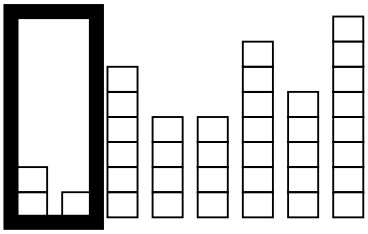
Sorting: After Step A7



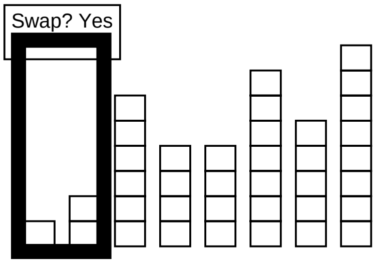
Q: Is the list sorted?

A: No

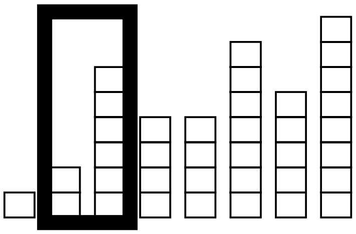
Sorting: Step B1



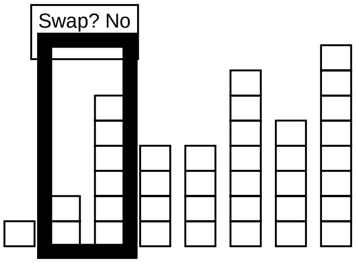
Sorting: Step B1



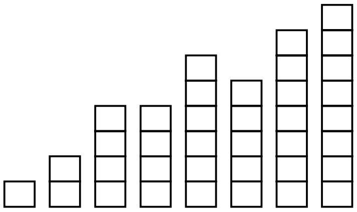
Sorting: Step B2



Sorting: Step B2



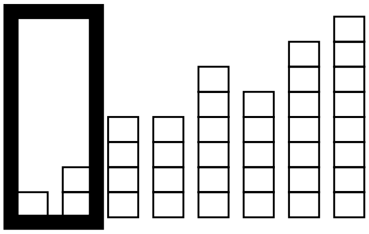
Sorting: After Step B7



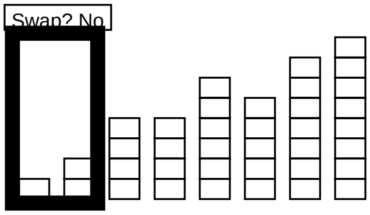
Q: Is the list sorted?

A: No

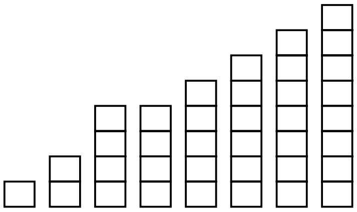
Sorting: Step C1



Sorting: Step C1



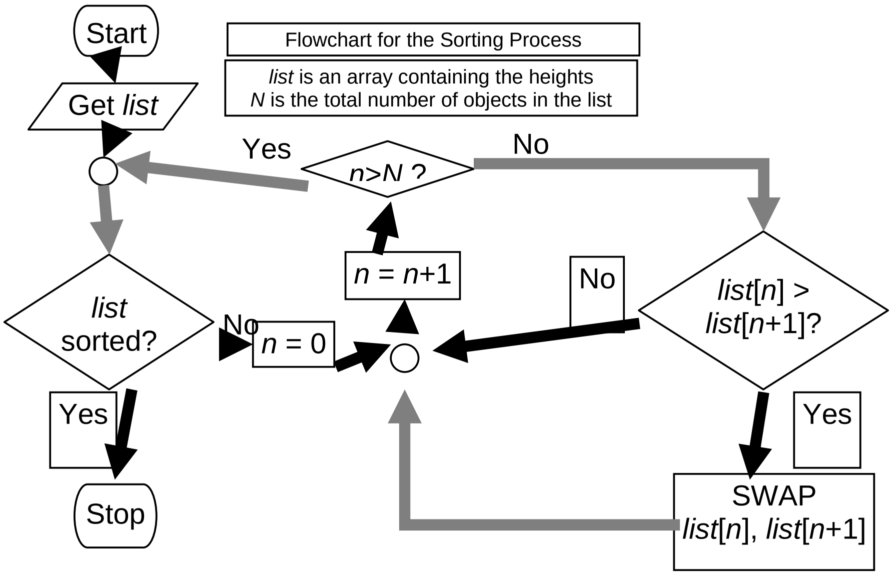
Sorting: After Step C7



Q: Is the list sorted?

A: Yes
STOP
Let’s now look at this same process of sorting being applied to a bigger list

---FLASH MOVIE FOR BUBBLE SORT GOES HERE---


Start
Flowchart for the Sorting Process
list is an array containing the heights
N is the total number of objects in the list
Get list
Yes
n>N ?
No
list sorted?
No
n = 0
n = n+1
No
list[n] > list[n+1]?
Yes
Stop
Yes
SWAP
list[n], list[n+1]

```
Dim swapFlag As Boolean, list(8) As Integer
readList( list() ) ‘this needs to be defined
swapFlag = True
Do While swapFlag = True
    For n = 1 To 8
        If list(n) > list(n + 1) Then
            temp = list(n)
            list(n) = list(n + 1)
            list(n + 1) = temp
            swapFlag = True
        End If
    Next
Loop
For n = 1 To 8
  Debug.Print list(n)
Next
```

Q: Is this the only possible algorithm for sorting a list?
A: Certainly not! In fact this one (called the “Bubble sort”) is probably the worst (reasonable) algorithm for sorting a list – it is just too slow
You will learn a lot more about sorting in your future courses

17.4 Pros and Cons of Flowcharts (1)

I personally don't find flowcharts very useful

The process of writing an algorithm in the form of a flowchart is just too cumbersome

And then converting this graphical form into code is not straight forward

However, there is another kind of flowcharts – called Structured Flowcharts – that may be better suited for software developers

17.5 Pros and Cons of Flowcharts (2)

The good thing about flowcharts is that their symbols are quite intuitive and almost universally understood

Their graphical nature makes the process of explaining an algorithm to one's peers quite straightforward

17.6 Pros and Cons of Pseudo Code (1)

Quite suitable for SW development as it is closer in form to real code

One can write the pseudo code, then use it as a starting point or outline for writing real code

Many developers write the pseudo code first and then incrementally comment each line out while converting that line into real code

17.7 Pros and Cons of Pseudo Code (2)

Pseudo code can be constructed quite quickly as compared with a flowchart

Unlike flowcharts, no standard rules exist for writing pseudo code

With that we have reached the end of the materials that we wanted to cover today. However, I still need to tell you about your assignment #6

In Today's Lecture, We …

We continued our discussion on algorithms that we had started during the 16th lecture

In particular, we looked at the building blocks that are used in all algorithms

We also discussed the pseudo code and flowcharts for particular problems

In addition, we outlined the pros and cons of those two techniques

Focus of the Next Lecture: Programming Languages

To understand the role of programming languages in computing

To understand the differences among low- & high-level, interpreted & compiled, and structured & object-oriented programming languages

Lecture 18
Objects, Properties, Methods
(Web Development Lecture 6)

During the last lecture we continued our discussion on Interactive Forms

We got our first taste of JavaScript – the object-based language that we will be employing throughout the rest of the Web development part of this course

We developed a (simple) client-side script in JavaScript

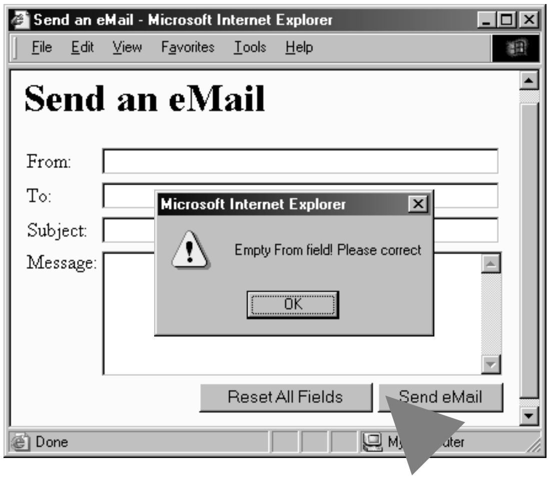
During Today's Lecture …

We will have a more formal introduction to JavaScript and client-side scripting

We will become able to appreciate the concept of objects in JavaScript

We will learn about the properties of those objects, and about how to read & modify them

We will become able to perform simple tasks through the application of methods


Send an eMail - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Send an eMail
From:
To:
Subject:
Message:
Microsoft Internet Explorer
OK
Reset All Fields
Send eMail
Done
My Computer

Empty From field! Please correct

Last time we looked at two distinct ways of performing the "form" field checking function.

From now onwards, we will be employing the 2nd way more often than not

In that 2nd way, we referred to a function in the HTML BODY, and but defined that function in the HTML HEAD

The main code segment that goes between the <SCRIPT>, </SCRIPT> tags in the HEAD:

```
function checkForm() {
   if ( document.sendEmail.sender.value.length < 1)
{
      window.alert( "Empty From field! Please correct"
);
   }
}
```

The JavaScript code included as an attribute of the "Send eMail" button:

```
onMouseOver="checkForm()"
```

```
<HTML>
<HEAD>
<TITLE>Send an eMail</TITLE>
<SCRIPT>
    function checkForm(){
        if (document.sendEmail.sender.value.length < 1) {
            window.alert('Empty From field! Please correct');
        }
    }
</SCRIPT>
</HEAD>
<BODY bgcolor="#FFFFCC">
<H1>Send an eMail</H1>
<FORM name="sendEmail" method="post" action=sendMailScriptURL>
  <TABLE><TR> <TD>From: </TD>
     <TD><INPUT type="text" name="sender" size="50" ></TD>
    </TR><TR> <TD>To: </TD>
     <TD><INPUT type="text" name="receiver" size="50"></TD>
    </TR><TR><TD>Subject: </TD>
     <TD><INPUT type="text" name="subject" size="50"></TD>
    </TR><TR><TD valign="top">Message: </TD>
     <TD><TEXTAREA name="message" cols="38"
rows="6"></TEXTAREA></TD>
    </TR><TR><TD colspan="2" align="right">
      <INPUT type="reset" name="reset" value="Reset All Fields">
      <INPUT type="submit" name="sendEmail" value="Send eMail"
onMouseOver="checkForm()">
</TD></TR></TABLE></FORM>
</BODY>
</HTML>
```

18.1 New Concept: Client-Side Scripts

Small programs that are a part of the Web page and run on the user's (client's) computer

They interact with the user to collect info or to accomplish other tasks

Once it has been collected, they may help pass the collected info on to a server-side script

We'll use JavaScript to do client-side scripting. However, there are many other languages that can be used for that purpose, e.g. VBScript

18.2 Advantages of Client-Side Scripting

Reduced server load as it does not have to send messages to the user's browser about missing or incorrect data

Reduced network traffic as the form's data is sent only once instead of many to's and fro's

18.3 Disadvantages

Client-side scripts do not work with all browsers

Some user intentionally turn scripting off on their browsers

This increases the complexity of the Web page, as it now has to support both situations: browsers with scripting capability, and those not having that capability

18.4 JavaScript

A programming language specifically designed to work with Web browsers

It is designed to be used for developing small programs – called scripts – that can be embedded in HTML Web pages

JavaScript:

Is an interpreted language

Supports event-driven programming

Is object-based

Some of things that JavaScript cannot do!

The following file operations on the client computer:

Read -- Modify

Rename -- Delete

Create

Create graphics (although, it does have the ability to format pages through HTML - including the placement of graphics)

Any network programming bar one function: the ability to download a file to the browser specified through an arbitrary URL

Some of the things that JavaScript can do!

Control the appearance of the browser

Control the content and appearance of the document displayed in the browser

Interact with the user through event handlers

Arbitrary calculations, including floating-point ones

Store & modify a limited amount of data about the user in the form of client-side "cookies"

18.5 Client-Side JavaScript

Everything that JavaScript manipulates, it Although a version of JavaScript exists that can be used to write server-side scripts, our focus in this course will only be on client-side scripting

Case Sensitivity

HTML is *not* case sensitive. The following mean the same to the browser:

<HTML> -- <html>

<Html> -- <htMl>

JavaScript *is* case sensitive. Only the first of the following will result in the desired function – the rest will generate an error or some other undesirable event:

onMouseClick -- OnMouseClick

onmouseclick -- ONMOUSECLICK

JavaScript

A programming language specifically designed to work with Web browsers

It is designed to be used for developing small programs – called scripts – that can be embedded in HTML Web pages

JavaScript:

Is an interpreted language

Supports event-driven programming

Is object-based

JavaScript is Object-Based

treats as an *object* – e.g. a window or a button

An object has *properties* – e.g. a window has size, position, status, etc.

Objects are modified with *methods* that are associated with that object – e.g. a resize a window with *resizeTo(150, 200)*

Not Object-Oriented!

JavaScript is not a true object-oriented language like C++ or Java

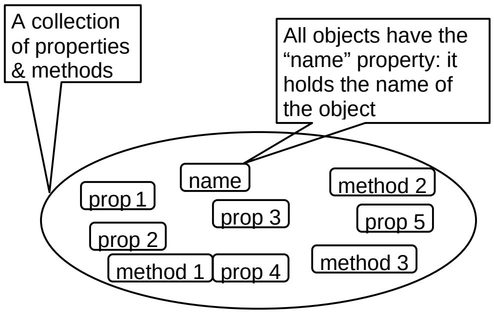
It is so because it lacks two key features:

A formal inheritance mechanism

Strong typing

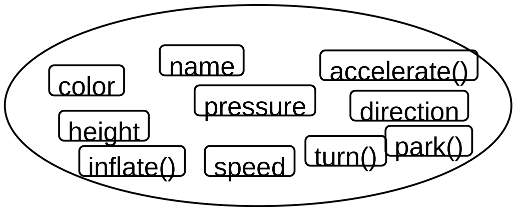
Nevertheless, it does include many key concepts that are part of almost all object-oriented languages, and therefore is referred as an *object-based* language

Object: A *named* collection of properties (data, state) & methods (instructions, behavior)

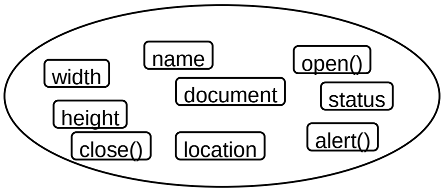

A collection of properties & methods

All objects have the "name" property: it holds the name of the object

Example: A Bicycle



Example: JavaScript's "window" Object



18.6 Properties

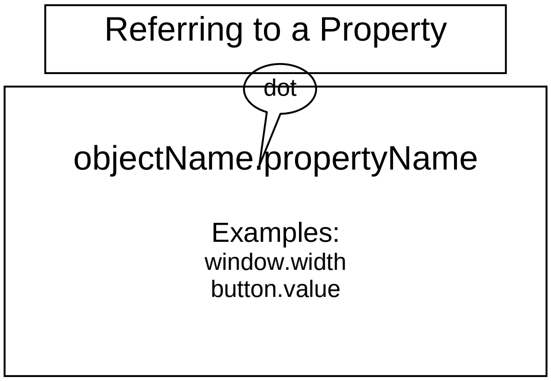
Objects may have a single or several properties

A property may have one of the following values:

Number -- Text -- Boolean

Array -- Functions

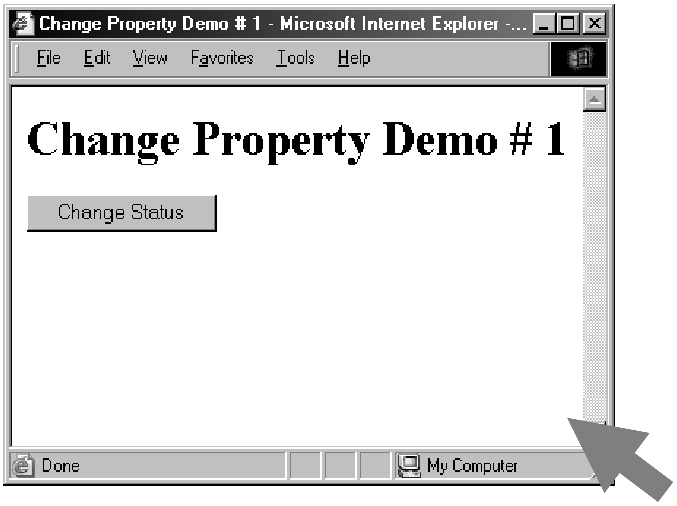
Objects (Example: "document" – a property of the "window" object – is an object in itself. A "document" in turn may contain a "form" object as a property, and then that "form" may contain a "button" property, which, once again, is an object in itself)



Referring to a Property

objectName.propertyName

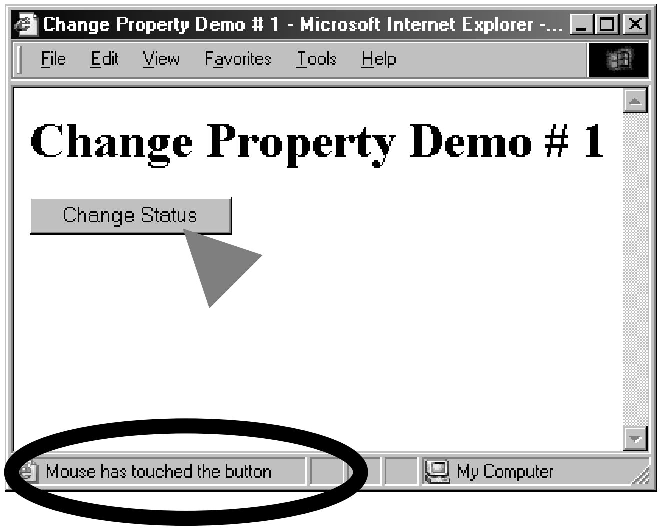
Examples:
window.width
button.value



Change Property Demo # 1 - Microsoft Internet Explorer -...
File Edit View Favorites Tools Help
Change Status
My Computer

Change Property Demo # 1

Done



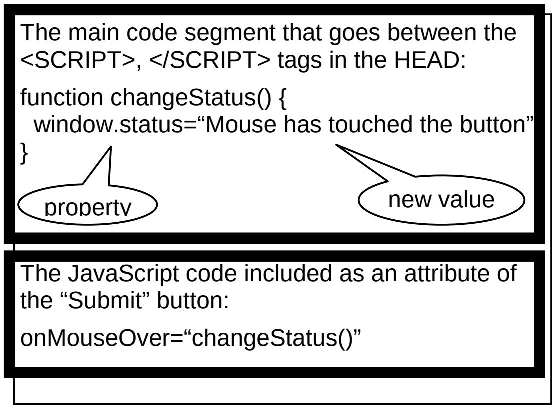
Change Property Demo # 1 - Microsoft Internet Explorer -...
File Edit View Favorites Tools Help
Change Status
My Computer

Change Property Demo # 1

Mouse has touched the button

```
<HTML>
<HEAD>
```

```
  <TITLE>Change Property Demo # 1</TITLE>
  <SCRIPT>
    function changeStatus() {
      window.status = “Mouse has touched the button”;
    }
  </SCRIPT>
</HEAD>
<BODY>
<H1>Change Property Demo # 1</H1>
<FORM name=“dummy” method=“” action=“”>
   <INPUT type=“submit” name=“” value=“Change Status“
      onMouseOver=“changeStatus()”>
</FORM>
</BODY>
</HTML>
```

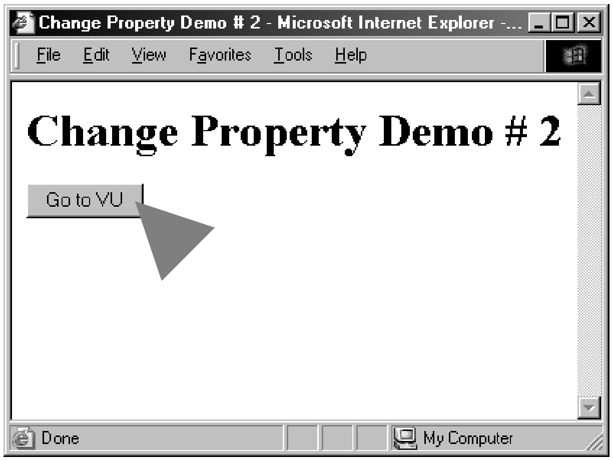


The main code segment that goes between the <SCRIPT>, </SCRIPT> tags in the HEAD:

```
function changeStatus() {
  window.status=“Mouse has touched the button”
}
```

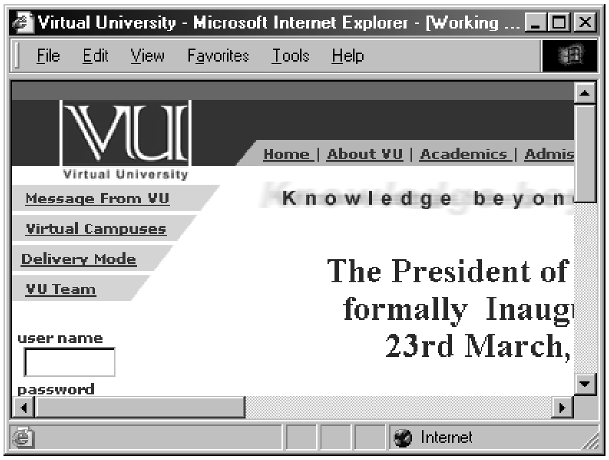
The JavaScript code included as an attribute of the “Submit” button:

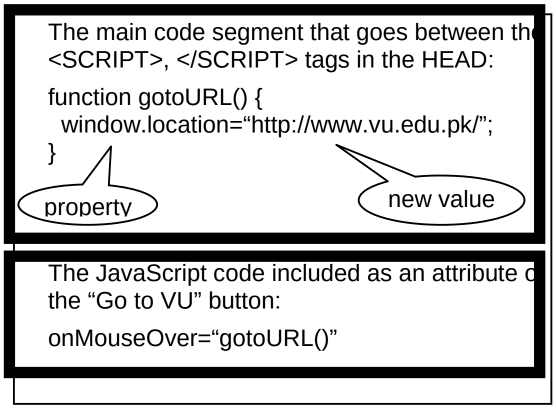
onMouseOver=“changeStatus()”



Change Property Demo # 2 - Microsoft Internet Explorer -...
File Edit View Favorites Tools Help
Go to VU
Done
My Computer

Change Property Demo # 2





The main code segment that goes between th <SCRIPT>, </SCRIPT> tags in the HEAD:

```
function gotoURL() {
  window.location=“http://www.vu.edu.pk/”;
}
```

The JavaScript code included as an attribute o the “Go to VU” button:

```
onMouseOver=“gotoURL()”
```

You should be connected to the Internet for the successful execution of the example that **we just discussed**

A Suggestion

Please try the pieces of code that I just demonstrated to you to change the status and location properties of the “window” object yourself

Also try changing the width, height properties of the “window” object

Types of Objects

JavaScript objects

Objects that are part of JavaScript

Examples: window, document

Browser objects

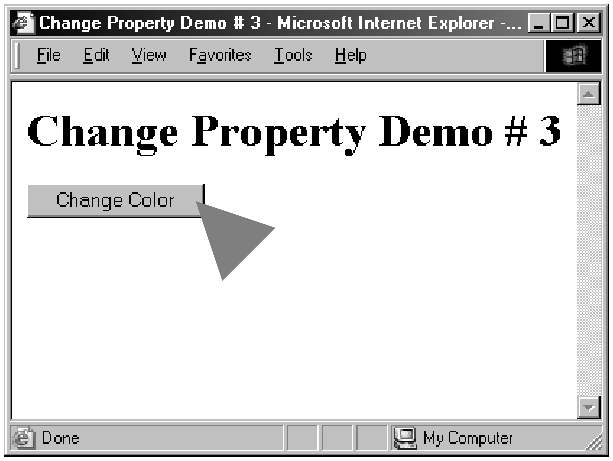
Objects that contain info *not* about the contents of the display, but the browser itself

Examples: history, navigator

User-defined object

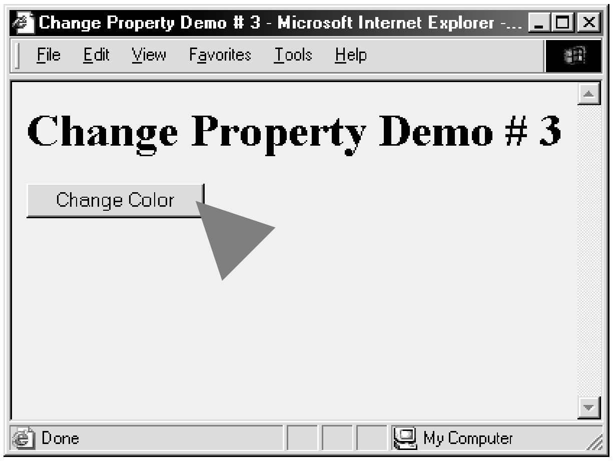
One More Example:

Let us try to change the background color of the window



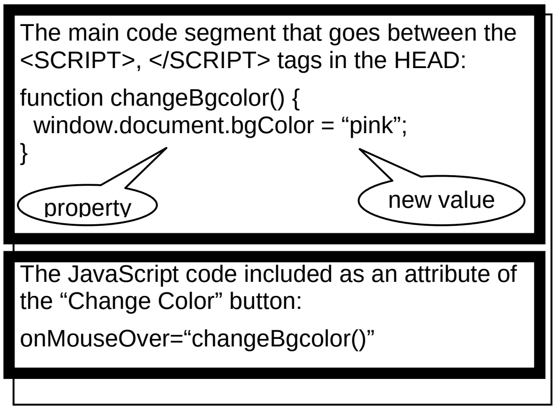
Change Property Demo # 3

Change Property Demo # 3 - Microsoft Internet Explorer -...
File Edit View Favorites Tools Help
Change Color
Done
My Computer



Change Property Demo # 3

Change Property Demo # 3 - Microsoft Internet Explorer -...
File Edit View Favorites Tools Help
Change Color
Done
My Computer



The main code segment that goes between the <SCRIPT>, </SCRIPT> tags in the HEAD:

```
function changeBgcolor() {
  window.document.bgColor = “pink”;
}
```

The JavaScript code included as an attribute of the “Change Color” button:

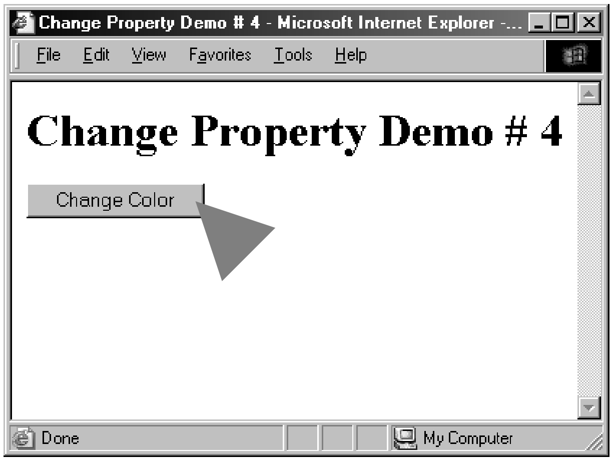
```
onMouseOver=“changeBgcolor()”
```

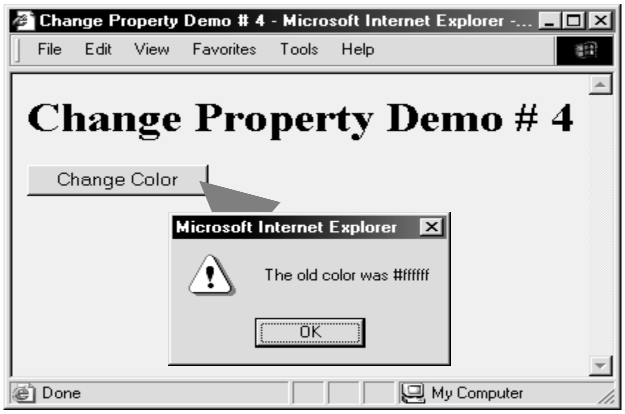
In addition to “bgColor”, there are many other properties of the “document” object, e.g.

fgColor linkColor	cookie forms[ ]

title	images[ ]
URL	links[ ]
referrer	…
lastModified	…
	…

We have learnt how to modify the properties of objects
But the properties are not there just so that we can modify them; we can also just read them – that is just find out their current value
Let us now look at an example where we first read a property, display the current value, and then change the property


Change Property Demo # 4 - Microsoft Internet Explorer -...
File Edit View Favorites Tools Help
Change Property Demo # 4
Change Color
Done
My Computer


Change Property Demo # 4 - Microsoft Internet Explorer -...
File Edit View Favorites Tools Help
Change Property Demo # 4
Change Color
Microsoft Internet Explorer
OK
Done
My Computer

The old color was #ffffff

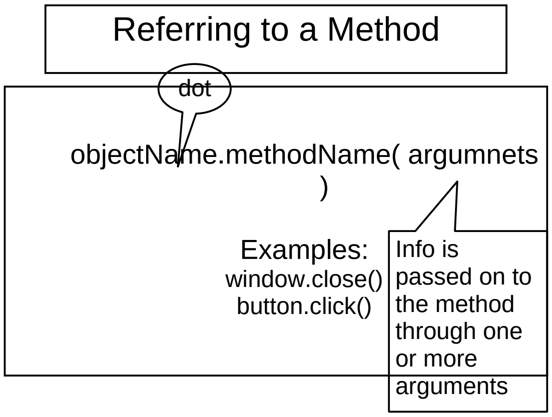
The main code segment that goes between the <SCRIPT>, </SCRIPT> tags in the HEAD:

```
function changeBgcolor() {
   oldColor = window.document.bgColor;
   window.document.bgColor = “pink”;
   window.alert("The old color was " + oldColor);
}
```

The JavaScript code included as an attribute of the “Change Color” button:

```
onMouseOver=“changeBgcolor()”
```

Now we have established what we mean by objects: a named collection of properties and methods
And that JavaScript treats everything that it manipulates as an object
We have also learnt how to change the properties of these objects by selecting a property and equating it to a new value
Methods: Functions (code, instructions, behavior) associated with objects
Methods are functions associated with an object that can be used to manipulate that object
Example:
window.close()
Here "close()" is a method that has been defined for the "window" object. Its function is to close the "window"



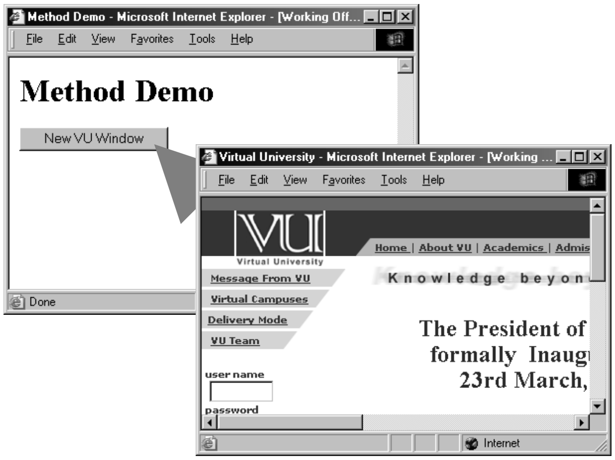
Referring to a Method

objectName.methodName(argumnets)

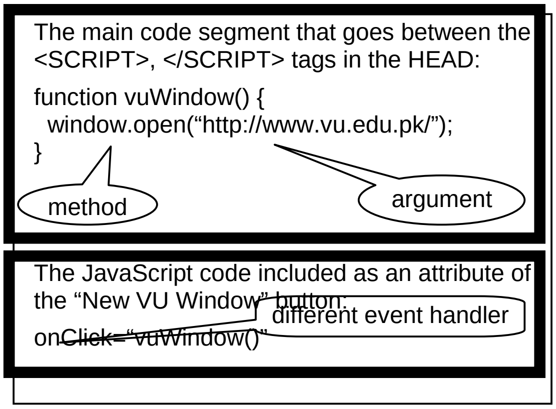
Examples:
window.close()
button.click()

Info is passed on to the method through one or more arguments

A few more methods associated with the "window" object
alert()
confirm()
prompt()
close()
open()
focus()
blur()
setTimeOut()



Method Demo - Microsoft Internet Explorer - [Working Off...
File Edit View Favorites Tools Help
Method Demo
New VU Window
Done
Virtual University - Microsoft Internet Explorer - [Working ...
File Edit View Favorites Tools Help
VU
Virtual University
Home | About VU | Academics | Admis
Knowledge beyon
Message From VU
Virtual Campuses
Delivery Mode
VU Team
The President of
formally Inaug
23rd March,
user name
password
Internet



The main code segment that goes between the <SCRIPT>, </SCRIPT> tags in the HEAD:

```
function vuWindow() {
  window.open(“http://www.vu.edu.pk/”);
}
```

The JavaScript code included as an attribute of the “New VU Window” button:

```
onClick=“vuWindow()”
```

18.7 Event Handlers

Objects are made up of properties and associated methods

Many objects also have “event handlers” associated with them

“Events” are actions that occur as a result of user’s interaction with the browser

We use “event handlers” [e.g. onMouseOver(), onClick()] to design Web pages that can react to those events

More on event handlers in a future lecture

During Today’s Lecture …

We had a more formal introduction to JavaScript and client-side scripting

We became able to appreciate the concept of objects in JavaScript

We learnt about the properties of those objects

We also became able to perform simple tasks through the application of methods

Next (the 7th) Web Dev Lecture:

Data Types and Operators

To find out about data types

To become familiar with JavaScript data types

To become able to use JavaScript statements and arithmetic operators

Lecture 19
Programming Languages

During the last lecture …

We continued our discussion on algorithms that we had started during the 16th lecture
In particular, we looked at the building blocks that are used in all algorithms
We also discussed the pseudo code and flowcharts for particular problems
In addition, we outlined the pros and cons of those two techniques
Last time we discussed what to implement
Today's Lecture
Today we are going to discuss the tool that is used to implement SW
To understand the differences among low- & high-level, interpreted & compiled, and structured & object-oriented programming languages
To understand the role of programming languages in computing

WHAT IS PROGRAMING (CODING) ?

The process of telling the computer what to do

TYPES OF PROGRAMS

Batch Programs
Event-Driven Programs

19.1 Batch Programs

These are typically started from a shell (or automatically via a scheduler) and tend to follow a pattern of:
Initialize internal data
Read input data
Process that data
Print or store results
Key feature: No user interaction with the computer while the program is running

Programming Language

A vocabulary and set of grammatical rules for instructing a computer to perform specific tasks

19.2 Event-Driven Programs

Examples: GUIs, microwave, camera
The system sends events to the program and the program responds to these as they arrive.
Events can include things a user does - like clicking the mouse - or things that the system itself does - like updating the clock.
These programs generally work as follows:
Initialize the internal data
Wait for events to arrive
Identify an incoming event and react accordingly

Programming Language

A vocabulary and set of grammatical rules for instructing a computer to perform specific tasks

All programs consists of:

Sequence of instructions
Conditionals
Loops

These may contain:

Data
Input/output (print, etc)
Operations (add, divide, etc)

Examples of Prog. Languages

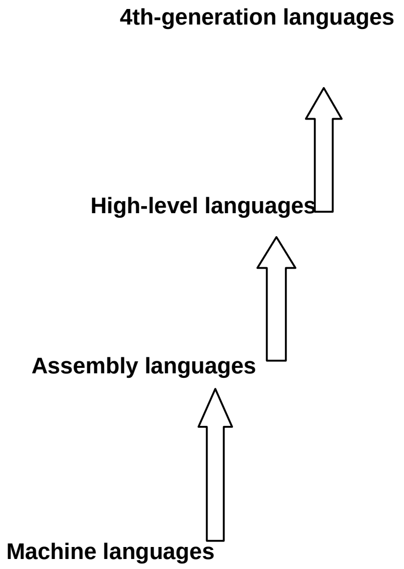
Machine Language	
Assembly Language (1956-63)	Perl (1987)
LISP (1956)	VisualBasic (1991)
PL/1(1964)	PowerBuilder
BASIC (1964)	Ada(1983)
Pascal (1970)	C++ (1983-85)
Smalltalk (1972)	QBasic (1986 Java (1995)
C (1972) Fortran (1957)	JavaScript
COBOL (1959)	C# (2001)

19.3 Types of Prog. Languages

High level Programming Languages
Low Level Programming Languages

High-level programming languages, while simple compared to human languages, are more complex than the languages the **uP** actually understands, called machine languages each different type of microprocessors has its own unique machine language lying between machine languages & high-level languages are languages called Assembly languages

Assembly languages are similar to machine languages, but are easier to program in as they allow a programmer to substitute names for numbers
Machine languages consist of numbers only.



Regardless of what language you use, you eventually need to convert your program into a language that the computer can understand
Two ways for doing that:
- compile the program or
- interpret the program

Interpreter is a program that executes instructions written in a high-level language
An interpreter translates high-level instructions into an intermediate form, which it then executes. In contrast, a compiler translates high-level instructions directly into machine language
Compiled programs generally run faster than interpreted programs.

The advantage of an interpreter, however, is that it does not need to go through the compilation stage during which the whole of the high-level code is translated into machine instructions in one go. This process can be time-consuming if the program is long.

The interpreter can immediately execute high-level programs, without waiting for the completion of the translation process

The choice of which language to use can also depend on the:
-Type of computer the program is to run on,
- Expertise of the programmer

Interpreters: immediate response, but execute code slowly.

Compilers: Takes longer to compile, but super-fast execution.

Both interpreters and compilers are available for most high-level languages. However, **BASIC** and **LISP** were especially designed to be executed by an interpreter.

Why are there so many different programming languages?

What are the advantages of particular languages?

The question of which language is best is one that consumes a lot of time and energy among computer professionals

Every language has its strengths and weaknesses

-Can a single language have all the good bits of other languages?

-Is there a perfect language?

-Do some good features force a language to also have bad features?

-What makes a feature good or bad?

FORTRAN is a particularly good language for processing numerical data, but it does not lend itself very well to large business programs

Pascal is very good for writing well-structured and readable programs, but it is not as flexible as the C programming language

C++ embodies powerful object-oriented features, but it is complex and difficult to learn

What **changes** in the field of computer languages can we expect in the near future?

-Which programming language should you learn?
Should you learn more than one?

19.4 Programming SW Development

- SW Design Methodology ?

The set of (often flexible) rules and guidelines a team of developers follow to construct reasonably complex SW systems

19.5 Object Oriented Design

OO SW is all about objects: a black box which receives messages & responds with those of its own

An object has 2 aspects:

State, also termed as properties, data

Example: For the bicycle: color, speed, pressure

Behaviors, also termed as methods, instructions

Example: For the same object: accelerate(), inflate()

In traditional design, these 2 aspects have been kept apart

The designer starts with any component (object) of the system; designs it as an independent, self-contained system, and then moves to the design of some other component . The over-all system is put together by fitting together a collection of these components.

Key feature: Details of the design of the component are kept independent of the over-all system.

Benefit: It can be easily re-used in other systems: design once; use multiple times

19.6 Structured Design

Also called top-down design

The designer starts by first conceiving a skeleton high-level design of the system, and then starts defining features of that over-all design in an ever-increasing detail

Making small changes in the functionality of the systems sometimes leads to major re-design exercise

Structured design emphasizes separating a program's data from its functionality

Separating data from functionality typically leads to SW that is difficult to maintain & understand - especially for large SW systems

19.7 Object-Oriented Languages

Programming languages specifically designed to make it easy to implement object-oriented designs

Examples: Smalltalk, C++, Java

Programming Languages

http://www.wikipedia.com/wiki/Programming_language

During Today's Lecture, We …

To understand the role of programming languages in computing

To understand the differences among low- & high-level, interpreted & compiled, and structured & object-oriented programming languages

Focus of the Next Lecture:

The SW Development Process

Development process of reasonably complex SW systems does not consist of "coding" only

We will become familiar with the various phases of the process that developers follow to develop SW systems of reasonable complexity

Lecture 20
SW Development Methodology

We discussed the role of programming languages in computing
We also discussed the differences among low- & high-level, interpreted & compiled, and structured & object-oriented programming languages
We also discussed the object-oriented and the structured methodologies for SW design

Any Other SW Design Methodologies?

-- Spaghetti Design Methodology
The most popular software design (programming) methodology

Today's Lecture

Development process of reasonably complex SW systems does not consist of "coding" only

We will become familiar with the various phases of the process that developers follow to develop SW systems of reasonable complexity

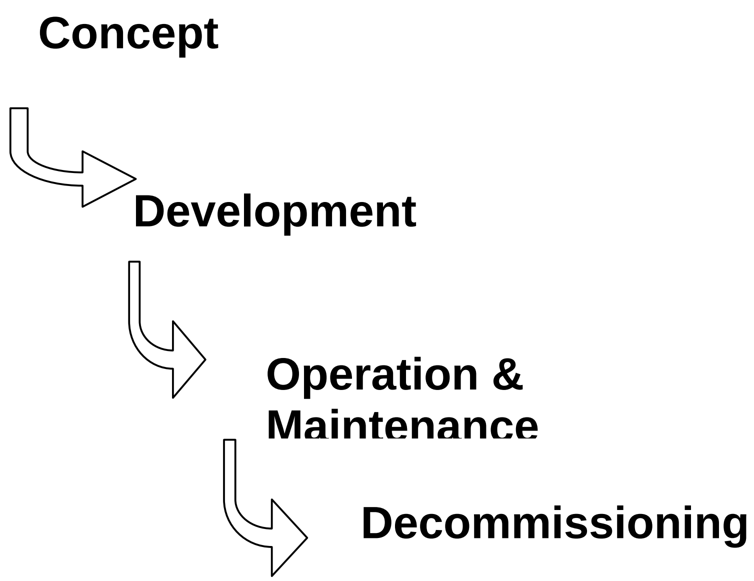
SW Life-Cycle

The sequence of phases a SW goes through from the concept to decommissioning
It is important to think about all those phases before the design work starts
Thinking about the future phases generally results in:
Shorter delivery times
Reduced costs of development
A system of higher quality

A Case in Point

I didn't discuss with the customer the specs of the HW & OS before developing a particular e-commerce SW.
I wrote it for the HW/OS that was easily available to me.
Unfortunately that HW/OS combination differed from what was easily available to the client Result: Huge amount of rework. Higher cost. Delayed delivery. Lower quality.
Therefore, now before designing a SW system, I first write down the installation manual, and get it OK'd by the customer. I do the same with the Operation & Maintenance manual as well.

Simple SW Life Cycle

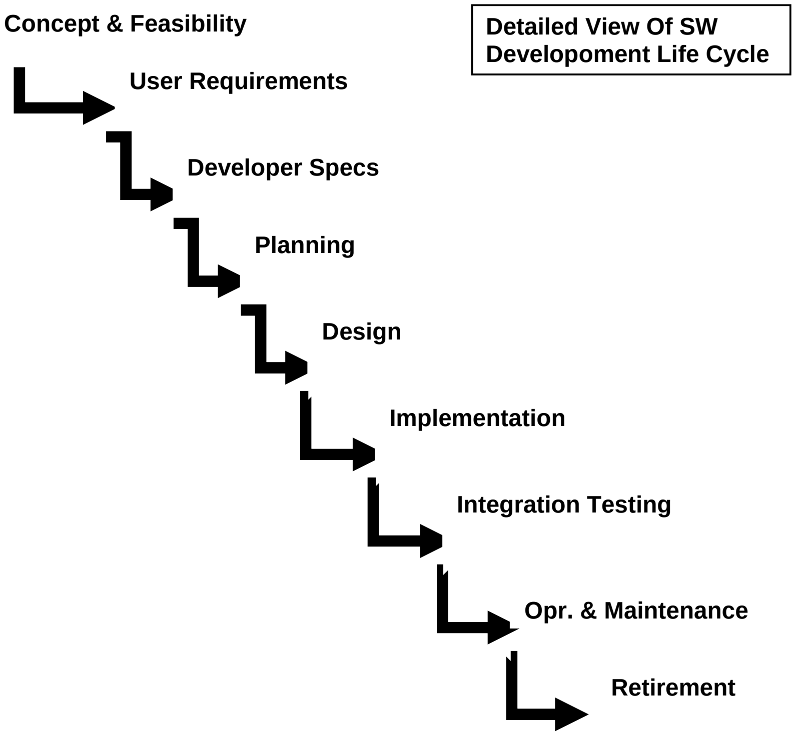


Concept

Development

Operation & Maintenance

Decommissioning



Detailed View Of SW Developoment Life Cycle

Concept & Feasibility

User Requirements

Developer Specs

Planning

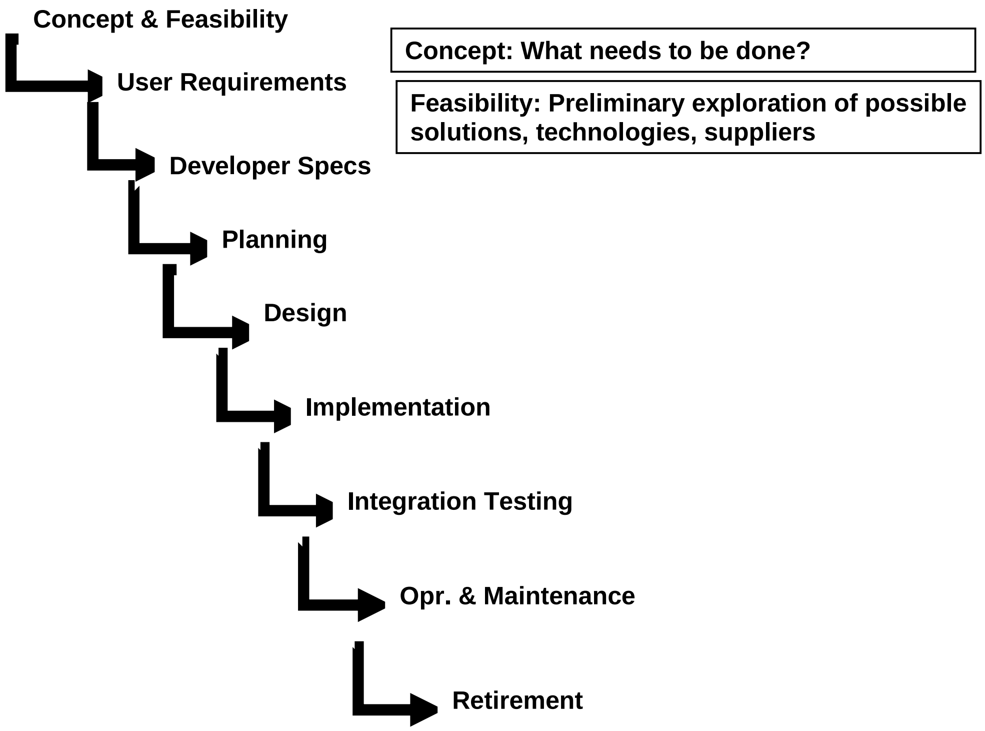
Design

Implementation

Integration Testing

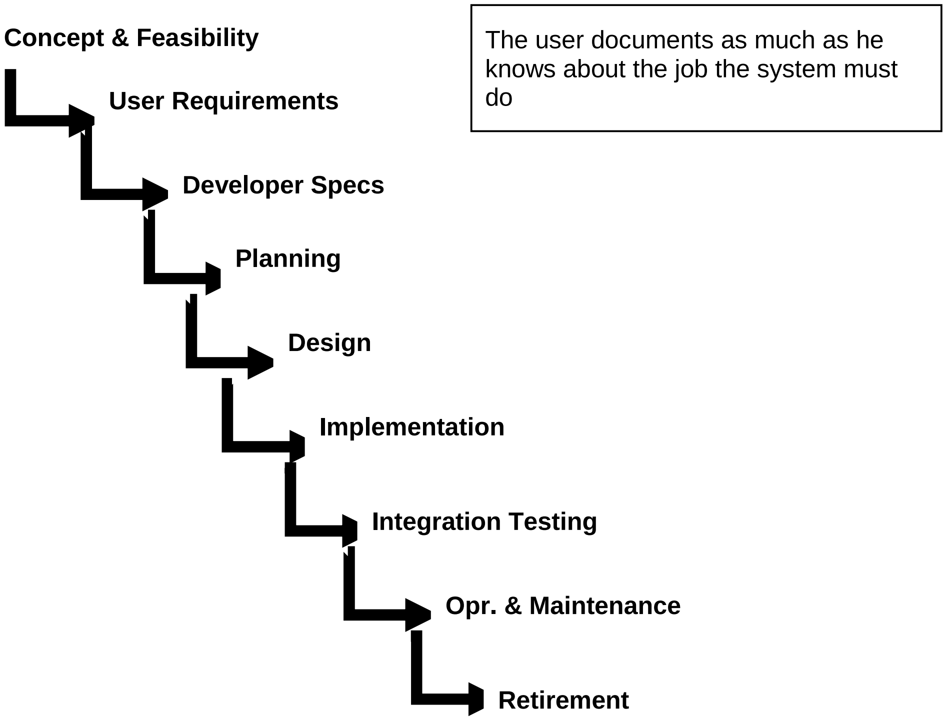
Opr. & Maintenance

Retirement



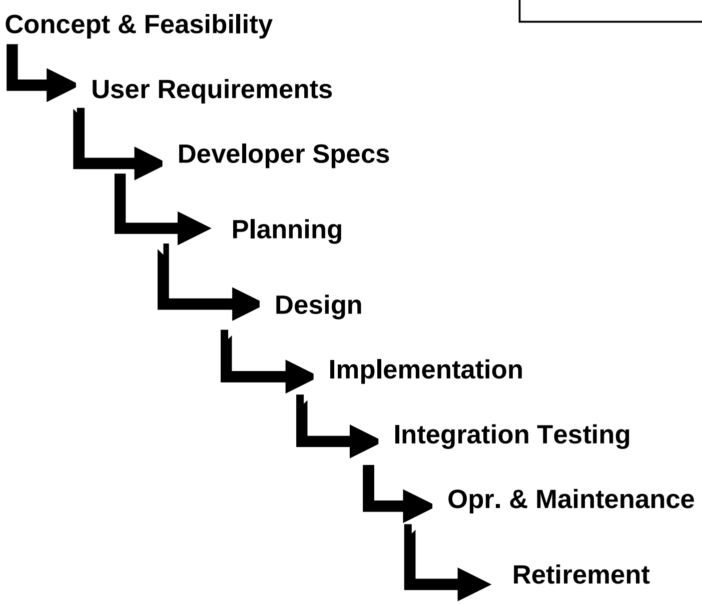
Concept: What needs to be done?

Feasibility: Preliminary exploration of possible solutions, technologies, suppliers

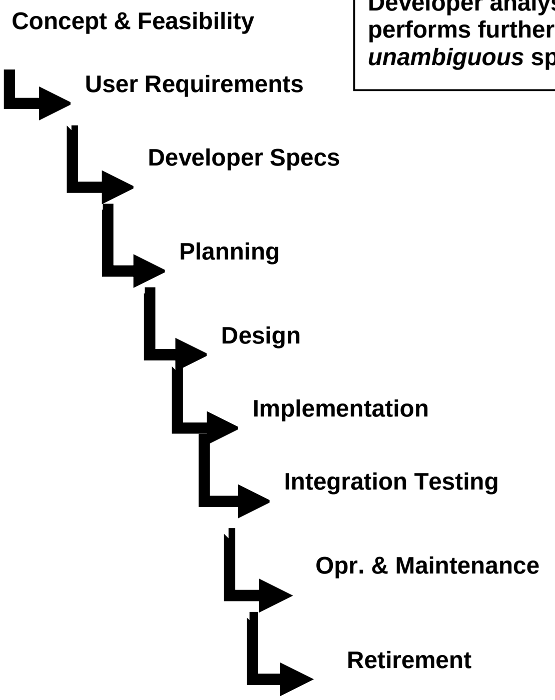


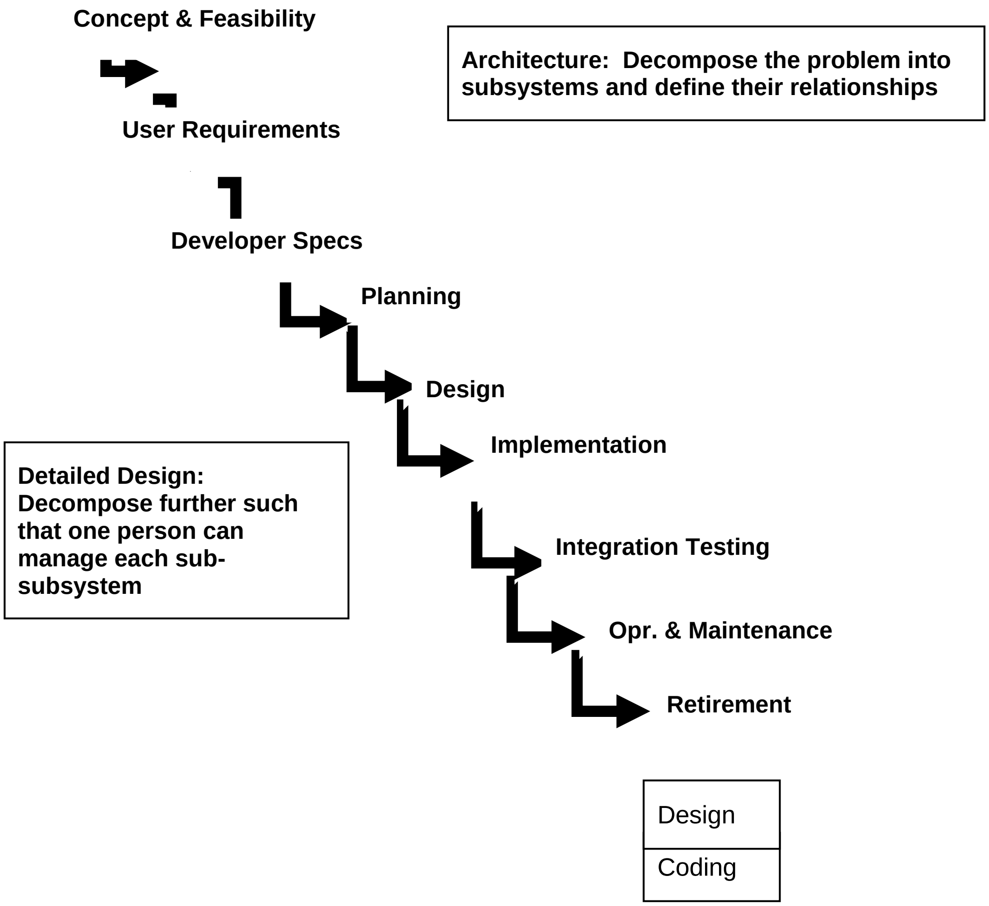
The user documents as much as he knows about the job the system must do

Detailed plan specifying the required resources and expected deliverables



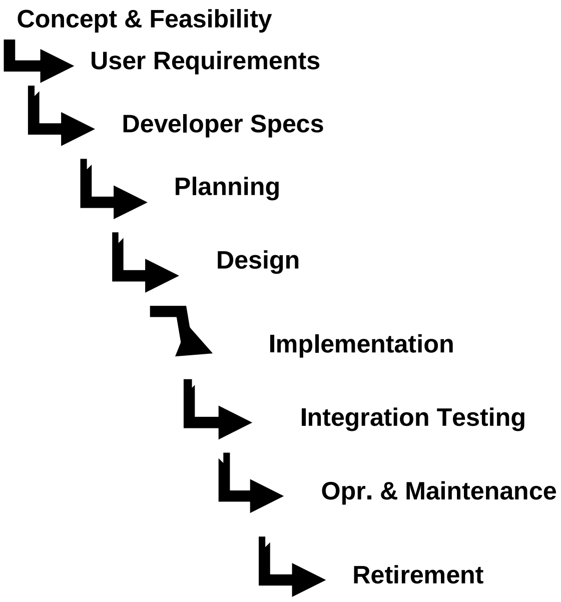
Developer analyses users requirement, performs further investigation, and produces *unambiguous* specifications



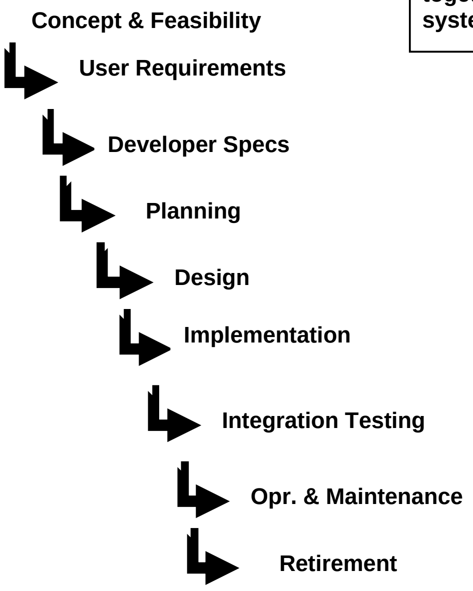


Architecture: Decompose the problem into subsystems and define their relationships

Detailed Design: Decompose further such that one person can manage each sub-subsystem



Bring the sub-subsystems together to form subsystems and test. Bring subsystems together to form the system and test

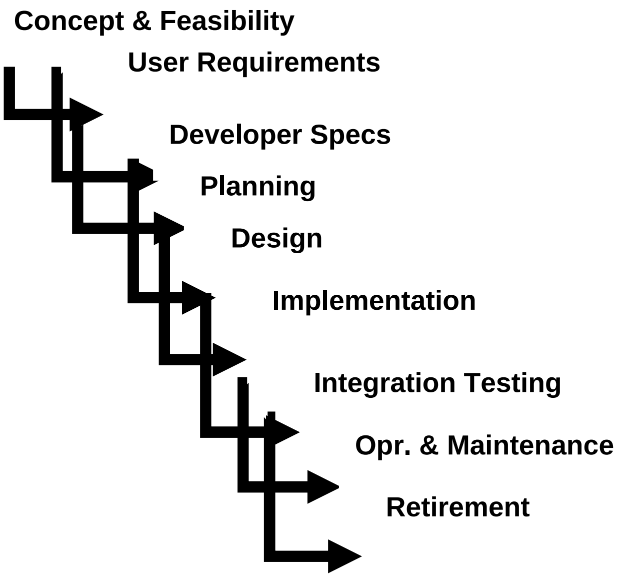


Use

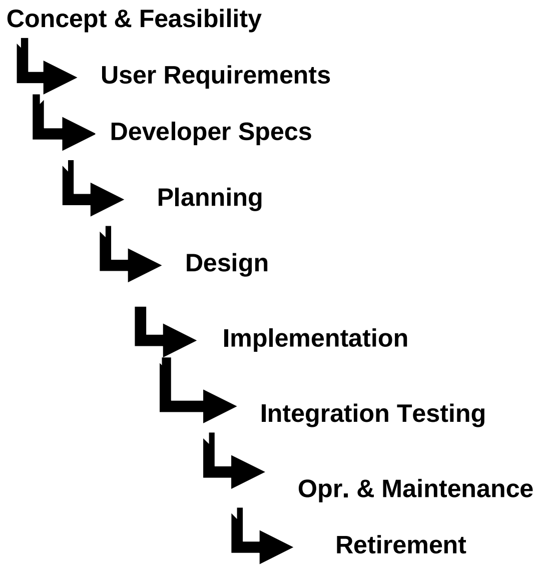
Enhance

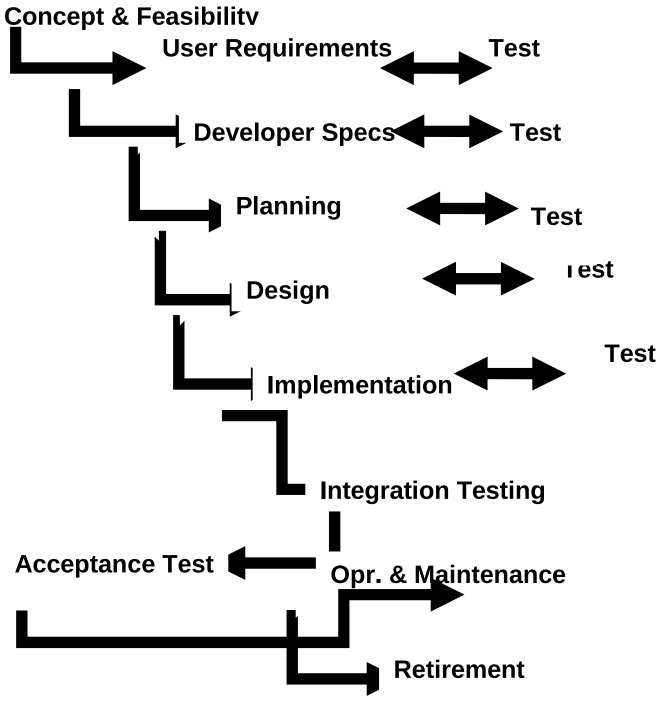
Adapt

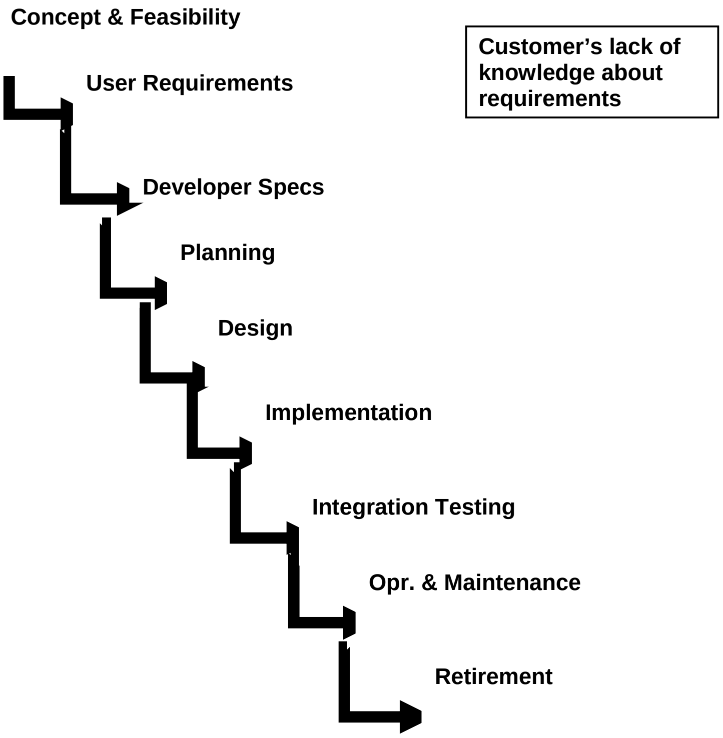
Correct

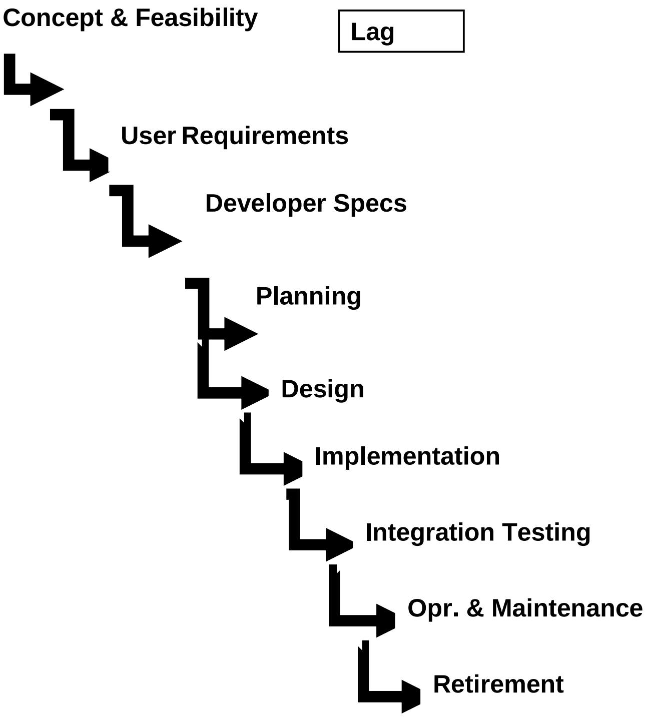


Phase it out when the time comes









Other Life-Cycle Models

The sequence of phases (or the life-cycle mode) that I showed is just one example of the several sequences that SW developers follow

This one is called the "Waterfall" model

You will learn about some more models (e.g. the Spiral model) in your future courses

The Waterfall Lifecycle Model and its Derivatives
www.cs.qub.ac.uk/~J.Campbell/myweb/misd/node3.html
In Today's Lecture
We became familiar with the various phases of the process that developers follow to develop SW systems of reasonable complexity
We looked at a couple of problems related to the Waterfall SW development model

Next Lecture: 2nd In the Productivity SW Series Spreadsheets

We will become familiar with the basic features and functions of spreadsheets
We will become able to perform simple data analysis using spreadsheet SW

Lecture 21

Data Types & Operators

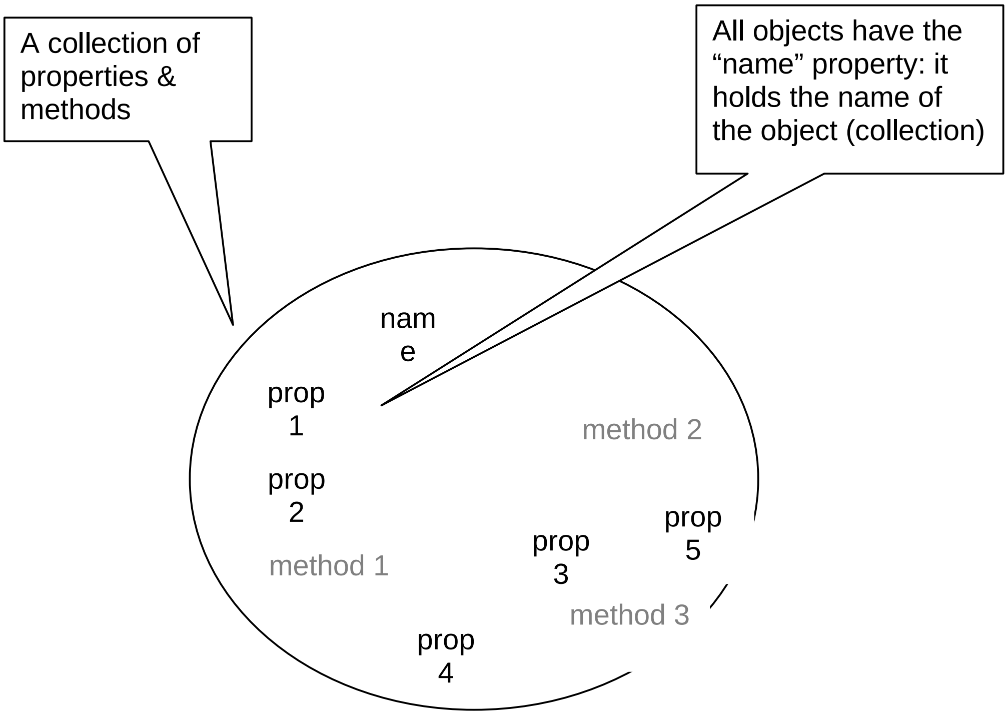
(Web Development Lecture 7)

- Everything that JavaScript manipulates, it treats as an *object* – e.g. a window or a button
- An object has *properties* – e.g. a window has size, position, status, etc.
- An object can be manipulated with *methods* that are associated with that object – e.g. a resize a window with *resizeTo(150, 200)*

Object: A *named* collection of properties (data, state) & methods (instructions, behavior)

During the last lecture

we had a discussion on Objects, Properties, Methods



A collection of properties & methods

All objects have the “name” property: it holds the name of the object (collection)

Types of Objects

- JavaScript objects
  - Objects that are part of JavaScript
  - Examples: window, document
- Browser objects
  - Objects that contain info *not* about the contents of the display, but the browser itself
  - Examples: history, navigator
- User-defined object

Object-Based, Not Object-Oriented!

- JavaScript is not a true object-oriented language like C++ or Java
- It is so because it lacks two key features:
  - A formal inheritance mechanism
  - Strong typing
- Nevertheless, JavaScript shares many similarities with object-oriented languages, and therefore is called an object-based language

The concept of objects and associated properties and methods is a very powerful idea, and we will be talking about it a lot during this course
However, today, our focus will be on some of the nitty-gritty details of JavaScript

During Today's Lecture …

- We will find out about JavaScript data types
- About variables and literals
- We will also discuss various operators supported by JavaScript

21.1 JavaScript Data Types

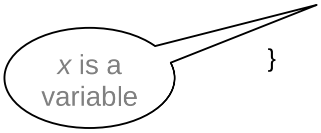
Unlike in C, C++ and Java, there are no explicit data types in JavaScript
Nevertheless, it recognizes & distinguishes among the following types of values:
Numbers, e.g., 23, 4.3, -230, 4.4e-24
Booleans, e.g., true, false
Strings, e.g., "hello", "What's the time?"
Undefined
We'll come back to these data types, but before that we have to define a few new terms
First, variables:

Variables

Variables give us the ability to manipulate data through reference instead of actual value.
Variables are names assigned to values.
Variables are containers that hold values (Example: Hotel guest name, Guest room no).
Generally, the value of a variable varies during code execution (that is why the term "variable.

Example

```
x = 1;
while (x < 6) {
    document.write (x);
    x = x + 1;
}
```



Try Doing the Same Without Using A Variable

5 lines of code replacing 5 lines of code!

Why use variables?

```
document.write (“1”); document.write (“2”);
document.write (“3”); document.write (“4”);
document.write (“5”);
```

Another Situation

```
x = 1;
while (x < 6000) {
       document.write (x);
       x = x + 1;
}
```

21.2 Declaring Variables

Many languages require that a variable be declared (defined) before it is first used

Although JavaScript allows variable declaration, it does not require it - except in the case when we want to declare a variable being local (more on local variables later in the course!)

However, it is good programming practice to declare variables before using them

Declaring Variables

var height

var name, address, phoneNumber

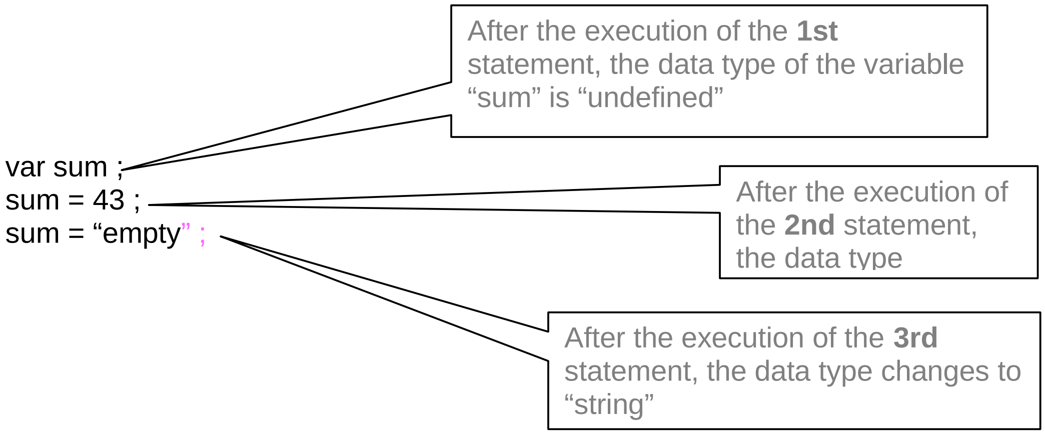
JavaScript Variables are *Dynamically Typed*

Any variable in JavaScript can hold any type of value, and that type can change midway through the program.

This is unlike the case for C, C++ and Java, where a variable’s type is defined before usage.

The untyped feature makes JavaScript simpler to program in when developing short programs. However, this feature brings in a few problems as well. Can you describe any?

JavaScript Variables are *Dynamically Typed*



```
var sum ;
sum = 43 ;
sum = “empty” ;
```

After the execution of the **1st** statement, the data type of the variable “sum” is “undefined”

After the execution of the **2nd** statement, the data type

After the execution of the **3rd** statement, the data type changes to “string”

Identifiers

- Identifiers are names used by JavaScript to refer to variables (as well as objects, properties, methods, and functions!)
- An identifier must begin with an alphabetical character (a-z or A-Z) or the underscore “_” character
- Subsequent characters can be an alphabetical (a-z or A-B) or numeric character (0-9) or an underscore

numberOneUniversity ,N99umber_one_University
_5numberoneuniversity,x,reallyReallyLongIndentifier12345678901234

Another Restriction on Identifiers

- Do not use any of the JavaScript keywords as identifiers
- For example, do not name a variable as “while”. When the browser sees this term in JavaScript code, it will get confused as it already knows this keyword as part of a loop statement. Same is the case for “var” or “if” or any of the other keywords.

JavaScript (Java) Reserved Words
Names that can't be used for variables, functions, methods, objects

finally	byte	import	throws	else
protected	goto	with	default	new
abstract	static	class	interface	var
float	case	in	transient	extends
public	if	this	do	null
Boolean	super	const	long	void
for	catch	instanceof	true	false
return	private	throw	double	package
break	switch	continue	native	while
function	char	int	try	final
synchronized		implements		????

Avoid These Special Names As Well (1)
Names that should not be used for variables, functions, methods, objects

close	confirm	assign	Window	JavaClass
History	Image	Form	java	onfocus
navigator	Number	location	onblur	Select
prompt	Radio	Package	Reset	Element
unescape	valueOf	sun	window	JavaObjec
closed	Date	blur	Docume	onload
history	isNaN	Frame	JavaArra	Self
netscape	Object	Math	onerror	untaint
prototype	ref	parent	scroll	taint
defaultStatus		clearTimeout		document

alert	Area	assign	Boolean	Checkbox
escape	FileUploa	Form	frames	getClass
status	Link	location	MimeType	navigate
onunload	opener	Package	parseFloa	Password
setTimeou	String	sun	Text	top
Anchor	Array	blur	Button	Submit
eval	focus	Frame	Function	Hidden
length	Location	Math	name	Navigator
open	Option	parent	parseInt	Plugin
JavaPackage		taint	Textarea	toString

Identifiers appear in JavaScript statements

Let us now discuss a few other elements that appear in those statements

Elements of JavaScript Statements

b = 2 ;
sum = sum + 49 ;
name = “Bhola” + “ Continental” ;
x = Math.floor (x)
	Identifiers
	Operators
	Literals
	Punctuation

JavaScript Literals

A data value that appears directly in a statement

Literals can be of several types. Some of them are:

Number

String

Boolean

Numeric Literals

24,-230000000000000000,9.80665,1.67e-27,

JavaScript stores all numbers, even integers, as floating-point numbers

String Literals

“” , ”Bhola” , “Where is the Bhola Continental Hotel?”

String literals are always enclosed in a matching pair of single or double quotes

Boolean Literals

True, false ,

if (tankFull == false)

addMoreWater = true

21.3 **JavaScript Operators**

Operators operate on operands to achieve the desired results

JavaScript has numerous operators, classified in many categories. We will look at only a few of them belonging to the following categories:

Assignment operators -- Arithmetic operators
Comparison operators -- String operators
Logical operators

We'll look at a few more during future lectures, but understand that there are many more. Even you text book does not cover all of them!

Assignment Operator "="

Changes the value of what is on the LHS, w.r.t. what is on the RHS

total_number_of_students = 984 ;
title = "Understanding Computers" ;
swapFlag = false ;
x = y + 33 ;

Arithmetic Operators
Multiply 2 * 4 → 8
Divide 2 / 4 → 0.5
Modulus 5 % 2 → 1
Add 2 + 4 → 6
Subtract 2 - 4 → -2
Negate -(5) → -5

21.4 Comparison Operators

Not the same as the assignment "=" operator

The "equal to (==)" Comparison Operator

```
if ( today == "Sunday" )
document.write("The shop is closed");
```

The string "The shop is closed" will be written to the document only if the variable today has a value equal to "Sunday"

Comparison Operators
a == b True if a and b are the same
a != b True if a and b are not the same
a > b True if a is greater than b
a >= b True if a is greater than or equal to b
a < b True if a is less than b
a <= b True if a is less than or equal to b

Example

```
if ( x != 0 )
    result = y / x;
else
    result = "not defined";
```

21.5 **Logical Operators**

a && b AND True if both are true
a || b OR True of either or both are true
!a NOT True if a is false

The "AND (&&)" Logical Operator

```
  if ( (pitch == "hard") && (bowler == "fast") )
              myStatus = "Pulled muscle";
```

The value of the variable myStatus will be set to "Pulled muscle" if both of the conditions are true

Example

```
if ( x || y )
```

```
        document.write (“Either or both are true”);
else
        document.write (“Both are false”);
```

So far we have looked at the assignment operator, arithmetic operators, comparison operators and logical operators

The final category that we are going to look at is string operators

In that category, we look at only one, the concatenation operator

The “+” String Operator

The “+” operator can be used to concatenate two strings

```
                title = “bhola” + “continental”
```

The value of the variable title becomes “bholacontinental”

21.6 Elements of JavaScript Statements

Semicolon ;

Terminate all JavaScript statements with a semicolon. It is not always necessary, but highly recommended.

b = 2;	Identifiers
sum = sum + 49;	Operators
name = “Bhola” + “ Continental”;	Literals
x = Math.floor (x);	Punctuation

White Spaces & Line Breaks

White spaces: The space & the tab characters

JavaScript ignores any extra white spaces or line breaks that you put in the code

This gives you the freedom of using them for making your code appear neat and readable

```
while ( x > 0) {
        remaind = x % 2;
        y = remaind + y;
 }
while ( x > 0) {remaind = x % 2; y = remaind + y;}
while ( x > 0) {
remaind = x % 2;
                                y = remaind + y;
                                }
```

Now let’s talk about a very special type of JavaScript statement that does not really do anything, but is found in most pieces of code!

Comments

Comments are included on a Web page to explain how and why you wrote the page the way you did

Comments can help someone other than the author to follow the logic of the page in the author’s absence

The commented text is neither displayed in the browser nor does it have any effect on the logical performance of the Web page, and is visible only when the actual code is viewed

JavaScript Comments

Single-line comments (two options)

```
        // Author: Bhola
        <!-- Creation Date: 24 March 2003
```

Multi-line comments

```
        /* Author: Bhola
           Creation Date: 24 March 2003 */
```

HTML Comments

```
<!-- Author: Bhola
        Creation Date: 24 March 2003 -->
```

Note : comments let the code speak for itself!
Comments add clarity

Decimal to Binary Conversion in JavaScript

```
x = 75 ;   // x is the decimal number
y = “” ;   // y is the binary equivalent
while ( x > 0) {
        remainder = x % 2 ;
        quotient = Math.floor( x / 2 ) ;
        y = remainder + y ;
        x = quotient ;
}
document.write(“y = ” + y) ;
```

During Today’s Lecture …

We found out about JavaScript data types

About variables and literals

We also discussed several operators supported by JavaScript

<u>Next (the 8th) Web Dev Lecture:</u>

Flow Control and Loops

To be able to understand the concept of flow control using the “if” and “switch” structures

To be able to understand the concept of behind the “while” and “for” looping structures

To be able to solve simple problems using flow control and loop statements

Lecture 22

Spreadsheets

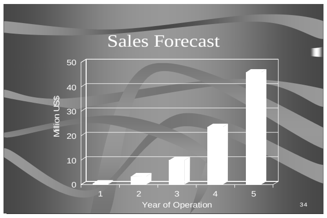
Today's Lecture:

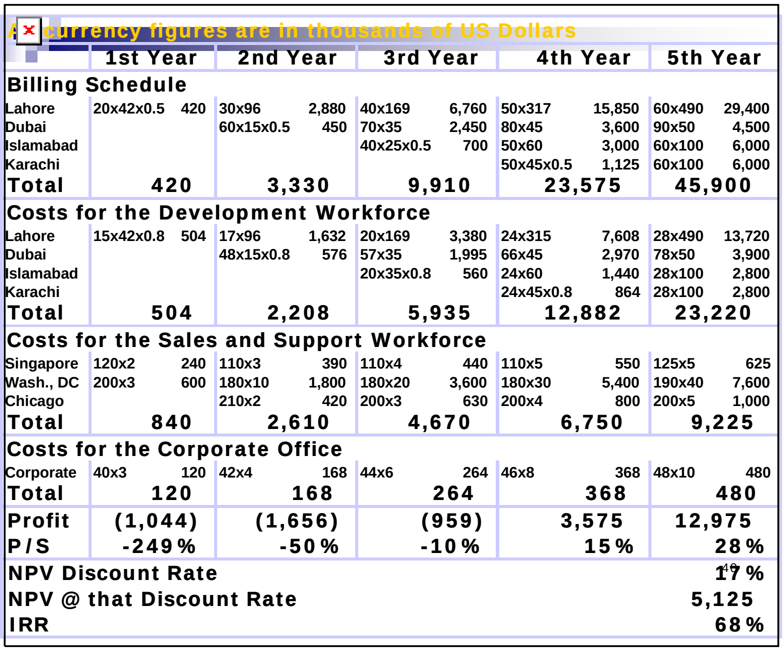
Spreadsheets

It was the first among the four lectures that we plan to have on productivity software
We learnt about what we mean by word processing and also desktop publishing
We also discussed the usage of various functions provided by common
Second among the four lectures that we plan to have on productivity software
This 2nd lecture is on spreadsheets
We'll learn about why we are interested in spreadsheets
We'll discuss the several common functions provided by popular spreadsheet SW programs

22.1 Business Plan for a New Software Development Company

The information provided in this business plan is confidential. Please do not disclose it without checking with me first. Thanks.





A currency figures are in thousands of US Dollars

	1st Year		2nd Year		3rd Year		4th Year		5th Year	
Billing Schedule										
Lahore	20x42x0.5	420	30x96	2,880	40x169	6,760	50x317	15,850	60x490	29,400
Dubai			60x15x0.5	450	70x35	2,450	80x45	3,600	90x50	4,500
Islamabad					40x25x0.5	700	50x60	3,000	60x100	6,000
Karachi							50x45x0.5	1,125	60x100	6,000
Total		**420**		**3,330**		**9,910**		**23,575**		**45,900**
Costs for the Development Workforce										
Lahore	15x42x0.8	504	17x96	1,632	20x169	3,380	24x315	7,608	28x490	13,720
Dubai			48x15x0.8	576	57x35	1,995	66x45	2,970	78x50	3,900
Islamabad					20x35x0.8	560	24x60	1,440	28x100	2,800
Karachi							24x45x0.8	864	28x100	2,800
Total		**504**		**2,208**		**5,935**		**12,882**		**23,220**
Costs for the Sales and Support Workforce										
Singapore	120x2	240	110x3	390	110x4	440	110x5	550	125x5	625
Wash., DC	200x3	600	180x10	1,800	180x20	3,600	180x30	5,400	190x40	7,600
Chicago			210x2	420	200x3	630	200x4	800	200x5	1,000
Total		**840**		**2,610**		**4,670**		**6,750**		**9,225**
Costs for the Corporate Office										
Corporate	40x3	120	42x4	168	44x6	264	46x8	368	48x10	480
Total		**120**		**168**		**264**		**368**		**480**
Profit		**(1,044)**		**(1,656)**		**(959)**		**3,575**		**12,975**
P/S		**-249%**		**-50%**		**-10%**		**15%**		**28%**
NPV Discount Rate										**17%**
NPV @ that Discount Rate										**5,125**
IRR										**68%**

40

Spreadsheets

Electronic replacement for ledgers
Used for automating engineering, scientific, but in majority of cases, business calculations
A spreadsheet - VisiCalc - was the first popular application on PC's.
It helped in popularizing PC's by making the task of financial-forecasting much simpler, allowing individuals to do forecasts which previously were performed by a whole team of financial wizard

What Can They Do? (1)

Can perform calculations repeatedly, accurately, rapidly
Can handle a large number of parameters, variables
Make it easy to analyze what-if scenarios for determining changes in forecasts w.r.t. change in parameters

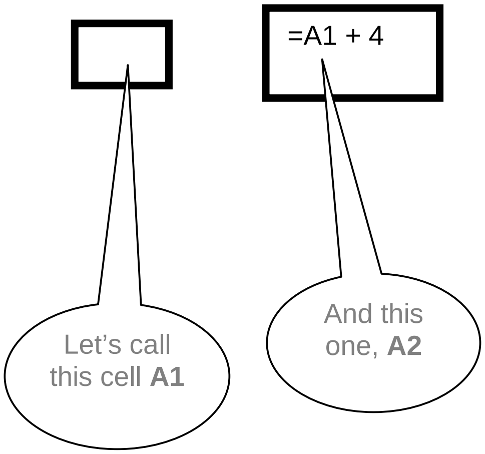
What Can They Do? (2)

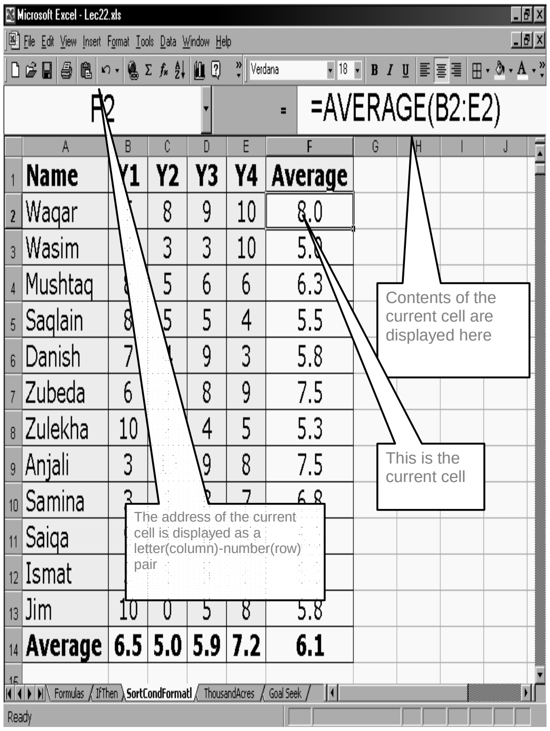
Are easy to interface with other productivity SW packages
Easy to store, recall, modify
Make it is easy to produce graphs:
Graphs reveal the knowledge contained in data with greater clarity and ease as compared with data arranged in rows and columns
Modern spreadsheet programs can be used to display data in a variety of graphical formats

22.2 The Structure of A Spreadsheet

Collection of cells arranged in rows and columns
Each cell can contain one of the following:Numbers
Text
Formulas
These cells display either the number or text that was entered in them or the value that is found by executing the formula

Connecting Two Cells





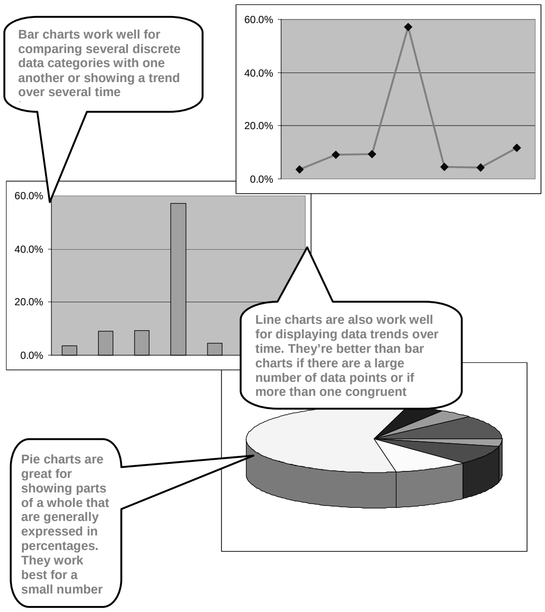
	A	B	C	D	E	F	G	H	I	J
1	Name	Y1	Y2	Y3	Y4	Average				
2	Waqar	[illegible]	8	9	10	8.0				
3	Wasim	[illegible]	3	3	10	5.[illegible]				
4	Mushtaq	8	5	6	6	6.3				
5	Saqlain	8	5	5	4	5.5				
6	Danish	7	4	9	3	5.8				
7	Zubeda	6	[illegible]	8	9	7.5				
8	Zulekha	10	[illegible]	4	5	5.3				
9	Anjali	3	[illegible]	9	8	7.5				
10	Samina	3	[illegible]	[illegible]	7	6.8				
11	Saiqa	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]				
12	Ismat	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]				
13	Jim	10	0	5	8	5.8				
14	Average	6.5	5.0	5.9	7.2	6.1				

Contents of the current cell are displayed here

This is the current cell

The address of the current cell is displayed as a letter(column)-number(row) pair

All currency figures are in thousands of US Dollars										
	1st Year		2nd Year		3rd Year		4th Year		5th Year	
Billing Schedule										
Lahore	20x42x0.5	420	30x96	2,880	40x169	6,760	50x317	15,850	60x490	29,400
Dubai			60x15x0.5	450	70x35	2,450	80x45	3,600	90x50	4,500
Islamabad					40x25x0.5	700	50x60	3,000	60x100	6,000
Karachi							50x45x0.5	1,125	60x100	6,000
Total	420		3,330		9,910		23,575		45,900	
Costs for the Development Workforce										
Lahore	15x42x0.8	504	17x96	1,632	20x169	3,380	24x315	7,608	28x490	13,720
Dubai			48x15x0.8	576	57x35	1,995	66x45	2,970	78x50	3,900
Islamabad					20x35x0.8	560	24x60	1,440	28x100	2,800
Karachi							24x45x0.8	864	28x100	2,800
Total	504		2,208		5,935		12,882		23,220	
Costs for the Sales and Support Workforce										
Singapore	120x2	240	110x3	390	110x4	440	110x5	550	125x5	625
Wash., DC	200x3	600	180x10	1,800	180x20	3,600	180x30	5,400	190x40	7,600
Chicago			210x2	420	200x3	630	200x4	800	200x5	1,000
Total	840		2,610		4,670		6,750		9,225	
Costs for the Corporate Office										
Corporate	40x3	120	42x4	168	44x6	264	46x8	368	48x10	480
Total	120		168		264		368		480	
Profit	(1,044)		(1,656)		(959)		3,575		12,975	
P/S	-249%		-50%		-10%		15%		28%	
NPV Discount Rate									17%	
NPV @ that Discount Rate									5,125	
IRR									68%	



22.3 Goal Seek

Goal Seek in Excel

When you use the Goal Seek command, Excel changes the value in one cell until the value in a second cell reaches a number that you desire. For instance, if you had a spreadsheet that calculated profit for the Bhola eService from a variety of inputs, including employee numbers, expenses, products sold, price of products, you might use goal seek to define your break-even price of products. You would tell the computer to change price of products until Profit was zero (break-even), and you would do that using Tools, Goal Seek.

To use Goal Seek, go to the Tools command. If Goal seek . . . is not an option, you must first go to Add-ins (also under Tools), and select Goal Seek. Once Goal Seek is loaded, choose it under Tools.

In Goal Seek there will be three boxes to fill in.

The first says "Set cell." Enter the cell address (or click on the cell) of the cell whose value you want to fix or set to a specific number (i.e. Profit cell). This cell must contain a formula or function. Otherwise it will not be linked to the cell you will be changing to obtain zero profit.

The second says "To value." Enter the appropriate value you wish to see in that "Set" cell (i.e. 0 if you want the Profit to come out zero).

The third says "By changing cell." Enter or click on the cell you want Goal Seek to change to obtain the zero profit. (i.e. milk price). This cell must not be a formula or function. Then click "okay."

At this point Goal Seek will show you the answer. For instance, Profit will now be zero and the Milk Price cell will have changed to another price (maybe 11.86) to make Profit=0. You can accept the change or you can cancel the Goal Seek and return to the previous numbers. Often you just want to take note of the new numbers and cancel. If you accept and change your mind, click Undo.

Things that you must remember!!

Make sure the "Set Cell" cell is a formula or function or cell reference.

Make sure you have set that sell to a reasonable number.

Make sure the "By Changing Cell" cell is a number or blank, and not a formula, function or cell reference like =C5.

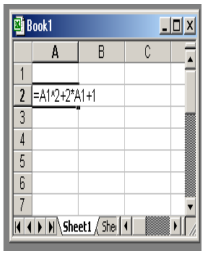
Make sure there is a link by formulas between the two cells you entered in the Goal Seek. However complicated the link might be, they must be related for the Set cell to be changed by the Change cell.
Finally, make sure your formula in the "Set Cell" cell is correct (as well as all others).

Simple Example

Assume the following cells. We will use Goal Seek to find a number to make the sum=150.
A2 = 25
A3 = 40
A4 = SUM(A2:A3) which is showing 65
In Goal Seek:
Set Cell: click on A4
To Value: enter 150
By Changing Cell: click on A3
The sum in A4 should now be 150, and A3 should have become 125 for that to happen.

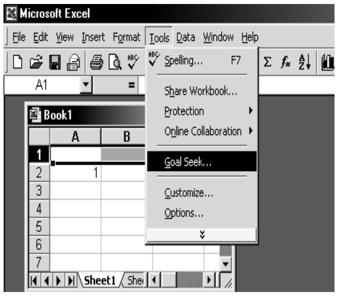
Solving Equation: *f(x) = x2 + 2x + 1 = 0*

- Write the formula in a cell e.g. A2

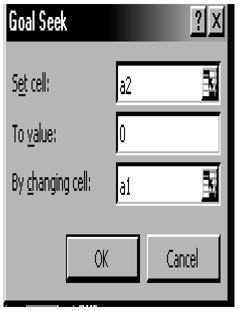


=A1^2+2*A1+1

- Select the goal seek option



- In the ‘set cell’ input field write the cell number that needs to be changed I.e. a2
- In the ‘to value’ field enter the value we want the cell a2 to have i.e. 0

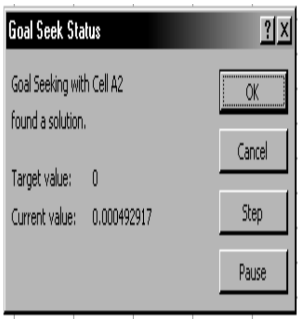


Set cell: a2

To value: 0

By changing cell: a1

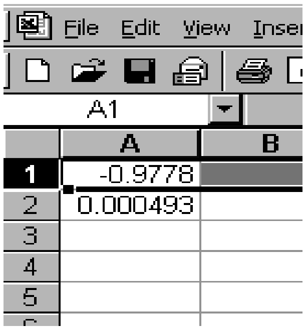
- This shows the that the target was to have 0 value but excel could calculate for 0.0004 value
- On pressing Ok we will get->
- Here the value of a1 is -0.97 which is almost equal



Goal Seeking with Cell A2 found a solution.

Target value: 0

Current value: 0.000492917



	A	B
1	-0.9778	
2	0.000493	
3		
4		
5		

Hence to get the value of the given function as 0 the value of x should be 1
Which is the solution of the equation

$f(x) = x^2 + 2x + 1 = 0$

links

Following are some urls for the goal seek ;
http://www.oootraining.com/QwikAndDirty/QwikAndDirtyExcelWeb/DataAnalysis/Using_Goal_Seek/Using_Goal_Seek.htm

The Best Feature: Undo
Allows you to recover from your mistakes
Allows you to experiment without risk
Getting On-Screen Help
All spreadsheets generally have some form of built-in help mechanism
To me, it seems like that many of those help-systems are designed to be “not-very-helpful”: they make finding answers to simple questions quite difficult
Nevertheless, do try them when you are searching for answers

I’ll now demonstrate the use of spreadsheets with the help of several examples

Formulas
Sorting
Conditional formatting
Graphs
Goal seek

Today's Lecture was the …
Second among the four lectures that we plan to have on productivity software
This 2nd lecture was on spreadsheets
We learnt about what we mean by spreadsheets
We discussed the usage of various functions provided by common spreadsheets
Focus of the Next Productivity SW Lecture: Presentations
To become familiar with the basics of multimedia presentations
To become able to develop simple presentation with the help of presentation software

Lecture 23

Flow Control & Loops

(Web Development Lecture 8)

During the last lecture we had a discussion on Data Types, Variables & Operators

We found out about JavaScript data types

About variables and literals

We also discussed several operators supported by JavaScript

JavaScript Data Types

JavaScript recognizes & distinguishes among the following types of values:

Numbers

Booleans

Strings

Undefined

Variables

Variables give us the ability to manipulate data through reference instead of actual valueVariables are containers that hold values

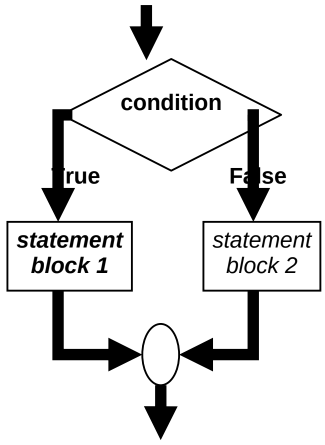
Declaring Variables

Although JavaScript allows variable declaration, it does not require it - except in the case when we want to declare a variable being local (more on local variables later in the course!)

JavaScript Variables are Dynamically Typed

Any variable in JavaScript can hold any type of value, and the that type can change midway through the program

FLOW CONTROL
Select between alternate courses of action depending upon the evaluation of a condition



JavaScript Flow Control Structures

- if … else
- switch

if: Example 1

```
if  ( day == “Sunday” )
                bhola = “Cool” ;
```

semicolon

The condition enclosed in

Set the value of the variable ‘bhola to ‘Cool’ if the ‘day’ is equal to ‘Sunday’

This was the case if we want to execute a single statement given that the condition is true
What if we want to execute multiple statements in case the

if: Example 2

```
if  ( day == “Sunday” )        {
        bhola = “Cool” ;
        mood = “Great” ;
        clothing = “Casual” ;
  }
```

Set the value of the variable ‘bhola to ‘Cool’, ‘mood’ to ‘Great’, and ‘clothing’ to ‘casual’ if the

These curly braces group the multiple statements into a single compound statement

if: Example 2

```
if  ( day == “Sunday” )  {
        bhola = “Cool” ;
        mood = “Great” ;
        clothing = “Casual” ;
}
```

Note: No semicolon after the closing curly brace

Set the value of th
‘mood’ to ‘G , and ‘clot e
‘day’ is equal to ‘Sunday’

Compound Statements

- At times, we need to put multiple statements at places where JavaScript expects only oneFor those situations, JavaScript provides a way of grouping a number of statements into a

2. This is done simply by enclosing any number of statements within curly braces, { }NOTE: Although the statements within the block end in semicolons, the block itself doesn’t

if: Example 3

```
if  ( (day == “Sunday”) || (day == “Saturday”) )  {
        bhola = “Cool” ;
        mood = “Great” ;
        clothing = “Casual” ;
}
```

if: Example 4

```
weekend = ( day == “Sunday” ) || ( day ==     “Saturday” ) ;

if  ( weekend )  {
          bhola = “Cool” ;
          mood = “Great” ;
          clothing = “Casual” ;
}
```

What is the data type of the variable “weekend”?

We now know how to execute a statement or a block of statements given that the condition is true
What if we want to include an alternate action as well, i.e. a statement or a block of statements to be executed in case the condition in not true

if … else: Example 1

```
if ( GPA >= 1.0 )
          bhola = “Pass” ;
else
          bhola = “Fail” ;
```

if … else: Example 2

```
if ( GPA >= 1.0 ) {
          bhola = “Pass” ;
}
Else
          bhola = “Fail” ;
```

if … else: Example 3

```
if ( GPA >= 1.0 ) {
          bhola = “Pass” ;
          mood = “Great”
          ;
} else
```

if … else: Example 4

```
if ( GPA >= 1.0 ) {
          bhola = “Pass” ;
          mood = “Great” ;
} else {
          bhola = “Fail” ;
          mood = “Terrible” ;  }
```

if … else: Example 5

```
if ( grade == “A” )
                points = 4.0 ;
if ( grade == “B” )
                points = 3.0 ;
if ( grade == “C” )
                points = 2.0 ;
if ( grade == “D” )
                points = 1.0 ;
if ( grade == “F” )
                points = 0.0 ;
```

This piece of code is correct, but not very efficient!

What can we do to improve it?

if … else: Example 5

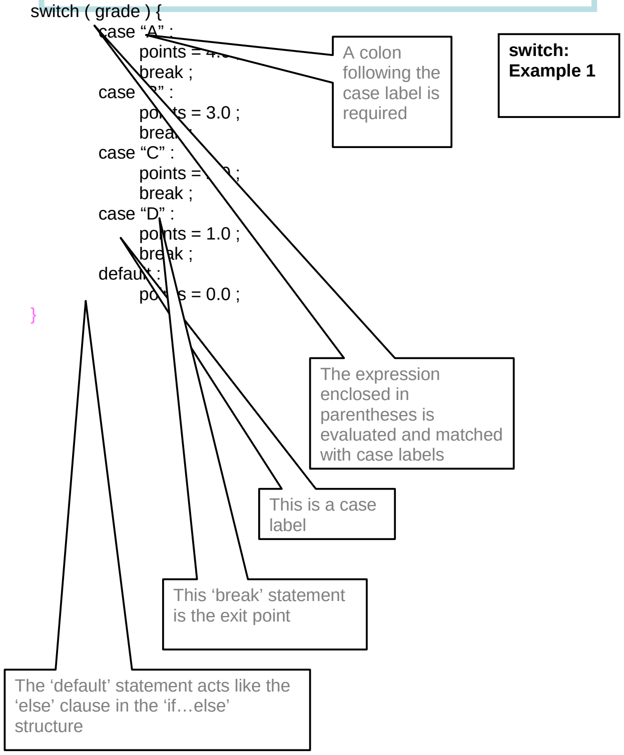
```
if ( grade == “A” )
        points = 4.0 ;
if ( grade == “B” )
        points = 3.0 ;
if ( grade == “C” )
        points = 2.0 ;
if ( grade == “D” )
        points = 1.0 ;
```

This piece of code is correct, but not very efficient!

What can we do to improve it?

if … else: Example 6

```
if ( grade == “A” )
        points = 4.0 ;
else {
        if ( grade == “B” )
                points = 3.0 ;
        else {
                if ( grade == “C” )
                        points = 2.0 ;
                else {
                        if ( grade == “D” )
                                points = 1.0 ;
                        else
                                points = 0.0 ;
                }
        }
}
```



switch: Example 1

```
switch ( grade ) {
        case “A” :
                points = 4.0 ;
                break ;
        case “B” :
                points = 3.0 ;
                break ;
        case “C” :
                points = 2.0 ;
                break ;
        case “D” :
                points = 1.0 ;
                break ;
        default :
                points = 0.0 ;
}
```

A colon following the case label is required

The expression enclosed in parentheses is evaluated and matched with case labels

This is a case label

This ‘break’ statement is the exit point

The ‘default’ statement acts like the ‘else’ clause in the ‘if…else’ structure

Switch Example 2

switch: Example 2

```
switch ( inquiry ) {
        case “apple” :
              document.write( “Apples are Rs 50/kg” ) ;
              break ;
        case “mangos” :
              document.write( “Mangos are Rs 90/kg” ) ;
              break ;
        case “grapes” :
              document.write( “Grapes are Rs 60/kg” ) ;
              break ;
        default :
              document.write( inquiry + “? Please retry!” ) ;
}
```

if…else --?-- switch

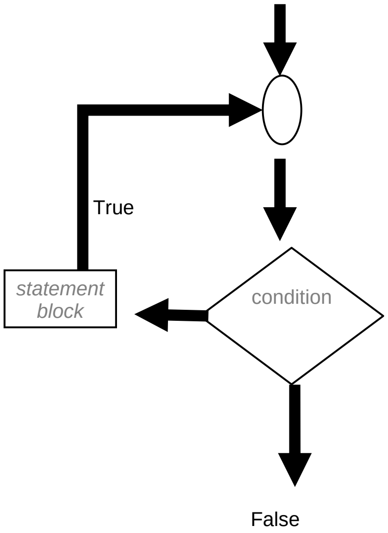
- If the action to be taken of the value of a single variable (or a single expression), use ‘switch’
- When the action depends on the values of multiple variables (or expressions), use the ‘if...else’ structure

if … else: Example 7

```
if ( ( GPA >= 1.0 ) && ( attendance >= 40 ) )

        bhola = “Pass” ;

else {

        if ( ( GPA >= 2.0 ) && ( attendance >= 36 ) )          bhola
        = “Probation” ;

        else

              bhola = “Fail” ;

}
```

LOOPS

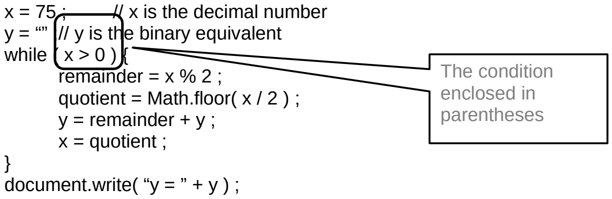
Loop through a set of statements as long as a condition is true



JavaScript’s Looping Structures

while
for
…

Decimal to Binary Conversion in JavaScript



```
x = 75 ;        // x is the decimal number
y = “”  // y is the binary equivalent
while ( x > 0 ) {
        remainder = x % 2 ;
        quotient = Math.floor( x / 2 ) ;
        y = remainder + y ;
        x = quotient ;
}
document.write( “y = ” + y ) ;
```

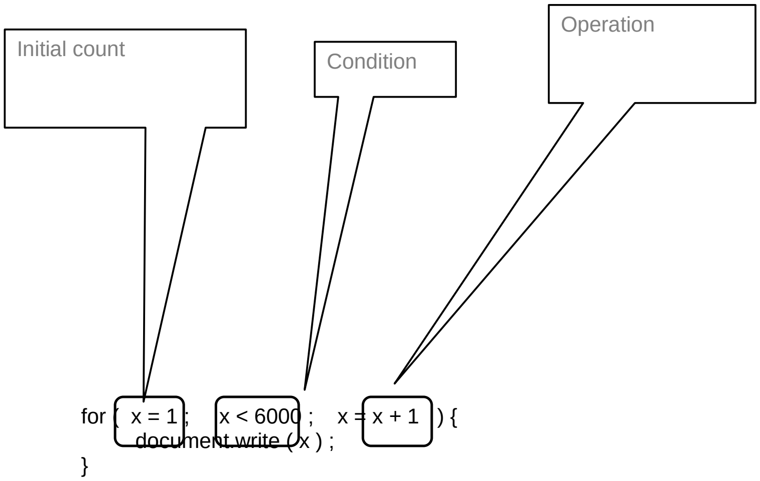
while: Example 2

```
while ( tankIsFull == false ) {
        tank = tank + bucket ;
}
document.write ( “Tank is full now” ) ;
```

while: Example 3

```
x = 1 ;
while ( x < 6000 ) {
        document.write ( x ) ;
        x = x + 1 ;
}
```

for: Example 1



```
for (  x = 1 ;    x < 6000 ;    x = x + 1  ) {
        document.write ( x ) ;
}
```

for: Description (1)

- The ‘for’ loop starts by initializing the *counter variable* (which in this case is x)
- The initial value in this case is ‘1’, but can be any other positive or negative number as well
- Next the ‘for’ loop checks the condition. If the condition evaluates to a ‘true’ value, the ‘for’ loop goes through the loop once

for: Description (2)

- After reaching the end of that iteration, the ‘for’ loop goes to the top once again, performs the operation, checks the condition
- If the condition evaluates to a ‘false’ value, the ‘for’ loop finishes looping
- Otherwise, the ‘for’ loop goes through the loop once again
- Repeat from step 4

for: Example 2

```
for ( x = 99 ; x < 6000 ; x = x + 1 ) {
        document.write ( x ) ;
}
```

for: Example 3

```
for ( x = 6000 ; x > 0 ; x = x - 1 ) {
        document.write ( x ) ;
}
```

How many iterations would this ‘for’ loop run for?

6000?

for: Example 3

```
for ( x = 6000 ; x > 0 ; x = x - 1 ) {
        document.write ( x ) ;
}
```

How many iterations would this ‘for’ loop run for?

6000?

for: Example 4

```
for ( x = 6000 ; x < 0 ; x = x - 1 ) {
        document.write ( x ) ;
}
```

How many iterations would this ‘for’ loop run for?

None?

for --?-- while

- When the exact number of iterations is known, use the ‘for’ loop

‘for’ loops become especially useful when used in conjunction with arrays

We’ll find out about arrays next time, and we’ll probe their usefulness as part of ‘for’ loop structures

During Today's Lecture …

We discussed the concept of flow control using the "if" and "switch" structures
And also the concept behind the "while" and "for" looping structures
We also solved simple problems using flow control and loop structures

Next (the 9th) Web Dev Lecture: Arrays

We will find out why we need arrays
We will become able to use arrays for solving simple problems

Lecture 24

Design Heuristics

During the last lecture …

We became familiar with the various phases of the process that developers follow to develop SW systems of reasonable complexity

We looked at a couple of problems related to the Waterfall SW development model

Today's Lecture

Heuristics for System Architecting

We will try to understand the role of heuristics in architectural (or high-level) design

We will become familiar with a few popular design heuristics

24.1 Heuristic

Rule of thumb learned through trial & error

Common sense lesson drawn from experience

Qualitative principle, guideline, general judgement

Natural language description of experience

24.2 System

A collection of elements which working together produces a result not achieved by the things alone

24.3 System Architecture

The structure

(in terms of components, connections, constraints) of a product or a process

24.4 Heuristics for system architecting

Rules and lessons learnt by system architects after long experiences which when followed result in sound, stable, practical systems

# 1 My favorite system architecting (and other relevant) heuristics --- in no particular order ---

#2 Given many parts of a system to be designed/built, do the hard part 1st

# 3 All the serious mistakes are made on the very first day

# 4 Simplify, simplify, simplify!

Probably the most useful heuristics for increasing reliability while decreasing cost & time-to-build

# 5 If you can't explain it in 5 minutes, either you don't understand it or it does not work

# 6 A system will develop & evolve much more rapidly if there are stable intermediate forms than if there

Build iteratively; add features gradually

# 7 Success is defined by the user, not the builder

# 8 It's more important to know what the customer needs instead of what he says he wants

# 9 If you think that your design is perfect, it is only because you have not shown to anyone else

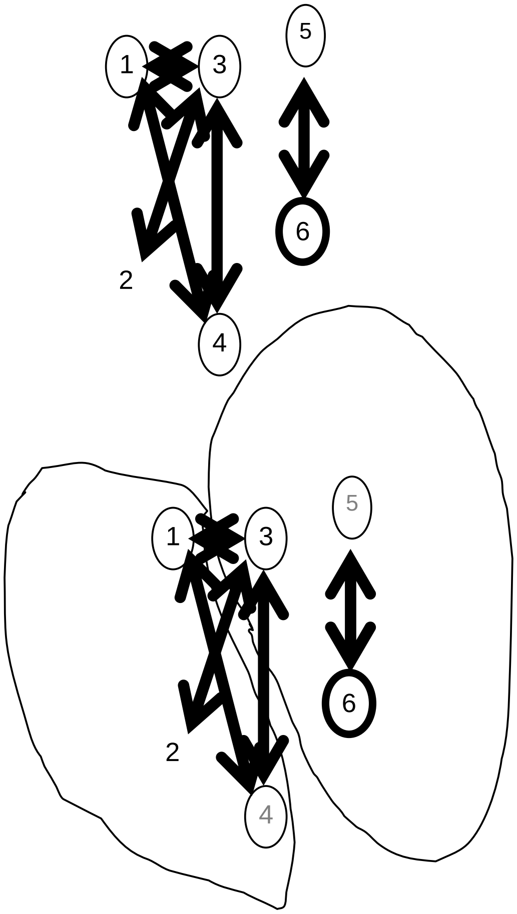
--- Get your designs reviewed ---

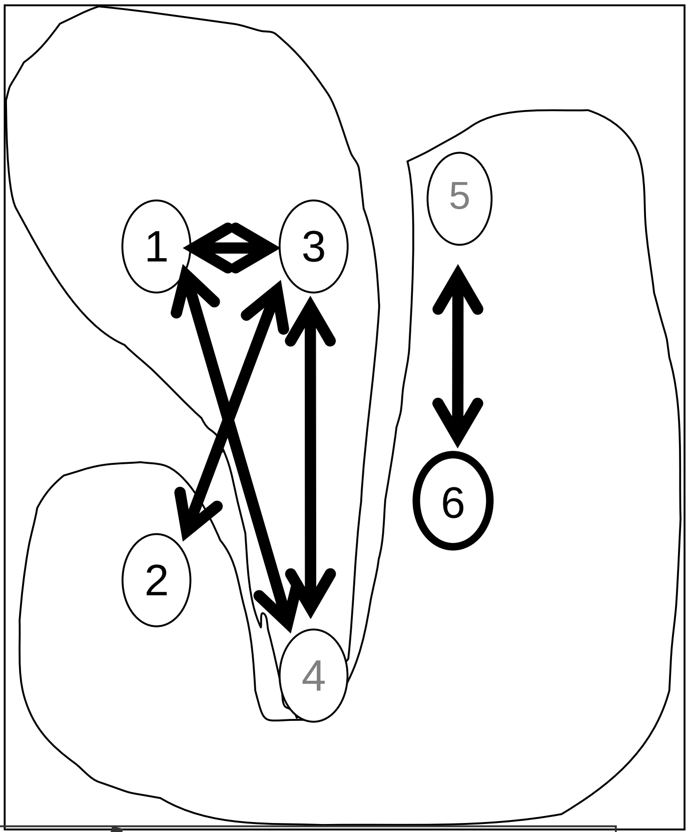
# 10 A good solution to a problem somehow looks nice & elegant

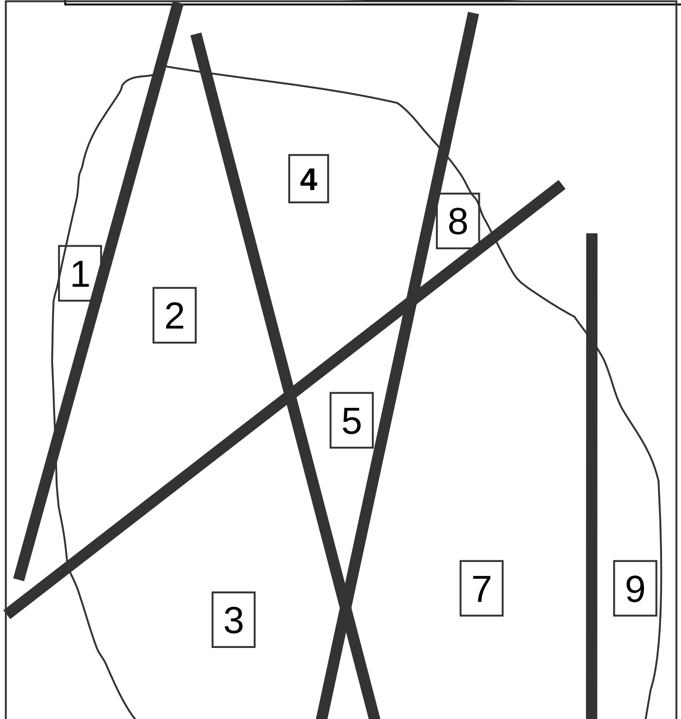
# 11 In partitioning, choose the chunks so that they are as independent as possible

Chunks should have low external complexity & high internal complexity

Organize personal tasks to minimize the time individuals face interfacing


5
1
3
6
2
4
5
1
3
6
2
4


5
1
3
6
2
4


4
8
1
2
5
7
9
3
6

# 12 Partition/repartition the problem until a model consisting of 7±2
chunks emerges
# 13 When choices must be made with unavoidably inadequate info:
Choose the best available & then watch to see:
whether further solutions appear faster than future problems .
If so, the choice was at least adequate .
If not, go back & choose again
**14** The Triage
1. Let the dying die
2. Ignore who'll recover on their own
3. Treat only those who'll die without your help
#**15** Don't just remove the defect; correct the process that caused it
# 16 The number of defects remaining in a system after a given level of tests is proportional to the number found during the test
# 17 Programmers deliver the same number of LOC/day regardless of the language they are writing in .
Use the Highest level Language
In Today's Lecture
We became familiar with the the role of heuristics in design
We also discussed a few well-known design heuristics for architectural design
In Today's Lecture
We became familiar with the the role of heuristics in design
We also discussed a few well-known design heuristics for architectural design
Next Lecture:

Web Design for Usability
To become able to appreciate the role of usability in Web design
To become able to identify some of the factors affecting the usability of a Web page

Lecture 25

Web Design for Usability

During the last lecture …:

We looked at the role of heuristics in architectural (or high-level) design
We also became familiar with a few popular design heuristics

Heuristic:

Rule of thumb learned through trial & error
Common sense lesson drawn from experience

Caution! Caution!

Heuristics don't always lead to the best results

At times they even lead to the wrong ones, but mostly to results that are good-enough

25.1 USABILITY

Today's Goal:

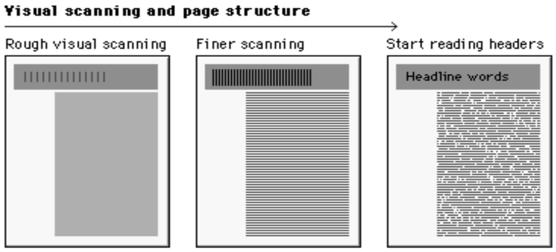
25.1 Web Design for Usability

To become able to appreciate the role of usability in Web design
To become able to identify some of the factors affecting the usability of a Web page
What's a Good Site?

The one that achieves the result that it was designed for. Generally, that result can only be achieved by giving the user what s/he wants, as quickly as possible, without her/him expending much effort. One definition of usability: Let the user have what s/he wants quickly, without much effort. "Quickly" is important!

25.2 SPEED:

Users don't read; they scan
Users don't make optimal choices; they look for the first good-enough solution
Users don't figure out how things work; they muddle through



Design is Important!

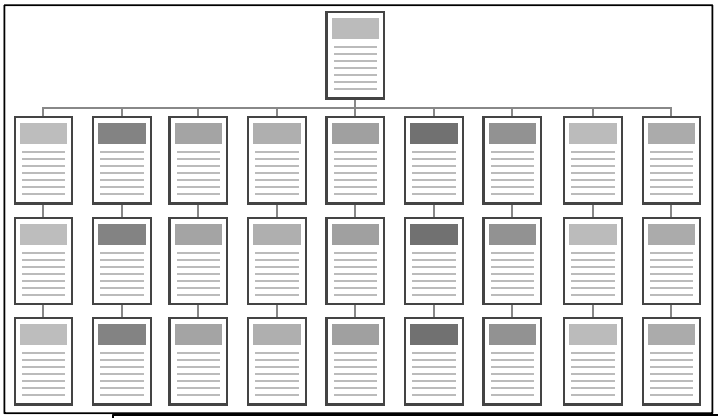
62% of shoppers gave up looking for the item they wanted to buy online (Zona Research)

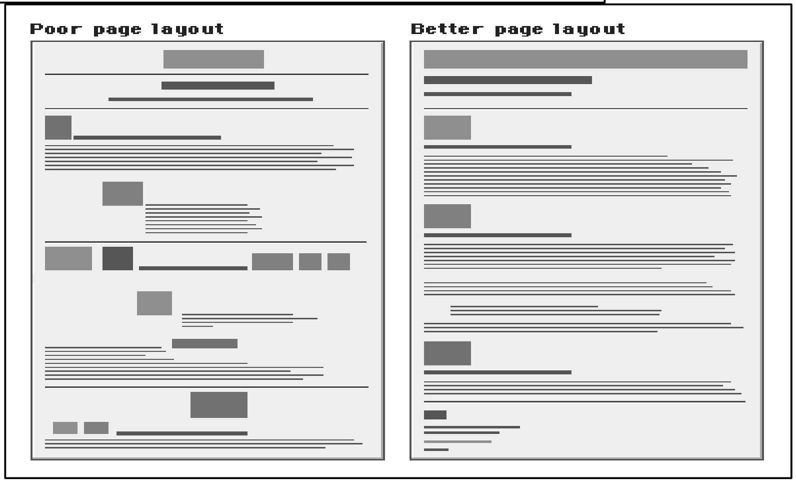
40% visitors don't return to a site if their first visit was a -ive experience (Forrester Research)

83% of users have left sites in frustration due to poor navigation, slowness (NetSmart Research)

Simple designs have greater impact: they can be understood immediately! (Mullet/Sano)

Designs should be consistent & predictable (unified)





25.3 Elements of Website Design:

Navigation scheme
Layout of information
Overall look and feel

25.4 Website Navigation:

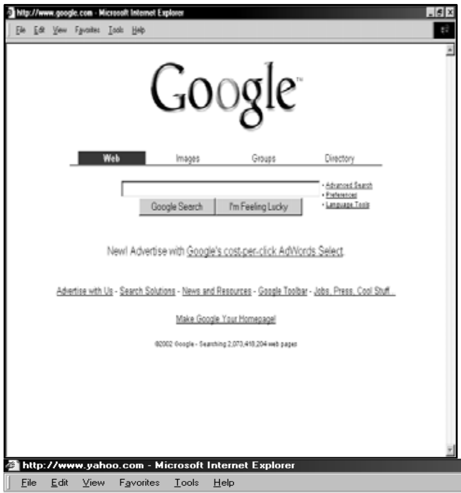
The interface/controls that a Website provides to the user for accessing various parts of the Website
It probably is the most important aspect of the design of a Website

25.5 A Few Navigation Design Heuristics:

Put the main navigation on the left of the page
It should be 'invisible' until it is wanted
It should require an economy of action & time
It should remain consistent
Use text for navigation labels. If you must use icons, put a description underneath each icon

25.6 Navigation Design Heuristics (contd.):

Labels should be clear, understandable
Labels should be legible
Do not play with standard browser buttons & features
Provide search capability

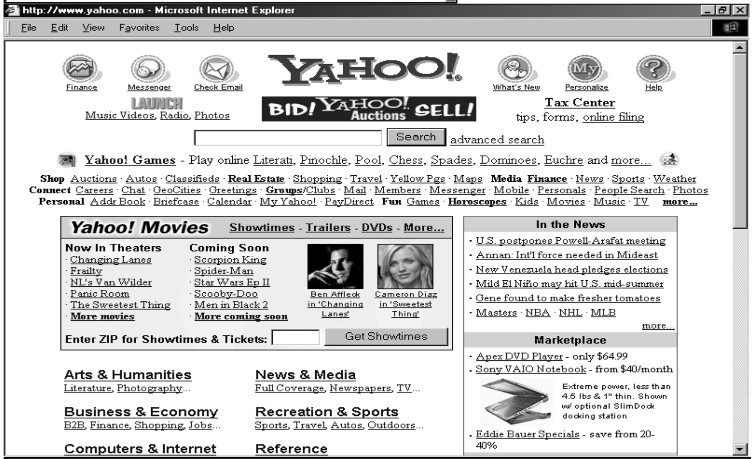


New! Advertise with Google's cost-per-click AdWords Select

Advertise with Us - Search Solutions - News and Resources - Google Toolbar - Jobs, Press, Cool Stuff...

Make Google Your Homepage!

©2002 Google - Searching 2,073,418,204 web pages


http://www.yahoo.com - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Finance
Messenger
Check Email
YAHOO!
What's New
Personalize
Help
LAUNCH
BID! YAHOO! Auctions SELL!
Search
Ben Affleck in 'Changing Lanes'
Cameron Diaz in 'Sweetest Thing'
Get Showtimes

Music Videos, Radio, Photos

Tax Center
tips, forms, online filing

advanced search

Yahoo! Games - Play online Literati, Pinochle, Pool, Chess, Spades, Dominoes, Euchre and more...

Shop Auctions · Autos · Classifieds · **Real Estate** · Shopping · Travel · Yellow Pgs · Maps **Media** **Finance** · News · Sports · Weather
Connect Careers · Chat · GeoCities · Greetings · **Groups**/Clubs · Mail · Members · Messenger · Mobile · Personals · People Search · Photos
Personal Addr Book · Briefcase · Calendar · My Yahoo! · PayDirect **Fun** Games · **Horoscopes** · Kids · Movies · Music · TV **more...**

Yahoo! Movies Showtimes - Trailers - DVDs - More...

Now In Theaters	Coming Soon
· Changing Lanes	· Scorpion King
· Frailty	· Spider-Man
· NL's Van Wilder	· Star Wars Ep II
· Panic Room	· Scooby-Doo
· The Sweetest Thing	· Men in Black 2
· **More movies**	· **More coming soon**

Enter ZIP for Showtimes & Tickets:

In the News

- U.S. postpones Powell-Arafat meeting
- Annan: Int'l force needed in Mideast
- New Venezuela head pledges elections
- Mild El Niño may hit U.S. mid-summer
- Gene found to make fresher tomatoes
- Masters · NBA · NHL · MLB

more...

Marketplace

- Apex DVD Player - only $64.99
- Sony VAIO Notebook - from $40/month

Extreme power, less than 4.5 lbs & 1" thin. Shown w/ optional SlimDock docking station

- Eddie Bauer Specials - save from 20-40%

Arts & Humanities
Literature, Photography...

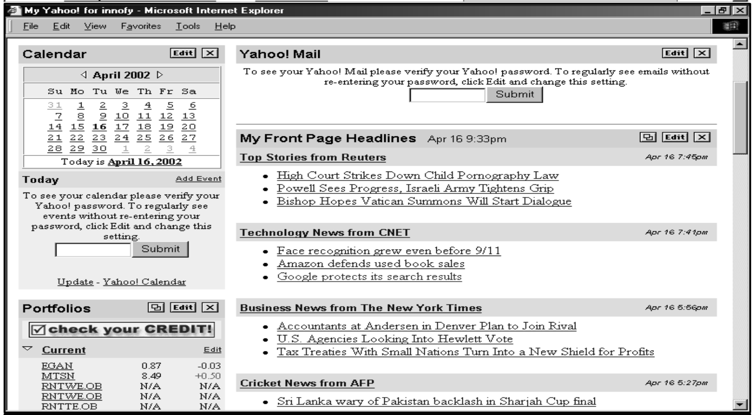
News & Media
Full Coverage, Newspapers, TV...

Business & Economy
B2B, Finance, Shopping, Jobs...

Recreation & Sports
Sports, Travel, Autos, Outdoors...

Computers & Internet

Reference


My Yahoo! for innofy - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Edit
Submit
Add Event
check your CREDIT!

Calendar

April 2002

Su	Mo	Tu	We	Th	Fr	Sa
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

Today is April 16, 2002

Today

To see your calendar please verify your Yahoo! password. To regularly see events without re-entering your password, click Edit and change this setting.

Update - Yahoo! Calendar

Portfolios

Current Edit

EGAN	0.87	-0.03
MTSN	8.49	+0.50
RNTWE.OB	N/A	N/A
RNTWE.OB	N/A	N/A
RNTTE.OB	N/A	N/A

Yahoo! Mail

To see your Yahoo! Mail please verify your Yahoo! password. To regularly see emails without re-entering your password, click Edit and change this setting.

My Front Page Headlines Apr 16 9:33pm

Top Stories from Reuters *Apr 16 7:45pm*

- High Court Strikes Down Child Pornography Law
- Powell Sees Progress, Israeli Army Tightens Grip
- Bishop Hopes Vatican Summons Will Start Dialogue

Technology News from CNET *Apr 16 7:41pm*

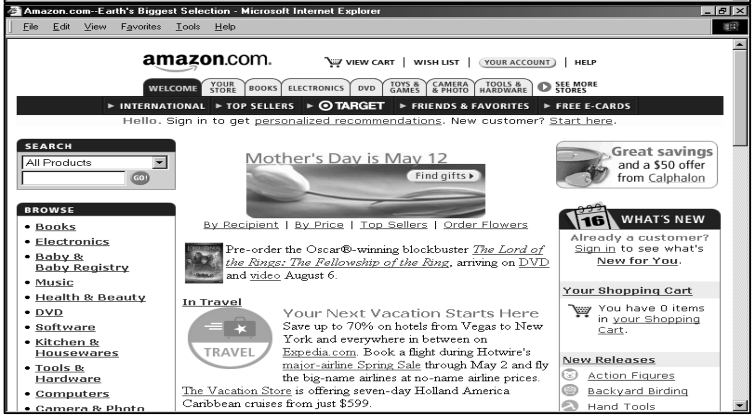
- Face recognition grew even before 9/11
- Amazon defends used book sales
- Google protects its search results

Business News from The New York Times *Apr 16 5:56pm*

- Accountants at Andersen in Denver Plan to Join Rival
- U.S. Agencies Looking Into Hewlett Vote
- Tax Treaties With Small Nations Turn Into a New Shield for Profits

Cricket News from AFP *Apr 16 5:27pm*

- Sri Lanka wary of Pakistan backlash in Sharjah Cup final



Hello. Sign in to get personalized recommendations. New customer? Start here.

BROWSE

- Books
- Electronics
- Baby & Baby Registry
- Music
- Health & Beauty
- DVD
- Software
- Kitchen & Housewares
- Tools & Hardware
- Computers
- Camera & Photo

By Recipient | By Price | Top Sellers | Order Flowers

Pre-order the Oscar®-winning blockbuster *The Lord of the Rings: The Fellowship of the Ring*, arriving on DVD and video August 6.

In Travel

Your Next Vacation Starts Here
Save up to 70% on hotels from Vegas to New York and everywhere in between on Expedia.com. Book a flight during Hotwire's major-airline Spring Sale through May 2 and fly the big-name airlines at no-name airline prices. The Vacation Store is offering seven-day Holland America Caribbean cruises from just $599.

Already a customer? Sign in to see what's New for You.

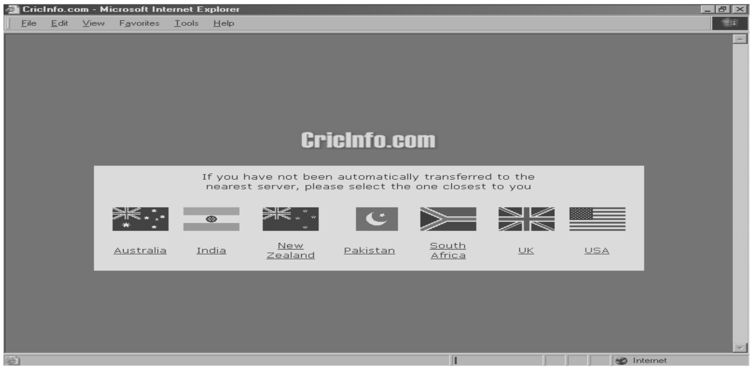
Your Shopping Cart

You have 0 items in your Shopping Cart.

New Releases

- Action Figures
- Backyard Birding
- Hand Tools

A good
Solution to
Problem
Is nice and
elegent

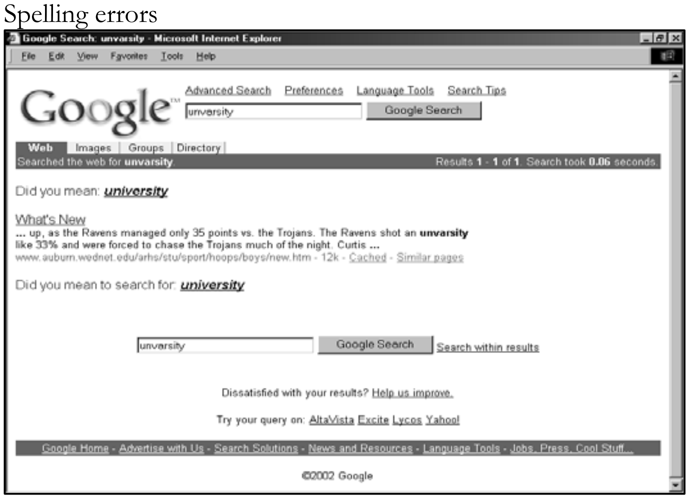

CricInfo.com - Microsoft Internet Explorer
File Edit View Favorites Tools Help
CricInfo.com
Australia
India
New Zealand
Pakistan
South Africa
UK
USA
Internet

If you have not been automatically transferred to the nearest server, please select the one closest to you

25.7 Good designs assist the user in recovering from errors

25.8 Assisting the User Recover from Errors:

Location, post code mismatch
Credit card number errors
Phone numbers
Spelling errors


Google Search: unvarsity - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Google™
Advanced Search Preferences Language Tools Search Tips
unvarsity
Google Search
Web Images Groups Directory
unvarsity
Google Search
Search within results
Google Home - Advertise with Us - Search Solutions - News and Resources - Language Tools - Jobs, Press, Cool Stuff...

Searched the web for **unvarsity**. Results **1** - **1** of **1**. Search took **0.06** seconds.

Did you mean: ***university***

What's New
... up, as the Ravens managed only 35 points vs. the Trojans. The Ravens shot an **unvarsity** like 33% and were forced to chase the Trojans much of the night. Curtis ...
www.auburn.wednet.edu/arhs/stu/sport/hoops/boys/new.htm - 12k - Cached - Similar pages

Did you mean to search for: ***university***

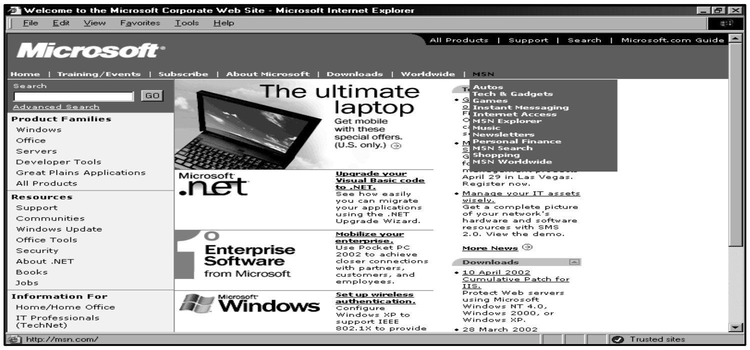
Dissatisfied with your results? Help us improve.

Try your query on: AltaVista Excite Lycos Yahoo!

©2002 Google

25.9 A few constructive recommendations

Let's look at a few Web sites and see how we can improve their usability



The ultimate laptop
Get mobile with these special offers. (U.S. only.)

Upgrade your Visual Basic code to .NET.
See how easily you can migrate your applications using the .NET Upgrade Wizard.

Mobilize your enterprise.
Use Pocket PC 2002 to achieve closer connections with partners, customers, and employees.

Set up wireless authentication.
Configure Windows XP to support IEEE 802.1X to provide

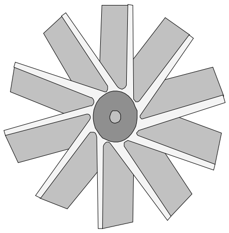
April 29 in Las Vegas. Register now.

- Manage your IT assets wisely.
Get a complete picture of your network's hardware and software resources with SMS 2.0. View the demo.

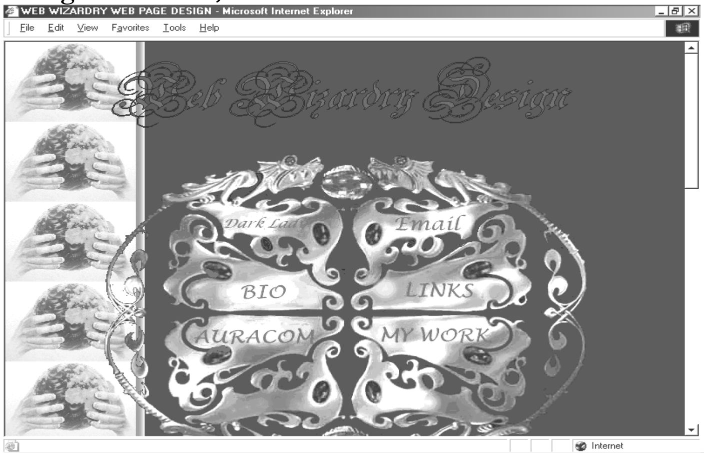
- 10 April 2002 Cumulative Patch for IIS.
Protect Web servers using Microsoft Windows NT 4.0, Windows 2000, or Windows XP.
- 28 March 2002

Enter

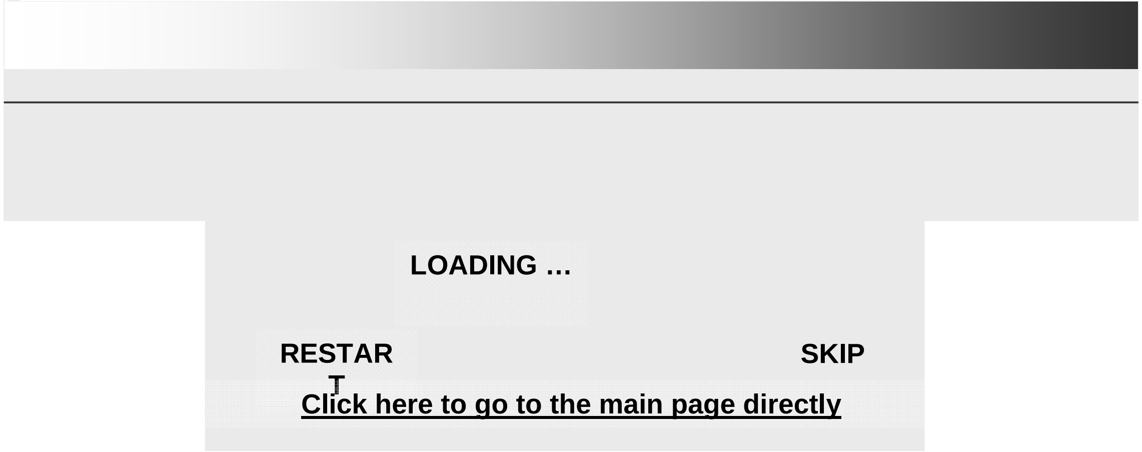
Dragon's Lair



All rights reserved, 2002.



WEB WIZARDRY WEB PAGE DESIGN - Microsoft Internet Explorer

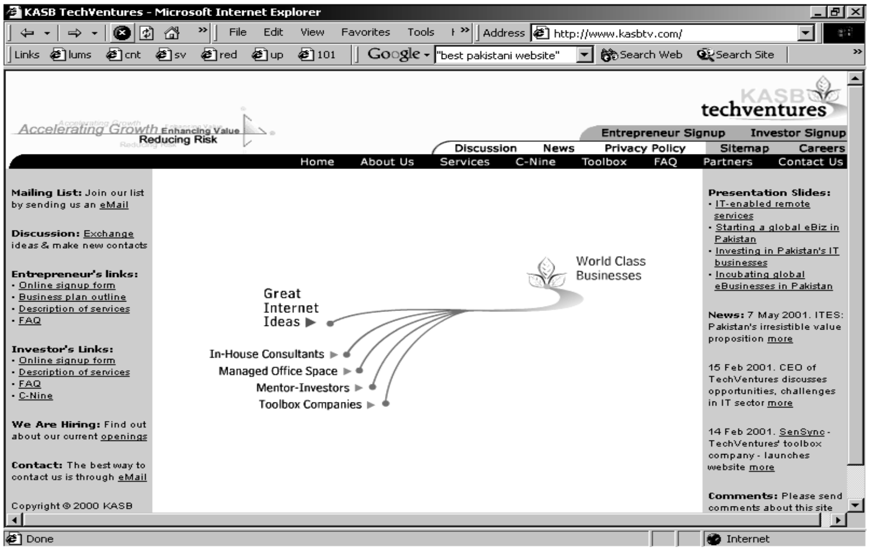


LOADING …

RESTAR
T

SKIP

Click here to go to the main page directly



KASB TechVentures - Microsoft Internet Explorer

File Edit View Favorites Tools
Address http://www.kasbtv.com/
Links lums cnt sv red up 101
Google "best pakistani website" Search Web Search Site
KASB techventures
Accelerating Growth Enhancing Value Reducing Risk
Entrepreneur Signup Investor Signup
Discussion News Privacy Policy Sitemap Careers
Home About Us Services C-Nine Toolbox FAQ Partners Contact Us
Great Internet Ideas
In-House Consultants
Managed Office Space
Mentor-Investors
Toolbox Companies
World Class Businesses
Done
Internet

Mailing List: Join our list by sending us an eMail

Discussion: Exchange ideas & make new contacts

Entrepreneur's links:
- Online signup form
- Business plan outline
- Description of services
- FAQ

Investor's Links:
- Online signup form
- Description of services
- FAQ
- C-Nine

We Are Hiring: Find out about our current openings

Contact: The best way to contact us is through eMail

Copyright © 2000 KASB

Presentation Slides:
- IT-enabled remote services
- Starting a global eBiz in Pakistan
- Investing in Pakistan's IT businesses
- Incubating global eBusinesses in Pakistan

News: 7 May 2001. ITES: Pakistan's irresistible value proposition more

15 Feb 2001. CEO of TechVentures discusses opportunities, challenges in IT sector more

14 Feb 2001. SenSync - TechVentures' toolbox company - launches website more

Comments: Please send comments about this site

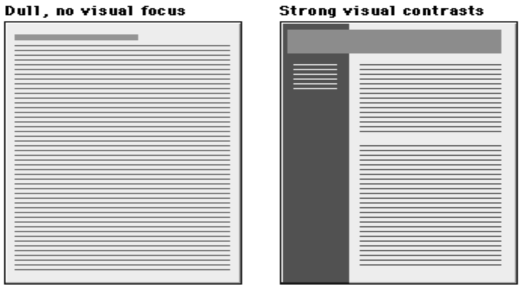
25.10 Making Display Elements Legible:

1. Designing (arranging) Display Elements

Elements must be large enough to be processed visually
Elements must contrast sufficiently with their backgrounds

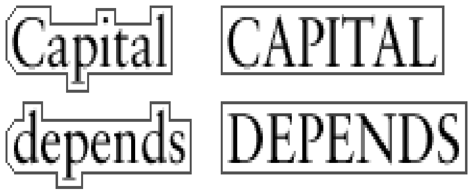
Making Display Elements Legible:

Related elements should be visually grouped through the use of space, color, or graphical boundaries
The relative levels of importance among elements in a display should be revealed graphically



25.11 Ensuring Text is Readable:

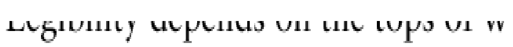
Use sans serif (e.g. Arial, Helvetica, Verdana) typefaces for display on screen
Display type intended for continuous reading at 10 to 14 points
Avoid the overuse of bold and italics
Avoid setting type in all caps



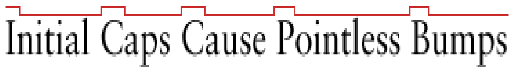
Downstyle headlines are more legible, because we primarily scan the tops of words as we read:

Legibility depends on the tops of w

Notice how much harder it is to read the bottom half of the same sentence:



If you use initial capital letters in your headlines you disrupt the reader's scanning of the word forms:



Arrange type intended for extended reading flush left, ragged right
Avoid lines of type shorter than 40 characters and longer than 60 characters

Mark the boundaries between paragraphs with blank lines rather than indentation

Use headings and subheadings to visually reveal the relationships among text elements they label – paragraphs after paragraphs of text do not work that well on the Web

25.12 Using Pictures & Illustrations:

Avoid using pictures that are strictly decorative

25.13 Using Motion

Use motion to attract the viewer's attention

Avoid the use of motion for "cosmetic" purposes

Success is defined by the user, not the builder

In Today's Lecture:

We looked at the role of usability in Web site design

We identified some of the factors affecting the usability of a Web page

Next Lecture:

Computer Networks

We will become able to appreciate the role of networks in computing

We will familiarize ourselves with various networking topologies and protocols

Lecture 26

Arrays

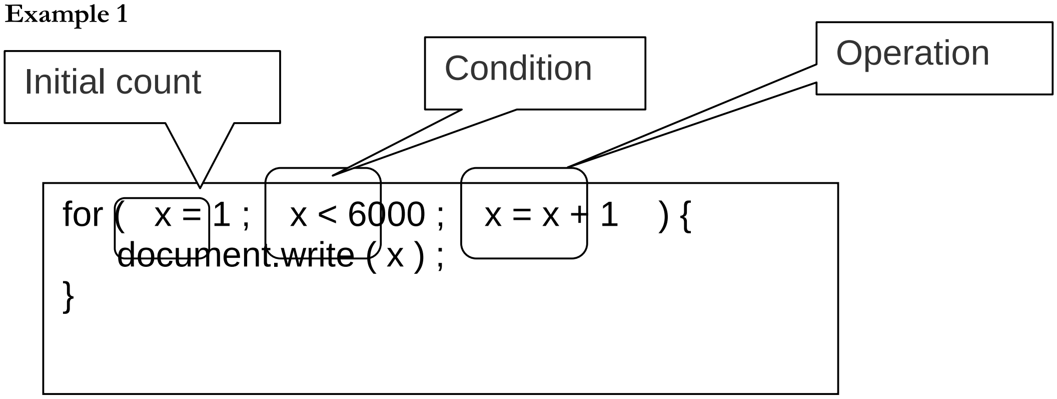
(Web Development Lecture 9)

During the last lecture we had a discussion on Flow Control & Loops

We discussed the concept of flow control using the "if" and "switch" structures
And also the concept behind the "while" and "for" looping structures
We also solved simple problems using flow control and loop structures
if…else --?-- switch
If the action to be taken of the value of a single variable (or a single expression), use 'switch'
When the action depends on the values of multiple variables (or expressions), use the 'if...else' structure

Compound Statements:

for: Example 1



```
for (   x = 1 ;    x < 6000 ;    x = x + 1    ) {
    document.write ( x ) ;
}
```

for --?-- while

When the exact number of iterations is known, use the 'for' loop
When the number of iterations depend upon a condition being met, use the 'while' loop
'for' loops become especially useful when used in conjunction with arrays We'll find out about arrays today, and we'll probe their usefulness as part of 'for' loop structures

Today's Topic:

Arrays

We will find out why we need arrays
We will become able to use arrays for solving simple problems

Array:

An indexed list of elements
We said that a variable is a container that holds a value.
Similarly, an Array can be considered a container as well, but this one can hold multiple values
Array:
An indexed list of elements

Example: There are many ways of assigning identifiers to the following fruit

strawberry
fruit1
fruit[0]

orange
fruit2
fruit[1]

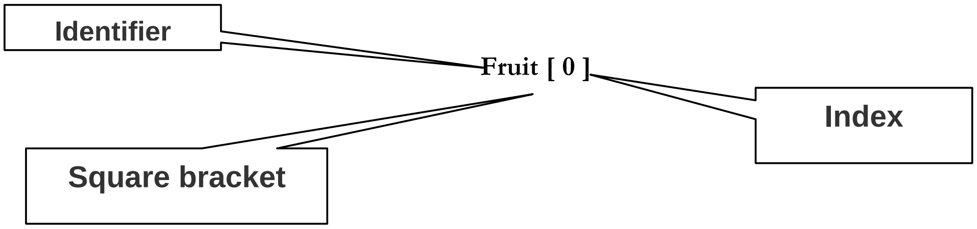
apple
fruit3
fruit[2]

watermelo
n
fruit4
fruit[3]

Array

An indexed list of elements
fruit[0], fruit[1], fruit[2], and fruit[3] are the elements of an array
‘fruit’ is the identifier for that array
The length of the ‘fruit’ array is 4, i.e. ‘fruit’ has four elements

Array



```
var student1, student2, student3, student4 ;
student1 = “Waseem” ;
student2 = “Waqar” ;
student3 = “Saqlain” ;
student4 = “Daanish” ;
document.write( student1 ) ;
document.write( student2 ) ;
document.write( student3 ) ;
document.write( student4 ) ;

student = new Array( 4 )  ;   //array declaration
student[ 0 ] = “Waseem” ;
student[ 1 ] = “Waqar” ;
student[ 2 ] = “Saqlain” ;
student[ 3 ] = “Daanish” ;
for ( x = 0 ; x < 4 ; x = x + 1 ) {
      document.write( student[ x ] ) ;
}
```

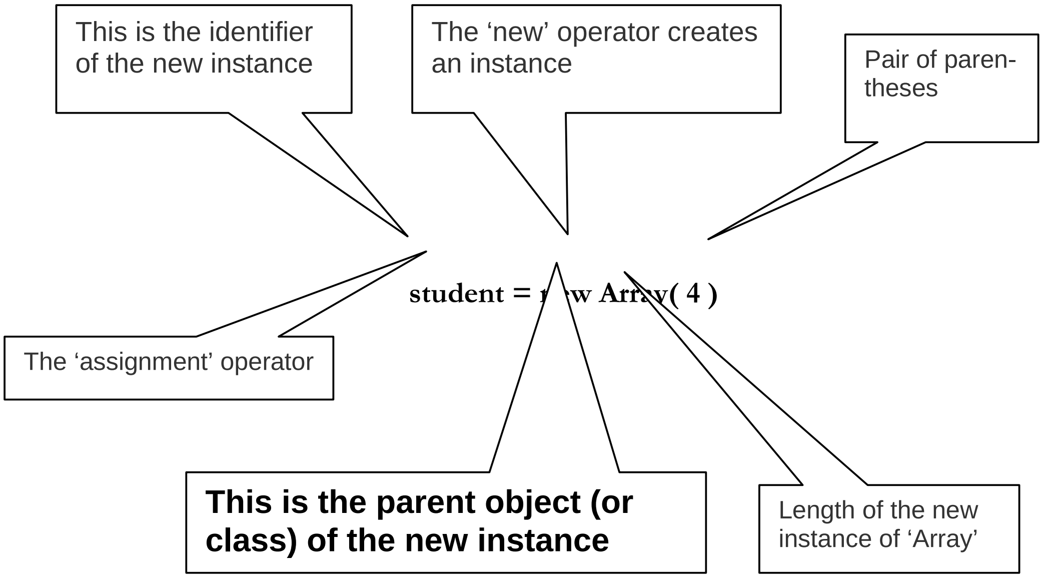
Can you see the advantage of using arrays along with the ‘for’ loop?

26.1 Arrays in JavaScript

- In JavaScript, arrays are implemented in the form of the ‘Array’ object
- The key property of the ‘Array’ object is ‘length’, i.e the number of elements in an array
- Two of the key ‘Array’ methods are:
  – reverse( )
  – sort( )
- Elements of an array can be of any type; you can even have an array containing other arrays

Declaring a New Instance of the Array Object

- ‘student’ is an instance of the ‘Array’ object
- ‘student’ was declared such that it is of length ‘4’
- That is, student is an array having 4 elements
- The four elements of the array are: ‘student[0]’, ‘student[1]’, ‘student[2]’, and ‘student[3]’



This is the identifier of the new instance

The ‘new’ operator creates an instance

Pair of paren-theses

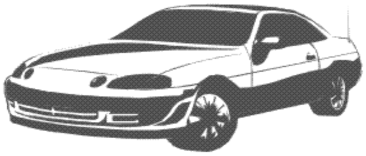
student = new Array(4)

The ‘assignment’ operator

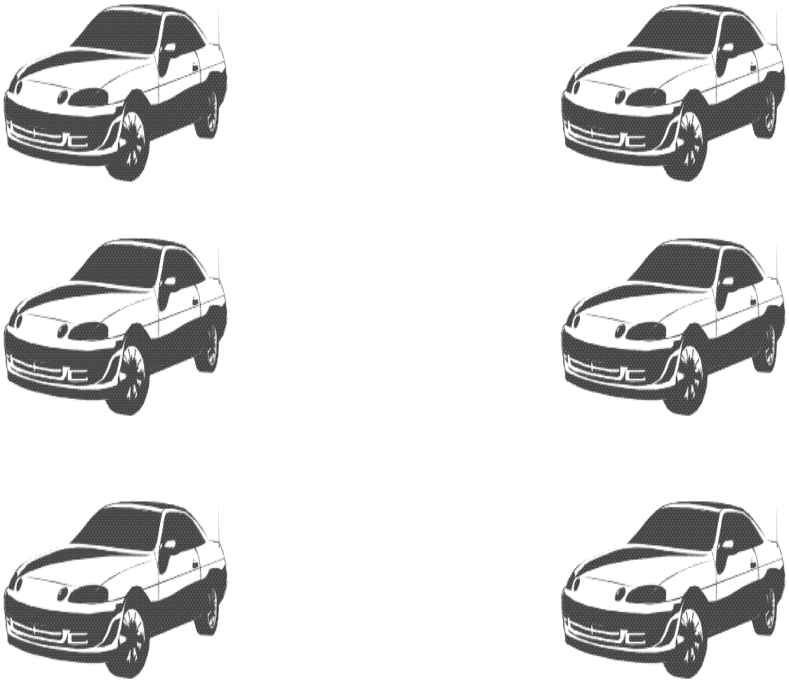
This is the parent object (or class) of the new instance

Length of the new instance of ‘Array’

An Object

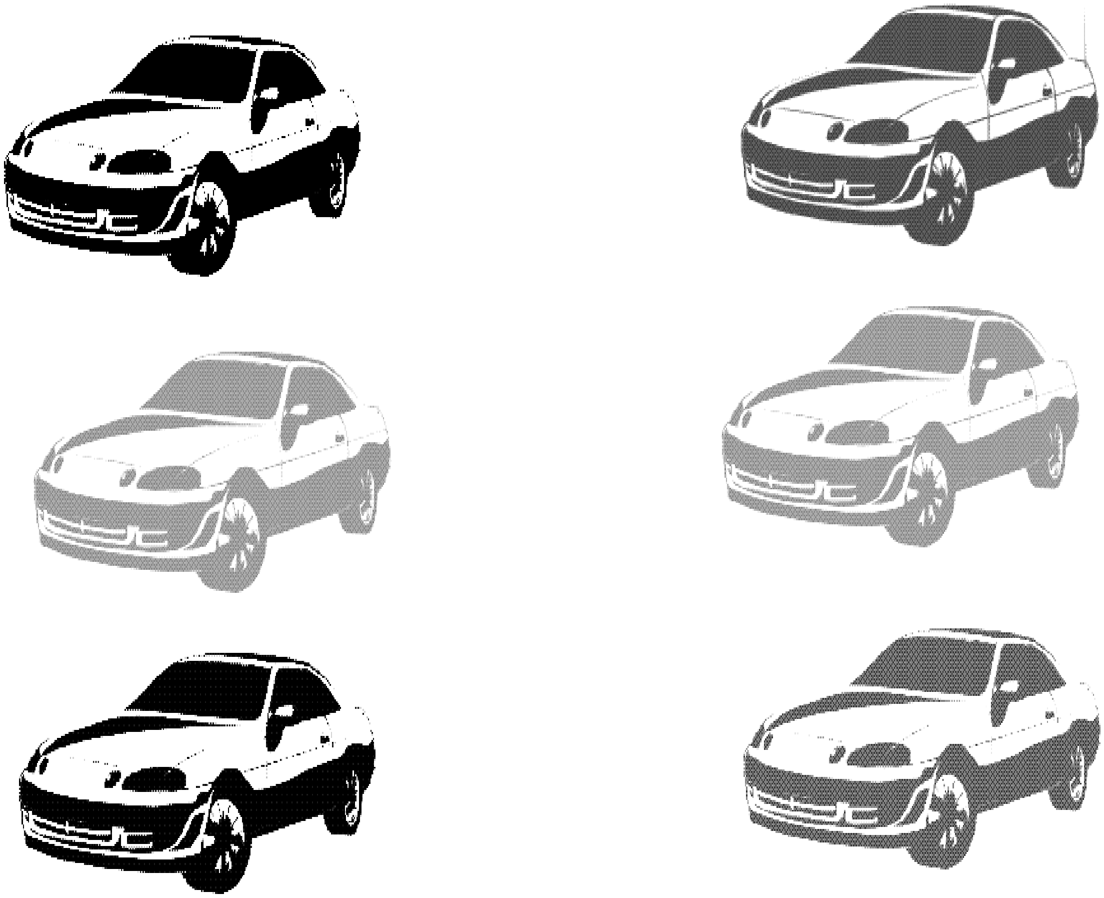


‘Instances’ of an Object



All ***instances*** **of an object are objects themselves!**

'Property' Values of the Instances May Differ



student = new Array(4)

26.2 Array Identifiers

The naming rules for Array identifiers are the same as were discussed for variable identifiers

Assigning Values to Array Elements

```
a[ 1 ] = 5 ;  //the second element
name[ 5 ] = "bhola" ;
number = 5 ;
name[ number ] = name[ 5 ] ;
for ( x = 0 ; x < 10 ; x = x + 1 ) {
	y[ x ] = x * x ;
}
```

Remember: just like C, C++ and Java, the first element of an array has an index number equal to zero

JavaScript Arrays are Heterogeneous

Unlike many other popular languages, a JavaScript Array can hold elements of multiple data types, simultaneously

```
a = new Array( 9 ) ;
b = new Array( 13 ) ;
b[ 0 ] = 23.7 ;
b[ 1 ] = "Bhola Continental Hotel" ;
```

```
b[ 2 ] = a ;
```

26.3 The 'length' Property of Arrays

'length' is a property of the object 'd'

'd' is an instance of the 'Array' object

```
d = new Array ( 5 ) ;
document.write( d.length ) ;
```

The 'length' Property of Arrays

```
x = new Array ( 10 ) ;
for ( x = 0 ; x < 10 ; x = x + 1 ) {
	y[ x ] = x * x ;
}
x = new Array ( 10 ) ;
for ( x = 0 ; x < x.length ; x = x + 1 ) {
	y[ x ] = x * x ;
}
```

What is advantage of using 'x.length' here instead of using the literal '10'?

26.4 Array Methods: sort( )

26.5 Sorts the elements in alphabetical order

```
x = new Array ( 4 ) ;
x[ 0 ] = "Waseem" ;
x[ 1 ] = "Waqar" ;
x[ 2 ] = "Saqlain" ;
x[ 3 ] = "Shoaib" ;
x.sort( ) ;
for ( k = 0 ; k < x.length; k = k + 1 ) { document.write( x[ k ] + "<BR>" ) ;
}
```

```
Saqlain
Shoaib
Waqar
Waseem
```

Were the elements sorted in ascending or descending order? What if you wanted to arrange them in the reverse order?

26.6 Array Methods: reverse( )

26.7 Reverses the order of the elements

```
x = new Array ( 4 ) ;
x[ 0 ] = "Waseem" ;
x[ 1 ] = "Waqar" ;
x[ 2 ] = "Saqlain" ;
x[ 3 ] = "Shoaib" ;
x.reverse( ) ;
x.sort( ) ;
for ( k = 0 ; k < x.length; k = k + 1 ) {   document.write( x[ k ] + "<BR>") ;
}
```

```
Saqlain
Shoaib
Waqar
Waseem
```

Is this the required result?

Array Methods: reverse( )

Reverses the order of the elements

```
x = new Array ( 4 ) ;
x[ 0 ] = "Waseem" ;
x[ 1 ] = "Waqar" ;
x[ 2 ] = "Saqlain" ;
x[ 3 ] = "Shoaib" ;
x.sort( ) ;
x.reverse( ) ;
for ( k = 0 ; k < x.length; k = k + 1 ) {   document.write( x[ k ] + "<BR>") ;
}
```

```
Waseem
Waqar
Shoaib
Saqlain
```

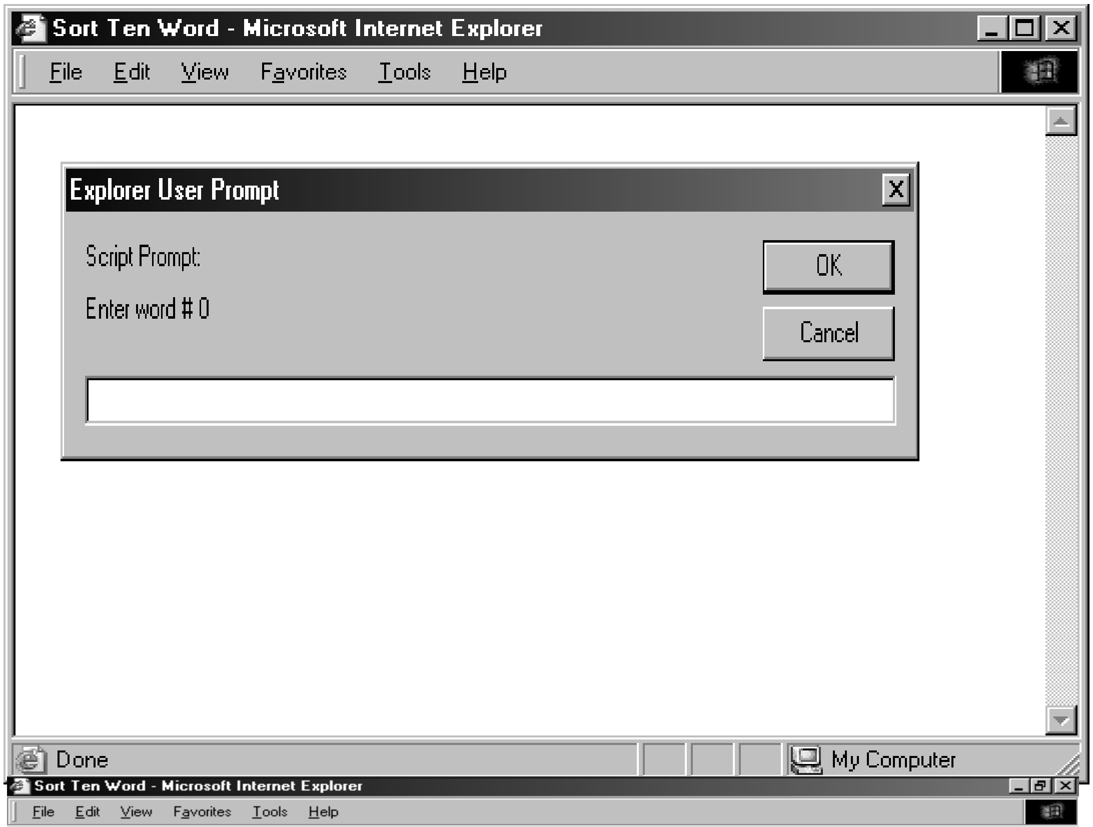
Let's Now Do a More Substantial Example

Develop a Web page that prompts the user for 10 words, and then displays them in form of a list in two different ways:

1.In the order in which the words were entered

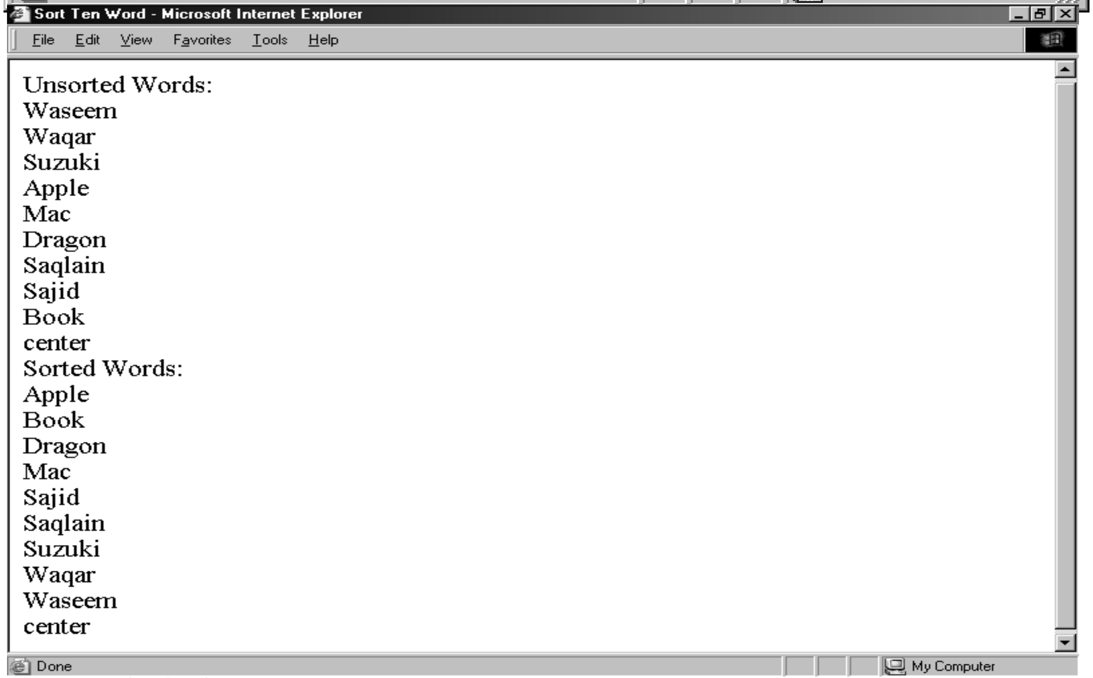
2.In a sorted order

We will try to show you the complete code - the JavaScript part as well as the HTML part - for this example



Script Prompt:

Enter word # 0



```
Unsorted Words:
Waseem
Waqar
Suzuki
Apple
Mac
Dragon
Saqlain
Sajid
Book
center
Sorted Words:
Apple
Book
Dragon
Mac
Sajid
Saqlain
Suzuki
Waqar
Waseem
center
```

26.7 Pseudo Code

1.Declare the array that will be used for storing the words

2.Prompt the user and read the user input into the elements of the array

3.Now write the array to the document
4.Sort the array
5.Write the sorted array to the document

```
<HTML>
    <HEAD>
        <TITLE>Sort Ten Words</TITLE>
        <SCRIPT>
            words = new Array ( 10 ) ;
            for ( k = 0 ; k < words.length ; k = k + 1 ) {
                words[ k ] = window.prompt( "Enter word # " + k, "" ) ;
            }
            document.write( "UNSORTED WORDS:" + "<BR>" ) ;
            for ( k = 0 ; k < words.length ; k = k + 1 ) {
                document.write( words[ k ] + "<BR>" ) ;
            }
            words.sort( ) ;
            document.write( "SORTED WORDS:" + "<BR>" ) ;
            for ( k = 0 ; k < words.length ; k = k + 1 ) {
                document.write( words[ k ] + "<BR>" ) ;
            }
        </SCRIPT>
    </HEAD>
    <BODY>
    </BODY>
</HTML>
<HTML>
    <HEAD>
        <TITLE>Sort Ten Words</TITLE>
        <SCRIPT>
            //JavaScript Code
        </SCRIPT>
    </HEAD>
    <BODY>
    </BODY>
</HTML>
```

The next three slides show the JavaScript code that goes between the <SCRIPT>, </SCRIPT> tags

Pseudo Code

•Declare the array that will be used for storing the words
•Prompt the user and read the user input into the elements of the array
•Now write the array to the document
•Sort the array
•Write the sorted array to the document

```
words = new Array ( 10 ) ;
for ( k = 0 ; k < words.length ; k = k + 1 ) {
    words[ k ] = window.prompt(
                        "Enter word # " + k, "" ) ;
}
```

This method is used for collecting data from the user. It can display a message and provides a field in which the user can enter data

Pseudo Code

```
1.Declare the array that will be used for storing the words
2.Prompt the user and read the user input into the elements of the array
3.Now write the array to the document
4.Sort the array
5.Write the sorted array to the document
```

```
words.sort( ) ;
document.write( "Sorted Words:" + "<BR>" ) ;
for ( k = 0 ; k < words.length ; k = k + 1 ) {   document.write( words[ k ] + "<BR>" ) ;
}
```

During Today's Lecture …

•We found out why we need arrays

•We became able to use arrays for solving simple problems

Next (the 10th) Web Dev Lecture:

Functions & Variable Scope

•To become familiar with some of JavaScript's built-in functions

•To be able to understand the concept of user-defined functions and their use for solving simple problems

•To become familiar with the concept of local and global variables

Lecture 27

Computer Networks

During the last lecture …
(Web Design for Usability)

- We looked at the role of usability in Web design
- We identified some of the factors affecting the usability of a Web page

Designs should be consistent & predictable (unified)

What's a Good Site?

- The one that achieves the result that it was designed for
- Generally, that result can only be achieved by giving the user what s/he wants, as quickly as possible, without her/im expending much effort
- One definition of usability: Let the user have what s/he wants, quickly, without much effort
- "Quickly" is important!

Website Navigation

- It probably is the most important aspect of the design of a Website

Good designs assist the user in recovering from errors

Today's Goals:
(Computer Networks)

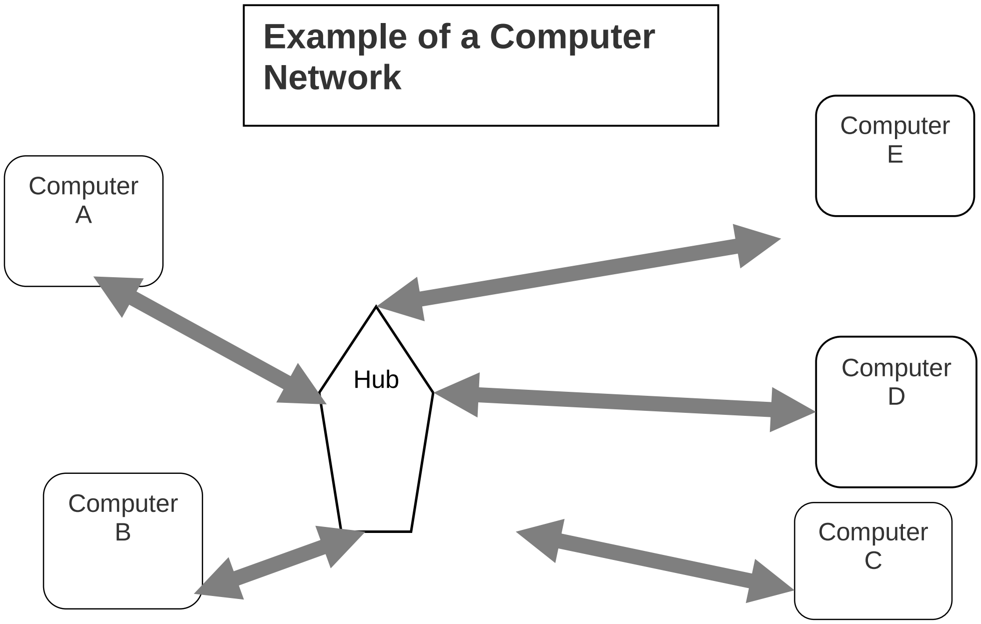
- We will become able to appreciate the role of networks in computing
- We will look at several different types of networks
- We will familiarize ourselves with networking topologies and protocols

Computer Network

Multiple computers that are connected together to share information and other resources

Examples of Computer Network Usage

- I can send an eMail message to a remote computer using the SMTP protocol
- I can browse documents residing on a remote computer using the HTTP protocol
- •I can download or upload files to a remote computer using the FTP protocol
- •I can run a program on a remote computer using the TELNET protocol



Components of Conventional Computer Networks

1. **Computers**
2. **Network Interface Cards (NIC)**

–I/O device that plugs into the computer
–Enables it to communicate over a network

3. Hub

–The network traffic controller
Components of Conventional Computer Networks

4.Cables

–Are either electrical or optical
–Not required at all for wireless networks

5.Protocol

–Rules governing communications over the network

How Does a Conventional Network Work?

1. Suppose computer A wants to send a message to D
2. Computer A sends the message to its NIC
3. The NIC translates the message into electrical pulses suitable for the computer network in use & transmits it to the hub through the cable
4. The hub receives them and forwards them to all computers connected to the it
5. The NICs of all computers connected to the hub receive the forwarded electrical pulses
6. The NIC of computer D decides that the message is for it, & translates the pulses back to a form suitable for the computer

Hub

- A device that is used to connect several computers to form a network
- A hub has several ports. The number generally is 8, 12, 16, 24, 32, or 48
- Each computer in a network is connected to one of those ports through a cable
- A computer wanting to send a message to one of the others in the network sends a message to the hub, which, in turn, broadcasts the message to all others connected to it

Packet

- The smallest unit of data transmitted over a computer network
- A message to be transferred over the network is broken up into small packets by the sending computer
- Each packet contains the following info:

–Sender's address
–Destination address
–Data
–Error-recovery info

- All packets travel independently
- When all packets are received by the destination computer, it reassembles them to form the original message

Types of Computer Networks according to the network access policy

- Private
- Public

27.1 Private Networks

- Organizations having many computers usually connect them in the form of private networks
- Access to these networks is restricted to authorized computers only

- This allows computers from within the organization to exchange info, but keeps the info private and protected from outsiders
- All equipment on a private network is generally for the exclusive use of that organization

27.2 Public Networks

- All networks that are not private, are … public
- **Example:** Internet

- Communication equipment used in these networks is generally being used by users belonging to several (possibly thousands of) organizations as well as those belonging to no organization

27.3 VPN: Virtual Private Network (1)

- From the user's point-of-view, a VPN looks like a secure, private network
- VPNs use public telecom infrastructure, maintaining privacy through security procedures
- VPNs provide secure network connections for distance computers without using dedicated, private channels to supply the connection
- Key benefit of VPNs over conventional PNs: Lower cost

Types of Computer Networks according to the distance between nodes

LAN: Local Area Network)

WAN: Wide Area Network)

LAN

A network of computers located in the same building or a handful of nearby buildings

Examples:

–Computer network at your PVC

–Computer network of a University campus

WAN

A network in which computers are separated by great distances, typically across cities or even continents

May consist of several interconnected LANs

Example:

–The network connecting the ATM of a bank located in various cities

–A network connecting the local and oversea offices of a SW house

–Internet

Connecting LANs to other Networks:

Special-purpose devices are used to link LANs to other networks

They may belong to one of the following categories:

–Routers

–Bridges

–Gateways

–Modems

Router

A special-purpose computer that directs data traffic when several paths are available

A router examines the destination info in each arriving packet and then routes it through the most efficient path available

The router either delivers the packet to the destination computer across a local network or forwards the packet to another router that is closer to the final destination

Bridge

Used to form a connection between two separate, but similar networks

In a way, it creates an extended LAN by passing information between two or more LANs

Gateway

A special-purpose computer that connects and translates between networks that use different communications protocols

LAN's may use a gateway (or router) to connect to the Internet

Modem

I/O device used for connecting two computers over telephone lines

modem = modulator + demodulator

Modulator converts computer messages to electrical pulses that are suitable for transmission over the telephone lines

Demodulator converts electrical pulses received over telephone lines into messages that are comprehensible for computers

27.4 Network Topologies

The pattern in which computers are connected to form a network

Popular patterns:

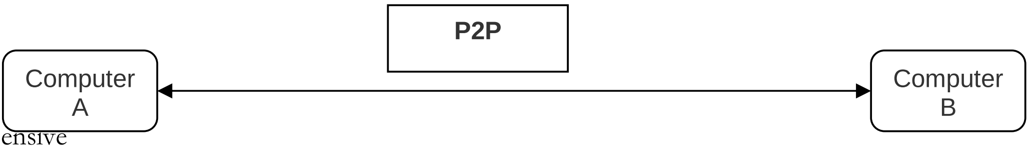
–Point-to-point

–Star

–Bus

–Ring

Networks are also formed by combining 2 or more of these 4 basic patterns



Inexpensive

Limited connectivity

Quite often used for connecting two LANs to form a WAN

Star

A computer sends the address of the intended receiver and the data to the server

The server then sends the message to the intended receiver

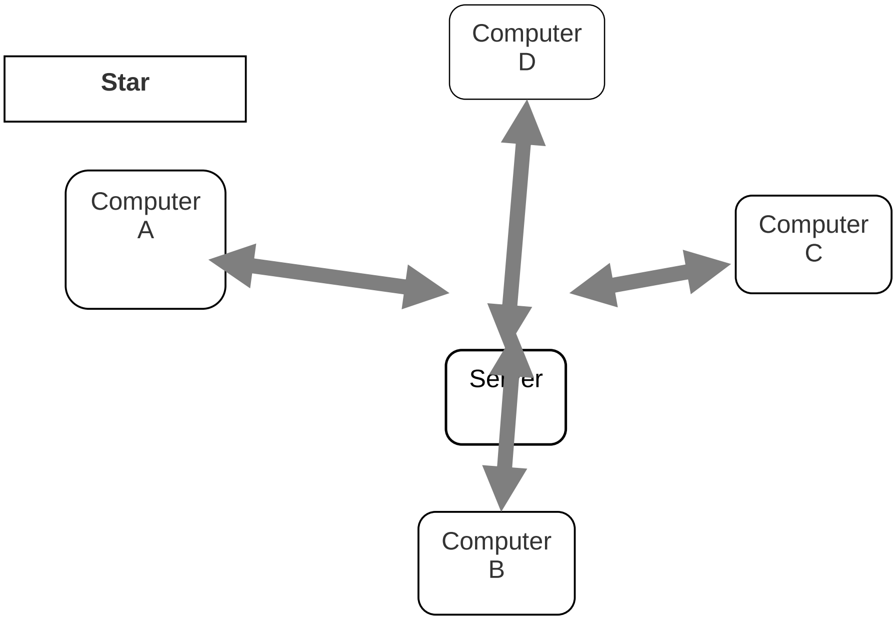
This topology allows multiple messages to be sent simultaneously

Costly, because it uses an additional computer to direct the data

Costly, because each node is individually wired to the hub

If the server goes down, so does the network

If any of the nodes goes down, the rest of the network is not affected



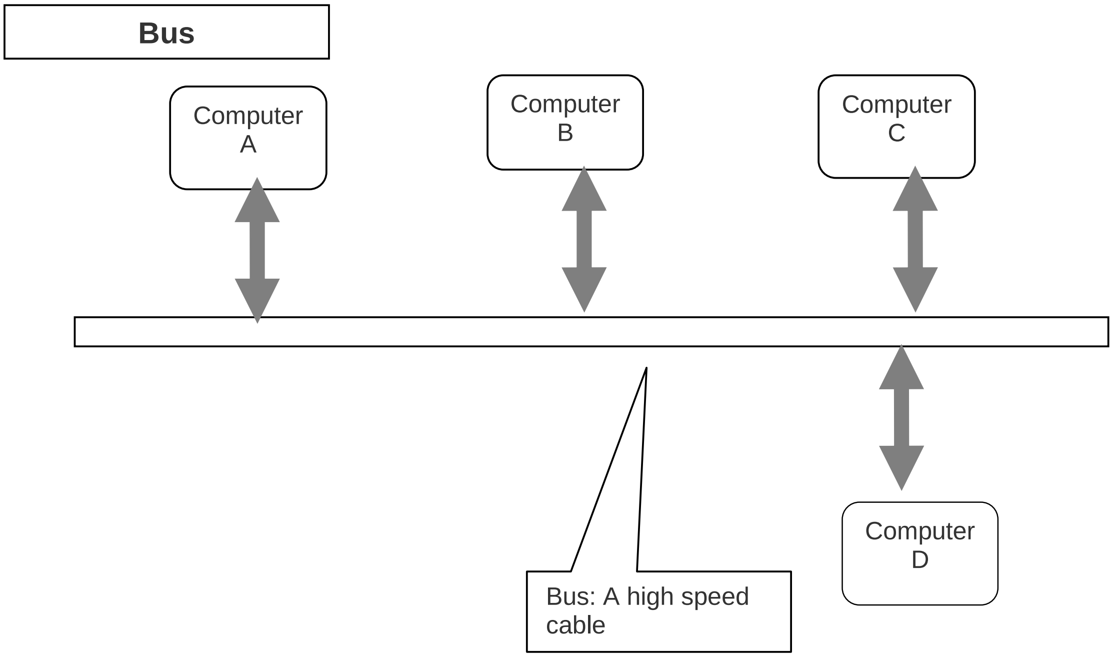
Bus

No server is required

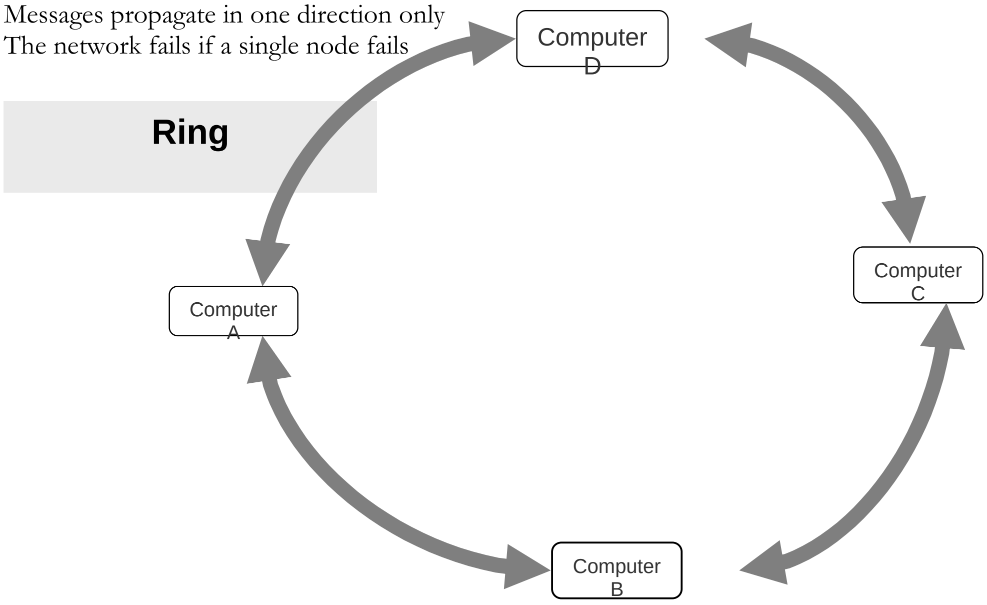
One computer sends data to another by broadcasting the address of the receiver and the data over the bus

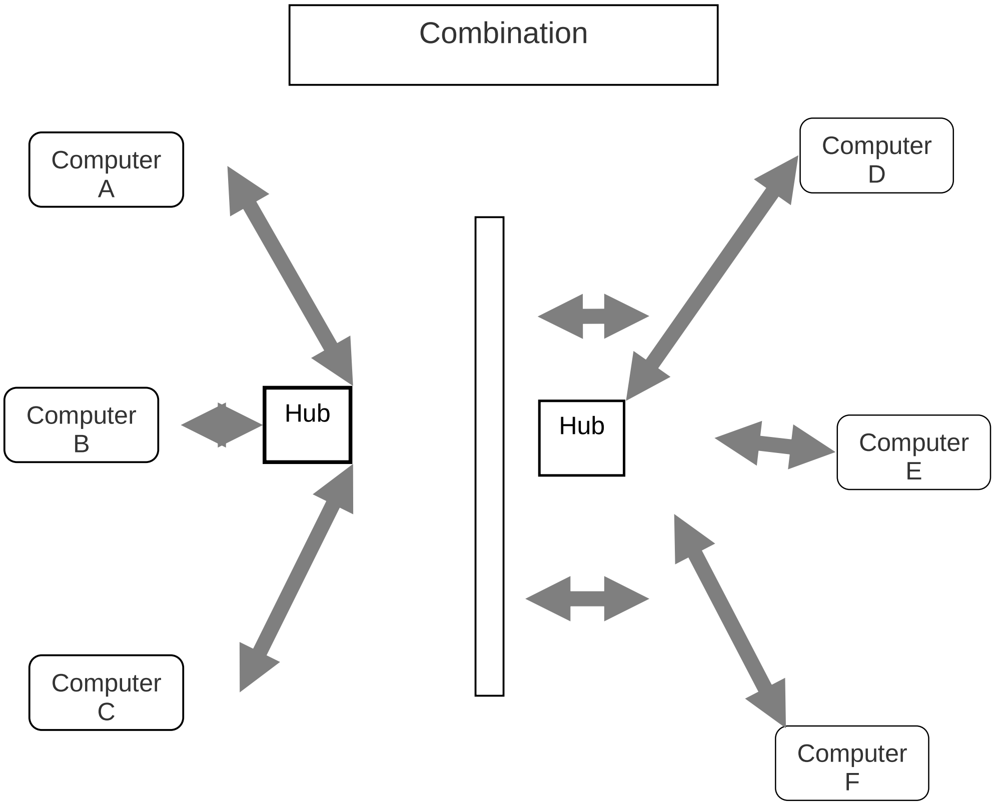
All the computers in the network look at the address simultaneously, and the intended recipient accepts the data

A bus network, unlike ring or star networks, allows data to be sent directly from one computer to another
However, only one computer at a time can transmit data. The others must wait to until the bus gets idle
If any of the nodes goes down, the rest of the network is not affected



Ring
No server is required
A computer sends the message to its neighbor. The neighbor examines the message to determine if it is the intended recipient
If the data are not intended for that particular neighbor, it passes the message to the next computer in the ring
This process is repeated until the data arrive at their intended recipient
This topology allows multiple messages to be carried, simultaneously
Data transmission is slow since each message is checked by each computer
New nodes are difficult to add
Messages propagate in one direction only
The network fails if a single node fails





27.5 Networking Protocols

Networks use protocols, or rules, to exchange info through shared channels
Protocols prevent collisions of packets caused by simultaneous transmission between two or more computers
Several protocols are available for various types of networks. Here we discuss two that are popular for LANs: Ethernet; Token Ring

Ethernet Protocol

A computer using this protocol checks if a shared connection is in use before transmitting a message
If not, the computer transmits data
Two computers may sense an idle connection and may send packets simultaneously. To account for such situations, transmitting computers continue to monitor the connection and re-transmit if a packet collision occurs

Token Ring Protocol

This protocol passes a special message called a token through the network
A computer that receives the token is given permission to send a packet of information
If the computer has no packet to send, it passes the token to the next computer
Computer Networks = Computers + Communications
Types of Communication Channels
1 . Wire
2 . Wireless
A key characteristic of these channels is bandwidth

Bandwidth

Capacity of a communication channel for carrying data
Measured in bits/s (bps), kb/s, Mb/s, Gb/s, Tb/s
Optical fiber channels have the highest (1 Tb/s)

Telephone lines the lowest (56 kb/s)

27.6 Types of Communication Channels

Wire

- Copper
  - Twisted-pair
  - Coaxial cable
- Optical fiber

Wireless

- Line-of-sight
  - Microwave
  - Optical
- Non-line-of-sight
  - Satellite
  - Radio
  - Cellular

Wireless (Radio) LANs Are Becoming Popular

Key benefits:

- Set-up time
- Set-up cost
- Maintenance cost
- Cost

Key challenges:

- Security & privacy
- Quality of service
- Cost

27.7 Network Security

Keeping an eye on the security of private networks (e.g. LANs) is relatively easy

However, their connections to other networks (e.g. the Internet) pose a security risk because the one has no control over users on those networks

Network Security

Applications transferred from the Internet to the LAN may contain computer viruses

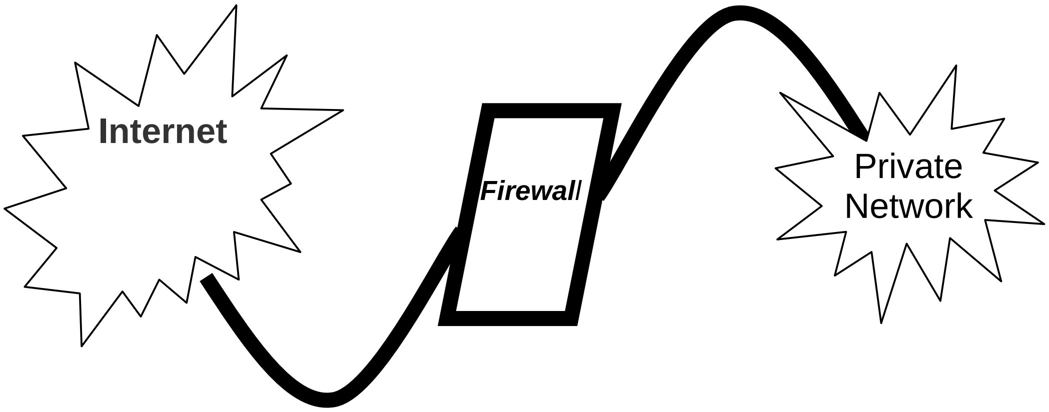
External, unauthorized users may gain access to sensitive data

A special type of gateway - a firewall – can keep external users from accessing resources on the LAN while letting LAN users access the external info

Firewall

A system that that guards a private network, enforcing an access/deny policy to all traffic going to and coming from the Internet

It keeps an eye on all the packets that go in and out of the private network and blocks them or allows them to continue to their destination according to the policy



Firewall Policy: Example

One can configure a firewall to allow only eMail to enter the private network, thus shielding it from any malicious attacks except for those via eMail

In Today's Lecture:

We looked at the role of networks in computing

We looked at several different types of networks

We familiarized ourselves with networking topologies and protocols

Next Lecture:

Introduction to the Internet

To become able to appreciate the role of the Internet in today's computing

To become familiar with the history and evolution of the Internet

Lecture 28

Introduction to the Internet

During the Last Lecture
(Computer Networks)

We looked at the role of networks in computing
We looked at several different types of networks
We familiarized ourselves with networking topologies and protocols

Computer Network

Multiple computers that are connected together to share information and other resources
Types of Computer Networks according to the network access policy
Private
Public

Types of Computer Networks
according to the distance between nodes

LAN: Local Area Network
WAN: Wide Area Network

Network Topologies

The pattern in which computers are connected to form a network

Popular patterns:

–Point-to-point
–Star
–Bus
–Ring

Networks are also formed by combining 2 or more of these 4 basic patterns

Networking Protocols

Networks use protocols, or rules, to exchange information through shared channels. These protocols prevent collisions of data caused by simultaneous transmission between two or more computers. Several protocols are available for various types of networks. Here we discuss two that are popular for LANs: Ethernet; Token Ring

Types of Communication Channels

Wire
Wireless
Wireless (Radio) LANs Are Becoming Popular

Key benefits:

– Set-up time
– Set-up cost
– Maintenance cost
– Cost

Key challenges:

– Security & privacy
– Quality of service
– Cost

Today's Goal:

Introduction to the Internet

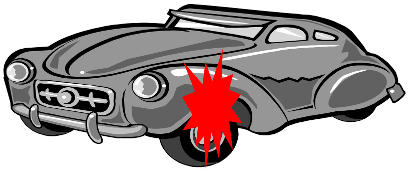
To become able to appreciate the role of the Internet in today's computing
To become familiar with the history and evolution of the Internet

an accident!

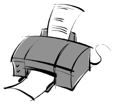
•This car was involved in that accident

It belongs to …

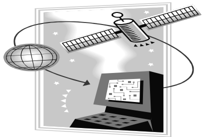
Mr. Tom Peters of Palo Alto, California



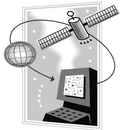
After the accident, Mr. Peters …
filled out a form, giving info about:
Himself
–The circumstances of the accident
–Estimated repair expenses
& then …



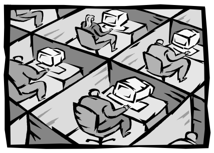
1/ 7
• Mr. Peters's fax machine



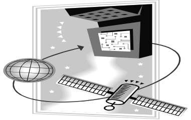
2/ 7 ACME Insurance Group's server in New York



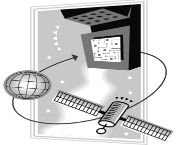
3/ 7
Bhola eServices (Pvt) Ltd's server at Davis Rd, Lahore



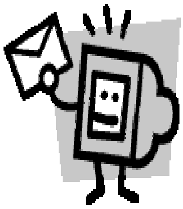
4/ 7
Claims processing in Lahore



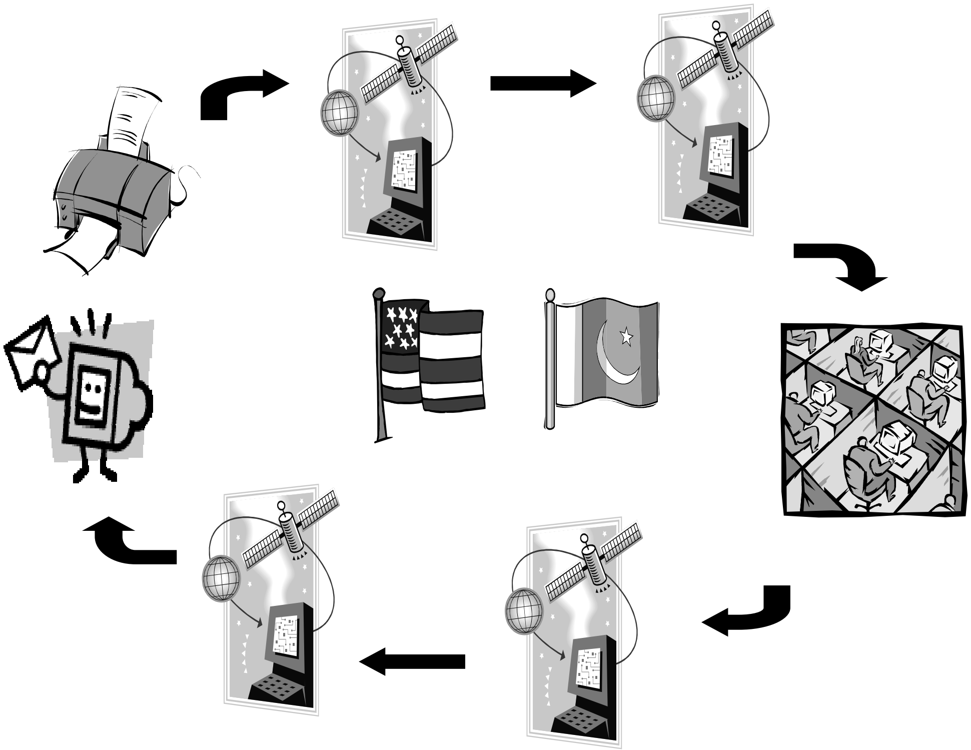
5/ 7
Bhola eServices (Pvt) Ltd's server at Davis Rd, Lahore



6/ 7
ACME Insurance Group's server in New York



7/ 7
Mr. Peters's home PC



Key Question!

Why process the insurance claim in Pakistan?

Answer: Everybody Wins!

Tom Peters
ACME Insurance
Bhola eServices

Answer: Everybody Wins!

Tom Peters
ACME Insurance
Bhola eServices

Lower premium
&
Quicker turnaround

Answer: Everybody Wins!
Tom Peters
ACME Insurance
Bhola eServices

Better margins due to 50% saving on claim processing costs

Answer: Everybody Wins!

Tom Peters
ACME Insurance
Bhola eServices

Internal rate of return (IRR) of 60-80%

The Key Point …
Bhola eServices (Pvt) Ltd is …

… supplying a service
… using local, attractively-priced workers
… to a remote, overseas client
… *over the Internet* … & making good money in the process!

Internet

Enables users located at far-way locations to easily share information with others located all over the world
Enables users to easily and inexpensively communicate with others located all over the world
Enables the users to operate and run programs on computers located all over the world
The Internet is unlike any previous human invention. It is a world-wide resource, accessible to all of the humankind.

Internet Users Worldwide

673M in 2002
1B+ in 2005
(48% wireless)
1.2M Internet users in Pakistan in 5/2000
(1% of population)
In early 2002,
54% of Australian population
51% of Singaporean population
39% of Japanese population
3% of Chinese population

Key Characteristics

Geographic Distribution
Global - reaches around the world
Robust Architecture
Adapts to damage and error
Speed
Data *can* travels at near '*c*' on copper, fiber, airwaves

Key Characteristics

Universal Access
Same functionality to everyone
Growth Rate
The fastest growing technology ever
Freedom of Speech
Promotes freedom of speech
The Digital Advantage
Is digital: can correct errors

28.1 Internet: Network of Networks

A large number of networks, interconnected physically
Capable of communicating and sharing data with each other
From the user's point view, Internet – a collection of interconnected networks – looks like a single, unified network

28.2 Internet Networking Protocols

Communications on the Internet is controlled by a set of two protocols: TCP and IP
TCP/IP Transmission Control Protocol/Internet Protocol
Networking protocol used by *all* computers and networks on the Internet
Originally developed by the US DoD for Unix, but now available for most other OSes
TCP breaks down the message to be sent over the Internet into packets
IP routes these packets through the Internet to get them to their destination
When the packets reach the destination computer, TCP reassembles them into the original message

Tools & Services Available on the Internet

Electronic mail (POP, IMAP, SMTP)

Instant messaging (ICQ, MSN)
Remote login (telnet)
File transfer (ftp)
Network news (nntp)
WWW (http)

1960's

1969 - DoD-ARPA creates an experimental network – ARPANET – as a test-bed for emerging networking technologies

ARPANET originally connected 4 universities & enabled scientists to share info & resources across long distances

ARPANET continued to expand throughout the 70's and 80's

1970's

Networking tools developed in the 70's include:

1972 - The National Center for Supercomputing Apps. (NCSA) develops the telnet application for remote login, making it easier to connect to a remote computer

1973 - FTP (File Transfer Protocol) is introduced, standardizing the transfer of files between networked computers

1980's

1983 - The TCP/IP protocols becomes the only set of protocols used on the ARPANET

This sets a standard for all networks, and generates the use of the term Internet as the net of nets

ARPANET splits into two nets to keep military & non-military network sites separate: ARPANET and MILNET

1980's

In **1982** and **1983**, the first desktop computers begin to appear

Many are equipped with an OS called Berkeley Unix, which includes networking SW, allowing easy connection to the Internet using telnet

The PC revolution continues through the 80's, making access to computer resources & net-worked info increasingly available to public

1985-86: NSF connects the US's six supercomputing centers together, calling it the the NSFNET, or NSFNET backbone

To expand access to the Internet, NSF developed regional nets, which were then connected to the NSFNET backbone

Plus, NSF supported institutions (universities, etc.) in their efforts to connect to the regional nets

1987 - NSF awards a grant to Merit Network, Inc. to operate & manage future development of the NSFNET

Merit collaborates with IBM & MCI on R&D for fast networking technologies

1989 - The backbone network is upgraded to T1, making it able to transmit data at speeds of 1.5 Mb/s (approx. 60 pages of text/second)

1990's1990 - The ARPANET is dissolved

1991 - Gopher is developed at the U of MN

It provides a hierarchical, menu-based method for providing & locating info on the Internet

1993 - CERN releases WWW, developed by Tim Berners-Lee

It uses HTTP and hypertext, revolutionizing the way info is presented & accessed on Internet

1993 - The NSFNET is upgraded to T3 (45 Mb/s or about 1800 pages/s)

1993-1994 - Web browsers Mosaic & Netscape Navigator are introduced

Their GUI makes WWW & Internet more appealing to the general public

1995 - NSFNET is replaced by a new architecture, called vBNS which utilizes regional networks and Network Access Points

Who runs the Internet?

Who owns it?

Today's Goal:

Introduction to the Internet

We looked at the role Internet plays in today's computing
We reviewed some of the history and evolution of the Internet

Next Lecture:

Internet Services

We will try to familiarize ourselves with with some of the Internet services:
–http (surfing, shopping, searching)
–eMail
–ftp
–News groups, message boards, forums
–Instant messaging
–Multimedia delivery

Lecture 29
Functions & Variable Scope
(Web Development Lecture 10)

During the last lecture we had a discussion on Arrays

We found out why we need arrays

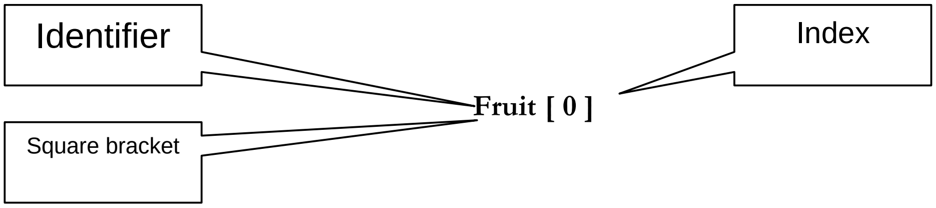
We became able to use arrays in conjunction with the 'for' loop for solving simple problems

Array

An indexed list of elements

A variable is a container that holds a value

Similarly, an Array can be considered a container as well, but this one is more interesting as it can hold multiple values



Arrays in JavaScript

In JavaScript, arrays are implemented in the form of the 'Array' object

The key property of the 'Array' object is 'length', i.e the number of elements in an array

Two of the key 'Array' methods are:

–reverse( )

–sort( )

lements of an array can be of any type; you can even have an array containing other arrays

Today's Goal:

Functions & Variable Scope

To be able to understand the concept of functions and their use for solving simple problems

To become familiar with some of JavaScript's built-in functions

To become familiar with the concept of local and global variables

29.1 Function

A group of statements that is put together (or defined) once and then can be used (by reference) repeatedly on a Web page

Also known as subprogram, procedure, subroutine

From the last lecture …

```
words = new Array ( 10 ) ;
for ( k = 0 ; k < words.length ; k = k + 1 ) {
      words[ k ] = window.prompt( "Enter word # " + k, "" ) ;
}
document.write( "UNSORTED WORDS:" + "<BR>" ) ;
for ( k = 0 ; k < words.length ; k = k + 1 ) {
      document.write( words[ k ] + "<BR>" ) ;
}
words.sort( ) ;
document.write( "SORTED WORDS:" + "<BR>" ) ;
for ( k = 0 ; k < words.length ; k = k + 1 ) {
      document.write( words[ k ] + "<BR>" ) ;
}
```

```
words = new Array ( 10 ) ;
```

```
for ( k = 0 ; k < words.length ; k = k + 1 ) {
        words[ k ] = window.prompt( "Enter word # " + k, "" ) ;
}
document.write( "UNSORTED WORDS:" + "<BR>" ) ;
for ( k = 0 ; k < words.length ; k = k + 1 ) {
        document.write( words[ k ] + "<BR>" ) ;
}
words.sort( ) ;
document.write( "SORTED WORDS:" + "<BR>" ) ;
for ( k = 0 ; k < words.length ; k = k + 1 ) {
        document.write( words[ k ] + "<BR>" ) ;
}
function writeList( heading, words ) { ß----------------------------
        document.write( heading + "<BR>" ) ;
        for ( k = 0 ; k < words.length ; k = k + 1 ) {
                document.write( words[ k ] + "<BR>" ) ;
        }
}
words = new Array ( 10 ) ;
for ( k = 0 ; k < words.length ; k = k + 1 ) {
        words[ k ] = window.prompt( "Enter word # " + k, "" ) ;
}
writeList( "Unsorted Words:", words ) ; ß----------------------
words.sort( ) ;
writeList( "Sorted List:", words ) ; < --------------------------------
```

Function definition

Let's us see if we can redouble the advantage

Function call

Function call

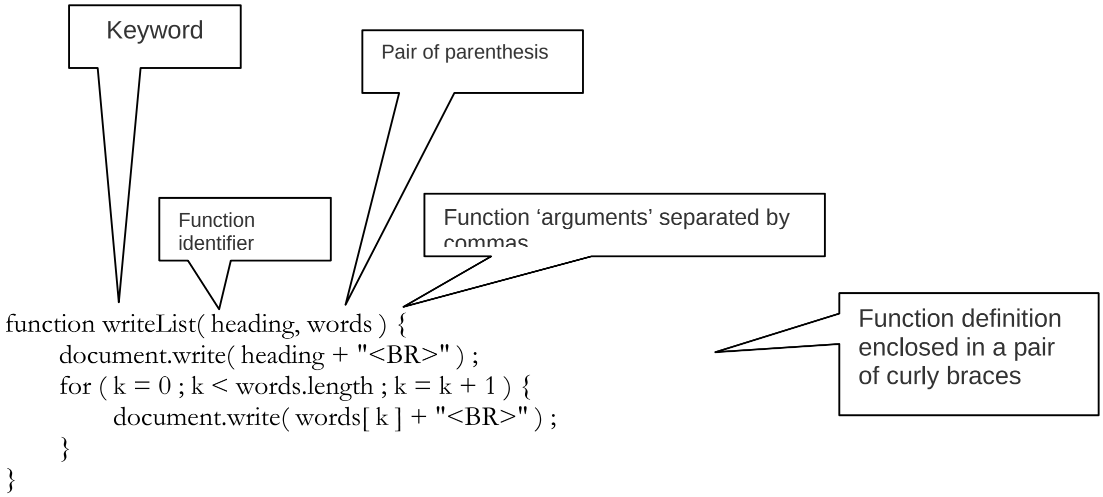
29.2 Advantages of Functions

Number of lines of code is reduced

Code becomes easier to read & understand

Code becomes easier to maintain as changes need to be made only at a single location instead multiple locations

```
function writeList( heading, words ) {
        document.write( heading + "<BR>" ) ;
        for ( k = 0 ; k < words.length ; k = k + 1 ) {
                document.write( words[ k ] + "<BR>" ) ;
        }
}
words = new Array ( 10 ) ;
for ( k = 0 ; k < words.length ; k = k + 1 ) {
        words[ k ] = window.prompt( "Enter word # " + k, "" ) ;
}
writeList( "Unsorted Words:", words ) ;
words.sort( ) ;
writeList( "Sorted List:", words ) ;
```


Keyword
Pair of parenthesis
Function identifier
Function 'arguments' separated by commas
Function definition enclosed in a pair of curly braces

```
function writeList( heading, words ) {
        document.write( heading + "<BR>" ) ;
        for ( k = 0 ; k < words.length ; k = k + 1 ) {
                document.write( words[ k ] + "<BR>" ) ;
        }
}
```

29.3 Function Identifiers

The naming rules for function identifiers are the same as were discussed for variable and array identifiers

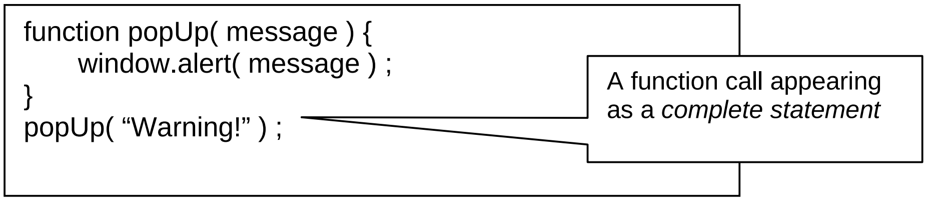
29.4 Arguments of a Function

A comma-separated list of data

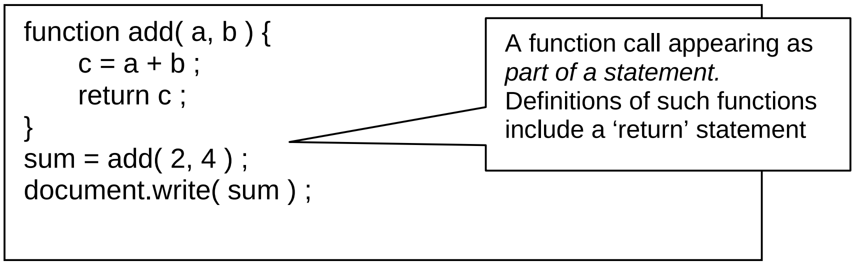
Arguments define the interface between the function and the rest of the Web page

Arguments values are passed to the function by value (some popular languages pass arguments 'by reference' as well)

To ensure that a function is defined before it is called up, define all functions in the HEAD portion of Web pages



```
function popUp( message ) {
        window.alert( message ) ;
}
popUp( "Warning!" ) ;
```


A function call appearing as *part of a statement.* Definitions of such functions include a 'return' statement

```
function add( a, b ) {
        c = a + b ;
        return c ;
}
sum = add( 2, 4 ) ;
document.write( sum ) ;
```

```
function popUp( message ) {
        window.alert( message ) ;
}
popUp( "Warning!" ) ;
```

```
function add( a, b ) {
      c = a + b ;
      return c ;
}
sum = add( 2, 4 ) ;
document.write( sum ) ;
```

What would this modifica-tion do?

```
function add( a, b ) {
      c = a + b ;
      return c ;
}
document.write( add( 2, 4 ) ) ;
```

Another Example

5! = 120

0! = ?

```
•function factorial( n ) {
•     product = 1 ;
•     for ( k = 1 ; k <= n ; k = k + 1 ) {
•           product = product * k
•     }
•     return product ;
•}

•n = window.prompt( "Enter a number ", "" ) ;

•document.write(n, "! = ", factorial( n ) ) ;
```

What Would this Statement Do?

factorial(factorial (3)) ;

This is termed as the

recursive

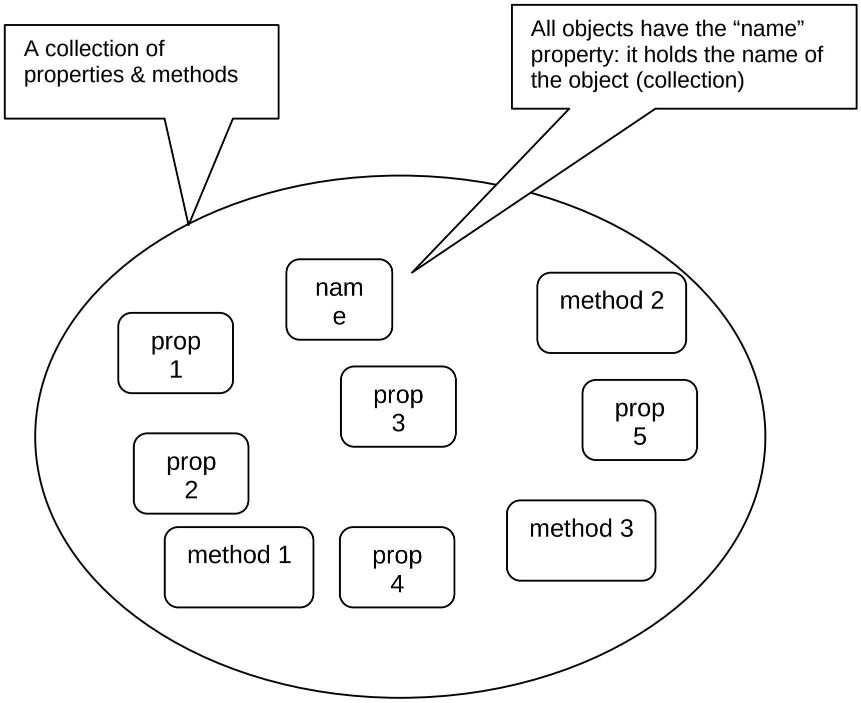
use of a function

Methods

Methods *are* functions

They are *unusual* in the sense that they are stored as properties of objects

Object: A *named* collection of properties (data, state) & methods (instructions, behavior)


A collection of properties & methods
All objects have the "name" property: it holds the name of the object (collection)
nam e
method 2
prop 1
prop 3
prop 5
prop 2
method 3
method 1
prop 4

29.5 Event Handlers

Special-purpose functions that come predefined with JavaScript

They are *unusual* in the sense that they are many times called in the HTML part of a Web page and not the <SCRIPT> … </SCRIPT> part

More on event handlers in a future lecture

Predefined, Top-Level or Built-In Functions

Event handlers are not the only functions that come predefined with JavaScript. There are many others.

Practically, there is no difference between predefined functions and those that are defined by the programmer (termed as user-defined or custom functions)

There are many of them, but here we discuss only two: parseInt( ), parseFloat( )

The dictionary meaning of 'Parse': *To breakdown into simpler components and analyze*

parseInt( )

Syntax: parseInt (string)

```
string1 = “3.14159” ;
document.write( parseInt( string1 ) ) ;
document.write( “<BR>” ) ;
string2 = “$100.00” ;
document.write( parseInt( string2 ) ) ;
document.write( “<BR>” ) ;
```

```
string3 = “ 23 ” ;
document.write( parseInt( string3 ) ) ;
```

```
3
NaN
23
```

1. Parses the string argument; returns an integer
2. If it encounters a non-numeral - anything other than (+,-) or (0-9) - it ignores it and all succeeding characters, and returns the integer value parsed up to that point
3. If the first character cannot be converted to a number, parseInt returns NaN
4. parseInt truncates numbers to integer values
5. Leading and trailing spaces are ignored

parseFloat( )

Syntax: parseFloat (string)

```
string1 = “3.14159” ;
document.write( parseFloat( string1 ) ) ;
document.write( “<BR>” ) ;
string2 = “$100.00” ;
document.write( parseFloat( string2 ) ) ;
document.write( “<BR>” ) ;
string3 = “ 23E-15 ” ;
document.write( parseFloat( string3 ) ) ;
```

1. Parses the string argument; returns a FP number
2. If it encounters a character other than

A sign (+,-)
A numeral (0-9)
A decimal point
An exponent

it returns the value up to that point, ignoring that and all succeeding characters

3. If the first character cannot be converted to a number, parseFloat returns NaN
4. Leading and trailing spaces are ignored

29.6 Scope of Variable

Defining the space in which a variable is effective is known as defining the scope of a variable. A variable can be either *local* or *global* in scope.

Local and Global Variables

Local or Function-level Variable

Effective only in the function in which they are declared

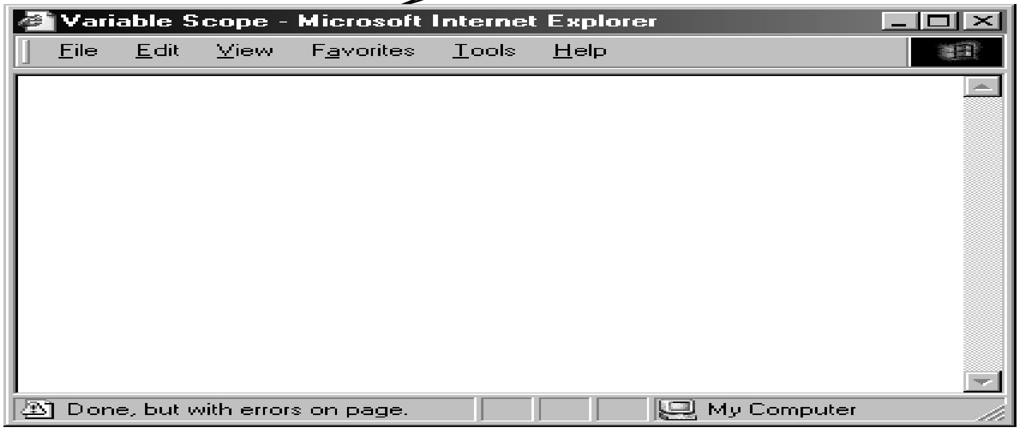
Global Variables

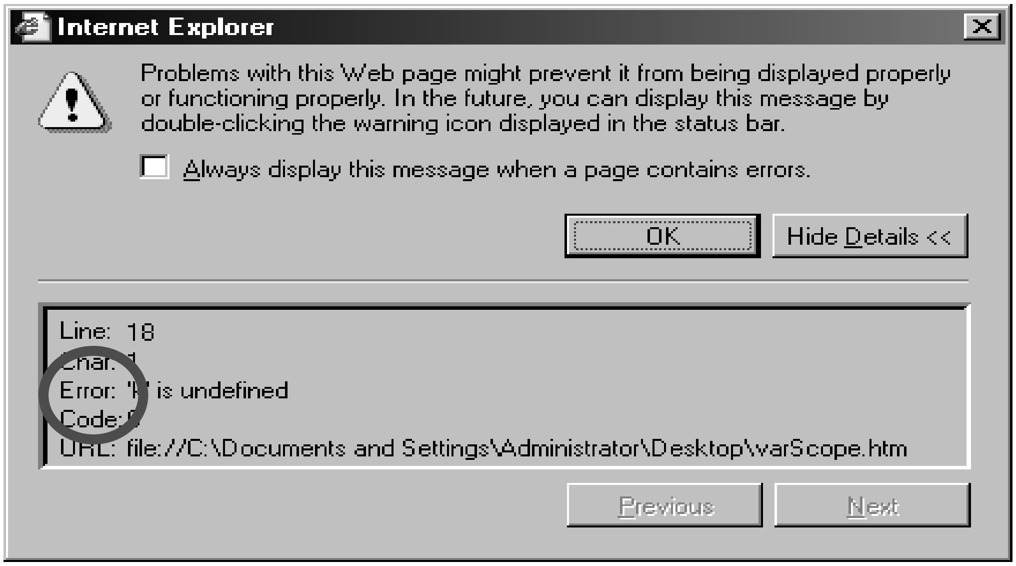
Visible everywhere on the Web page

Example

```
function factorial( n ) {
        product = 1 ;
        for ( k = 1 ; k <= n ; k = k + 1 ) {
                product = product * k
        }
        return product ;
}
n = window.prompt( "Enter a number ", "" ) ;
document.write( “k = ”, k ) ;
document.write( “<BR>” ) ;
document.write(n, "! = ", factorial( n ) ) ;
```

What would this statement write?


Variable Scope - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Done, but with errors on page.
My Computer



Problems with this Web page might prevent it from being displayed properly or functioning properly. In the future, you can display this message by double-clicking the warning icon displayed in the status bar.

□ Always display this message when a page contains errors.

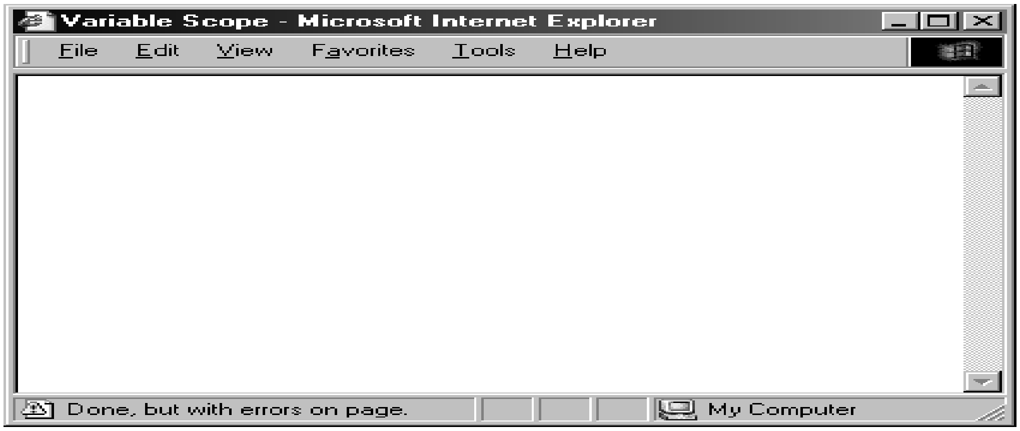
```
Line: 18
Char: 1
Error: 'k' is undefined
Code: 0
URL: file://C:\Documents and Settings\Administrator\Desktop\varScope.htm
```

```
function factorial( n ) {
	product = 1 ;
	for ( k = 1 ; k <= n ; k = k + 1 ) {
		product = product * k
	}
	return product ;
}
n = window.prompt( "Enter a number ", "" ) ;
document.write( “k = ”, k ) ;
document.write( “<BR>” ) ;
document.write(n, "! = ", factorial( n ) ) ;
```

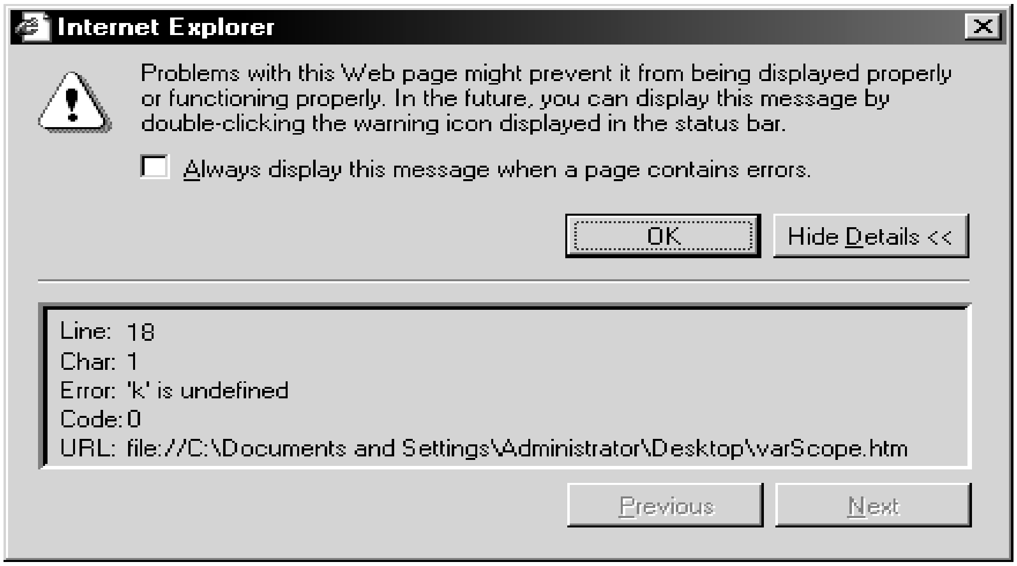
```
function factorial( n ) {
        product = 1 ;
        for ( k = 1 ; k <= n ; k = k + 1 ) {
                product = product * k
        }
        return product ;
}
n = window.prompt( "Enter a number ", "" ) ;
document.write(n, "! = ", factorial( n ) ) ;
document.write( "<BR>" ) ;
document.write( "k = ", k ) ;
```

```
10! = 3628800
k = 11
```

```
function factorial( n ) {
        var k ;
        product = 1 ;
        for ( k = 1 ; k <= n ; k = k + 1 ) {
                product = product * k
        }
        return product ;
}
n = window.prompt( "Enter a number ", "" ) ;
document.write(n, "! = ", factorial( n ) ) ;
document.write( "<BR>" ) ;
document.write( "k = ", k ) ;
```



Done, but with errors on page.



Problems with this Web page might prevent it from being displayed properly or functioning properly. In the future, you can display this message by double-clicking the warning icon displayed in the status bar.

□ Always display this message when a page contains errors.

```
Line: 18
Char: 1
Error: 'k' is undefined
Code: 0
URL: file://C:\Documents and Settings\Administrator\Desktop\varScope.htm
```

‘k’ is a Local Variable

‘k’ is not declared or used in the main code
Instead, it is declared within the function ‘factorial’ *only*
‘k’ is local to the ‘factorial’ function, and does not hold any meaning outside that function

```
function factorial( n ) {
        var k, product ;
        product = 1 ;
        for ( k = 1 ; k <= n ; k = k + 1 ) {
                product = product * k
        }
        return product ;
}
n = window.prompt( "Enter a number ", "" ) ;
document.write(n, "! = ", factorial( n ) ) ;
document.write( “<BR>” ) ;
document.write( product ) ;
```

Here ‘product’ has been made a local variable as well

What would this statement write?

Local Variables

Declaring variables (using the var keyword) within a function, makes them *local*•They are available only within the function and hold no meaning outside of it

Global Variables

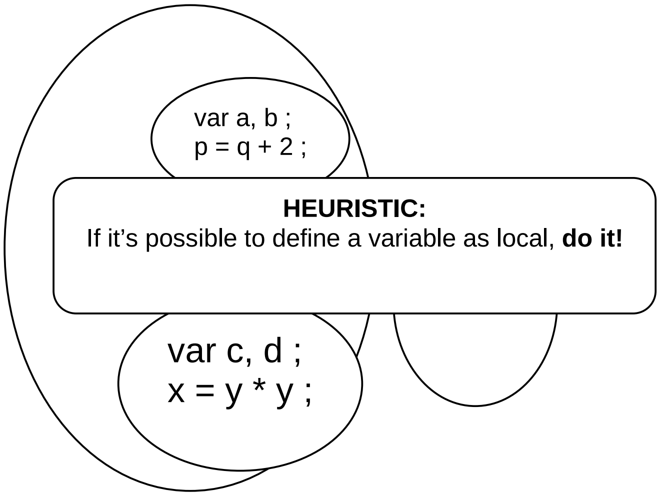
All other variables used in a Web page (or window) are *global*
They can be manipulated from the main code as well as from any of the functions
They include:
–All variables declared in the main code
–All variables used but not declared in the main code
–All variables used but not declared in any of the functions defined on the Web page (or window)

```
function writeList( heading, words ) {
        document.write( heading + "<BR>" ) ;
        for ( k = 0 ; k < words.length ; k = k + 1 ) {
                document.write( words[ k ] + "<BR>" ) ;
        }
}
words = new Array ( 10 ) ;
for ( k = 0 ; k < words.length ; k = k + 1 ) {
        words[ k ] = window.prompt( "Enter word # " + k, "" ) ;
}
writeList( “Unsorted Words:”, words ) ;
words.sort( ) ;
writeList( “Sorted List:”, words ) ;
words.reverse( ) ;
writeList( “Reverse-Sorted List:”, words ) ;
```

Would the functionality change if we delete the argument ‘words’ from these 4 places?

Local –vs- Global

Global variables can make the logic of a Web page difficult to understand
Global variables also make the reuse and maintenance of your code much more difficult



HEURISTIC:
If it's possible to define a variable as local, **do it!**

Glob	Local
u	a
m	b
p	c
q	d
x	
y	
r	
s	

Variables declared within functions are local; all others global

<u>During Today's Lecture …</u>

We looked at functions and their use for solving simple problems
We became familiar with a couple of JavaScript's built-in functions
We became familiar with the concept of local and global variables

<u>Next Web Dev Lecture:</u>

Event Handling

We'll learn to appreciate the concept of event driven programming
We will produce solutions for simple problems using various event handlers

Lecture 30

Internet Services

During the last lecture …

(Introduction to the Internet)

We looked at the role Internet plays in today's computing

We reviewed some of the history and evolution of the Internet

Internet: The Enabler

Enables attractively-priced workers located in Pakistan to provide services to overseas clients

Enables users to easily share information with others located all over the world

Enables users to easily, inexpensively communicate with others remote users

Enables the users to operate and run programs on computers located all over the world

The Internet is unlike any previous human invention. It is a world-wide resource, accessible to all of the humankind.

Key Characteristics

Geographic Distribution	Global - reaches around the world
Robust Architecture	Adapts to damage and error
Speed	Data *can* travels at near '*c*' on copper, fiber, airwaves

Universal Access

Same functionality to everyone

Growth Rate

The fastest growing technology ever

Freedom of Speech

Promotes freedom of speech

The Digital Advantage

Is digital: can correct errors

Internet: Network of Networks

A large number of networks, interconnected physically

Capable of communicating and sharing data with each other

From the user's point view, Internet – a collection of interconnected networks – looks like a single, unified network

TCP/IP Transmission Control Protocol/Internet Protocol

TCP breaks down the message to be sent over the Internet into packets

IP routes these packets through the Internet to get them to their destination

When the packets reach the destination computer, TCP reassembles them into the original message

1960's

1969 - DoD-ARPA creates an experimental network – ARPANET – as a test-bed for emerging networking technologies

ARPANET originally connected 4 universities & enabled scientists to share info & resources across long distances

1980's

1983 - The TCP/IP protocols becomes the only set of protocols used on the ARPANET

This sets a standard for all networks, and generates the use of the term Internet as the net of nets

1990's1993 - CERN releases WWW, developed by Tim Berners-Lee

It uses HTTP and hypertext, revolutionizing the way info is presented & accessed on Internet

1990's1993-1994 - Web browsers Mosaic & Netscape Navigator are introduced

Their GUI makes WWW & Internet more appealing to the general public

Today's Goal: Internet Services

To look at several services provided by the Internet

–FTP

–Telnet
–Web
–eMail
–Instant messaging
–VoIP

- *But first, we need to find out about the addressing scheme used on the Internet*

30.1 Internet Addressing

Regular post cannot be delivered unless we write a destination address on the envelope
Same is true for the Internet
Regular post can be delivered at the intended address even if the given address is not precise. That is not the case for Internet addressing

www.vu.edu.pk
203.215.177.33

IP Address

A unique identifier for a computer on a TCP/IP network
Format: four 8-bit numbers separated by periods. Each 8-bit number can be 0 to 255
Example:
–203.215.177.33 (IP address of the VU Web server)
Networks using TCP/IP route messages based on the IP address of the destination
Any IP addresses (as long as they are unique) can be assigned within a PN
However, connecting a PN to the Internet requires using unique, registered IP addresses

Domain Names

IP addresses are fine for computers, but difficult to recognize and remember for humans
A domain name is a meaningful, easy-to-remember 'label' for an IP address
Examples:
www.vu.edu.pk
216.239.33.101 www.google.com

30.2 DNS: Domain Name System

DNS is the way that Internet domain names are located & translated into IP addresses
Maintaining a single, central table of domain name/IP address relationships is impractical
–Billions of DNS-IP translations take place every day
–The DNS-IP tables get updated continuously
Tables of DNs & IP addresses are distributed throughout the Internet on numerous servers
There is a DNS server at most ISPs. It converts the domain names in our Internet requests to actual IP addresses
In case it does not have a particular domain name in its table, it makes a request to another DNS server on the Internet

30.3 Internet Services

There are many, but we will look at only the following:
FTP
Telnet
Web
eMail
Instant messaging
VoIP

FTP: File Transfer Protocol

Used to transfer files between computers on a TCP/IP network (e.g Internet)
Simple commands allow the user to:

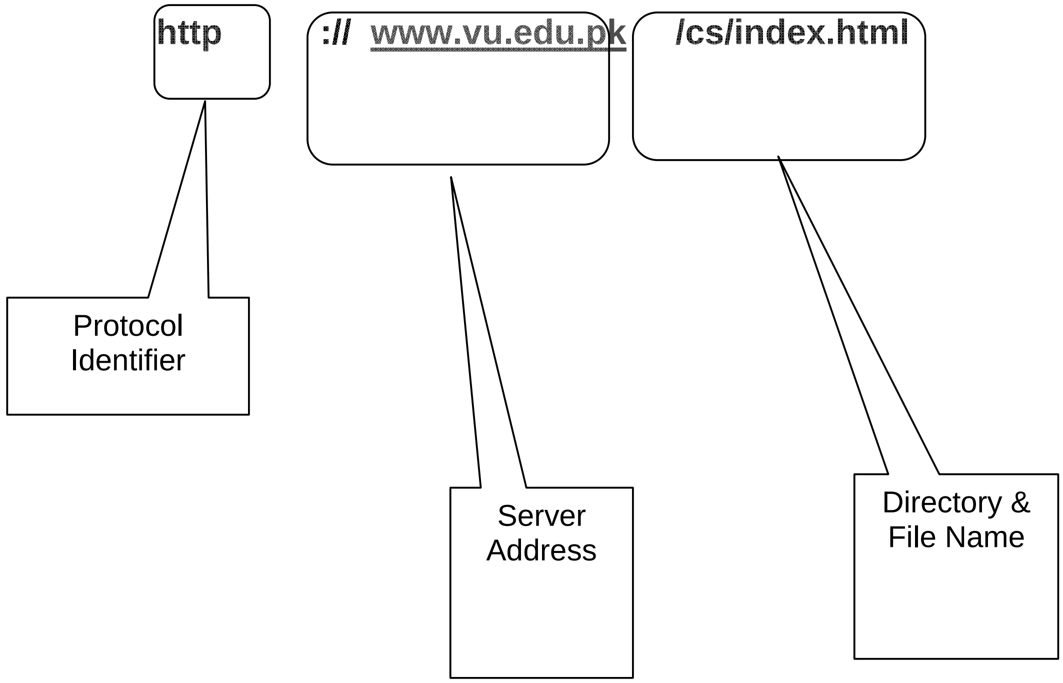
List, change, creat[redacted]rs on a remote computer
Upload and downes
Typical use: Transferring Web content from the developer’s PC to the Web server

Telnet Protocol

Using Telnet, a user can remotely log on to a computer (connected to the user’s through a TCP/IP network, e.g. Internet) & have control over it like a local user, including control over running various programs
In contrast, FTP allows file operations only
Typical use: Configuring and testing of a remote Web server

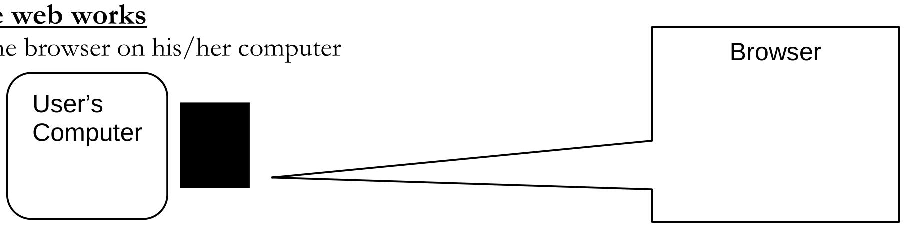
The Web

The greatest, shared resource of information created by humankind
A user may access any item on the Web through a URL, e.g.
http://www.vu.edu.pk/cs/index.html
Before, going any further, let us dissect this URL



How does the web works

User launches the browser on his/her computer

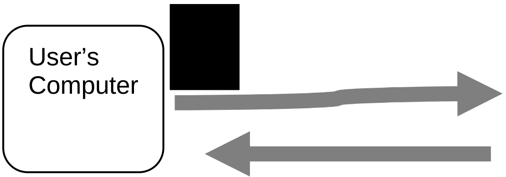


User types in the URL into the browser
The browser breaks down the URL into 3 parts :

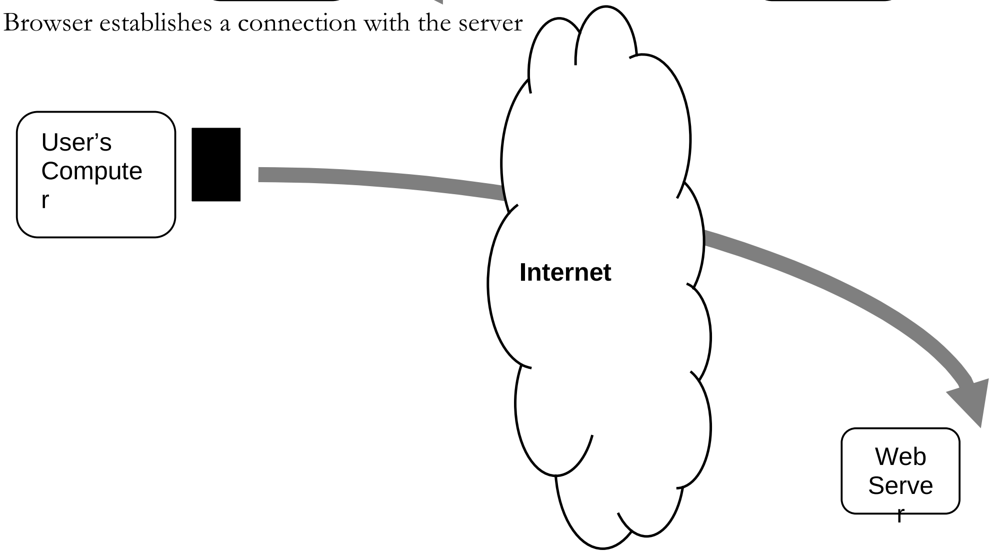
Protocol Identifier
Server Address
Directory & File Name

Browser sends server’s name to the DNS server

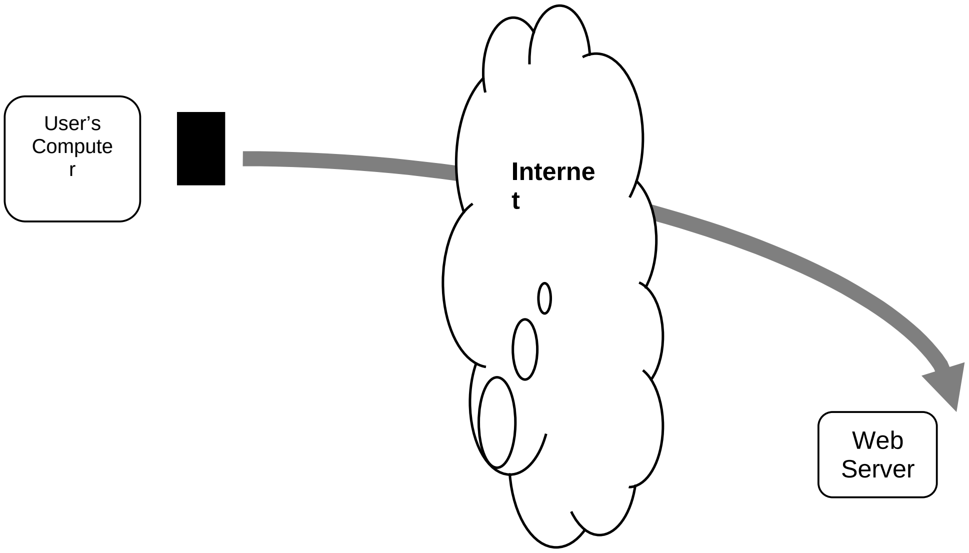
Domain Name



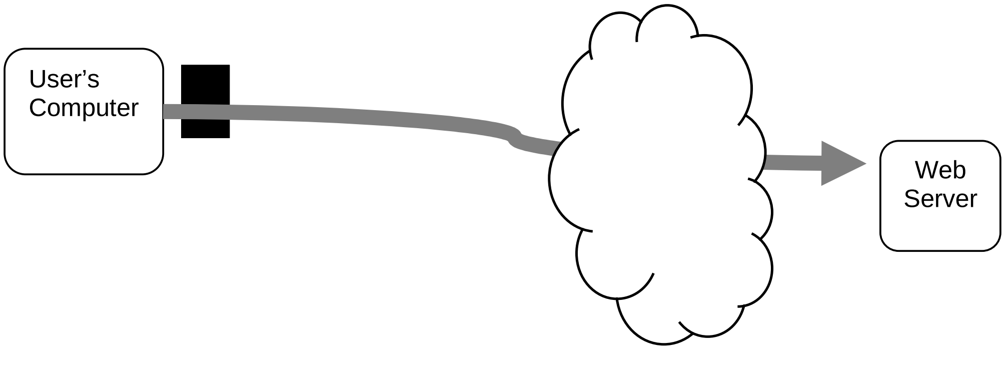
Browser establishes a connection with the server



<u>Browser sends a 'GET' request for cs/index.html</u>



<u>Server sends the requested file to the browser</u>



Browser displays index.html

email

Computer-to-computer messaging

Inexpensive, and quite quick, but not instant!

The most popular service on the Internet, even more than surfing, but soon to be overtaken by instant messaging

Billions are sent every day

30.3 How does an eMail system work?

But first, the components:

eMail client

SMTP server

POP3 server

eMail Clients

Programs used for writing, sending, receiving, and displaying eMail messages

Examples: Outlook, Communicator, Hotmail, YahooMail

SMTP: Simple Mail Transfer Protocol

A protocol used to send and receive eMail messages over a TCP/IP network

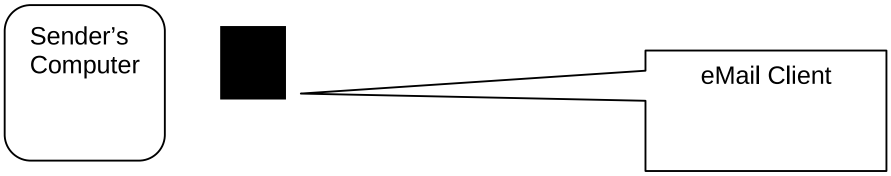
POP3: Post Office Protocol

A protocol used for receiving eMail messages

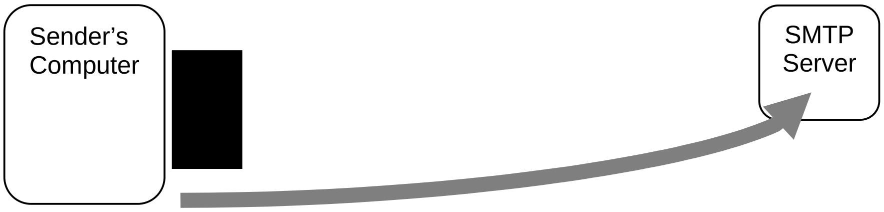
A POP3 server maintains text files (one file per user account) containing all messages received by a user

eMail client interacts with the POP3 server for discovering and downloading new eMail messages

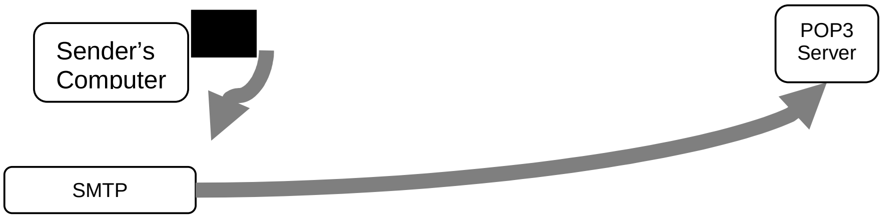
The message is prepared using the eMail client



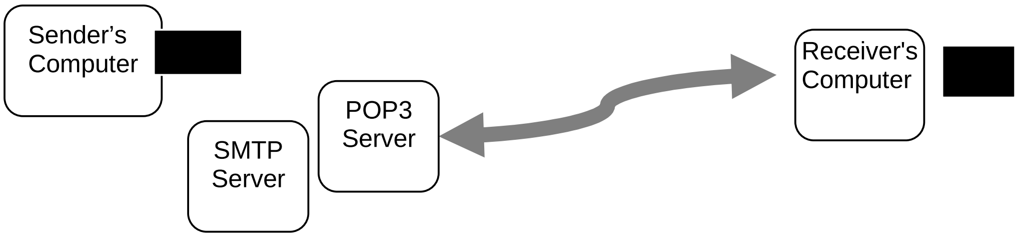
The eMail client sends it to the SMTP server



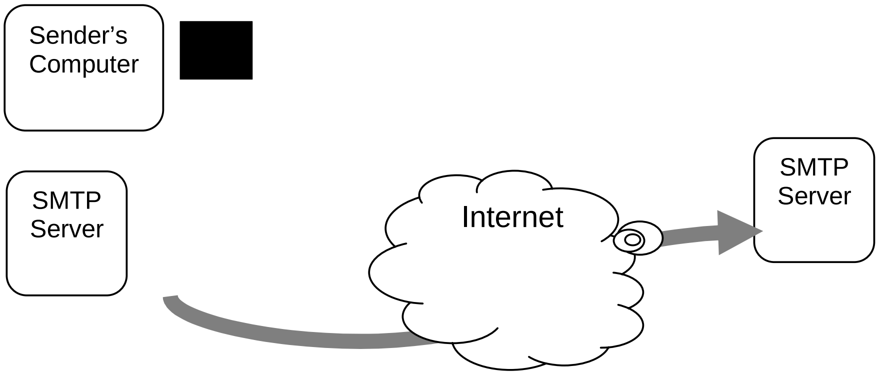
If the receiver is local, it goes to the POP3 server



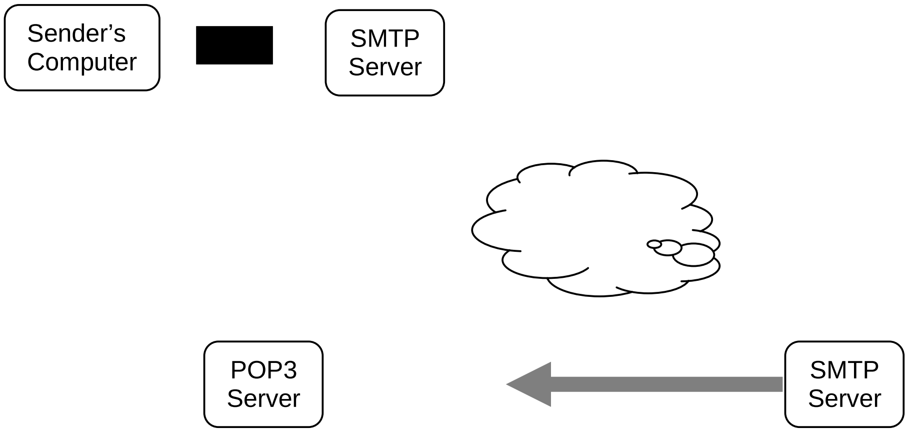
The receiver picks it at his/her convenience



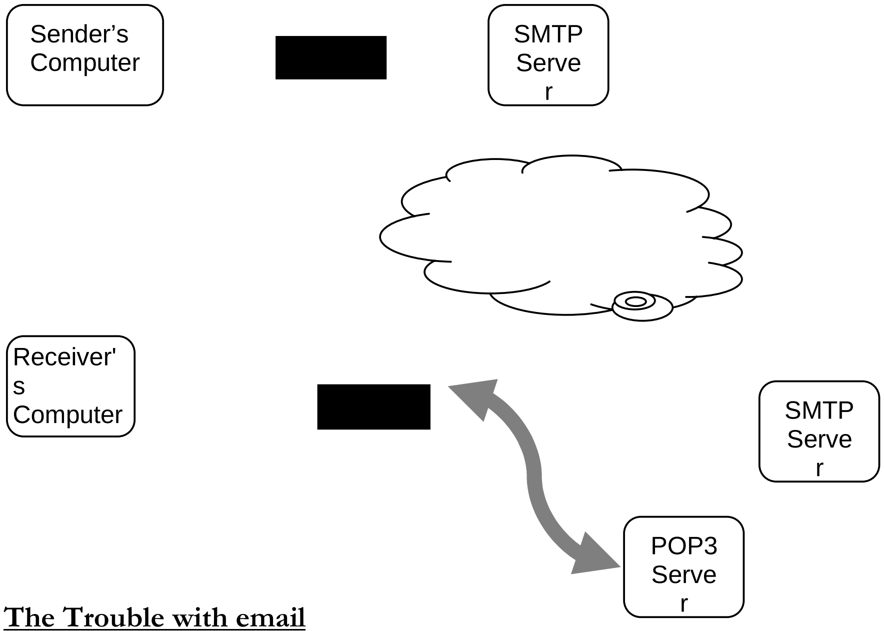
Otherwise, it is sent to receiver's SMTP server



Which forwards it to the local POP3 server



The receiver picks it at his/her convenience



The Trouble with email

Slow response times

No way of knowing if the person we are sending eMail to is there to read it

The process of having a conversation through eMail by exchanging several short messages is too cumbersome

Instant messaging (IM) solves these problems

Instant Messaging

- The IM services available on the Internet (e.g. ICQ, AIM, MSN Messenger, Yahoo! Messenger) allow us to maintain a list of people (contacts) that we interact with regularly
- We can send an instant messages to any of the contacts in our list as long as that contact is online

30.4 Using Instant Messaging

Whenever a contact in our list comes online, the IM client informs us through an alert message and by playing a sound

To send an instant message to a contact, just click on the contact in the IM client, and start typing the message

The selected contact will receive that message almost immediately after you press 'Enter'

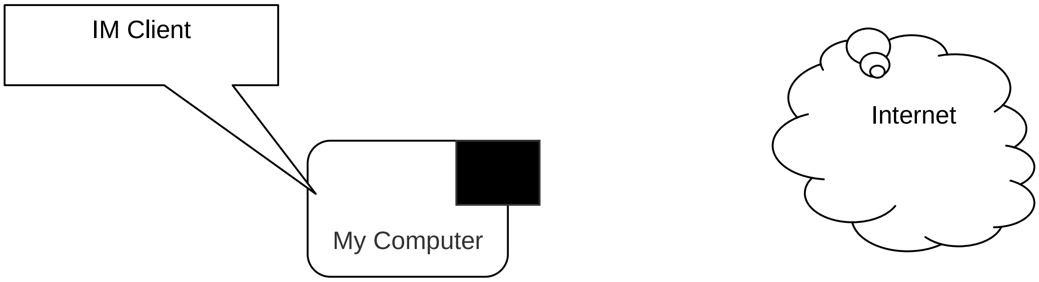
When the contact's IM client receives the message, it alerts the contact with a blinking message and by playing a sound

That contact then can type a response to the received message, and send it instantly

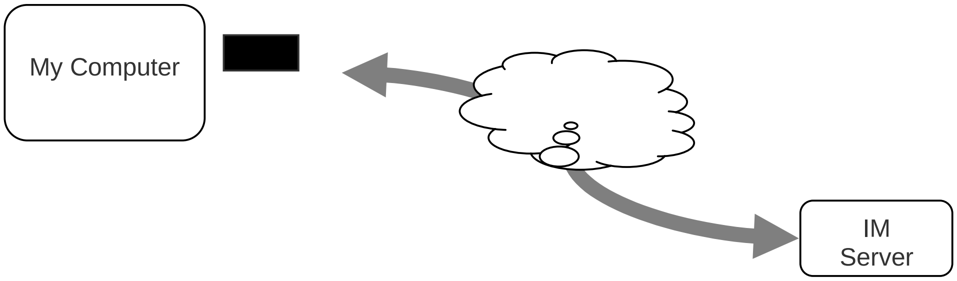
Several such conversations can be carried out simultaneously, each occupying a separate IM windows

How instant messaging works?

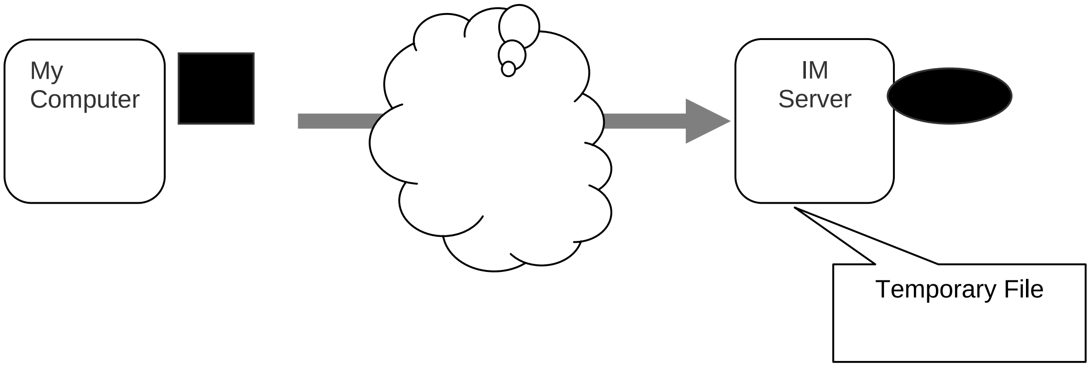
User launches the IM client



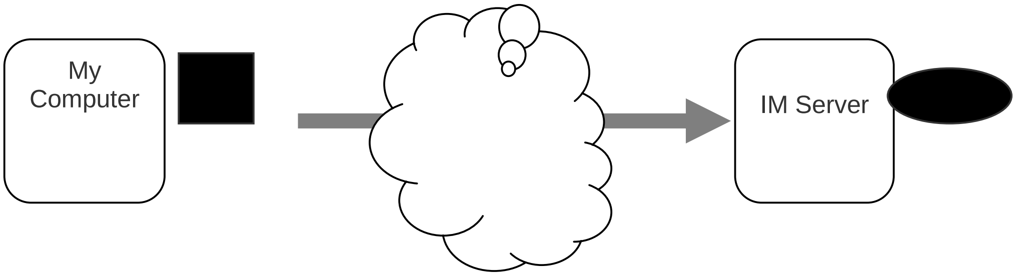
IM client finds the IM server & logs in



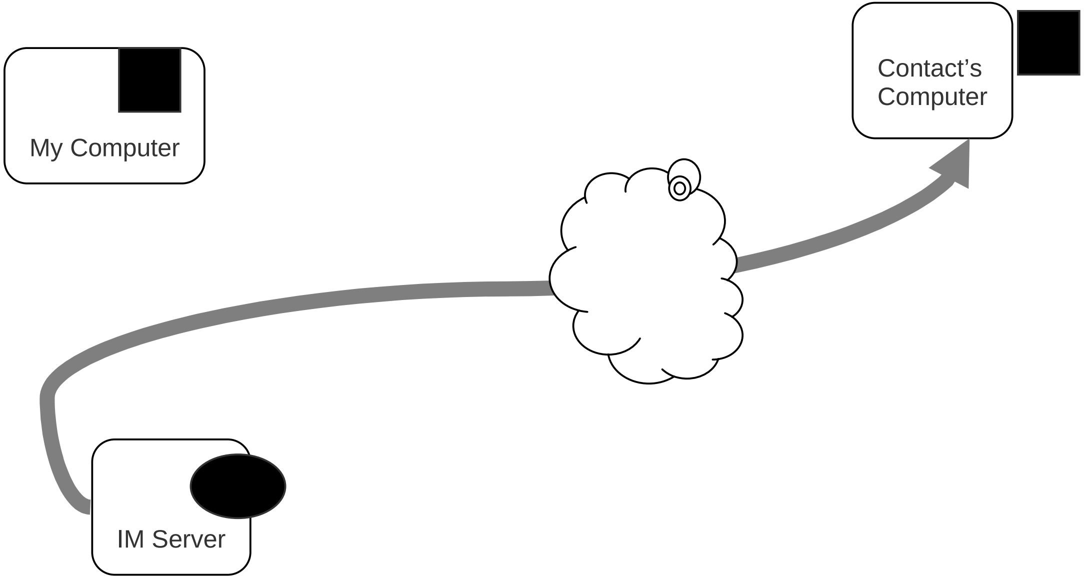
It sends communication info (IP address, etc) to the IM server



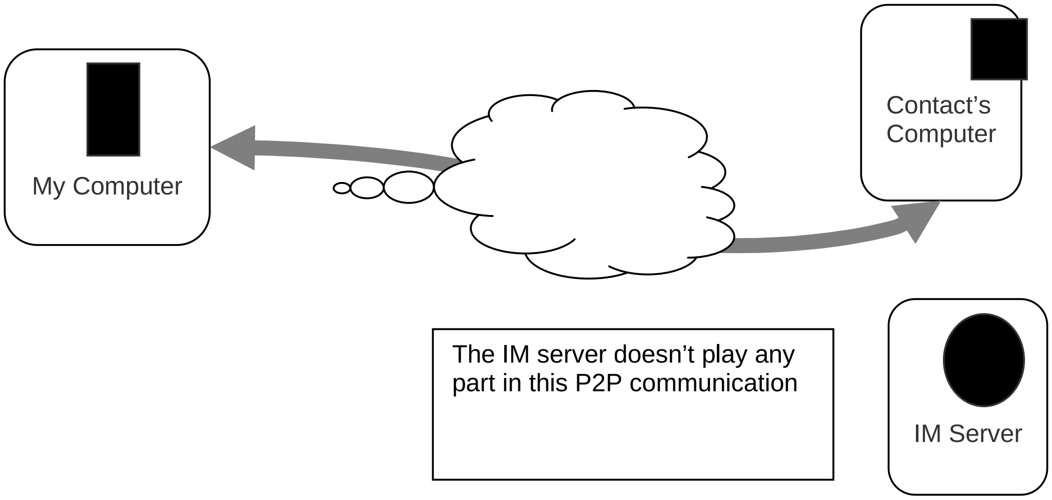
IM server finds user's contacts & sends him/her the communication info for the ones online



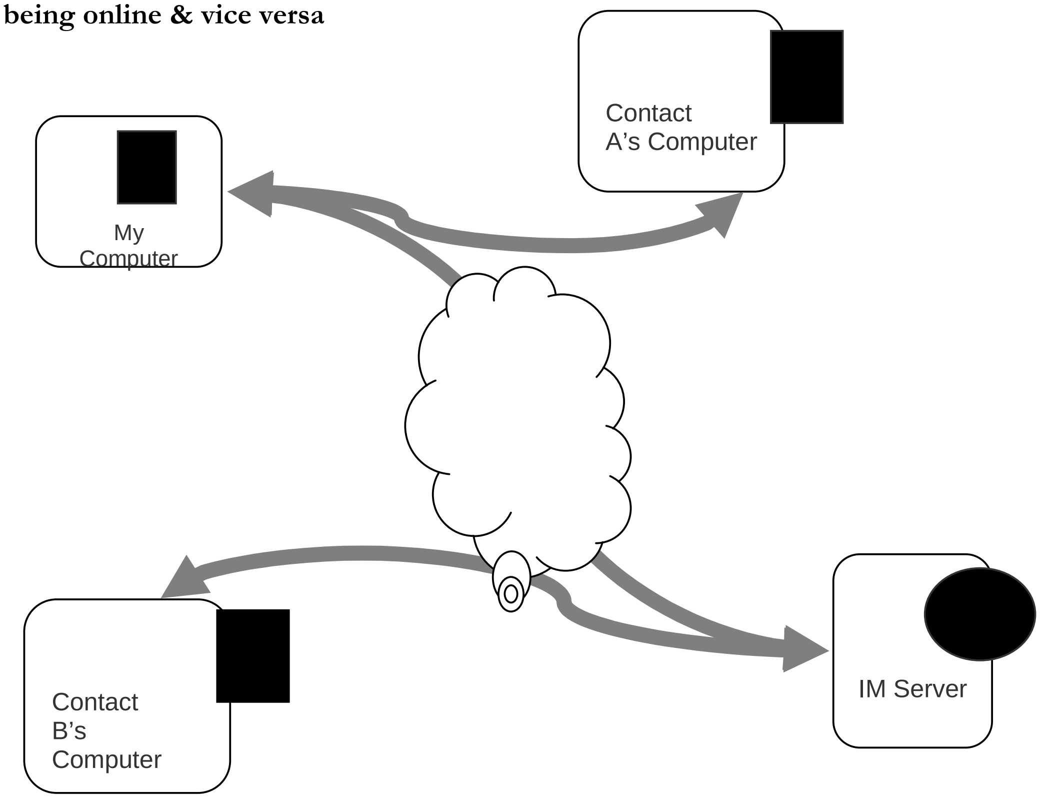
IM server also tells the contacts that the user is online; sends his/her communication info to them



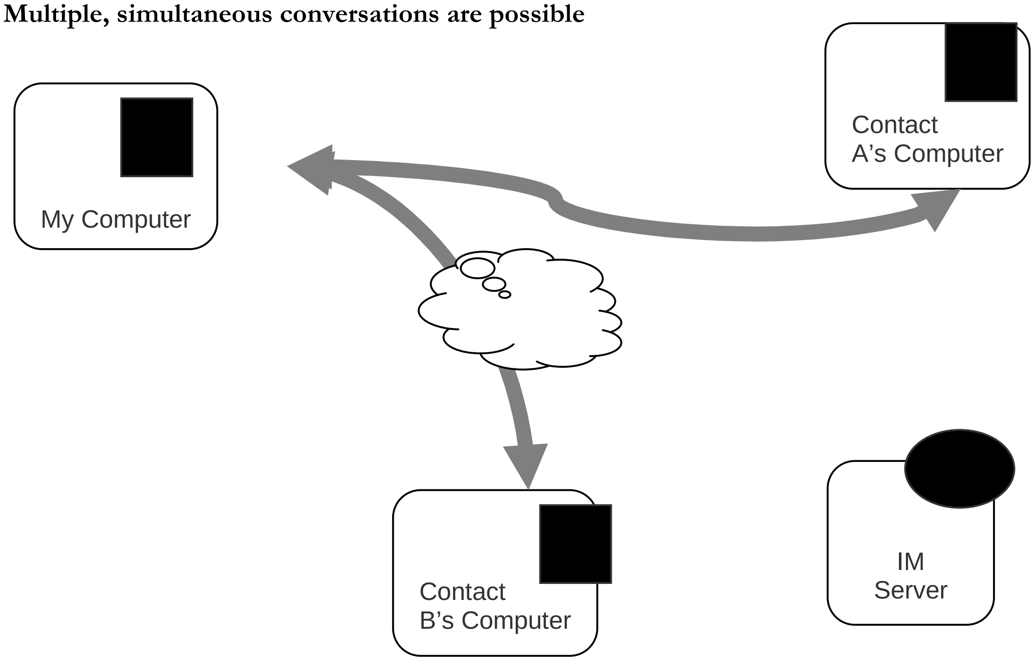
Now the user's & the contact's IM clients are ready to communicate directly (P2P)


Contact's Computer
My Computer
The IM server doesn't play any part in this P2P communication
IM Server

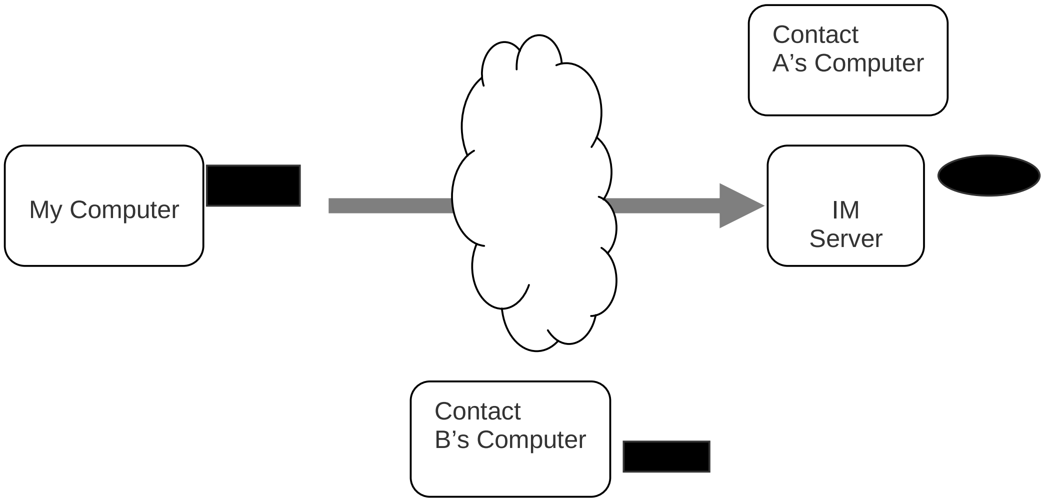
As new contact's come online, IM server informs them about the user being online & vice versa



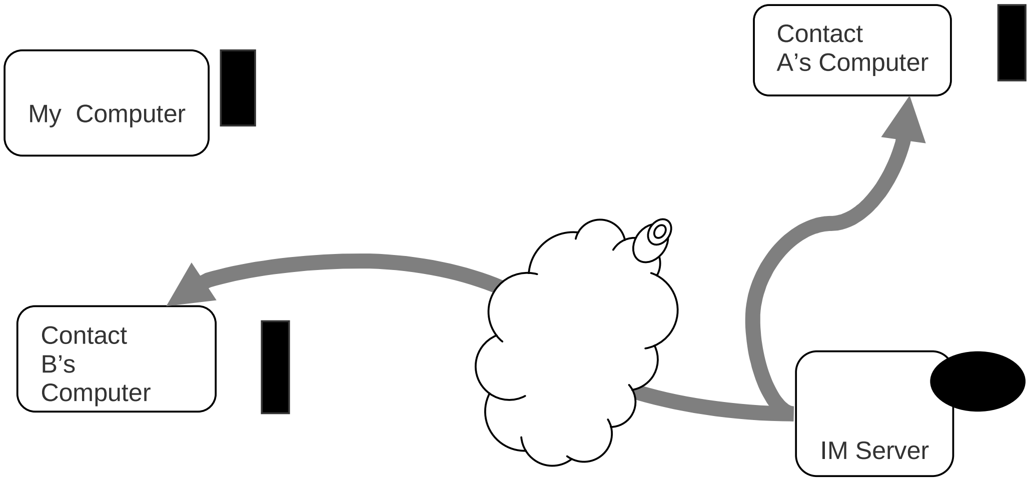
Multiple, simultaneous conversations are possible



When the user logs-off, his/her IM client informs the IM server



IM server erases the temporary file and informs the user's contact's about his/her 'offline' status



Key Point

Once the IM server provides the communication info to the user and his/her contact's IM client, the two are able to communicate with each other without the IM server's assistance

This server-less connection is termed as a P2P connection

Question

Why do we require the server in the first place?

Why doesn't my IM client look for the user's contact's IM client without the IM server's help?

Answer

Many users (including almost all home users) do not have permanent IP addresses. They are assigned temporary IP addresses by their ISP each time they connect to the Internet

The server-based IM scheme removes the need of having permanent IP numbers

It also gives IM users true mobility, allowing them the use of IM from any Internet-connected computer

30.5 VoIP: Voice over IP

Voice delivered from one device to another using the Internet Protocol

Voice is first converted into a digital form, is broken down into packets, and then transmitted over a TCP/IP network (e.g. Internet)

Four modes:

C2C

C2T

T2C

T2T (with a TCP/IP net somewhere in between)

Pro

Much cheaper than traditional phone service

Con

Noticeably poor quality of voice as compared with land-line phone service, but not much worse than cell phone service

Today's Goal: Internet Services

We looked at several services provided by the Internet

FTP

Telnet

Web

eMail

Instant messaging

VoIP

We also found out about the addressing scheme used on the Internet

Next Lecture:

Next lecture (Lecture 31) - the third one in the four-lecture productivity SW sequence - will be on developing presentations

However, during lecture 33, we will become familiar with the role that graphics and animations play in computing

Lecture 31

Developing Presentations

Focus of the 22th Lecture was on Spreadsheets:

Second among the four lectures that we plan to have on productivity software
We learnt about what we mean by spreadsheets
We discussed the usage of various functions provided by common spreadsheets

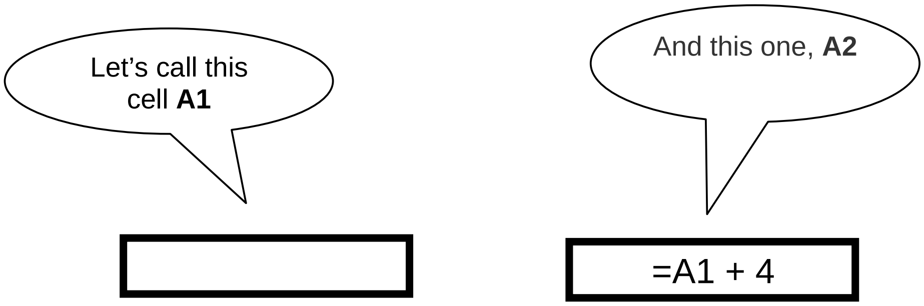
Spreadsheets:

Electronic replacement for ledgers
Used for automating engineering, scientific, but in majority of cases, business calculations.
A spreadsheet - VisiCalc - was the first popular application on PC's.
What Can They Do?
Can perform calculations repeatedly, accurately, rapidly.
Can handle a large number of parameters, variables
Make it easy to analyze what-if scenarios for determining changes in forecasts w.r.t. change in parameters.
Are easy to interface with other productivity SW packages
Easy to store, recall, modify
Make it is easy to produce graphs:

The Structure of A Spreadsheet:

Collection of cells arranged in rows and columns
Each cell can contain one of the following:
Numbers
Text
Formulas
These cells display either the number or text that was entered in them or the value that is found by executing the formula.

Connecting Two Cells:



Today's Lecture:

Developing Presentation:

Third among the four lectures that we plan to have on productivity software
We will discuss several design guidelines for making effective multimedia presentations
We will become able to develop simple presentation with the help of presentation making software

31.1 Presentations:

I used to use transparencies in conjunction with overhead projectors for making presentations
Some time back, I used to write on transparencies with felt-tip markers
Then I moved on to developing presentations on a PC, and printing the final version on transparencies with a laser printer
Some of my contemporaries used color inkjet printers instead of the laser printer
Another option was to develop them on a computer and then transfer to 35mm slides using a camera, and display it using a slide projector

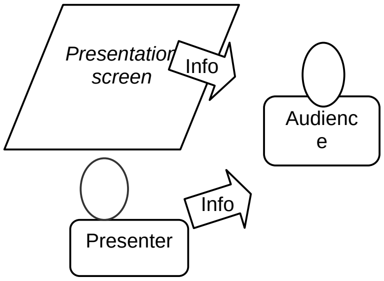
Problems With All Those Modes:

It was difficult and often costly to make changes, especially last minute changes
No sound, no animation, no video
Electronic transmission, in some cases, was not easy
It was difficult keeping track of old ones and making sure of their proper storage

Solution: Multimedia Presentations :

Great tool for effectively communicating ideas to an audience
All electronic
Easy to make last minute changes
The undo feature encourages experimentation
More attractive; commanded more interest
May include animations, sound, video
Easy to catalog, store, and recall
Great tool for making presenter-free interactive material (e.g. self-learning tutorials)

The Presentation Scenario:



The Goal of the Presenter:

Maximize the (sum of the 2 types of) info that needs to be transferred to the audience.
Recommended Approach
Put together a presentation that is:
Simple,clear ,consistent, design guidelines for simplicity, clarity, consistency

Layout Guidelines:

Keep layouts simple
Vary the look of successive slides. Mix up graphics with bulleted lists with animations
Avoid cluttering the slides with too much text or graphics. Your audience should hear what you have to say and not be distracted by a busy layout
Put a title on each slide. As soon as the audience see the slide, the title should make it clear as to the point of that slide

Slide Background:

Keep the backgrounds simple. You want a background that shows off your info, not one that makes it illegible
Avoid bright background colors. Light colored text against a dark background works best
Keep colors, patterns, and text styles consistent (not necessarily the same) for all slides in a presentation

Color Usage Guidelines:

Use color sparingly to highlight a point, but don't get carried away
Choose them with care; at times, the wrong choice may convey an unintended message
Select background colors that are easy on the eye for several minutes of viewing, e.g. don't go for a bright yellow or red or other warm colors for background
Instead, use cool colors like blues and greens as backgrounds

Writing Text:

Limit text to a few phrases on a screen. A good rule of thumb is 5±2 lines on a slide
Write short phrases - not sentences - in the form of bulleted points: if you display sentences on your slides, you have nothing to add!

Have every bullet on a slide begin with a verb, or alternatively, have each begin with a noun

Text Usage Guidelines:

Normal text is easier to read than ALL CAPS
Avoid ornate typefaces
Use a clean & readable typeface, e.g. sans serif ones (Arial, Verdana, Helvetica)
Use at least a 24-point size, with the normal text size being 28-32
Be consistent in type size throughout the presentation
Keep text simple and easy to read by not using many different text styles (bold, italics, underline) different typefaces, different font sizes, varying font colors within a sentence

A Word of Caution on Guidelines:

These guidelines are not 'Laws of Nature'
For example, if I keep on repeating the same type face and font size and background throughout a long presentation, I'll put the audience to sleep
At times, I use a warm background color or a very large (or small!) font size on a slide or two just to wake the audience up, or to make an important point

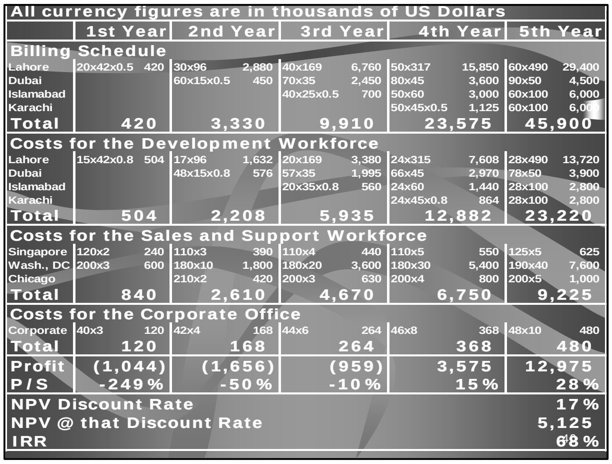
Graphics & Images:

Use simple graphics or images in place of text
Example:
Components of an OS diagram (lecture 11)
It not only listed the components in the form of colored discs, but also gave info visually about their interactions (through overlaps) and relative importance (through the size of each disc)

Animations & Transitions:

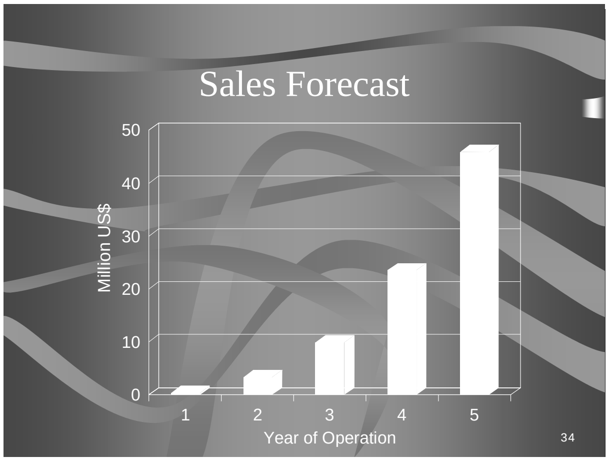
Use simple slide transitions. Too many different transitions are distractive
Animation is especially suitable for displaying:
Steps of a process: Waterfall model
Flow of info in a system: How does IM works?

Graphics and Images examples:



All currency figures are in thousands of US Dollars

	1st Year		2nd Year		3rd Year		4th Year		5th Year	
Billing Schedule										
Lahore	20x42x0.5	420	30x96	2,880	40x169	6,760	50x317	15,850	60x490	29,400
Dubai			60x15x0.5	450	70x35	2,450	80x45	3,600	90x50	4,500
Islamabad					40x25x0.5	700	50x60	3,000	60x100	6,000
Karachi							50x45x0.5	1,125	60x100	6,000
Total		**420**		**3,330**		**9,910**		**23,575**		**45,900**
Costs for the Development Workforce										
Lahore	15x42x0.8	504	17x96	1,632	20x169	3,380	24x315	7,608	28x490	13,720
Dubai			48x15x0.8	576	57x35	1,995	66x45	2,970	78x50	3,900
Islamabad					20x35x0.8	560	24x60	1,440	28x100	2,800
Karachi							24x45x0.8	864	28x100	2,800
Total		**504**		**2,208**		**5,935**		**12,882**		**23,220**
Costs for the Sales and Support Workforce										
Singapore	120x2	240	110x3	390	110x4	440	110x5	550	125x5	625
Wash., DC	200x3	600	180x10	1,800	180x20	3,600	180x30	5,400	190x40	7,600
Chicago			210x2	420	200x3	630	200x4	800	200x5	1,000
Total		**840**		**2,610**		**4,670**		**6,750**		**9,225**
Costs for the Corporate Office										
Corporate	40x3	120	42x4	168	44x6	264	46x8	368	48x10	480
Total		**120**		**168**		**264**		**368**		**480**
Profit		**(1,044)**		**(1,656)**		**(959)**		**3,575**		**12,975**
P / S		**-249%**		**-50%**		**-10%**		**15%**		**28%**
NPV Discount Rate										**17%**
NPV @ that Discount Rate										**5,125**
IRR										**68%**



31.2 The Structure of A Presentation:

Title slide
Overview slide
Main body
Slide 1
Slide 2
Slide 3
…
…
Summary slide
Divide long presentations into sections, and have separate title, overview, summary, body slides for each section

31.3 Presentation Development SW:

One can use a word processor to develop presentations of reasonable quality
However, using a SW package especially designed for developing presentation can:
Speed-up the task
Make available features not available in standard word processors

Presentation development SW lets users:

Choose from a variety of ready-made presentation designs
Create original designs as well as change colors, background, fonts in ready-made designs
Add, delete, move slides within a presentation
Insert graphics & images, or create their own

Presentation development SW lets users:

Import from other applications or create new tables/plots
Create simple animations
Incorporate sound and videos
Add hyperlinks, custom navigational controls
Save work in HTML, PDF, graphics formats
The Best Feature: Undo
Allows you to recover from your mistakes
Allows you to experiment without risk

Popular SW:

Microsoft PowerPoint
CA Harvard Graphics
Lotus Freelance Graphics
Corel Presentation

Let's now demonstrate the use of the presentation making SW:

We will create a new presentation
Enter text
Add, delete, and move slides
View slide show

Today's Lecture was the …:

Third among the four lectures that we plan to have on productivity software
We discussed several design guidelines for making effective multimedia presentations
We became able to develop simple presentation with the help of presentation software

Focus of the Final Productivity SW Lecture: Database SW:

To become familiar with the basic functions and features of desktop data management software
To become able to build a small application with the help of database software

Lecture 32
Event Handling
(Web Development Lecture 11)

During the last lecture we discussed Functions & Variable Scope:

We looked at functions and their use for solving simple problems

We became familiar with a couple of JavaScript's built-in functions

We became familiar with the concept of local and global variables

Function:

A group of statements that is put together (or defined) once and then can be used (by reference) repeatedly on a Web page

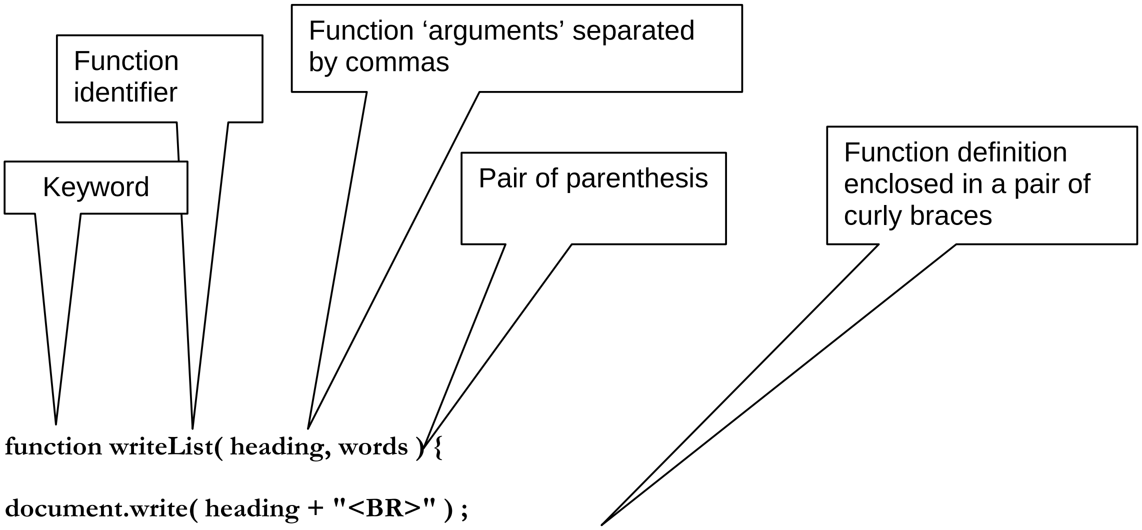
Also known as subprogram, procedure, subroutine

Advantages of Functions:

Number of lines of code is reduced

Code becomes easier to read & understand

Code becomes easier to maintain as changes need to be made only at a single location instead multiple locations



Function 'arguments' separated by commas

Function identifier

Keyword

Pair of parenthesis

Function definition enclosed in a pair of curly braces

```
function writeList( heading, words ) {

document.write( heading + "<BR>" ) ;

for ( k = 0 ; k < words.length ; k = k + 1 ) {
document.write( words[ k ] + "<BR>" ) ;
}
}
```

Arguments of a Function:

A comma-separated list of data

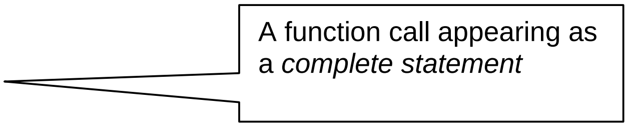
Arguments define the interface between the function and the rest of the Web page

Arguments values are passed to the function by value (some popular languages pass arguments 'by reference' as well)

To ensure that a function is defined before it is called up, define all functions in the HEAD portion of Web pages

Two Ways of Calling Functions:

```
function popUp( message ) {
    window.alert( message ) ;
}
popUp( "Warning!" ) ;
```



A function call appearing as a *complete statement*

```
function add( a, b ) {
      c = a + b ;
      return c ;
}
sum = add( 2, 4 ) ;
document.write( sum ) ;
```

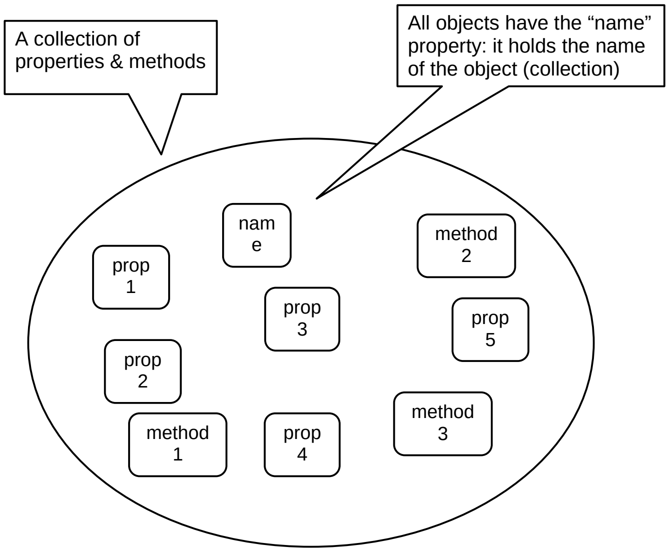
A function call appearing as *part of a statement.* Definitions of such functions include a 'return' statement

What Would this Statement Do?
factorial(factorial (3)) ;

This is termed as the *recursive* use of a function.

Methods:

Methods *are* functions
They are *unusual* in the sense that they are stored as properties of objects


A collection of properties & methods
All objects have the "name" property: it holds the name of the object (collection)
nam e
method 2
prop 1
prop 3
prop 5
prop 2
method 3
method 1
prop 4

Predefined, Top-Level or Built-In Functions:

Event handlers are not the only functions that come predefined with JavaScript. There are many others.
Practically, there is no difference between predefined functions and those that are defined by the programmer (termed as user-defined or custom functions)
There are many of them, but here we discuss only two: parseInt( ), parseFloat( )

Local Variables:

Declaring variables (using the var keyword) within a function, makes them *local*
They are available only within the function and hold no meaning outside of it.

Local –vs– Global:

Global variables can make the logic of a Web page difficult to understand
Global variables also make the reuse and maintenance of your code much more difficult

HEURISTIC:
If it's possible to define a variable as local,
do it!

Event Handlers:

Special-purpose functions that come predefined with JavaScript
They are *unusual* in the sense that they are mostly called from the HTML part of a Web page and not the <SCRIPT> … </SCRIPT> part

Today’s Goal:

Event Handlers

To become able to appreciate the concept of event handlers:
What are they?
What do they do?
How do we benefit from them?
To learn to write simple programs that use event handlers

32.1 What is Event Handling?

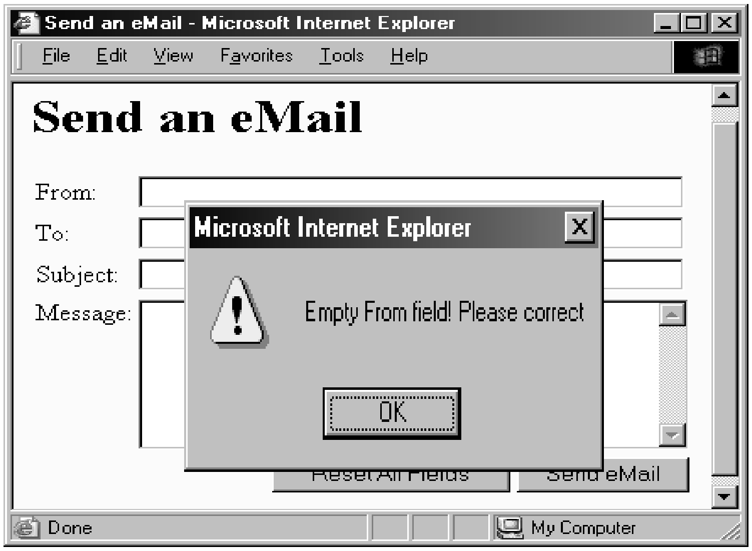
Capturing events and responding to them
The system sends events to the program and the program responds to them as they arrive
Events can include things a user does - like clicking the mouse - or things that the system itself does - like updating the clock. Today we will exclusively focus on user-events

Event Driven Programs:

Programs that can capture and respond to events are called ‘event-driven programs’
JavaScript was specifically designed for writing such programs
Almost all programs written in JavaScript are event-driven

JavaScript Handling of Events:

Events handlers are placed in the BODY part of a Web page as attributes in HTML tags
Events can be captured and responded to directly with JavaScript one-liners embedded in HTML tags in the BODY portion
Alternatively, events can be captured in the HTML code, and then directed to a JavaScript function for an appropriate response


Send an eMail - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Send an eMail
From:
To:
Subject:
Message:
Microsoft Internet Explorer
OK
Reset All Fields
Send eMail
Done
My Computer

Empty From field! Please correct

```
<INPUT
 type=“submit”
 name=“sendEmail”
 value=“Send eMail”
 onMouseOver=
  “if (document.sendEmail.sender.value.length < 1)
    window.alert(‘Empty From field! Please correct’)”
>
```

Additional JavaScript code for the *smart* ‘Send eMail’ button that does not allow itself to be clicked if the “From” text field is left blank
That was event handling through what we may call ‘in-line JavaScript’
That is, the event was captured and handled with a JavaScript one-liner that was embedded in the HTML tag

32.2 In-Line JavaScript Event Handling :

Event handlers are placed in the BODY portion of a Web page as attributes of HTML tags

The event handler attribute consists of 3 parts:

The identifier of the event handler

The equal sign

A string consisting of JavaScript statements enclosed in double or single quotes

Multiple JavaScript statements (separated by semicolons) can be placed in that string, but all have to fit in a single line; no newline characters are allowed in that string

Due to this limitation, sophisticated event handling is not possible with in-line event handling

Another - more sophisticated - way of accomplishing the same task:

JavaScript that goes between the <SCRIPT>, </SCRIPT> tags:

```
function checkForm() {
   if ( document.sendEmail.sender.value.length < 1)  {
      window.alert( “Empty From field! Please correct” );
   }
}
```

JavaScript included as an attribute of the “Send eMail” button:

```
onMouseOver=“checkForm( )”
```

Usage Guideline:

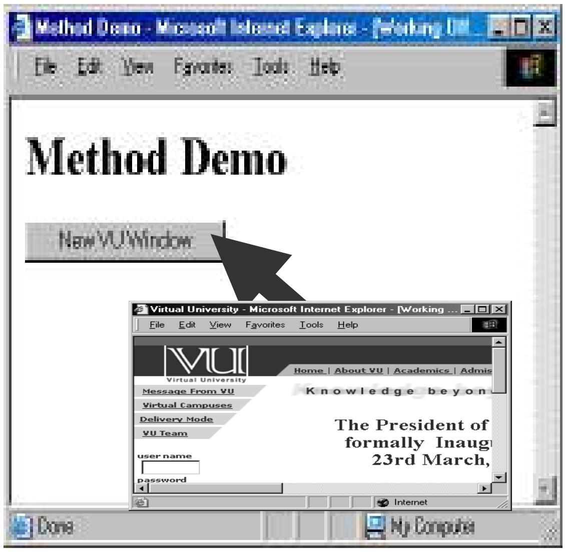
For very short scripts, “all code in the tag” works well

The “code in the HEAD portion” is the right choice for developing larger JavaScript scripts

It makes the code easier to read

It allows the reuse of a function for multiple event handlers

Another event-handling example; this time from lecture 18


Method Demo - Microsoft Internet Explorer - [Working Off...
File Edit View Favorites Tools Help
Method Demo
New VU Window
Virtual University - Microsoft Internet Explorer - [Working ...
File Edit View Favorites Tools Help
VU
Virtual University
Home | About VU | Academics | Admis
Message From VU
Virtual Campuses
Delivery Mode
VU Team
Knowledge beyon
The President of
formally Inaug
23rd March,
user name
password
Internet
Done
My Computer

JavaScript that goes between the <SCRIPT>, </SCRIPT> tags:

```
function vuWindow() {
  window.open(“http://www.vu.edu.pk/”) ;
}
```

JavaScript included as an attribute of the “New Window” button:

```
onClick=“vuWindow()”
```

A Few of My Favorite Event Handlers:

onClick
onDblClick
onMouseOver
onMouseDown
onFocus

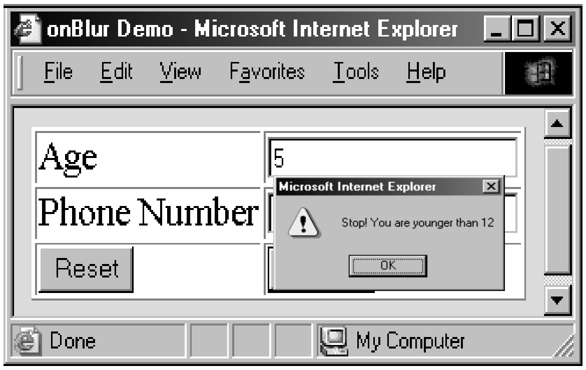
onBlur
onReset
onSubmit
onLoad
onUnload

There are many more: there is an expanded, but still incomplete list in your book.
Now let’s look at some of these error handlers in a bit more detail

onFocus & onBlur:

onFocus executes the specified JavaScript code when a window receives focus or when a form element receives input focus

onBlur executes the specified JavaScript code when a window loses focus or a form element loses focus


onBlur Demo - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Done
My Computer

Age 5

Phone Number

Reset

Microsoft Internet Explorer

Stop! You are younger than 12

OK

JavaScript that goes between the <SCRIPT>, </SCRIPT> tags:

```
function checkAge( ) {
   if( parseInt( document.form1.age.value ) < 12 )  {
      window.alert( "Stop! You are younger than 12" ) ;
   }
}
```

JavaScript included as an attribute of the INPUT tag:

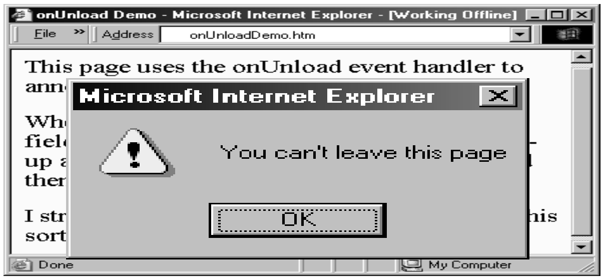
```
<INPUT type="text" name="age"
                        onBlur="checkAge( ) "
      >
<HTML><HEAD>
<TITLE>onBlur( ) Demo</TITLE>
<SCRIPT>
function checkAge() {
if( parseInt(document.form1.age.value) < 12) {
window.alert("Stop! You are younger than 12" ) ;
}
}
</SCRIPT></HEAD>
<BODY bgcolor="#66FFCC">
<FORM name="form1" method="post" action="">
<TABLE border="1">
<TR> <TD>Age</TD>
<TD><INPUT type="text" name="age" onBlur="checkAge()">
</TD></TR><TR> <TD>Phone Number</TD>
<TD><INPUT type="text" name="phNo"></TD>
</TR><TR> <TD><INPUT type="reset" value="Reset"></TD>
<TD><INPUT type="submit" value="Submit"></TD></TR>
</TABLE></FORM></BODY></HTML>
```

<u>onLoad & onUnload:</u>

onLoad executes the specified JavaScript code when a new document is loaded into a window

onUnload executes the specified JavaScript code when a user exits a document

What is the key difference between these 2 and the 4 event handlers (onMouseOver, onClick, onFocus, onBlur) that we have used so far?


onUnload Demo - Microsoft Internet Explorer - [Working Offline]
File
Address
onUnloadDemo.htm
Microsoft Internet Explorer
OK
Done
My Computer

This page uses the onUnload event handler to
ann
Wh
fiel
up a
ther
I str
his
sort

You can't leave this page

```
<HTML>
<HEAD>
<TITLE>onUnload Demo</TITLE>
<SCRIPT>
function annoyUser( ) {
        currentUrl = window.location ;
        window.alert( "You can't leave this page" ) ;
        window.location = currentUrl ;
}
</SCRIPT></HEAD>
<BODY onUnload="annoyUser( )">
This page uses the onUnload event handler …
</BODY></HTML>
```

```
<HTML>
<HEAD>
<TITLE>onUnload Demo</TITLE>
<SCRIPT>
function annoyUser( ) {
        currentUrl = window.location ;
        window.alert( "You can't leave this page" ) ;
        window.location = currentUrl ;
}
</SCRIPT></HEAD>
<BODY onUnload="annoyUser( )">
This page uses the onUnload event handler …
</BODY></HTML>
```

More Uses for onLoad/onUnload?

onLoad can be used to open multiple Windows when a particular document is opened

onUnload can be used to say “Thank you for the visit” when a user is leaving a Web page

At times, a user opens multiple inter-dependent windows of a Web site (e.g. VULMS). onUnload can be used to warn that all child Windows will become inoperable if the user closes the parent Window

A Note on Syntax:

Mixed-case capitalization of event handlers (e.g. onClick) is a convention (but not a requirement) for JavaScript event handlers defined in HTML code. Using ‘ONCLICK’ or ‘onclick’ as part of a an HTML tag is perfectly legal as well

At times, you may wish to use event handlers in JavaScript code enclosed in <SCRIPT>, </SCRIPT> tags

In those cases you have to strictly follow the JavaScript rule for all event handler identifiers: they must all be typed in small case, e.g. ‘onclick’ or ‘onmouseover’

A misleading statement from Lecture 18:

I stated:

JavaScript is case sensitive. Only the first of the following will result in the desired function – the rest will generate errors or other undesirable events:

onMouseClick – *OnMouseClick*
onmouseclick – *ONMOUSECLICK*

That statement is incorrect in two ways:

All four will work fine as part of HTML tags

Only the 'all small case' version will be interpreted as intended in JavaScript code

<u>During Today's Lecture …:</u>

We looked at the concept of event-driven programs and event handlers

What are they?

What do they do?

How do we benefit from them?

We wrote simple programs to demonstrate the capabilities of a few event handlers

<u>Next (the 12th) Web Dev Lecture:</u>
Mathematical Methods

We'll look at JavaScript's Math object

We will produce solutions for simple problems using various methods of the Math object

Lecture 33

Graphics & Animation

During the last lecture …(Internet Services):

We looked at several services provided by the Internet
FTP
Telnet
Web
eMail
Instant messaging
VoIP
We also found out about the addressing scheme used on the Internet

IP Address:

A unique identifier for a computer on a TCP/IP network
Format: four 8-bit numbers separated by periods. Each 8-bit number can be 0 to 255

Domain Names:

IP addresses are fine for computers, but difficult to recognize and remember for humans
A domain name is a meaningful, easy-to-remember 'label' for an IP address

DNS: Domain Name System:

DNS is the way that Internet domain names are located & translated into IP addresses

FTP:

Used to transfer files between computers on a TCP/IP network (e.g Internet)

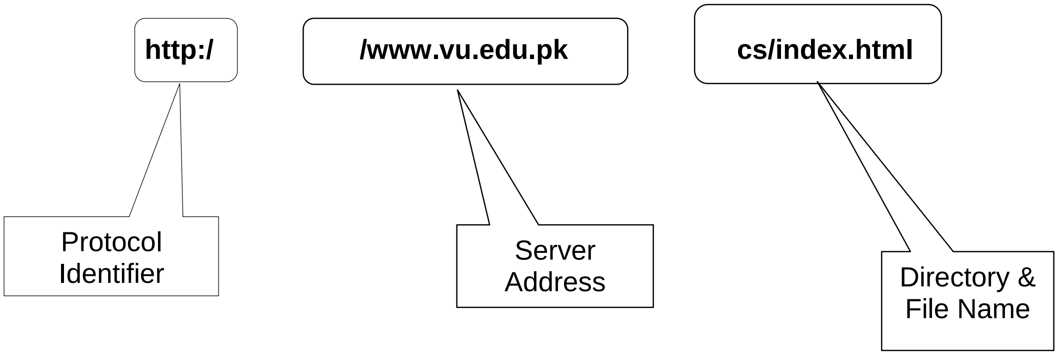
Telnet Protocol:

Using Telnet, a user can remotely log on to a computer (connected to the user's through a TCP/IP network, e.g. Internet) & have control over it like a local user, including control over running various programs

The Web :

The greatest, shared resource of information created by humankind
A user may access any item on the Web through a URL, e.g.

http://www.vu.edu.pk/cs/index.html



eMail:

Computer-to-computer messaging
Inexpensive, and quite quick, but not instant!

But first, the components:

eMail client
SMTP server
POP3 server

The Trouble with eMail:

Slow response times
No way of knowing if the person we are sending eMail to is there to read it

The process of having a conversation through eMail by exchanging several short messages is too cumbersome

Instant messaging (IM) solves these problems

Instant Messaging:

The IM services available on the Internet (e.g. ICQ, AIM, MSN Messenger, Yahoo! Messenger) allow us to maintain a list of people (contacts) that we interact with regularly

We can send an instant messages to any of the contacts in our list as long as that contact is online

Key Point:

Once the IM server provides the communication info to the user and his/her contact's IM client, the two are able to communicate with each other without the IM server's assistance

This server-less connection is termed as a P2P connection

VoIP: Voice over IP:

Voice delivered from one device to another using the Internet Protocol

Inexpensive, but of poor quality

Today's Goal:

Graphics & Animation

We will become familiar with the role that graphics and animations play in computing

We will look at how graphics & animation are displayed

We will look at a few popular formats used for storing graphics and animation

33.1 Computer Graphics:

Images created with the help of computers

2-D and 3-D (displayed on a 2-D screen but in such a way that they give an illusion of depth)

Used for scientific research, artistic expression, or for industrial applications

Graphics have made the computer interfaces more intuitive by removing the need to memorize commands

33.2 Displaying Images:

Most all computer displays consist of a grid of tiny pixels arranged in a regular grid of rows and columns

Images are displayed by assigning different colors to the pixels located in the desired portion of the computer display

Let's discuss the pixel a bit more …

Pixel:

The smallest image forming element on a computer display

The computer display is made up of a regular grid of these pixels

The computer has the capability of assigning any color to any of the individual pixels on the display

Let's now see how the computer displays a square

33.3 Pixel Colors :

The color of each pixel is generally represented in the form a triplet

In a popular scheme – the RGB scheme – each part of the triplet represents the intensity of one of out of three primary colors: red, green, blue

Often, the intensity of each color is represented with a byte, resulting in 256x256x256 (16+ million) unique color combinations

If this scheme is used to display an image that is equal to the size of an XGA (1024x768 pixels) display, the image will require 2.25MB of storage, which is just too much

A number of clever schemes have been invented to reduce the number of bytes that are required for storing graphics. 2 popular ones:

Color mapping

Dithering

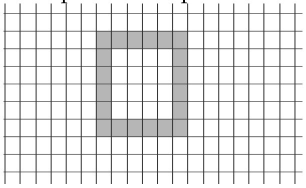
33.4 Color Mapping :

Instead of letting each pixel assume one out of 16 million possible colors, only a limited number of colors – called the platelet – are allowed

For example, the platelet may be restricted to 256 colors (requiring 1 byte/pixel instead of 3)

Each value, from 0 to 255, is mapped to a selected RGB color through a table, reducing the size of a 2.25MB graphic to 0.75MB

The quality of the displayed image will not suffer at all if the image only uses colors that are a part of the platelet



Color Platelet Example:

Color Platelet Code	Actual Color in RGB
1	255, 255, 000 (yellow)
2	255, 000, 000 (red)
3	000, 255, 255 (cyan)
4	255, 153, 051 (orange)
…	…
…	…
…	…

33.5 Dithering:

In this scheme, pixels of alternating colors are used to simulate a color that is not present in the platelet

For example, red and green pixels can be alternated to give the impression of bright yellow

The quality of the displayed image is poorer

33.6 Aliasing:

The computer screen consists of square-ish pixels arranged in a fixed grid

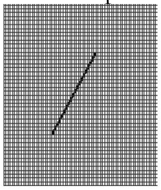
At times, when a diagonal line is drawn on this grid, it looks more like a staircase, instead of a straight line

This effect – called aliasing – can be managed by reducing the size of pixels

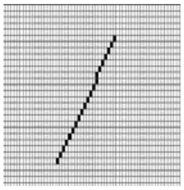
33.7 Anti-Aliasing:

Anti-aliasing is another technique used for managing the 'staircase' effect

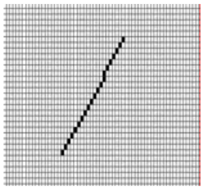
Let's say that we need to draw a white straight-line such that it overlaps 60% with one pixel, and 40% with another initially, and near the end, 58%, 41%, and 1%, respectively, with three pixels



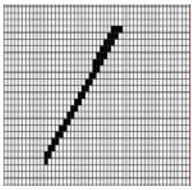
1



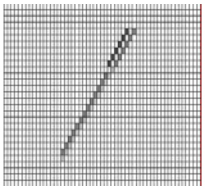
2



3



4



5

The staircase effect is caused because the proper drawing of the line requires a pixel that does not exist

There are three options in this case:

Assign the white color to the pixel corresponding to the largest overlap

Assign the white color to both pixels

Either of these will cause the staircase effect

The 3rd option is to color the pixel with 60% overlap to a 40% gray color & the other one to 60% gray

Result: A smoother - pleasing to the eye - look

33.8 Graphics File Formats:

The choice of the format generally depends upon the nature of the image. For example:

An image of natural scenery contains many irregular, non-gemetric shapes, therefore is stored in bit-map format

A CAD drawing consists of many geometric shapes like straight lines, arcs, etc. and therefore is stored in a vector format

A third situation arises when dealing with graphics that contain both regular and irregular shapes

33.9 Vector or Object-Oriented Graphics:

Treats everything that is drawn as an object

Objects retain their identity after they are drawn

These objects can later be easily moved, stretched, duplicated, deleted, etc

Are resolution independent

Relatively small file size

Examples: swf, svg, wmf, ps

33.10 Bit-Mapped or Raster Graphics:

Treats everything that is drawn as a bit-map

If an object is drawn on top of another, it is difficult to move just one of them while leaving the other untouched

Changing the resolution often requires considerable touch-up work

Relatively large file size

Examples: gif, jpg, bmp

33.11 File Formats Popular on the Web (1):

gif (Graphical Interchange Format)

Bit-map images compressed using the LZW algo.

The number of colors is first reduced to 256 and then consecutive pixels having the same color are encoded in a [color, numberOfPixels] format

Works well with computer-generated graphics (e.g. CAD, block diagrams, cartoons) but not with natural, realistic images

Loss-less for images having 256 colors or less

jpg (JPEG – Joint Photographic Experts Group)

Compressed, full-color and gray-scale bit-map images of natural, real-world scenes, where most every pixel differs in color from its neighbor

It does not work as well as gif with non-realistic images, such as cartoons or line drawings

Does not handle compression of B&W images

Lossy

swf (Shockwave Flash)

Designed for 2-D animations, but can also be used for storing static vector images as well

A special program (called a plug-in) is required to view swf files in a Web browser

svg (Structured Vector Graphics)

New format; may become more popular than swf

33.12 Image Processing:

A branch of computer science concerned with manipulating and enhancing computer graphics

Examples:
Converting 2-D satellite imagery into a 3-D model of a terrain
Restoring old, faded photographs into something closer to the original
Determining the amount of silting in Tarbela lake from a satellite image

33.13-D Graphics:

Flat images enhanced to impart the illusion of depth
We perceive the world and the objects in it in 3-D - breadth, width, depth - although the images formed on the retinas of our eyes are 2-D
The secret of 3-D perception: stereo vision
The two eyes are spaced a few cm apart
Result: The images formed on the two retinas are slightly different
The brain combines these two into a single 3-D image, enabling us to perceive depth

3-D Graphics: Applications:

Games
Medical images
3-D CAD

3-D Rendering:

The process of converting information about 3-D objects into a bit-map that can be displayed on a 2-D computer display
Computationally, very expensive!
Steps:
Draw the wire-frame (skeleton, made with thin lines)
Fill with colors, textures, patterns
Add lighting effects (reflections, shadows)

33.14 Animation:

Graphics in motion, e.g. cartoons
Illusion of motion is created by showing the viewer a sequence of still images, rapidly
Drawing those images - each slightly different from the previous one - used to be quite tedious work
Computers have helped in cutting down some of the tediousness
See next slides

Computer Animation: Examples

Games
Cartoons, movies
Visualization of processes, e.g the IM process
Displaying the results of scientific experiments, e.g. nuclear fusion

Tweening:

Creating a reasonable illusion of motion requires the drawing of 14-30 images per second of animation – very tedious!
In practice, only 4-5 images (called key images) instead of 14-30 are drawn, and then the computer is asked to create the remaining in-between images
This process of creating these in-between images from key images is called in-betweening (or tweening for short)
The simplest algorithm for tweening calculates the position of a particular segment of an image by calculating the average of the positions of that same image segment belonging to adjacent key images

The Future of Graphics & Animation:

New graphic-file storage formats will appear with better compression efficiencies
3-D animation will become more popular as computers become faster and algorithms become smarter
More realistic games; better realism in movies – may, one day, make the human actors extinct

Today's Goal:Graphics & Animation

We became familiar with the role that graphics and animations play in computing

We discussed how graphics & animation are displayed
We also looked at several formats used for storing graphics and animation

<u>Next Lecture:</u>(Intelligent Systems)

To become familiar with the distinguishing features of intelligent systems with respect to other software systems
To become able to appreciate the role of intelligent systems in scientific, business and consumer applications
To look at several techniques for designing intelligent systems

Lecture 34

Intelligent Systems

During the last lecture …:(Graphics & Animation)

We became familiar with the role that graphics and animations play in computing
We discussed how graphics & animation are displayed
We also looked at several formats used for storing graphics and animation

Computer Graphics:

Images created with the help of computers
2-D and 3-D (displayed on a 2-D screen but in such a way that they give an illusion of depth)
Used for scientific research, artistic expression, or for industrial applications
Graphics have made the computer interfaces more intuitive by removing the need to memorize commands

Displaying Images:

Most all computer displays consist of a grid of tiny pixels arranged in a regular grid of rows and columns
Images are displayed by assigning different colors to the pixels located in the desired portion of the computer display
Let's discuss the pixel a bit more …

Pixel:

The smallest image forming element on a computer display
The computer display is made up of a regular grid of these pixels
The computer has the capability of assigning any color to any of the individual pixels on the display
Let's now see how the computer displays a square
The color of each pixel is generally represented in the form a triplet
In a popular scheme – the RGB scheme – each part of the triplet represents the intensity of one of out of three primary colors: red, green, blue
Often, the intensity of each color is represented with a byte, resulting in 256x256x256 (16+ million) unique color combinations
Instead of letting each pixel assume one out of 16 million possible colors, only a limited number of colors – called the platelet – are allowed
For example, the platelet may be restricted to 256 colors (requiring 1 byte/pixel instead of 3)

Dithering:

In this scheme, pixels of alternating colors are used to simulate a color that is not present in the platelet
For example, red and green pixels can be alternated to give the impression of bright yellow
The quality of the displayed image is poorer

Aliasing:

The computer screen consists of square-ish pixels arranged in a fixed grid
At times, when a diagonal line is drawn on this grid, it looks more like a staircase, instead of a straight line
This effect – called aliasing – can be managed by reducing the size of pixels
Anti-aliasing is another technique used for managing the 'staircase' effect
Let's say that we need to draw a white straight-line such that it overlaps 60% with one pixel, and 40% with another initially, and near the end, 58%, 41%, and 1%, respectively, with three pixels

Vector or Object-Oriented Graphics:

Treats everything that is drawn as an object
Objects retain their identity after they are drawn
These objects can later be easily moved, stretched, duplicated, deleted, etc
Are resolution independent
Relatively small file size

Examples: swf, svg, wmf, ps

Bit-Mapped or Raster Graphics:

Treats everything that is drawn as a bit-map

If an object is drawn on top of another, it is difficult to move just one of them while leaving the other untouched

Changing the resolution often requires considerable touch-up work

Relatively large file size

Examples: gif, jpg, bmp

3-D Graphics:

Flat images enhanced to impart the illusion of depth

We perceive the world and the objects in it in 3-D - breadth, width, depth - although the images formed on the retinas of our eyes are 2-D

The secret of 3-D perception: stereo vision

3-D Rendering:

The process of converting information about 3-D objects into a bit-map that can be displayed on a 2-D computer display

Computationally, very expensive!

Steps:

Draw the wire-frame (skeleton, made with thin lines)

Fill with colors, textures, patterns

Add lighting effects (reflections, shadows)

Animation:

Graphics in motion, e.g. cartoons

Illusion of motion is created by showing the viewer a sequence of still images, rapidly

Drawing those images - each slightly different from the previous one - used to be quite tedious work

Computers have helped in cutting down some of the tediousness

This process of creating these in-between images from key images is called in-betweening (or tweening for short)

The simplest algorithm for tweening calculates the position of a particular segment of an image by calculating the average of the positions of that same image segment belonging to adjacent key images

Today's Goals:(Intelligent Systems)

To become familiar with the distinguishing features of intelligent systems with respect to other software systems

To become able to appreciate the role of intelligent systems in scientific, business and consumer applications

To look at several techniques for designing intelligent systems

34.1 (Artificial) Intelligent Systems:

SW programs or SW/HW systems designed to perform *complex* tasks employing strategies that mimic some aspect of human thought

One can debate endlessly about whether a certain system is intelligent or not

But to my mind, the key criterion is evolution: it is intelligent if it can learn (even if only a limited sense) and get better with time

Not a Suitable Hammer for All Nails!

if the nature of computations required in a task is not well understood
    or there are too many exceptions to the rules
    or known algorithms are too complex or inefficient
then AI has the potential of offering an acceptable solution

Selected Applications:

Games: Chess, SimCity

Image recognition

Medical diagnosis

Robots

Business intelligence

Sub-Categories of AI:

Expert systems

Systems that, in some limited sense, can replace an expert

Robotics

Natural language processing

Teaching computers to understand human language, spoken as well as written

Computer vision

Selected Techniques:

Artificial neural networks

Genetic algorithms

Rule-based systems

Fuzzy logic

Many times, any one of them can solve the problem at hand, but at others, only the right one will do.

Therefore, it is important to have some appreciation of them all .

Neural Networks:

Original inspiration was the human brain; emphasis now on usefulness as a computational tool

Many useful NN paradigms, but scope of today's discussion limited to the feed-forward network, the most popular paradigm

Feed-forward Network:

It is a layered structure consisting of a number of homogeneous and simple (but nonlinear) processing elements

All processing is local to a processing element and is asynchronous

During training the FN is forced to adjust its parameters so that its response to input data becomes closer to the desired response

Based on Darwin's evolutionary principle of 'survival of the fittest'

GAs require the ability to recognize a good solution, but not how to get to that solution.

Genetic Algorithms (2):

The procedure:

An initial set of random solutions is ranked in terms of ability to solve the problem at hand

The best solutions are then crossbred and mutated to form a new set

The ranking and formation of new solutions is continued until a good enough solution is found or …

Rulebased Systems (1):

Based on the principles of the logical reasoning ability of humans

Components of an RBS:

Rulebase

Working memory

Rule interpreter

The design process:

An RBS engineer interviews the expert to acquire the comprehensive set of heuristics that covers the situations that may occur in a given domain

This set is then encoded in the form of IF-THEN structures to form the required RBS

34.2 Fuzzy Logic:

Based on the principles of the approximate reasoning faculty that humans use when faced with linguistic ambiguity

The inputs and outputs of a fuzzy system are precise, only the reasoning is approximate

Parts of the knowledgebase of a fuzzy system:

Fuzzy rules

Fuzzy sets

The output of a fuzzy system is computed by using:

The MIN-MAX technique for combining fuzzy rules

The centroid method for defuzzification
Now we know about a few techniques
Let's now consider the situation when we are given a particular problem and asked to find an AI solution to that problem.
How do we determine the right technique for that particular problem?

Selection of an Appropriate AI Technique:

A given problem can be solved in several ways
Even if 2 techniques produce solutions of a similar quality, matching the right technique to a problem can save on time & resources
Characteristics of an optimal technique:
The solution contains all of the required information
The solution meets all other necessary criteria
The solution uses all of the available (useful) knowledge
How do we determine the suitability of a particular AI technique for a given task. We look at the task's requirements and then see which technique fulfils those requirements more completely – the one which does, is the one we use!
Here are a few aspects of the task and the techniques that we need to be aware off …

- Accuracy
- Explainability
- Response speed
- Scalability
- Compactness
- Flexibility
- Embedability
- Ease of use
- Learning curve
- Tolerance for complexity
- Tolerance for noise in data
- Tolerance for sparse data
- Independence from experts
- Development speed
- Computing ease

Credit Card Issuance:

Challenge. Increase the acceptance rate of card applicants who will turn out to be good credit risks
Inputs. Applicant's personal and financial profiles
Output. Estimated yearly loss if application is accepted
Expert knowledge. Some rules of thumb are available
Data. Profiles & loss data available for 1+ million applicants
Suitable technique?

Determination of the Optimal Drug Dosage:

Challenge. Warn the physician if she prescribes a dosage which is either too high or too low
Inputs. Patient's medical record. Pharmaceutical drug dosage instructions
Output. Warning along with reasons for the warning
Data. Medical records of thousands of patients. Drug dosage instructions on dozens of medicines
Suitable technique?

Prediction of Airline Cabin Crew's Preferences:

Challenge. Predict the future base/status preferences of the cabin crew of an airline. The predicted preferences will be used by the airline for forecasting its staffing and training requirements
Inputs. Crew's personal profiles. Preference history. Other data.
Output. Predicted preference card for a date one year in the future
Expert knowledge. Some rules of thumb are available
Data. Available for the last four years for 8000 crew members
Suitable technique?

The Right Technique:

Selection of the right AI technique requires intimate knowledge about the problem as well as the techniques under consideration

Real problems may require a combination of techniques (AI and/or nonAI) for an optimal solution

34.3 Robotics:

Automatic machines that perform various tasks that were previously done by humans

Example:

Pilot-less combat airplanes

Land-mine hunters

Autonomous vacuum-cleaners

Components: Body structure, actuators, power-source, sensors, controller (the AI-based part)

Autonomous Web Agents:

Also known as mobile agents, softbots

Computer program that performs various actions continuously, autonomously on behalf of their principal!

Key component of the Semantic Web of tomorrow

Multi-agent communities are being developed in which agents meet and represent the interests of their principals in negotiations or collaborations. **Example:**

Agents of a patient and a doctor get together to negotiate and select a mutually agreeable time, cost

Decision Support Systems:

Interactive software designed to improve the decision-making capability of their users

Utilize historical data, models to solve problems

The do not make decisions - just assist in the process

They provide decision-makers with information via easy to manage reports, what-if scenarios, and graphics

The Future?

Get ready to see robots playing a bigger role in our daily lives

Robots will gradually move out of the industrial world and into our daily life, similar to the way computers did in the 80's

Decision support systems will become a bigger part of the professional life of doctors, managers, marketers, etc

Autonomous land, air, sea vehicles controlled from 1000's of miles away from the war zone

Today's Summary:**Intelligent Systems**

We looked at the distinguishing features of intelligent systems w.r.t. other software systems

We looked at the role of intelligent systems in scientific, business, consumer and other applications

We discussed several techniques for designing intelligent systems

Next Lecture:**(Data Management)**

To become familiar with the issues and problems related to data-intensive computing

To become able to appreciate data management concepts and their evolution over the years

Lecture 35

Mathematical Methods (Web Development Lecture 12)

During the last lecture we discussed Event handling:

We looked at the concept of event-driven programs and event handlers

What are they?

What do they do?

How do we benefit from them?

We wrote simple programs to demonstrate the capabilities of a few event handlers

What is Event Handling?

Capturing events and responding to them

The system sends events to the program and the program responds to them as they arrive

Events can include things a user does - like clicking the mouse - or things that the system itself does - like updating the clock. Today we will exclusively focus on user-events.

Event Driven Programs:

Programs that can capture and respond to events are called ‘event-driven programs’

JavaScript was specifically designed for writing such programs

JavaScript’s Handling of Events:

Events handlers are placed in the BODY part of a Web page as attributes in HTML tags

Events can be captured and responded to directly with JavaScript one-liners embedded in HTML tags in the BODY portion

Alternatively, events can be captured in the HTML code, and then directed to a JavaScript function for an appropriate response

In-Line JavaScript Event Handling:

Event handlers are placed in the BODY portion of a Web page as attributes of HTML tags

The event handler attribute consists of 3 parts:

The identifier of the event handler

The equal sign

A string consisting of JavaScript statements enclosed in double or single quotes

Multiple JavaScript statements (separated by semicolons) can be placed in that string, but all have to fit in a single line; no newline characters are allowed in that string

Due to this limitation, sophisticated event handling is not possible with in-line event handling

Usage Guideline:

For very short scripts, “all code in the tag” works well

The “code in the HEAD portion” is the right choice for developing larger JavaScript scripts

It makes the code easier to read

It allows the reuse of a function for multiple event handlers

onFocus & onBlur:

onFocus executes the specified JavaScript code when a window receives focus or when a form element receives input focus

onBlur executes the specified JavaScript code when a window loses focus or a form element loses focus

onLoad & onUnload:

onLoad executes the specified JavaScript code when a new document is loaded into a window

onUnload executes the specified JavaScript code when a user exits a document.

Mixed-case capitalization of event handlers (e.g. onClick) is a convention (but not a requirement) for JavaScript event handlers defined in HTML code

At times, you may wish to use event handlers in JavaScript code enclosed in <SCRIPT>, </SCRIPT> tags

A Note on Syntax:

In those cases you have to strictly follow the JavaScript rule for all event handler identifiers: they must all be typed in small case, e.g. ‘onclick’ or ‘onmouseover’

Today’s Goal:(Mathematical Methods)

We will look at JavaScript’s Math object

We will look at solutions for simple problems using various methods of the Math object

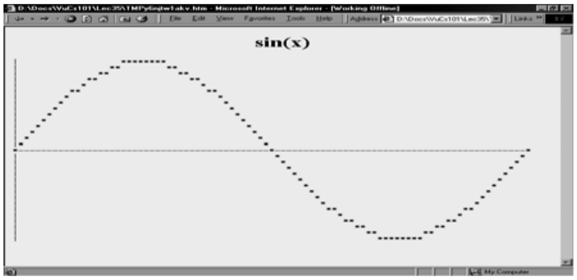
35.1 Problems & Solutions:

JavaScript doesn’t support drawing of graphics

However, crude graphics can be put together with the help of various text characters or tables

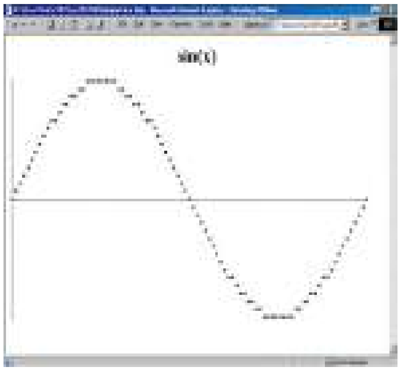
One cannot write a character at a random location on the screen using JavaScript

Instead, the graph has to be drawn from top to bottom, one row at a time – just like when regular text is written to a document



```
<HTML>
<HEAD>
<TITLE>Sine Function Plot</TITLE>
<SCRIPT>
function plotSine( ) {
          …
     }
     …
</SCRIPT>
</HEAD>
<BODY onLoad="plotSine( )">
</BODY>
</HTML>

function plotSine( ) {
     var ht, wd, rowN ; // rowN is the row number
     ht = 15 ; // height of the half cycle
     wd = 90 ; // width of the plot

     document.write(
          "<H1 align = 'center'>sin(x)</H1>" ) ;

     for( rowN = ht; rowN >= -ht; rowN = rowN - 1 ) {
          plotRow( rowN, ht, wd ) ;

}
}
function writeRow( row, wd ) {
     var rowE ;
     document.write(
     "<FONT face = 'courier' size = '-2'>" ) ;
     for( rowE = 0; rowE <= wd; rowE = rowE + 1 ) {
        document.write ( row[ rowE ] ) ;
        }
        document.write( "<BR></FONT>" ) ;
}

function plotRow( rowN, ht, wd ) {
        var theta, rowE ; // rowE is the row element
        var row = new Array( wd ) ;
        for ( rowE=0; rowE <= wd; rowE = rowE + 1 ) {
        theta = 2 * Math.PI * rowE / wd ;
        if( rowN == Math.round(ht * Math.sin( theta )))
        row[ rowE ] = "*" ;
        else
        row[ rowE ] = "&nbsp;" ;
        }
        writeRow ( row, wd ) ;
}

function plotRow( rowN, ht, wd ) {
        var theta, rowE ;
        var row = new Array( wd ) ;
        for ( rowE=0; rowE <= wd; rowE = rowE + 1 ) {
        theta = 2 * Math.PI * rowE / wd ;
        if( rowN == Math.round(ht * Math.sin( theta )))
            row[ rowE ] = "*" ;
        else

 row[ rowE ] = "&nbsp;" ;
        }
        writeRow ( row, wd ) ;
}
```

```
if( rowE == 0 )
        row[ rowE ] = "|" ;
else
        if( rowN == 0 )
        row[ rowE ] = "-" ;
        else
```



That is a sine wave.
How about a cosine?
Or a tangent?
Or, even, the natural logarithm?

<u>Today We Have Seen 3 New Elements:</u>

Math.PI
A property that gave us the value of Pi
Math.round( )
A method that rounded a number to its nearest integer
Math.sin( )

A method that gave us the sine of an angle
All 3 belong to JavaScript's Math object

35.2 Mathematical Functions in JavaScript:

In addition to the simple arithmetic operations (e.g. +, *, etc.) JavaScript supports several advanced mathematical operations as well
Notationaly, these functions are accessed by referring to various methods of the Math object
Moreover, this object also contains several useful mathematical constants as its properties
This object has no use, but of a placeholder

Properties:

Math.PI
Math.E
Math.LN2
Math.LN10
Math.LOG2E
Math.LOG10E
Math.SQRT2
Math.SQRT1_2

Note the CAPITAL lettering of all properties

Methods:

sin(r)
cos(r)
tan(r)
asin(x)
acos(x)
atan(x)
atan2(x, y)

sqrt(x)
pow(x, y)

exp(x)
log(x)

round(x)
floor(x)
ceil(x)

abs(x)

max(x, y)
max(x, y)

sin(r), cos(r), tan(r):

Standard trigonometric functions
Returns the sine, cosine or tangent of 'r', where 'r' is specified in radians

EXAMPLE

```
document.write( Math.cos( Math.PI / 4 ) )
```

```
0.707106781186547
```

asin(x), acos(x), atan(x):

Standard inverse-trigonometric functions
Returns the arcsine, arccosine or arctangent of 'r' in radians

EXAMPLE

```
document.write( Math.asin( 1 ) )
```

```
1.5707963267948965
```

sqrt(x)	pow(x, y)
Returns the square root of x	Returns x raised to the power y
0.5 → 0.7071	2, 32 → 4294967296

exp(x)	log(x)
Returns Math.E raised to the power x	Returns the the natural logarithm of x
1 → 2.718281	Math.E → 1

round(x)	floor(x)	ceil(x)
Returns integer nearest to x	Returns largest integer that is less than or equal to x	Returns smallest integer that is greater than or equal to x
1.1 → 1 12.5 → 13 -13.9 → -14	1.1 → 1 12.5 → 12 -13.9 → -14	1.1 → 2 12.5 → 13 -13.9 → -13

abs(x)
Returns the absolute value of x
1.1 → 1.1 -12.5 → 12.5 0 → 0

min(x, y)	max(x, y)
Returns the smaller of x and y	Returns the larger of x and y
2, 4 → 2 -12, -5 → -12	2, 4 → 4 -12, -5 → -5

random( ):

Returns a randomly-selected, floating-point number between 0 and 1

EXAMPLE

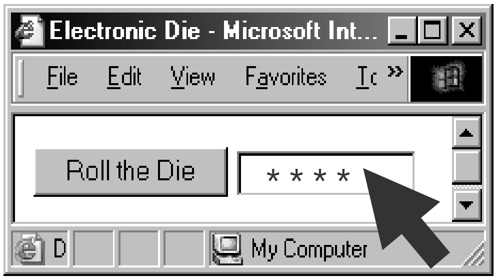
```
document.write( Math.random( ) )
```

```
0.9601111965589273
```

random( ):

Example

Design a Web page that displays the result of the rolling of a 6-sided die on user command


Electronic Die - Microsoft Int...
File Edit View Favorites Tc
Roll the Die
* * * *
D
My Computer

```
<HTML>
<HEAD>
<TITLE>Electronic Die</TITLE>
<SCRIPT>
function rollDie( ) { … }
</SCRIPT>
</HEAD>
<BODY>
<FORM … > … </FORM>
</BODY>
</HTML>
```

```
<FORM name="form1" method="post" action="">
<INPUT type="submit" name="Submit"
value="Roll Die" onMouseOver="rollDie( )">
<INPUT type="text" name="die" size="12">
</FORM>
```

```
function rollDie( ) {
        var dieN, dieDots, dots ;
        dieDots = "* " ;
        dieN = Math.round( 6 * Math.random( ) ) ;
        for( dots = 2; dots <= dieN; dots = dots + 1 ) {
                dieDots = dieDots + "* " ;
        }
        document.form1.die.value = dieDots ;
}
```

Asterisk

During Today's Lecture …:

We looked at the properties and methods of JavaScript's Math object

We produced solutions for simple problems using several methods of the Math object

Next (the 13th) Web Dev Lecture:

String Manipulation

To become familiar with a few methods used for manipulating strings

To become able to solve simple problems involving strings

Lecture 36

Data Management

During the last lecture …

(Intelligent Systems)

We looked at the distinguishing features of intelligent systems w.r.t. other software systems

We looked at the role of intelligent systems in scientific, business, consumer and other applications

We discussed several techniques for designing intelligent systems

(Artificial) Intelligent Systems:

SW programs or SW/HW systems designed to perform complex tasks employing strategies that mimic some aspect of human thought

Not a Suitable Hammer for All Nails!

if the nature of computations required in a task is not well understood
or there are too many exceptions to the rules
or known algorithms are too complex or inefficient
then AI has the potential of offering an acceptable solution

Selected Applications:

Games: Chess, SimCity
Image recognition
Medical diagnosis
Robots
Business intelligence

Neural Networks:

Original inspiration was the human brain; emphasis now on usefulness as a computational tool.

Genetic Algorithms (1):

Based on Darwin's evolutionary principle of 'survival of the fittest'

GAs require the ability to recognize a good solution, but not how to get to that solution

Rulebased Systems (1):

Based on the principles of the logical reasoning ability of humans.

Fuzzy Logic (1):

Based on the principles of the approximate reasoning faculty that humans use when faced with linguistic ambiguity

The Right Technique:

Selection of the right AI technique requires intimate knowledge about the problem as well as the techniques under consideration

Real problems may require a combination of techniques (AI and/or nonAI) for an optimal solution

Three exciting areas of AI applications Robotics:

Automatic machines that perform various tasks that were previously done by humans

Autonomous Web Agents (1):

Computer program that performs various actions continuously, autonomously on behalf of their principal!

Decision Support Systems:

Interactive software designed to improve the decision-making capability of their users

The do not make decisions - just assist in the process

Today's Goals:(Data Management)

First of a two-lecture sequence

Today we will become familiar with the issues and problems related to data-intensive computing

We will find out about flat-files, the simpleast databases

Next time, in our 4th lecture on productivity software, we will discuss relational databases and implement a simple relational database

Keeping track of a few dozen data items is straight forward

However, dealing with situations that involve significant number of data items, requires more attention to the data handling process

Dealing with millions - even billions - of inter-related data items requires even more careful thought

36.1 BholiBooks.com :

Consider the situation of a large, online bookstore

They have an inventory of millions of books, with new titles constantly arriving, and old ones being phased out on a regular basis

The price for a book is not a static feature; it varies every once in a while

Thousands of books are shipped each day, changing the inventory constantly

Some are returned, again changing the inventory situation constantly

The cost of each shipped order depends on:

Prices of individual books

Size of the order

Location of the customer

Mode of shipment

For each order, the customer's particulars –_ name, address, phone number, credit card number – are required

Generally, that data is not deleted after the completion of the transaction; instead, it is kept for future reference

All the transaction activity and the inventory changes result in:

Thousands of data items changing every day

Thousands of additional data items being added everyday

Keeping track & taking care (i.e. management) of all that constantly changing and expanding data is not a trivial task and requires disciplined attention and actions for ensuring the smooth & profitable operation of the bookstore

36.2 Issues in Data Management:

Data entry

Data updates

Data integrity

Data security

Data accessibility

Data Entry:

New titles are added every day

New customers are being added every day

Some of the above *may* require manual entry of new data into the computer systems

That new data needs to be added accurately

That can be achieved, for one, by user-interfaces that prevent the input of invalid data

Data Updates :

Old titles are deleted on a regular basis

Inventory changes every instant

Book prices change

Shipping costs change

Customers' personal data change

Various discount schemes are always commencing and concluding

All those actions require updates to existing data

Those changes need to be entered accurately

That can also be achieved by user-interfaces that prevent the input of invalid data

Data Security :

All the data that BholiBooks has in its computer systems is quite critical to its operation

The security of the customers' personal data is of utmost importance. Hackers are always looking for that type of data, especially for credit card numbers

Enough leaks of that type, and customers will stop doing business with BholiBooks

This problem can be managed by using appropriate security mechanisms that provide access to authorized persons/computers only

Security can also be improved through:

Encryption

Private or virtual-private networks

Firewalls

Intrusion detectors

Virus detectors

Data Integrity:

Integrity refers to maintaining the correctness and consistency of the data

Correctness: Free from errors

Consistency: No conflict among related data items

Integrity can be compromised in many ways:

Typing errors

Transmission errors

Hardware malfunctions

Program bugs

Viruses

Fire, flood, etc.

Ensuring Data Integrity:

Type Integrity is implemented by specifying the type of a data item:

Example: A credit card number consists of 12 digits. An update attempting to assign a value with more or fewer digits or one including a non-numeral should be rejected

Limit Integrity is enforced by limiting the values of data items to specified ranges to prevent illegal values

Example: Age of person should not be negative

Referential Integrity requires that an item referenced by the data for some other item must itself exist in the database

Example: If an airline reservation is requested for a particular flight, then the corresponding flight number must actually exist

Physical Integrity is ensured through hardware redundancy, backups, etc

Data Accessibility:

If the transaction and inventory data is placed in a disorganized fashion on a hard disk, it becomes very difficult to later search for a stored data item

What is required is that:

Data be stored in an organized manner

Additional info about the data be storedso that the data access times are minimized

What if two customers check on the aavailability of a certain title simultaneously?

On seeing its availability, they both order the title – for which, unfortunately, only a single copy is available

Same is the case when two airline customers try booking the only available seat

A solution to this *concurrency control* problem: Lock access to data while someone is using it

We can write our own SW that can take care of all the issues that we just discussed

OR

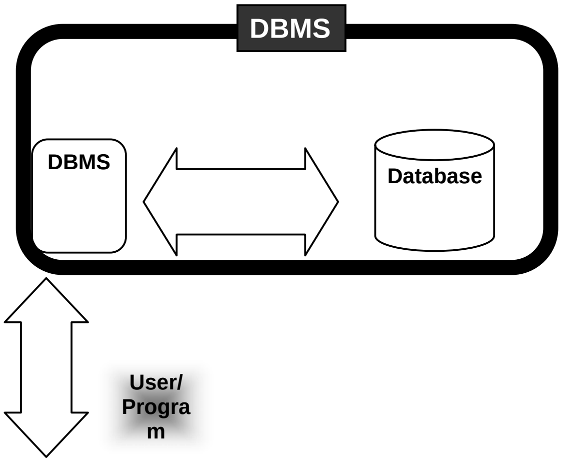
We can save ourselves lots of time, cost, and effort by buying ourselves a Database Management System (DBMS) that takes care of most, if not all, of the issues

36.3 DBMS :

DBMSes are popularly, but incorrectly, also known as 'Databases'

A DBMS is the SW system that operates a database, and is not the database itself

Some people even consider the database to be a component of the DBMS, and not an entity outside the DBMS



A DBMS takes care of the storage, retrieval, and management of large data sets on a database

It provides SW tools needed to organize & manipulate that data in a flexible manner

It includes facilities for:

Adding, deleting, and modifying data

Making queries about the stored data

Producing reports summarizing the required contents

Database:

A collection of data organized in such a fashion that the computer can quickly search for a desired data item

All data items in it are generally related to each other and share a single domain

They allow for easy manipulation of the data

They are designed for easy modification & reorganization of the information they contain

They generally consist of a collection of interrelated computer files

Example: VU Student Database:

Student's name

Student’s photograph

Father’s name

Phone number

Street address

eMail address

Courses being taken

Courses already taken & grades

Pre-VU educational record

Example: BholiBooks’ Customer DB:

Name, address, phone & fax, eMail

Credit card type, number, expiration date

Shipping preference

Books on order

All books that were ever shipped to the customer

Book preference

Example: BholiBooks’ Inventory DB:

Book title, author, publisher, binding, date of publication, price

Book summary, table of contents

Customers’, editors’, newspaper reviews

Number in stock

Number on order

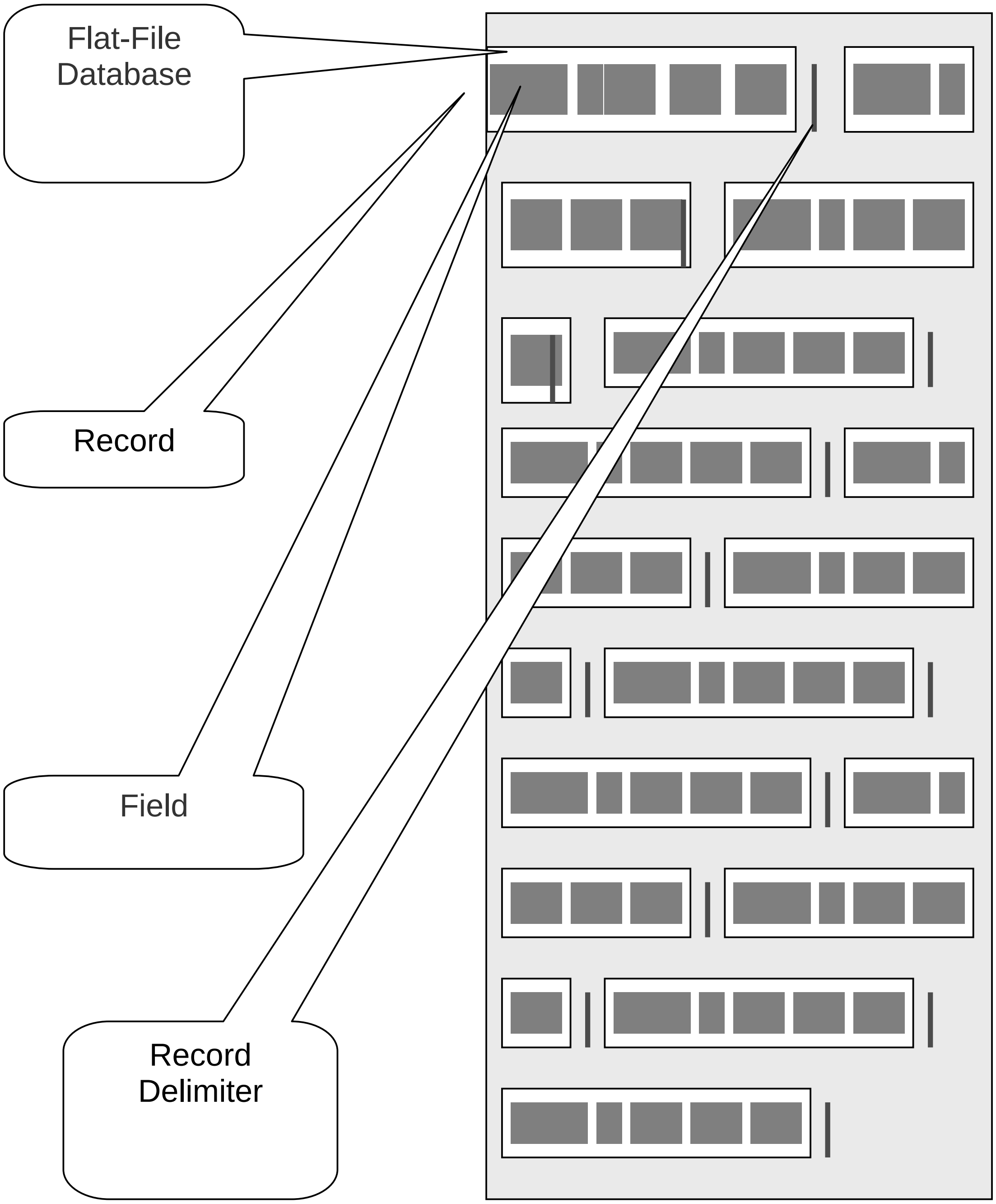
Special offer details

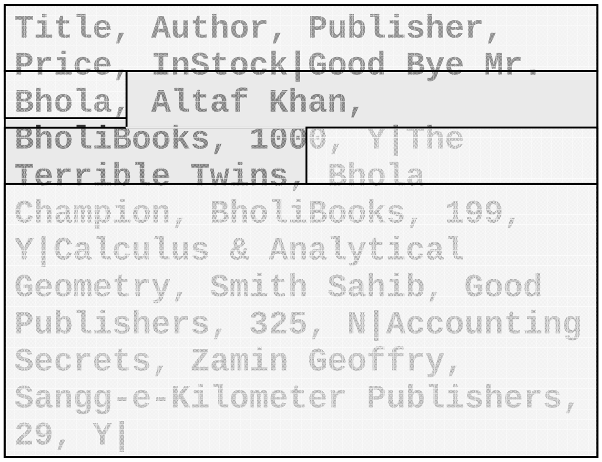
36.4 OS Independence:

DBMS stores data in a database, which is a collection of interrelated files
Storage of files on the computer is managed by the computer OS's file system
Intimate knowledge of the OS & its file system is required to provide rapid access to the data
The DBMS takes care of those details
It hides the actual storage details of data files from the user
It provides an OS-independent view of the data to the user, making data manipulation and management much more convenient
What can be stored in a database?
In the old days, databases were limited to numbers, Booleans, and text
These days, anything goes
As long as it is digital data, it can be stored:
Numbers, Booleans, text
Sounds
Images
Video

In the very, very old days …:

Even large amounts of data was stored in text files, known as flat-file databases
All related info was stored in a single long, tab- or comma-delimited text file
Each group of info – called a record - in that file was separated by a special character; vertical bar '|' was a popular option
Each record consisted of a group of fields, each field containing some distinct data item


Flat-File Database
Record
Field
Record Delimiter



```
Title, Author, Publisher,
Price, InStock|Good Bye Mr.
Bhola, Altaf Khan,
BholiBooks, 1000, Y|The
Terrible Twins, Bhola
Champion, BholiBooks, 199,
Y|Calculus & Analytical
Geometry, Smith Sahib, Good
Publishers, 325, N|Accounting
Secrets, Zamin Geoffry,
Sangg-e-Kilometer Publishers,
29, Y|
```

36.5 The Trouble with Flat-File Databases:

The text file format makes it hard to search for specific information or to create reports that include only certain fields from each record

Reason: One has to search sequentially through the entire file to gather desired info, such as 'all books by a certain author'

However, for small sets of data – say, consisting of several tens of kB – they can provide reasonable performance

Consider this tabular approach …
(same records, same fields, but in a different format)

Title	Author	Publisher	Price	InStock
Good Bye Mr. Bhola	Altaf Khan	BholiBooks	1000	Y
The Terrible Twins	Bhola Champion	BholiBooks	199	Y
Calculus & Analytical Geometry	Smith Sahib	Good Publishers	325	N
Accounting Secrets	Zamin Geoffry	Sung-e-Kilometer Publishers	29	Y

Tabular Storage: Features & Possibilities:

Similar items of data form a column

Fields placed in a particular row – same as a flat-file record – are strongly interrelated

One can sort the table w.r.t. any column

That makes searching – e.g., for all the books written by a certain author – straight forward

Tabular Storage: Features & Possibilities:

Similarly, searching for the 10 cheapest/most expensive books can be easily accomplished through a sort

Effort required for adding a new field to all the records of a flat-file is much greater than adding a new column to the table

CONCLUSION: Tabular storage is better than flat-file storage
We will continue on this theme next time

Today's Summary:(Data Management)

First of a two-lecture sequence

Today we became familiar with the issues and problems related to data-intensive computing

We also found out about flat-file and tabular storage

Next Lecture:(Database SW)

Next time, in our 4th lecture on productivity SW, we will continue our discussion on data management

We will find out about relational databases

We will also implement a simple relational database

Lecture 37
Database Software

Focus of the last Lecture was on Data Management

- First of a two-lecture sequence
- We became familiar with the issues and problems related to data-intensive computing
- We also found out about flat-file and tabular storage

Data Management

- Keeping track of a few dozen data items is straight forward
- However, dealing with situations that involve significant number of data items, requires more attention to the data handling process
- Dealing with millions - even billions - of inter-related data items requires even more careful thought

Issues in Data Management

Data Entry

- New titles are added every day
- New customers are being added every day
- That new data needs to be added accurately

Data Updates

- All those actions require updates to existing data
- Those changes need to be entered accurately

Data Security

- All the data that BholiBooks has in its computer systems is quite critical to its operation
- The security of the customers' personal data is of utmost importance. Hackers are always looking for that type of data, especially for credit card numbers
- This problem can be managed by using appropriate security mechanisms that provide access to authorized persons/computers only
- Security can also be improved through:

– Encryption
– Private or virtual-private networks
– Firewalls
– Intrusion detectors
– Virus detectors

Data Integrity

- Integrity refers to maintaining the correctness and consistency of the data

– Correctness: Free from errors
– Consistency: No conflict among related data items

- Integrity can be compromised in many ways:

– Typing errors
– Transmission errors
– Hardware malfunctions
– Program bugs
– Viruses
– Fire, flood, etc.

Ensuring Data Integrity

- *Type Integrity*
- *Limit Integrity*
- *Referential Integrity*
- *Physical Integrity*

Data Accessibility

- What is required is that:

– Data be stored in an organized manner
– Additional info about the data be stored so that the data access times are minimized

- A solution to this *concurrency control* problem: Lock access to data while someone is using it

DBMS

- A DBMS takes care of the storage, retrieval, and management of large data sets on a database
- It provides SW tools needed to organize & manipulate that data in a flexible manner
- It includes facilities for:

– Adding, deleting, and modifying data
– Making queries about the stored data
– Producing reports summarizing the required contents

Database

- A collection of data organized in such a fashion that the computer can quickly search for a desired data item

OS Independence

- It provides an OS-independent view of the data to the user, making data manipulation and management much more convenient

What can be stored in a database?

- As long as it is digital data, it can be stored:

– Numbers, Booleans, text
– Sounds
– Images
– Video

In the very, very old days …

- Even large amounts of data was stored in text files, known as *flat-file databases*
- All related info was stored in a single long, tab- or comma-delimited text file
- Each group of info – called a *record* - in that file was separated by a special character; vertical bar '|' was a popular option
- Each record consisted of a group of *fields*, each field containing some distinct data item

The Trouble with Flat-File Databases

- The text file format makes it hard to search for specific info or to create reports that include only certain fields from each record
- Reason: One has to search sequentially through the entire file to gather desired info, such as 'all books by a certain author'
- However, for small sets of data – say, consisting of several tens of kB – they can provide reasonable performance

Tabular Storage: Features & Possibilities

1.Similar items of data form a column
2.Fields placed in a particular row – same as a flat-file *record* – are strongly interrelated
3.One can sort the table w.r.t. any column
4.That makes searching – e.g., for all the books written by a certain author – straight forward
5.Similarly, searching for the 10 cheapest/most expensive books can be easily accomplished through a sort
6.Effort required for adding a new field to all the records of a flat-file is much greater than adding a new column to the table

CONCLUSION: Tabular storage is better than flat-file storage
We will continue on with tables' theme today

Today's Lecture:

Database SW

- In our 4th & final lecture on productivity software, we will continue our discussion from last week on data management
- We will find out about relational databases
- We will also implement a simple relational database

Let's continue on with the tabular approach. We stored data in a table last time, and liked it. Let's revisit that table and then put together another one

Table from the Last Lecture

Title	Author	Publisher	Pric	InStoc
Good Bye Mr. Bhola	Altaf Khan	BholiBooks	1000	Y
The Terrible Twins	Bhola Champion	BholiBooks	199	Y
Calculus & Analytical Geometry	Smith Sahib	Good Publishers	325	N
Accounting Secrets	Zamin Geoffry	Sung-e-Kilometer Publishers	29	Y

Another table …

Customer	Title	Shipment	Type
Aadil Ali	Good Bye Mr. Bhola	2002.12.26	Air
Aadil Ali	The Terrible Twins	2002.12.26	Air
Miftah Muslim	Calculus & Analytical Geometry	2002.12.25	Surface
Karen Kaur	Good Bye Mr. Bhola	2002.12.24	Air

This & the previous table are related

- They share a column, & are related through it
- A program can match info from a field in one table with info in a corresponding field of another table to *generate* a 3rd table that combines requested data from both tables
- That is, a program can use matching values in 2 tables to *relate* info in one to info in the other

Q: Who is BholiBooks' best customer?

- That is, who has spent the most money on the online bookstore?
- To answer that question, one can process the inventory and the shipment tables to generate a third table listing the customer names and the prices of the books that they have ordered

The *generated* table

Customer	**Price**
Aadil Ali	1000
Aadil Ali	199
Miftah Muslim	325
Karen Kaur	1000

Can you now process this table to find the answer to our question

Relational Databases

- Databases consisting of two or more related tables are called *relational databases*
- A typical relational database may have anywhere from 10 to over a thousand tables
- Each column of those tables can contain only a single type of data (contrast this with spreadsheet columns!)
- Table rows are called records; row elements are called fields
- A relational database stores all its data inside tables, and nowhere else
- All operations on data are done on those tables or those that are generated by table operations
- Tables, tables, and nothing but tables!

37.1 RDBMS

- Relational DBMS software
- Contains facilities for creating, populating, modifying, and querying relational databases
- Examples:

–Access
–FileMaker Pro
–SQL Server
–Oracle

– DB2
– Objectivity/DB
– MySQL
– Postgres

The Trouble with Relational DBs

- Much of current SW development is done using the object-oriented methodology
- When we want to store the object-oriented data into an RDBMS, it needs to be translated into a form suitable for RDBMS

The Trouble with Relational DBs

- Then when we need to read the data back from the RDBMS, the data needs to be translated back into an object-oriented form before use
- These two processing delays, the associated processing, and time spent in writing and maintaining the translation code are the key disadvantages of the current RDBMSes

Solution?

- Don't have time to discuss that, but try searching the Web on the following terms:
- Object-oriented databases

– Object-relational databases

Classification of DBMS w.r.t. Size

- Personal/Desktop/Single-user (MB-GB)

– Examples: Tech. papers' list; Methai shop inventory
– Typical DMBS: Access

- Server-based/Multi-user/Enterprise (GB-TB)

– Examples: HBL; Amazon.com
– Typical DMBS: Oracle, DB2

- Seriously-huge databases (TB-PB-XB)

– Examples: 2002 – BaBar experiment at Stanford (500TB); 2005 – LHC database at CERN (1XB)

– Typical DMBS: Objectivity/DB

37.2 Some Terminology

- *Primary Key* is a field that uniquely identifies each record stored in a table
- *Queries* are used to view, change, and analyze data. They can be used to:

– Combine data from different tables, efficiently

– Extract the exact data that is desired

- *Forms* can be used for entering, editing, or viewing data, one record at a time
- *Reports* are an effective, user-friendly way of presenting data. All DBMSes provide tools for producing custom reports.
- *Data normalization* is the process of efficiently organizing data in a database. There are two goals of the normalization process:

– Eliminate redundant data

– Storing only related data in a table

Before we do a demo, let me just mention my favorite database application:

Data Mining

- The process of analyzing large databases to identify patterns
- Example: Mining the sales records from a BholiBooks could identify interesting shopping patterns like "53% of customers who bought book A also bought book B". This pattern can be put to good use!
- Dat a mining often utilizes intelligent systems' techniques

Let's now demonstrate the use of a desktop RDBMS

- We will create a new relational database
- It will consist of two tables
- We will populate those tables
- We will generate a report after combining the data from the two tables

Access Tutorial

http://www.microsoft.com/education/DOWNLOADS/tutorials/classroom/office2k/acc2000.doc

Today's Lecture:

- In this final lecture on productivity software, we continued our discussion from last week on data management
- We found out about relational databases
- We also implemented a simple relational database

Next Lecture' Goals

(Cyber Crime)

- To know the different types of computer crimes that occur over cyber space
- To familiarize ourselves with with several methods that can be used to minimize the effect of these crimes
- To get familiar with a few policies and legislation designed to tackle cyber crime

Lecture 38
String Manipulations
(Web Development Lecture 13)

During the last lecture we discussed Mathematical Methods

- We looked at the properties and methods of JavaScript's Math object
- We produced solutions for simple problems using several methods of the Math object

Problems & Solutions

- JavaScript doesn't support drawing of graphics
- However, crude graphics can be put together with the help of various text characters or tables
- One cannot write a character at a random location on the screen using JavaScript
- Instead, the graph has to be drawn from top to bottom, one row at a time – just like when regular text is written to a document

Mathematical Functions in JavaScript

- In addition to the simple arithmetic operations (e.g. +, *, etc.) JavaScript supports several advanced mathematical operations as well
- Notationaly, these functions are accessed by referring to various methods of the Math object
- Moreover, this object also contains several useful mathematical constants as its properties
- This object has no use, but of a placeholder

Properties

Math.PI
Math.E
Math.LN2
Math.LN10
Math.LOG2E
Math.LOG10E
Math.SQRT2
Math.SQRT1_2

Note the CAPITAL lettering of all properties

Methods

sin(r)
cos(r)
tan(r)
asin(x)
acos(x)
atan(x)
atan2(x, y)

sqrt(x)
pow(x, y)

exp(x)
log(x)

round(x)
floor(x)
ceil(x)

abs(x)

max(x, y)
max(x, y

random( )

sin(r), cos(r), tan(r)

Standard trigonometric functions
Returns the sine, cosine or tangent of 'r',
where 'r' is specified in radians

EXAMPLE
document.write(Math.cos(Math.PI / 4))

0.7071067811865476

asin(x), acos(x), atan(x)

Standard inverse-trigonometric functions
Returns the arcsine, arccosine or arctangent of 'r'
in radians
EXAMPLE
document.write(Math.asin(1))

1.5707963267948965

sqrt(x)

Returns the square root of x

0.5 → 0.7071

pow(x, y)

Returns x raised to the power y

2, 32 → 4294967296

exp(x)

Returns Math.E raised to the power x

1 → 2.718281

log(x)

Returns the the natural logarithm of x

Math.E → 1

round(x)

Returns integer nearest to x

1.1 → 1
12.5 → 13
-13.9 → -14

floor(x)

Returns largest integer that is less than or equal to x

1.1 → 1
12.5 → 12
-13.9 → -14

ceil(x)

Returns smallest integer that is greater than or equal to x

1.1 → 2
12.5 → 13
-13.9 → -13

abs(x)

Returns the absolute value of x

1.1 → 1.1
-12.5 → 12.5
0 → 0

min(x, y)	max(x, y)
Returns the smaller of x and y	Returns the larger of x and y
2, 4 → 2 -12, -5 → -12	2, 4 → 4 -12, -5 → -5

random( )

Returns a randomly-selected, floating-point number between 0 and 1

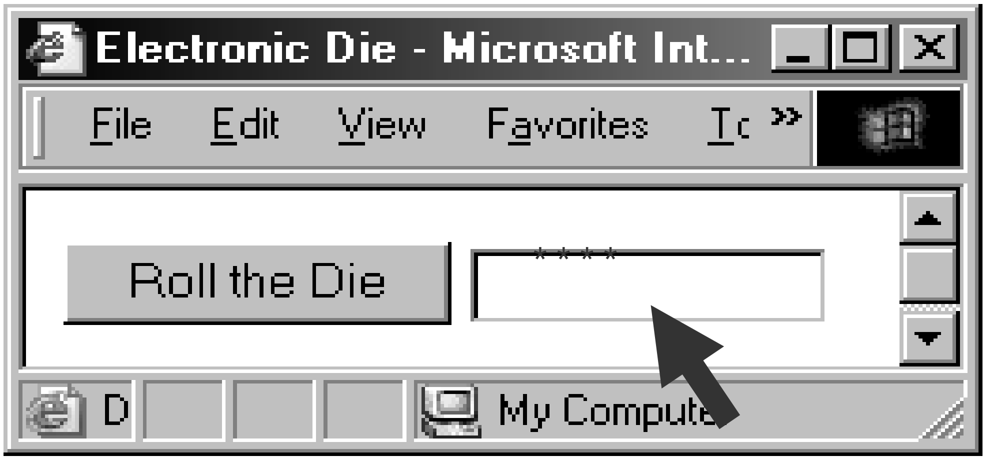
EXAMPLE

document.write(Math.random( ))

```
0.9601111965589273
```

random( ): Example

Design a Web page that displays the result of the rolling of a 6-sided die on user command



Today's Goal

(String Manipulation)

- To become familiar with methods used for manipulating strings
- To become able to solve simple problems involving strings

String Manipulation Examples

- Combine these words into a sentence i.e. take these strings and concatenate them into one
- Break this string into smaller ones

- Convert this string into upper case
- See if a particular character exists in a string
- Find the length of a string
- Convert this string into a number

38.1 String Manipulation in JavaScript

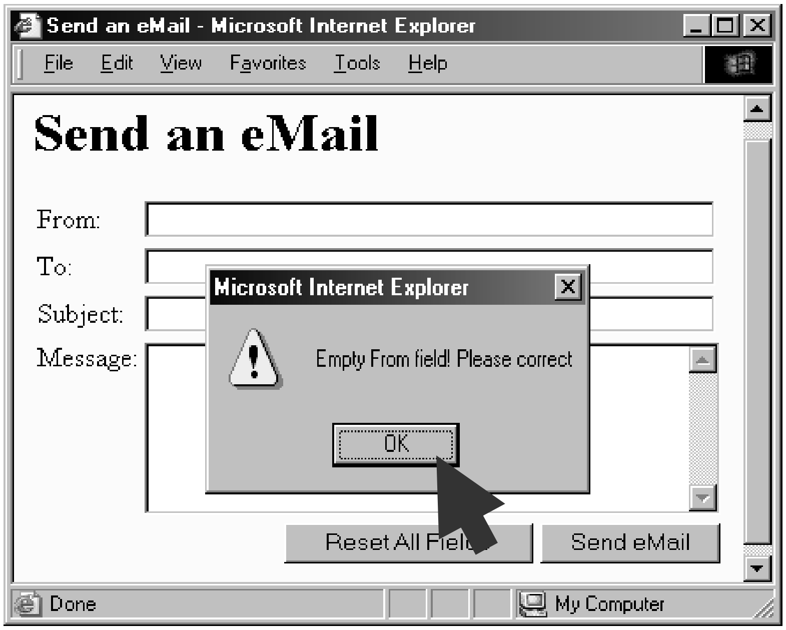
- **In addition to the concatenation operator (+) JavaScript supports several advanced string operations as well**
- **Notationaly, these functions are accessed by referring to various methods of the String object**
- **Moreover, this object also contains the 'length' property**

Example

```
name = "BHOLA" ;
document.write     ( "The length of the     string 'name' is ", name.length ) ;
```

```
The length of the string 'name' is 5
```

Let us now revisit an example that we first discussed in the 18th lecture Let us see how we put the 'length' property of a string to good use



Empty From field! Please correct

```
<HTML>
    <HEAD>
        <TITLE>Send an eMail</TITLE>
        <SCRIPT>
            function checkForm( ) { … }
        </SCRIPT>
    </HEAD>
    <BODY bgcolor="#FFFFCC">
```

```
        <TABLE><FORM …>…</FORM></TABLE>
    </BODY>
</HTML>

<TABLE>
    …
    <FORM …>
        <INPUT
            type="submit"
            name="sendEmail"
            value="Send eMail"
            onMouseOver="checkForm( )"
        >
        …
    </FORM>
</TABLE>

function checkForm( ) {
  if( document.sendEmail.sender.value.length < 1 ) {
        window.alert(
            "Empty From field! Please correct" ) ;
    }
}
```

This is a string

Other Uses of the 'length' Property

- To restrict the length of login name or password to specified bounds, i.e. no less than 4 and no more than 8 characters
- ???

String Methods

FORMAT

string.methodName( )

EXAMPLE:

```
    name = "Bhola" ;
    document.write( name.toUpperCase( ) ) ;
    document.write( name.bold( ) ) ;
```

BHOLA**Bhola**

Two Types of String Methods

1.HTML Shortcuts

2.All Others

String Methods: HTML Shortcuts

big( )
small( )
fontsize(*n*)

bold( )
italics( )
strike( )

link(URL)

sub( )
sup( )

fixed( )
fontcolor(*color*)

big( ), small( ), fontsize(*n*)

```
person = "Bhola" ;
document.write( person ) ;
document.write( person.big( ) ) ;
document.write( person.small( ) ) ;
document.write( person.fontsize( 1 ) ) ;
document.write( person.fontsize( 7 ) ) ;
```

```
BholaBholaBholaBholaBhola
```

sub( ), sup( )

```
person = "Bhola" ;
document.write( name ) ;
document.write( name.sub( ) ) ;
document.write( name ) ; document.write( name.sup( ) ) ;
```

```
BholaBholaBholaBholaBhola
```

bold( ), italics( ), strike( )

```
name = "Bhola" ;
document.write( name ) ;
document.write( name.bold( ) ) ;
document.write( name.italics( ) ) ;
document.write( name.strike( 1 ) ) ;
```

```
BholaBholaBholaBholaBhola
```

fixed( ), fontcolor(*color*)

```
person = "Bhola" ;
document.write( person ) ;
document.write( person.fixed( ) ) ;
document.write( person.fontcolor( “blue” ) ) ;
document.write( person.fontcolor( “orange” ) ) ;
```

```
BholaBholaBholaBholaBhola
```

link(*URL*)

```
hotel = "Bhola Continental" ;
document.write( hotel ) ;
document.write( hotel.link(
      “http://www.bholacontinental.com” ) ) ;
```

```
BholaBholaBholaBholaBhola
```

What was common among all those methods that we just discussed?

big( )	<BIG> … </BIG>
small( )	<SMALL> … </SMALL>
sub( )	<SUB> … </SUB>
sup( )	<SUP> … </SUP>
bold( )	<B> … </B>
italics( )	<I> … </I>
strike( )	<S> … </S>

fontsize(*n*)	<FONT size=*n*> … </FONT>
fontcolor(*color*)	<FONT color=*color*> … </FONT>
fixed( )	<PRE> … </PRE>
link(*URL*)	<A href=*URL*> …</A>

String Methods: All Others

toLowerCase( )
toUpperCase( )

charAt(*n*)
substring(*n, m*)

indexOf(*substring, n*)
lastIndexOf(*substring, n*)

split(*delimiter*)

toLowerCase( ), toUpperCase( )

```
person = "Bhola" ;
document.write( person ) ;
document.write( person.toLowerCase( ) ) ;
document.write( person.toUpperCase( ) ) ;
```

```
BholabholaBHOLA
```

charAt(*n*)

Returns a string containing the character at position *n* (the position of the 1st character is 0)

```
mister = "Bhola" ;
document.write( mister ) ;
document.write( mister.charAt( 0 ) ) ;
document.write( mister.charAt( 8 ) ) ;
document.write( mister.charAt( 2 ) ) ;
```

```
Bo
```

substring(*n, m*)

Returns a string containing characters copied from positions *n* to *m - 1*

```
s = "Bhola" ;
document.write( s.substring( 1, 3 ) ) ;
document.write( s.substring( 0, s.length ) ) ;
```

```
hoBhola
```

indexOf(*substring, n*)

Returns the position of the first occurrence of *substring* that appears on or after the *nth* position, if any, or -1 if none is found

```
s = "Bhola" ;
```

document.write(s.indexOf(“ola”, 1)) ;
document.write(s.indexOf(“z”, 3)) ;

```
2-1
```

lastIndexOf(*substring, n*)
Returns the position of the last occurrence of *substring* that appears on or before the n^{th} position, if any, or -1 if none is found
s = "Bhola" ;
document.write(s.lastIndexOf(“ola”, 5)) ;
document.write(s.lastIndexOf(“b”, 0)) ;

```
2-1
```

split(*delimiter*)
Returns an array of strings, created by splitting string into substrings, at *delimiter* boundaries

```
s = "Hello: I must be going!" ;
a = new Array( 5 ) ;
b = new Array( 5 ) ;
a = s.split( " " ) ;
b = s.split( "e" ) ;
document.write( "<TABLE>" ) ;
for( k = 0; k < 5; k = k + 1 )
      document.write( "<TR><TD>", a[ k ], "</TD><TD>", b[ k ], "</TD></TR>"
) ;
document.write( "</TABLE>" ) ;
```

```
Hello:    H
I      llo: I must b
must      going!
be   undefined
going!   undefined
```

Automatic Conversion to Strings

- Whenever a non-string is used where JavaScript is expecting a string, it converts that non-string into a string
- Example:
  - The document.write( ) method expects a string (or several strings, separated by commas) as its argument
  - When a number or a Boolean is passed as an argument to this method, JavaScript automatically converts it into a string before writing it onto the document

The ‘+’ Operator

- When ‘+’ is used with numeric operands, it adds them
- When it is used with string operands, it concatenates them
- When one operand is a string, and the other is not, the non-string will first be converted to a string and then the two strings will be concatenated

The ‘+’ Operator: Examples

document.write(2 + Math.PI) ;

document.write("2" + "3") ;

document.write("2" + Math.PI) ;

document.write("Yes" + false) ;

```
5.141592653589793
```

```
23
```

```
23.141592653589793
```

```
Yesfalse
```

```
6.283185307179586
```

```
NaN
```

__Strings In Mathematical Expressions__

When a string is used in a mathematical context, if appropriate, JavaScript first converts it into a number. Otherwise, a “NaN” is the result

```
document.write( "2" * Math.PI ) ;

document.write( "Yes" ^ 43 ) ;
```

```
100.55
```

__The ‘toString’ Method__

Explicit conversion to a string

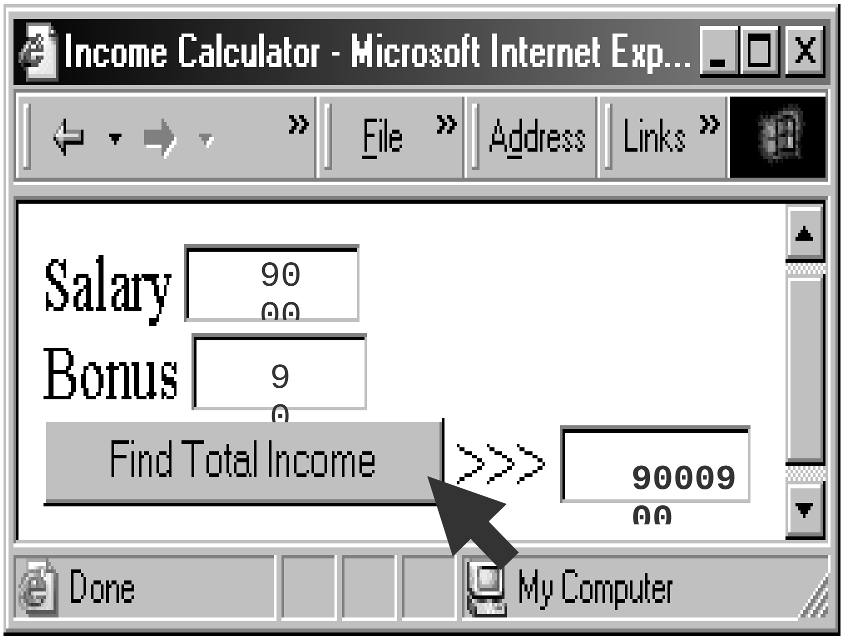
EXAMPLE:

Convert 100.553478 into a currency format

```
a = 100.553478 ;
b = a.toString( ) ;
decimalPos = b.indexOf( ".", 0 ) ;
c = b.substring( 0, decimalPos + 3 ) ;
document.write( c ) ;
```

Conversion from Strings

parseInt( ) and parseFloat( ) methods



Salary 9000

Bonus 90

900090

```
function calc( ) {
        document.myForm.total.value =
                document.myForm.salary.value +
                document.myForm.bonus.value ;
}
function calc( ) {
        document.myForm.total.value =
                parseFloat( document.myForm.salary.value ) +
                parseFloat( document.myForm.bonus.value ) ;
}
```

Why not use parseInt( ) here?

During Today's Lecture …

- We become familiar with methods used for manipulating strings
- We became able to solve simple problems involving strings

Next (the 14th) Web Dev Lecture:

Images & Animation

- To become able to add and manipulate images and animations to a Web page

Lecture 39

Cyber Crime

Focus of the last Lecture was on Database SW

- In our final lecture on productivity SW, we continued our discussion on data management
- We found out about relational databases
- We also implemented a simple relational database

Relational Databases

- Databases consisting of two or more related tables are called *relational databases*
- Each column of those tables can contain only a single type of data (contrast this with spreadsheet columns!)
- Table rows are called records; row elements are called fields
- A relational database stores all its data inside tables, and nowhere else
- All operations on data are done on those tables or those that are generated by table operations
- Tables, tables, and nothing but tables!

RDBMS

- Relational DBMS software
- Examples:
– Access
– FileMaker Pro
– SQL Server
– Oracle

Classification of DBMS w.r.t. Size

- Personal/Desktop/Single-user (MB-GB)
- Server-based/Multi-user/Enterprise (GB-TB)
- Seriously-huge databases (TB-PB-XB)

The Trouble with Relational DBs

- **Much of current SW development is done using the object-oriented methodology**
- **When we want to** store **the object-oriented data into an RDBMS, it needs to be translated into a** form **suitable for RDBMS**
- Then when we need to read the data back from the RDBMS, the data needs to be translated back into an object-oriented form before use
- These two processing delays, the associated processing, and time spent in writing and maintaining the translation code are the key disadvantages of the current RDBMSes

Some Terminology

- *Primary Key* is a field that uniquely identifies each record stored in a table
- *Queries* are used to view, change, and analyze data. They can be used to:
– Combine data from different tables, efficiently
– Extract the exact data that is desired
- *Forms* can be used for entering, editing, or viewing data, one record at a time
- *Reports* are an effective, user-friendly way of presenting data. All DBMSes provide tools for producing custom reports

Desktop RDBMS Demo

- We will create a new relational database
- It will consist of two tables
- We will populate those tables
- We will generate a report after combining the data from the two tables

Today's Lecture:

Cyber Crime

- To find out about several types of crimes that occur over cyber space (i.e. the Internet)
- To familiarize ourselves with with several methods that can be used to minimize the ill effects of those crimes

39.1 07 February 2000

- Users trying to get on to the Web sites of Yahoo, couldn't!
- Reason: Their servers were extremely busy!
- They were experiencing a huge number of hits
- The hit-rate was superior to the case when a grave incident (e.g. 9/11) occurs, and people are trying to get info about what has happened
- The only problem was that nothing of note had taken place!

What was going on?

- A coordinated, distributed DoS (Denial of Service) attack was taking place
- Traffic reached 1 GB/s; many times of normal!
- In the weeks leading to the attack, there was a noticeable rise in the number of scans that Internet servers were receiving
- Many of these scans appeared to originate from IP addresses that traced back to Korea, Indonesia, Taiwan, Australia

Three Phases of the DoS

1.Search
2.Arm
3.Attack

1. Search for Drones

- The attackers set about acquiring the control over the computers to be used in the attack …
- by scanning – using e.g. Sscan SW – a large numbers of computers attached to the Internet
- Once a computer with a weak security scheme is identified, the attackers try a break-in
- Once conquered, that computer – called a drone – will be used to scan others

2. Arming the Drones

- After several drones have been conquered, the DoS SW is loaded on to them
- Examples: Tribal Flood Network, Trinoo, TFN2K
- Like a time-bomb, that SW can be set to bring itself into action at a specified time
- Alternatively, it can wait for a commencement command from the attacker

3. The Actual Attack

- At the pre-specified time or on command, the SW implanted on all of the drones wakes-up and starts sending a huge number of messages to the targeted servers
- Responding to those messages overburdens the targeted servers and they become unable to perform their normal functions

Neutralizing the Attack

- The engineers responsible for monitoring the traffic on the Yahoo Web sites quickly identified the key characteristics of the packets originating from those drones
- Then they setup filters that blocked all those packets
- It took them around 3 hours to identify and block most of the hostile packets
- BTW, the sender's IP address can be spoofed, making it impossible to block the attack just by blocking the IP addresses

The Aftermath

- None of the Yahoo computers got broken-into; The attackers never intended to do that
- None of the user data (eMail, credit card numbers, etc.) was compromised
- Ill-effects:

– Yahoo lost a few million's worth of business
– Millions of her customers got annoyed as they could not access their eMail and other info from the Yahoo Web sites

Who Done It?

- The DoS SW is not custom SW, and can be downloaded from the Internet. Therefore, it is difficult to track the person who launched the attack by analyzing that SW

- After installing the DoS SW on the drones, setting the target computer and time, the attackers carefully wipe away any info on the drone that can be used to track them down
- End result: Almost impossible to track and punish clever attackers

How to stop DoS attacks from taking place?

- Design SW that monitors incoming packets, and on noticing a sudden increase in the number of similar packets, blocks them
- Convince system administrators all over the world to secure their servers in such a way that they cannot be used as drones
- BTW, the same type of attack brought down the CNN, Buy, eBay, Amazon Web sites the very next day of the Yahoo attack

39.2 DoS Attack: A Cyber Crime

- DoS is a crime, but of a new type - made possible by the existence of the Internet
- A new type of policing and legal system is required to tackle such crimes and their perpetrators
- Internet does not know any geographical boundaries, therefore jurisdiction is a key issue when prosecuting the cyber-criminal

Cyber crime can be used to …

- Damage a home computer
- Bring down a business
- Weaken the telecom, financial, or even defense-related systems of a country

Cyberwar!

- In 1997, blackouts hit New York City, Los Angeles
- The 911 (emergency help) service of Chicago was shut down
- A US Navy warship came under the control of a group of hackers
- What was happening? A cyber attack!
- All of the above did not happen in reality, but in a realistic simulation
- The US National Security Agency hired 35 hackers to attack the DoD's 40,000 computer networks
- By the end of the exercise, the hackers had gained root-level (the highest-level!) access to at least 3 dozen among those networks

Cyberwarfare:

A clear and present threat as well opportunity for all of the world's armed force!

39.3 More cybercrimes …

Mail Bombing

- Similar in some ways to a DoS attack
- A stream of large-sized eMails are sent to an address, overloading the destination account
- This can potentially shut-down a poorly-designed eMail system or tie up the telecom channel for long periods
- Defense: eMail filtering

Break-Ins

- Hackers are always trying to break-in into Internet-connected computers to steal info or plant malicious programs
- Defense: Intrusion detectors

Credit Card Fraud

- A thief somehow breaks into an eCommerce server and gets hold of credit numbers and related info
- The thief then uses that info to order stuff on the Internet
- Alternatively, the thief may auction the credit card info on certain Web sites setup just for that purpose
- Defense: Use single-use credit card numbers for your Internet transactions

Software Piracy

•Using a piece of SW without the author's permission or employing it for uses not allowed by the author is SW piracy

•For whatever reason, many computer users do not consider it to be a serious crime, but it is!
•Only the large rings of illegal SW distributors are ever caught and brought to justice
•Defense: Various authentication schemes. They, however, are seldom used as they generally annoy the genuine users

Industrial Espionage

- Spies of one business monitoring the network traffic of their competitors
- They are generally looking for info on future products, marketing strategies, and even financial info
- Defense: Private networks, encryption, network sniffers

Web Store Spoofing

- A fake Web store (e.g. an online bookstore) is built
- Customers somehow find that Web site and place their orders, giving away their credit card info in the process
- The collected credit card info is either auctioned on the Web or used to buy goods and services on the Web

39.4 Viruses

- Self-replicating SW that eludes detection and is designed to attach itself to other files
- Infects files on a computers through:

– Floppy disks, CD-ROMs, or other storage media
– The Internet or other networks

- Viruses cause tens of billions of dollars of damage each year
- One such incident in 2001 – the LoveBug virus – had an estimated cleanup/lost productivity cost of US$8.75 billion
- The first virus that spread world-wide was the *Brain* virus, and was allegedly designed by someone in Lahore

One Way of Classifying Viruses

- Malicious

– The type that grabs most headlines
– May destroy or broadcast private data
– May clog-up the communication channels
– May tie-up the uP to stop it from doing useful work

Neutral
– May display an annoying, but harmless message
Helpful
– May hop from one computer to another while searching for and destroying malicious viruses

Anatomy of a Virus

- A virus consists of 2 parts:
- Transmission mechanism
- Payload

Transmission Mechanism

- Viruses attach themselves to other computer programs or data files (termed as *hosts*)
- They move from one computer to another with the *hosts* and spring into action when the *host* is executed or opened

Payload

- The part of the virus that generally consists of malicious computer instructions
- The part generally has two further components:

– Infection propagation component:

- This component transfers the virus to other files residing on the computer

– Actual destructive component:

- This component destroys data or performs or other harmful operations

Commonsense Guidelines

- Download SW from trusted sites only

- Do not open attachments of unsolicited eMails
- Use floppy disks and CDROMs that have been used in trusted computers only
- When transferring files from your computer to another, use the write-protection notches
- Stay away from pirated SW
- Regularly back your data up
- Install Antivirus SW; keep it and its virus definitions updated

Antivirus SW

- Designed for detecting viruses & inoculating
- Continuously monitors a computer for known viruses and for other tell-tale signs like:
  - Most – but, unfortunately not all – viruses increase the size of the file they infect
  - Hard disk reformatting commands
  - Rewriting of the boot sector of a hard disk
- The moment it detects an infected file, it can automatically inoculate it, or failing that, erase it

39.5 Other Virus-Like Programs

- There are other computer programs that are similar to viruses in some ways but different in some others
- Three types:
  - Trojan horses
  - Logic- or time-bombs
  - Worms

Trojan Horses

- Unlike viruses, they are stand-alone programs
- The look like what they are not
- They appear to be something interesting and harmless (e.g. a game) but when they are executed, destruction results

Logic- or Time-Bombs

- It executes its payload when a predetermined event occurs
- Example events:
- A particular word or phrase is typed
  - A particular date or time is reached

Worms

- Harmless in the sense that they only make copies of themselves on the infected computer
- Harmful in the sense that it can use up available computer resources (i.e. memory, storage, processing), making it slow or even completely useless

Designing, writing, or propagating malicious code or participating in any of the fore-mentioned activities can result in criminal prosecution, which in turn, may lead to jail terms and fines!

Today's Lecture:

- We found out about several types of computer crimes that occur over cyber space
- We familiarized ourselves with with several methods that can be used to minimize the ill effects of these crimes

Next Lecture' Goals

(Social Implications of Computing)

We will explore the impact of computing on:

Business
Work
Living
Health
Education

Lecture 40
Social Implications of Computing

Focus of the last Lecture was on Cyber Crime

- We found out about several types of computer crimes that occur over cyber space
- We familiarized ourselves with with several methods that can be used to minimize the ill effects of these crimes

Three Phases of the DoS

1.Search
2.Arm
3.Attack

Neutralizing the Attack

- The engineers responsible for monitoring the traffic on the Yahoo Web sites quickly identified the key characteristics of the packets originating from those drones
- Then they setup filters that blocked all those packets

How to stop DoS attacks from taking place?

- Design SW that monitors incoming packets, and on noticing a sudden increase in the number of similar packets, blocks them
- Convince system administrators all over the world to secure their servers in such a way that they cannot be used as drones

Cyber crime can be used to …

- Damage a home computer
- Bring down a business
- Weaken the telecom, financial, or even defense-related systems of a country

Mail Bombing

- A stream of large-sized eMails are sent to an address, overloading the destination account
- This can potentially shut-down a poorly-designed eMail system or tie up the telecom channel for long periods
- Defense: eMail filtering

Break-Ins

- Hackers are always trying to break-in into Internet-connected computers to steal info or plant malicious programs
- Defense:
  - Firewalls
  - Intrusion detectors
  - Other effective security policies

Credit Card Fraud

- A thief somehow breaks into an eCommerce server and gets hold of credit numbers and related info
- The thief then uses that info to order stuff on the Internet
- Alternatively, the thief may auction the credit card info on certain Web sites setup just for that purpose
- Defense: Use single-use credit card numbers for your Internet transactions

Software Piracy

- Using a piece of SW without the author's permission or employing it for uses not allowed by the author is SW piracy
- Defense: Various authentication schemes. They, however, are seldom used as they generally annoy the genuine users

Industrial Espionage

- Spies of one business monitoring the network traffic of their competitors
- They are generally looking for info for future products, marketing strategies, and even financial info
- Defense: Private networks, encryption, network sniffers

Viruses

- Self-replicating SW that eludes detection and is designed to attach itself to other files
- Infects files on a computers through:
  - Floppy disks, CD-ROMs, or other storage media
  - The Internet or other networks

Anatomy of a Virus

A virus consists of 2 parts:

- Transmission mechanism
- Payload

Other Virus-Like Programs

- There are other computer programs that are similar to viruses in some ways but different in some others
- Three types:
  - Trojan horses
  - Logic- & time-bombs
  - Worms

Today's Goals:

(Social Implications of Computing)

- We will try to understand the impact of computing on:
  - Business
  - Work
  - Living
  - Health
  - Education

40.1 Introduction

- It should be clear to you that - for better or worse - the future of computing and the future of humankind are highly interdependent
- Computers have solved many problems for the humankind but have created a few tricky ones as well
- Today we will discuss both, but first …
- Why is it important to discuss the social implication of computing?

Why should we, as computing professionals, be interested in studying the social implications of our creations?

- Computing technology has changed our way of life like no other technology
- We need to study how it has done it to highlight the mistakes and success stories of the past
- We need to do it so that we can learn from them and select our future direction accordingly

Let's Start with the Dilemma of Computing

- Computers keep on becoming more and more powerful and gaining more and more autonomy
- They are being equipped with fail-safe and self-healing technologies
- Are we heading towards a future where the role of the masters and the slaves will be switched?
- Should we slow down or even reverse some of the technology advances to avoid that dark scenario?

40.2 Powerful Global Corporations

- Internet-based communication is allowing business entities to coordinate the activities of their globally-spread units with greater accuracy
- The knowledge gained by one unit becomes available to all others very quickly
- All this has made these business entities very powerful, even more powerful than many nation-states

40.3 The Network Organization

- The network paradigm (all connected to many others) is becoming the preferred organizational structure of more and more organizations as time goes by
- This new organization is replacing the old-style layered, tree-structured organizational model
- The organizations are learning that business can be done in a more effective manner if emphasis is placed upon cooperation, shared responsibility and networking:
  - Within the organization
  - And also with their customers and suppliers

The structure of the networked organization is flexible (although, at times a bit chaotic!), and changes according to the demand of the times

- It shares knowledge and decentralizes the control of the operation so that network works effectively to meet the business goals of the organization
- The workers :
  - Can spend more time doing creative work as they have immediate access to all of the required info through various computer-based technologies
  - Have a sense of ownership in the organization

While old professions are being eliminated …

- Typists
- Bank tellers
- Telephone operators

Number of Temporary Workers is on the Rise

- Even technical professionals of high-quality must now define themselves as temporary consultants, able to move from project to project within in an organization as well as among different organizations
- In the old days, loyalty was important, now professionalism and ability to perform are the watch words!
- The focus now in many computer-centric organizations is not belonging to the organization, but on professional competency and quality of work

Businesses Monitoring Their Employees

- Systems are available that monitor almost every key stroke that an employee makes on a computer
- Systems are available that read and censor all incoming and outgoing eMail
- It is quite straight forward to monitor where you surf, and when

40.4 Working from Home

- Computing has made it possible for some to avoid going the office for their work
- They can do their work from home and communicate their ideas, questions, answers to their colleagues through the Internet
- This gives them more time to spend with their families due to the time they save on commuting to their place of work

Working from Home: Disadvantages

- Contact with the colleagues and the quality of communication is reduced, which may result in a poorer quality of work
- Lack of interaction may also result in slower professional growth
- Family life may suffer as well, as some never turn off, and keep on working through out the day, evening and night!

40.5 From Mass- to Personalized-Marketing

- In the old days demographical data was analyzed and mass-marketing campaigns were launched to influence a reasonable portion of the population
- The Web has changed marketing forever, redirecting it from a mass focus to a single-person focus
- Our Web surfing data are captured. We are asked questions about our lifestyle in return of randomly awarded prizes
- All the collected data is then analyzed to determine patterns in our behavior, and individualized offers for services and goods are displayed in front of us on the Web or eMail

40.6 The Political Process

- We no longer need to gather the public's opinion through expensive referenda or public meetings
- Through computer discussion forums, newsgroups and mailing-lists, public and politicians may engage in a free, open exchange of ideas without leaving the comfort of their not-so-comfortable and very comfortable homes, respectively

Distances Have Contracted

- Because of the ever-decreasing costs of verbal, text, video communications, it is becoming easier to stay in touch of anyone, regardless of their physical location
- This has had a profound effect on small businesses, especially in developing countries like Pakistan
- It has also made it possible for families and friends to become closer in spite of the physical distance between them

Distances Are Increasing

- Television was bad enough; Video games and the Web has made the situation even worse
- Families are spending less & less time together in spite of the physical closeness. This may have a very detrimental effect on the emotional well-being of the children, … and parents
- Solitude is the order of the day as many children & adults spend their free time surfing, chatting, playing computer games, instead of spending it on interacting with friends or family

Virtual Communities

- Interest-based, instead of geography-based communities
- Ex: Ta'suv'voof, tennis, telepathy, cancer
- Members with common interest share ideas, ask questions, post answers and make announcements through mailing-lists, news groups or message boards
- These communities are definitely very different from traditional ones
- There are generally no bars on membership based on gender, race or religion
- However, they may lack the respect for the individual and civility that are the norm within conventional, geography-based communities

A Society Under Surveillance

- While surfing, we are being watched, constantly
- Our every click is recorded and analyzed to extract patterns and behaviors
- Those patterns are then used to persuade us to do things that those Web sites want us to do
- Webcams are becoming common. Providing a cheap way for parents to watch their children's every move

The Changing Face of Education

- Distance learning has received a boost due to the low-price of Internet communication and the availability of Web-based interactive content
- It has also become possible for students to interact in real-time with other students as well as teachers located a long distance from them
- Physical location is less of a hindrance now
- Students enrolled in distance-education programs have more control over what they want to learn, how they want to learn, and when
- The lack of face-to-face interaction and immediate to-and-fro questions and answers may, however, reduce the amount of knowledge that can be transferred from the teacher to the student
- In spite of that problem, computer-based distance education may be the only source of high-quality education for many, especially those in remote locations
- The fact, however, remains that the best mode of education is the conventional one, which has become more effective with the augmentation of computer-based learning aids

Info Gathering

- We are turning more and more towards online resources of info
- The info that just a few years back involved effort and time to pull together before it could be used is now literally a few key strokes away
- The time and effort spent on gathering info can now be spent on using it
- This capability has made the computer an active (and integral) part of our creative process

Telemedicine

- How can we place a doctor specializing in, for example, skin-related diseases or neurology in every district's hospital?
- We cannot! What then? Ignore all those not residing in big cities?
- Solution: Internet-based telemedicine
- An audio/video/text connection combined with a few remote medical instruments and a trained assistant can enable a remote doctor to examine and prescribe medicine to a patient far, far away

Is Progress Necessary?

- Progress is being made every day in the field of computing. The question that we need to ask is: "Are we going in the right direction?"
- Is it OK to make available all sorts of info to everyone? Does everyone needs to know how to build an H-bomb?
- Is it OK to keep on investing in surveillance technologies? Do personal privacy have no place in our technologically advanced future?
- Is it OK to automate everything that we lay our eyes on? Or certain things (e.g. caring for an infant) should remain with us old-fashioned human beings

Closure

- Your answers to the questions that I just raised may differ from mine, and I respect your opinion. All I say is, yes, progress is inventible, however, you – the creators of my future – should be a bit thoughtful about what you do

I command you to "go and invent the future," it is your duty and you may not desist from it, but, please, do think about the social implications and consequences of what you are doing before actually doing it

Today's Lecture:

(Social Implications of Computing)

We discussed the impact of computing on:

- Business
- Work
- Living
- Health
- Education

Next Lecture' Goals:

(The Computing Profession)

- Roles & responsibilities of a modern computer professionals
- The ethical issues facing the computing profession

Lecture 41
Images & Animation
(Web Development Lecture 14)

During the last lecture we discussed String Manipulation

- We became familiar with methods used for manipulating strings
- We became able to solve simple problems involving strings

String Manipulation in JavaScript

- In addition to the concatenation operator (+) JavaScript supports several advanced string operations as well
- Notationaly, these functions are accessed by referring to various methods of the String object
- Moreover, this object also contains the 'length' property

String Methods

FORMAT

string.methodName( )

EXAMPLE:

```
	name = "Bhola" ;
	document.write( name.toUpperCase( ) ) ;
	document.write( name.bold( ) ) ;
```

BHOLA**Bhola**

Two Types of String Methods

1.HTML Shortcuts
2.All Others

String Methods: HTML Shortcuts

big( )
small( )
fontsize(*n*)

bold( )
italics( )
strike( )

link(URL)

sub( )
sup( )

fixed( )
fontcolor(*color*)

String Methods: All Others

toLowerCase( )
toUpperCase( )

charAt(*n*)
substring(*n, m*)

indexOf(*substring, n*)
lastIndexOf(*substring, n*)

split(*delimiter*)

Automatic Conversion to Strings

- Whenever a non-string is used where JavaScript is expecting a string, it converts that non-string into a string
- **Example:**

– The document.write( ) method expects a string (or several strings, separated by commas) as its argument

– When a number or a Boolean is passed as an argument to this method, JavaScript automatically converts it into a string before writing it onto the document

The '+' Operator

- When '+' is used with numeric operands, it adds them
- When it is used with string operands, it concatenates them
- When one operand is a string, and the other is not, the non-string will first be converted to a string and then the two strings will be concatenated

Strings In Mathematical Expressions

When a string is used in a mathematical context, if appropriate, JavaScript first converts it into a number. Otherwise, a "NaN" is the result

```
document.write( "2" * Math.PI ) ;
```

```
6.283185307179586
```

```
document.write( "Yes" ^ 43 ) ;
```

```
NaN
```

The 'toString' Method

Explicit conversion to a string

EXAMPLE:

Convert 100.553478 into a currency format

```
a = 100.553478 ;
b = a.toString( ) ;
decimalPos = b.indexOf( ".", 0 ) ;
c = b.substring( 0, decimalPos + 3 ) ;
document.write( c ) ;
```

```
100.55
```

Conversion fr g

parseInt( ) and parseFloat( ) methods

Today's Goal

(Images & Animation)

- To become able to add and manipulate images and simple animations to a Web page

Images in HTML

- It is quite straight forward to include gif and jpg images in an html Web page using the <IMG> tag

Format: <IMG src=*URL*, alt=*text* height=*pixels* width=*pixels* align="*bottom*|*middle*|*top*">

Plea: Don't use images just for the sake of it!

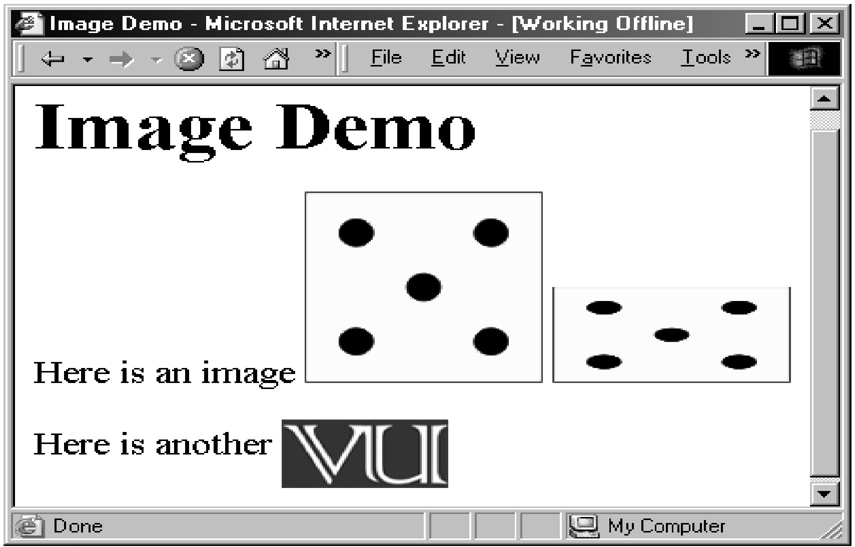


Image Demo - Microsoft Internet Explorer - [Working Offline]
File
Edit
View
Favorites
Tools
Done
My Computer

Image Demo

Here is an image

Here is another

```
<HTML><HEAD>
	<TITLE>Image Demo</TITLE>
</HEAD><BODY>
<H1>Image Demo</H1>
Here is an image <IMG src="die5.gif">
<IMG src="die5.gif" height="63" width="126"> <P>
Here is another <IMG align="middle" src=
"http://www.vu.edu.pk/images/logo/logotop.jpg">
</BODY></HTML>
```

41.1 Images in JavaScript

- Images in JavaScript can be manipulated in many ways using the built-in object Image
- Properties: name, border, complete, height, width, hspace, vspace, lowsrc, src
- Methods: None
- Event handlers: onAbort, onError, onLoad, etc.

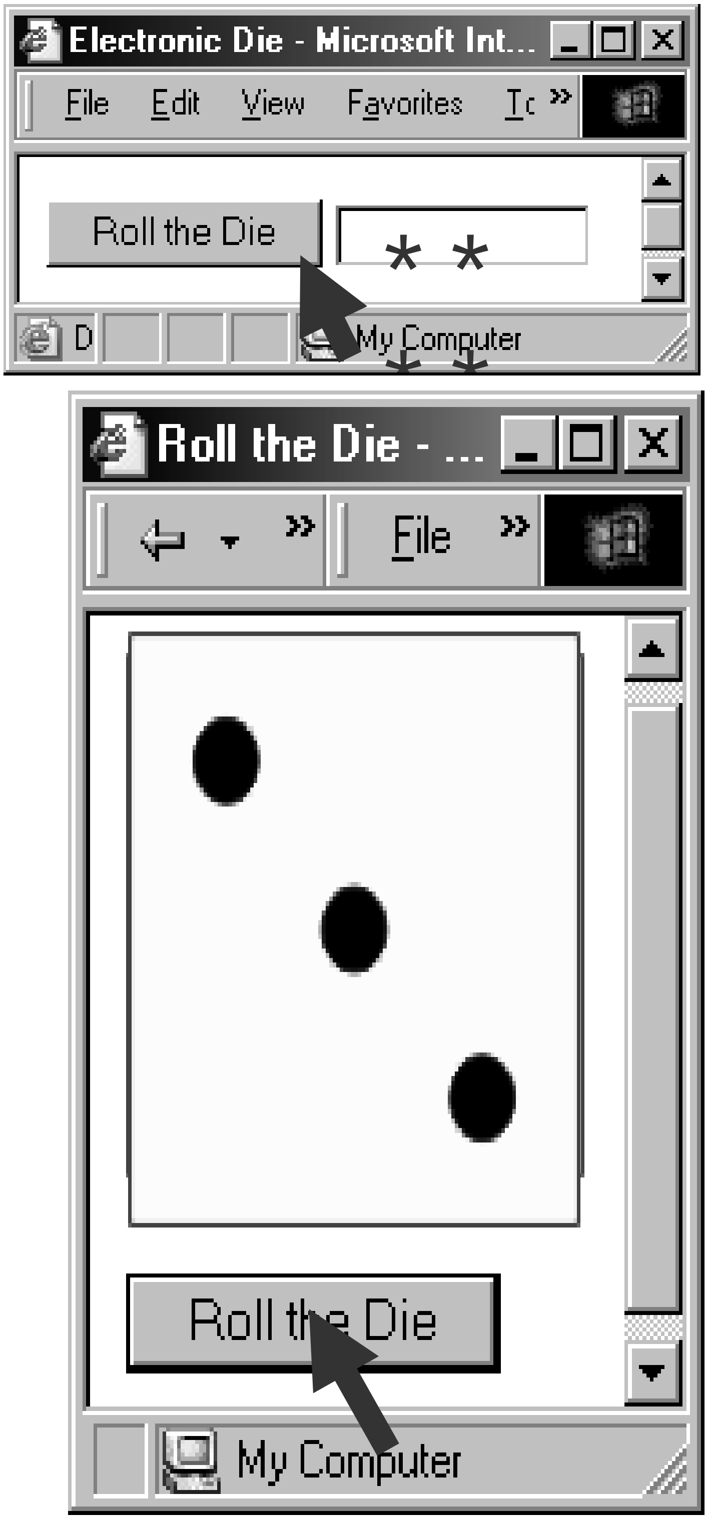
Image Preloading

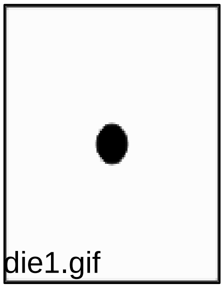
- The primary use for an Image object is to download an image into the cache before it is actually needed for display
- This technique can be used to create smooth animations or to display one of several images based on the requirement

The Image Pre-Loading Process

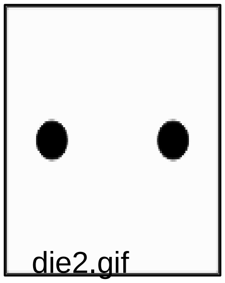
1. An instance of the Image object is created using the *new* keyword
2. The *src* property of this instance is set equal to the *filename* of the image to be pre-loaded
3. That step starts the down-loading of the image into the cache without actually displaying it
4. When a pre-loaded image is required to be displayed, the *src* property of the displayed image is set to the *src* property of the pre-fetched image

Let us revisit an example that we first saw in lecture 35

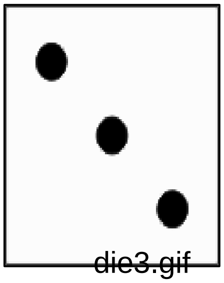




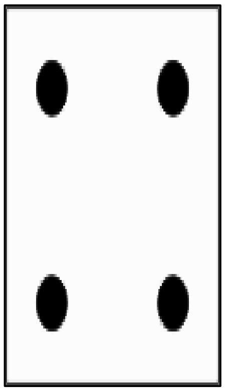
die1.gif



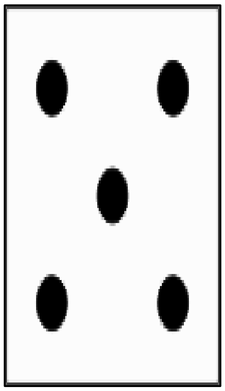
die2.gif



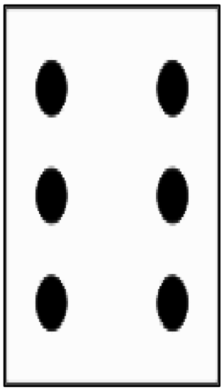
die3.gif



die4.gif



die5.gif



die6.gif

```
<HTML>
<HEAD>
      <TITLE>Roll the Die</TITLE>
      <SCRIPT>
            JavaScript Code
      </SCRIPT>
</HEAD>
<BODY >
      HTML Code
</BODY>
</HTML>
```

```
<IMG name="die" src="die6.gif">
<FORM>
      <INPUT type="button" value="Roll the Die"
            onClick="rollDie( )">
</FORM>
```

```
dieImg = new Array( 7 ) ;
for( k = 1; k < 7; k = k + 1 ) { //Preload images
      dieImg[ k ] = new Image( ) ;
      dieImg[ k ].src = "die" + k + ".gif" ;
}
function rollDie( ) {
      dieN = Math.ceil( 6 * Math.random( ) ) ;
      document.die.src = dieImg[ dieN ].src ;
}
```

Another Example

- Develop a Web page that displays six thumbnail images and a main image
- The main image should change to a larger version of the thumbnail as soon as the mouse moves over on a thumbnail image

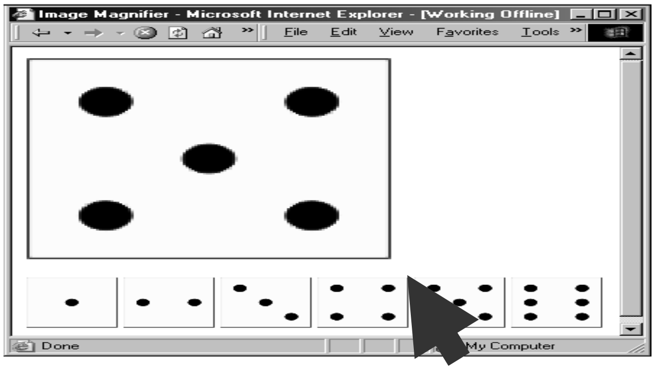


Image Magnifier - Microsoft Internet Explorer - [Working Offline]
File
Edit
View
Favorites
Tools
Done
My Computer

```
<HTML>
<HEAD>
      <TITLE>Image Selector</TITLE>
      <SCRIPT>
            JavaScript Code
      </SCRIPT>
</HEAD>
<BODY >
      HTML Code
</BODY>
</HTML>
```

```
dieImg = new Array( 7 ) ;
for( k = 1; k < 7; k = k + 1 ) { // Preload images
      dieImg[ k ] = new Image( ) ;
      dieImg[ k ].src = "die" + k + ".gif" ;
}
```

```
<IMG name="big" src="die6.gif" width="252" height="252"><P>
<IMG src="die1.gif" width="63" height="63"
      onMouseOver=
            "document.big.src=dieImg[ 1 ].src">
      …
```

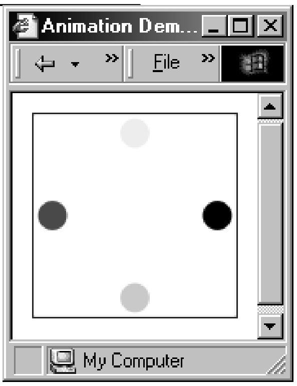
```
        …
<IMG src="die6.gif" width="63" height="63"
        onMouseOver=
                "document.big.src=dieImg[ 6 ].src">
```

Where Else Can We Use This?

- Automobile Web site
- ???

Animation Example 1

- Take 16 images and cycle through them to create an animation effect

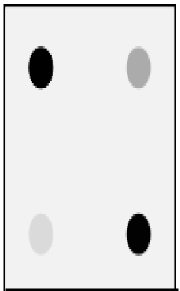


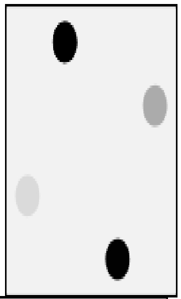
```
<HTML>
<HEAD>
        <TITLE>Animation 1</TITLE>
        <SCRIPT>
                JavaScript Code
        </SCRIPT>
</HEAD>
<BODY >
        HTML Code
</BODY>
</HTML>
```

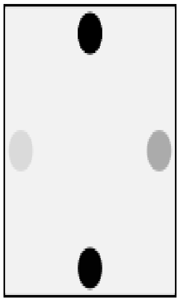
```
<CENTER>
        <IMG name="circle" src="circle1.gif"   onLoad="setTimeout(   'circulate(   )',   gap
)">
</CENTER>
```

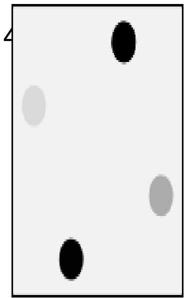
setTimeout( ) executes *circulate( )* once after a delay of *gap* milliseconds

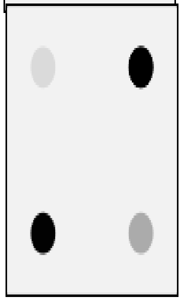
```
gap = 100 ;
imageN = 1 ;
circImg = new Array( 17 ) ;
for( k = 1; k < 17; k = k + 1 ) { // Preload images
        circImg[ k ] = new Image( ) ;
        circImg[ k ].src = "circle" + k + ".gif" ;
}
```

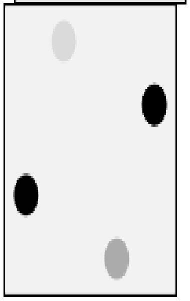
1


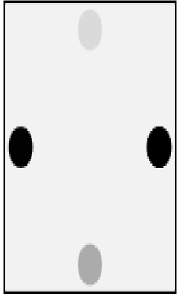
2


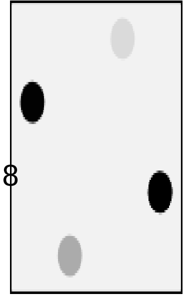
3


4


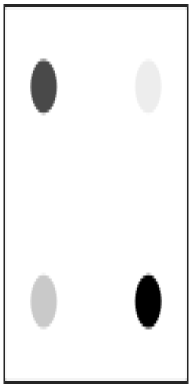
5


6


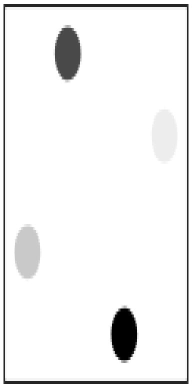
7


8


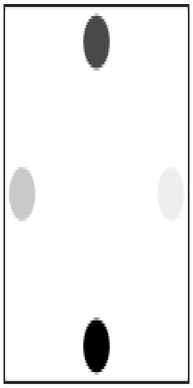
9



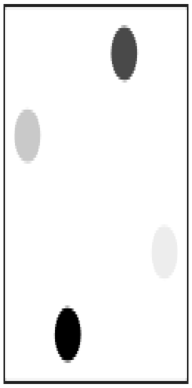
10



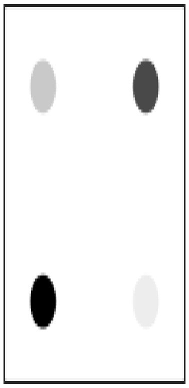
11



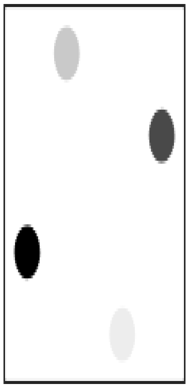
12



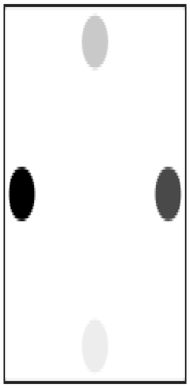
13



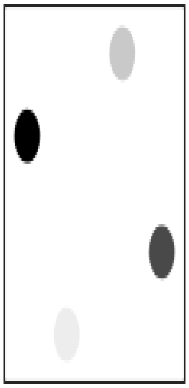
14



15



16



```
function circulate( ) {
        document.circle.src =
                circImg[ imageN ].src ;
        imageN = imageN + 1 ;
        if( imageN > 16 )
                imageN = 1 ;
}
```

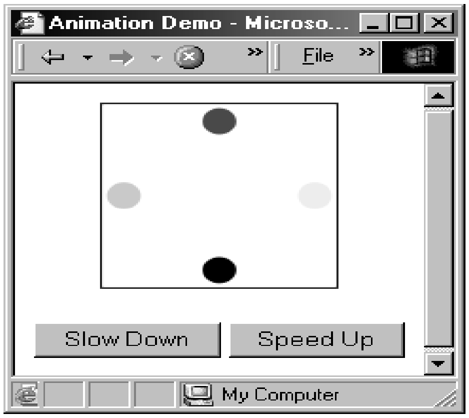
Animated Gifs

- We could have saved the 16 gif images of the previous example in a single file in the form of an animated gif, and then used it in a regular <IMG> tag to display a moving image
- However, JavaScript provides better control over the sequencing and the gap between the individual images

Animation Example 2

- Take 16 images and cycle through them to create an animation effect
- Provide buttons to slow down or speed up the animation

```
<HTML>
<HEAD>
        <TITLE>Animation 2</TITLE>
        <SCRIPT>
                JavaScript Code
        </SCRIPT>
</HEAD>
<BODY >
        HTML Code
</BODY>
</HTML>
```



```
<CENTER>
        <IMG name="circle" src="circle1.gif"   onLoad="setTimeout( 'circulate( )',
gap )">
</CENTER>
<FORM>
        <INPUT type="button" value="Slow Down"
                onClick="slowDown( )">
        <INPUT type="button" value="Speed Up"
                onClick="speedUp( )">
</FORM>


gap = 100 ;
imageN = 1 ;
circImg = new Array( 17 ) ;
for( k = 1; k < 17; k = k + 1 ) { // Preload images

        circImg[ k ] = new Image( ) ;

        circImg[ k ].src = "circle" + k + ".gif" ;
}
```

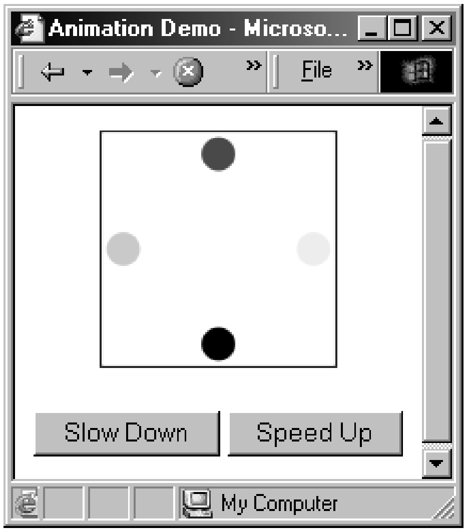
No change

```
function circulate( ) {
        document.circle.src =
                circImg[ imageN ].src ;
        imageN = imageN + 1 ;
        if( imageN > 16 )
                imageN = 1 ;
}
```

No change

```
function slowDown( ) {
        gap = gap + 20 ;
        if( gap > 4000 )
                gap = 4000 ;
}
function speedUp( ) {
        gap = gap - 20 ;
        if( gap < 0 )
                gap = 0 ;



}
```

Two new functions



41.2 Flash Animation

- Designed for 2-D animations, but can be used for storing static vector-images as well
- A special program (called a plug-in) is required to view Flash files in a Web browser
- Can be used to design complete, animated Web sites with hardly any HTML in it
- Binary-file storage

Structured Vector Graphics

- New format; may become more popular than Flash
- Plug-in required
- Text-file storage; search engine friendly

During Today's Lecture …

- We became able to add and manipulate images and simple animations to a Web page

Our 15th & Final Web Dev Lecture:

(Programming Methodology)

- To understand effective programming practices that result in the development of correct programs with minimum effort
- To become familiar with simple debugging techniques

Lecture 42
The Computing Profession

Focus of the last Lecture was on Social Implications of Computing

We discussed the impact of computing on:

– Business
– Work
– Living
– Health
– Education

Why should we, as computing professionals, be interested in studying the social implications of our creations?

- Computing technology has changed our way of life like no other technology
- We need to study how it has done it to highlight the mistakes and success stories of the past
- We need to do it so that we can learn from them and select our future direction accordingly

Dilemma of Computing

- Are we heading towards a future where the role of the masters and the slaves will be switched?
- Should we slow down or even reverse some of the technology advances to avoid that dark scenario?

Powerful Global Corporations

- Internet-based communication is allowing business entities to coordinate the activities of their globally-spread units with greater accuracy
- All this has made these business entities very powerful, even more powerful than many nation-states

The Network Organization

- The network paradigm (all connected to many others) is becoming the preferred organizational structure of more and more organizations as time goes by
- This new organization is replacing the old-style layered, tree-structured organizational model

Working from Home

- Computing has made it possible for some to avoid going the office for their work
- They can do their work from home and communicate their ideas, questions, answers to their colleagues through the Internet
- This gives them more time to spend with their families due to the time they save on commuting to their place of work

From Mass- to Personalized-Marketing

- The Web has changed marketing forever, redirecting it from a mass focus to a single-person focus

The Political Process

- Through computer discussion forums, newsgroups and mailing-lists, public and politicians may engage in a free, open exchange of ideas without leaving the comfort of their not-so-comfortable and very comfortable homes, respectively

Distances Have Contracted

- Because of the ever-decreasing costs of verbal, text, video communications, it is becoming easier to stay in touch with anyone, regardless of their physical location

Distances Are Increasing

- Solitude is the order of the day as many children & adults spend their free time surfing, chatting, playing computer games, instead of spending it on interacting with friends or family

Virtual Communities

- Interest-based, instead of geography-based

A Society Under Surveillance

- While surfing, we are being watched, constantly

The Changing Face of Education

- Distance learning has received a boost due to the low-price of Internet communication and the availability of Web-based interactive content

The Changing Face of Education

- The fact, however, remains that the best mode of education is the conventional one, which has become more effective with the augmentation of computer-based learning aids

Info Gathering

- The time and effort spent on gathering info can now be spent on using it

Telemedicine

- An audio/video/text connection combined with a few remote medical instruments and an on-site trained assistant can enable a doctor to examine and prescribe medicine to a patient far, far away

Closure

- I command you to "go and invent the future," it is your duty and you may not desist from it, but, please, do think about the social implications and consequences of what you are doing before actually doing it

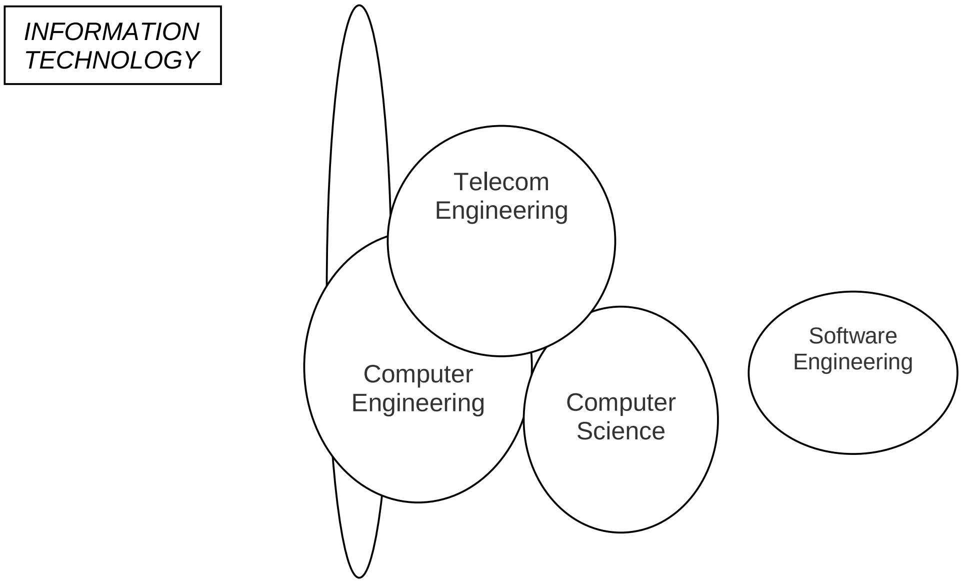
Today's Goals:

(The Computing Profession)

- To discuss several roles and associated responsibilities of modern computer professionals
- To discuss a few tricky situations where a knowledge of professional ethics would help

42.1 IT: Information Technology

The group of technologies concerned with the capture, processing and transmission of information in the digital-electronic form



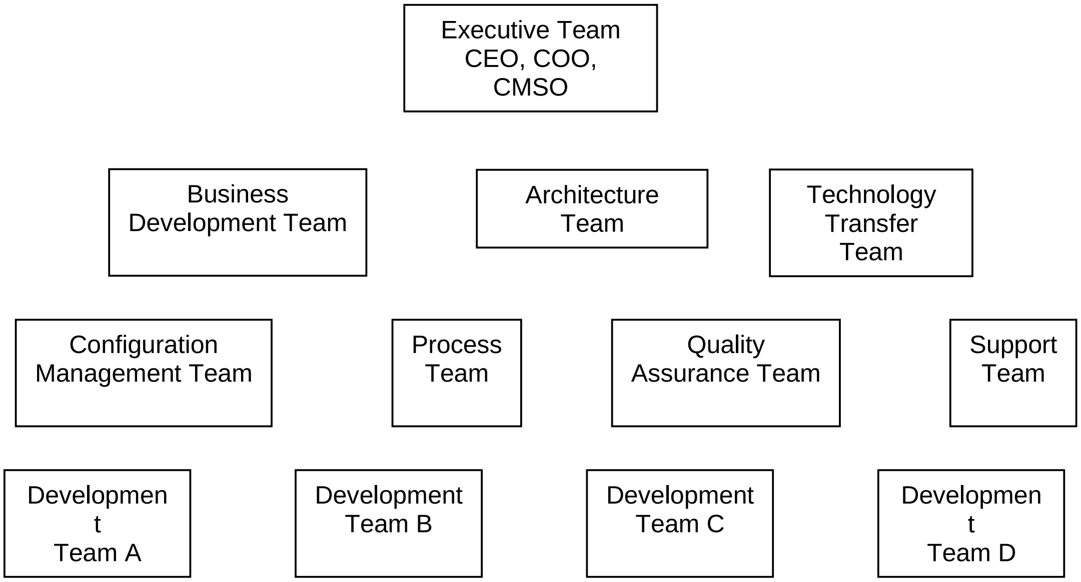
Who is a computing professional?

- Professionals involved in the development and/or maintenance of SW and/or computer HW
- Computer scientists, software engineers, computer engineers, and some of the telecom engineers are generally classified as computing professionals

Today's Focus Group

- Due to the limitation on time, today we will be focusing only on a subset of computing professionals: those involved in the development of SW
- Let us further restrict discussion to the computing professionals belonging to an organization focused solely on custom, SW development
- They work in a 100-person organization – pretty big on a local scale, but quite insignificant on an international one

42.2 Organization: A Collection of Teams



Development Team

- The number of development teams has varied between 3-7 at this organization
- Team-size has varied between 3-35
- Large teams are organized as a collection of sub-teams
- Lowest-level team: No more than 7 members
- Responsible for a project from after the specifications stage till the very end

Project Manager

- **Responsibilities:**

– Planning and tracking of the project
– Arranging of the appropriate resources
– lient relationship management

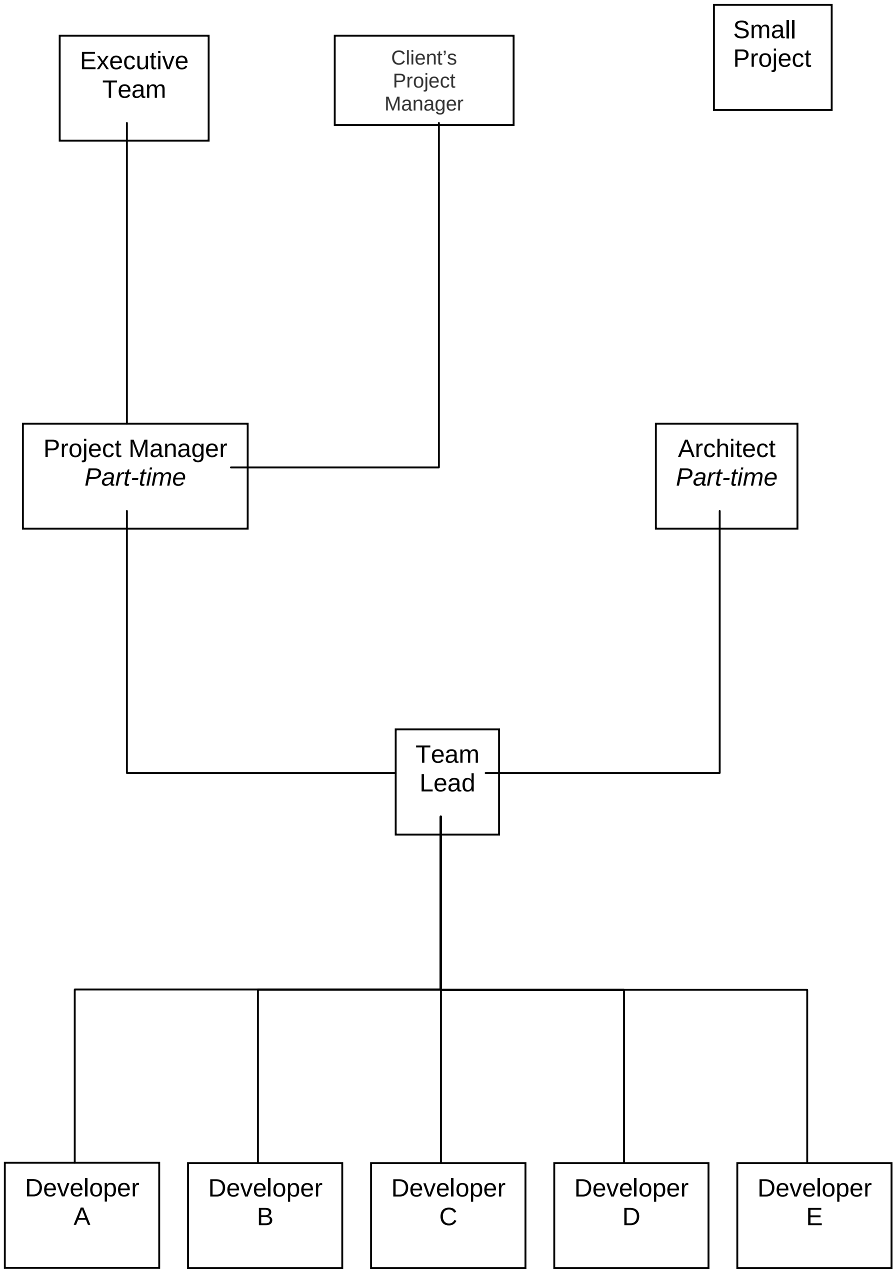
- **Profile:**

– 5+ years of team-lead experience
– Professional development course(s) in SW project management
– Technical MS and/or Technical BS + MBA

Architect

- **Responsibilities:**

- – Technology selection
- – High-level design
- – Makes certain that the implementation remains true to the design
- **Profile:**
- – 10-15 years of development experience
- – In-depth experience in several technologies
- – MS or PhD in a technical discipline



Team Lead

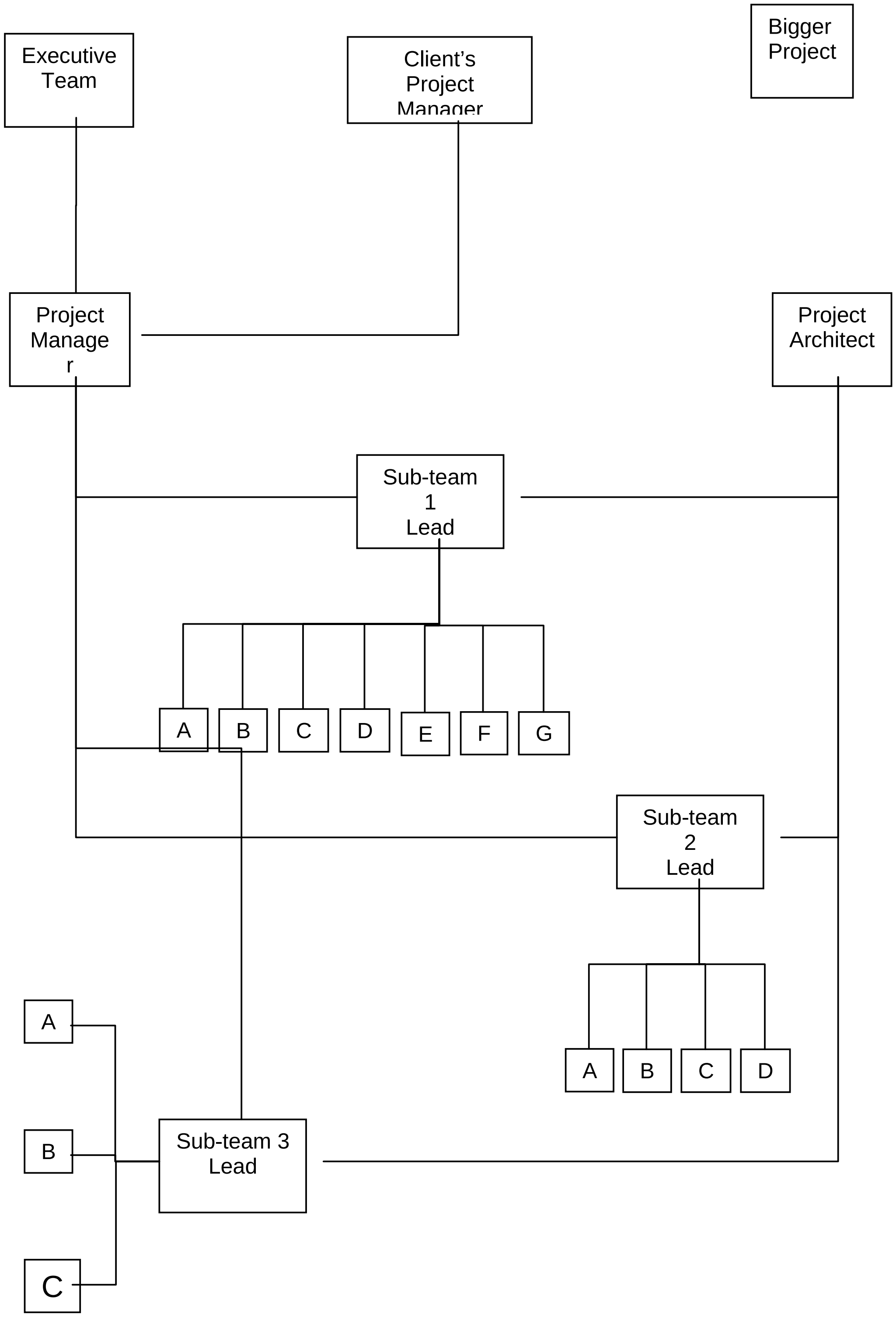
- **Responsibilities:**
  - Planning and tracking of the project
  - Detailed design
  - Professional development of team members
  - In case of small teams, development activities
- **Profile:**
  - 5+ years of development experience
  - Excellent interpersonal skills
  - Good planning skills
  - Good design skills

Developer

- **Responsibilities:**
  - Module-level design
  - Coding
  - Unit-testing
- **Profile:**
  - Technical BS

Executive Team

- CEO – Chief Executive Officer
  - Developer of the vision of the organization
  - Great PR skills
  - Great knack for spotting talent
- COO – Chief Operating Officer
  - Responsible for the day-to-day operations
  - Great organizational & interpersonal skills
- CMSO – Chief Marketing & Sales Officer
  - Responsible for bringing in work
  - Innovative


Executive Team
Client's Project Manager
Bigger Project
Project Manage r
Project Architect
Sub-team 1 Lead
A
B
C
D
E
F
G
Sub-team 2 Lead
A
B
C
D
A
B
C
Sub-team 3 Lead

Business Development Team

- 1-2 members
- Responsible for the development of detailed proposals for projects
- Profile of Members:
  - Combination of technical and business expertise
  - Good oral & written communication skills
  - Combination of technical & business degrees

Architecture Team

- 2-3 members
- Consists of the sharpest technical minds in the company

Configuration Management Team

- 2-3 members
- Keeps a vigilant eye on the process that keeps an extensive record of all versions of everything that is ever developed for a particular project: from proposals to specifications to plans to design to code

Process Team

- 1-2 members
- Team's goal: To continuously improve the SW development process to achieve improvements in cost, schedule, and quality
- Continuously monitors how SW is developed in the organization
- Encourages and assists all teams and team-members in improving their part in the SW development process

Quality Assurance Team

- Around 20 members
- Responsible for assuring the quality of all SW (i.e. making sure that it does what it is supposed to) that is produced at the organization
- Nothing goes to the customer without the approval of the QA team

Technology Transfer Team

- The size of this team varies with the amount of work at the organization – when the times are good, this team is quite small
- This team is responsible for:
  - Evaluating new technologies, products, processes
  - Selecting the ones that are right for the organization
  - Developing an expertise in their use
  - Introducing them in various ongoing/future projects

Support Team

- 2-3 members
- Members possess expertise in both HW & SW
- Responsible for the maintenance, expansion, improvement of the infrastructure consisting of:
  - Workstations, servers, printers
  - Networking equipment (router, switch, hub)
  - SW (OS, development SW, productivity SW, etc.)
  - Network security

That brings us to the end of our discussion on various roles and the associated responsibilities in the computing profession. Now we move on to another topic related to our profession, Ethics!

Ethics

- Ethics is a collection of heuristics that, when followed, improves our way of life
- I find them wonderful as they simplify my life
- For example, if you believe in the heuristic *always tell the truth,* your life becomes much simpler
- Now, you don't have to think before you make every statement that you make *"Shall I tell the truth, or lie?"*

Professional Ethics

- Professional ethics are a category of ethics, and here we discus the professional ethics relevant to computing
- Awareness of professional ethics is gaining importance with time as the decision-making process in the work place keeps on increasing in complexity
- The professional ethics provide a way of simplifying that decision making process

Let us now discuss a few situations where I will request you for your ethical opinions

Situation 1: Illegal Use

- A person is using a piece of SW without the author's permission and says: "I'm not really using it, I'm just evaluating it before I make a firm decision on buying"
- That person is "evaluating" that piece of SW for 13 months now!
- Is the conduct of that person ethical?

Situation 2: Vaporware

- A small company announces a new SW product
- A larger, more established competitor hears about that product, and starts a whispering campaign that she is also working on a similar product that will be released soon
- Potential customers decide to wait for the product instead of making the more riskier purchase from the smaller company
- The new company's sales become sluggish, and it fails to earn back the investment that it has put into developing that new product. That results in her closure
- The larger company never releases the promised product
- Is the conduct of that large company unethical or a reasonable business tactic?

Situation 3: Whistle Blower

- SW bugs, at times, have catastrophic consequences
- While Bhola sahib was working for a contractor at NASA, he found such a bug and reported it to his boss, Murphy sahib, who ordered him to never mention it to any one, or he will get fired
- Bhola sahib got scared, and did as he was told
- Did Bhola sahib's behave in an ethical manner? Would you hire him in your company?

Situation 4: Trade Secrets

- Bhola sahib was working at BholiSoft
- He leaves it to work for a competitor, SuperSoft
- Even before starting at SuperSoft, he already has divulged many of the trade secrets of BholiSoft during his interviews at SuperSoft, giving them an advantage over BholiSoft
- Do you agree with Bhola Sahib's ethics? Would you hire him in your company?

Today's Lecture:

(The Computing Profession)

- We discussed several roles and associated responsibilities of modern computer professionals
- We also discussed a few tricky situations where a knowledge of professional ethics would have helped

Next Lecture' Goals:

(The Future of Computing)

- To visualize the advances in computing that will take place in the future
- To visualize the impact of computing on our future

Lecture 43
The Future of Computing

Focus of the last Lecture was on the Computing Profession

- We discussed several roles and associated responsibilities of modern computer professionals
- We also discussed a few tricky situations where a knowledge of professional ethics would have helped

Who is a computing professional?

- Computer scientists, software engineers, computer engineers, and some of the telecom engineers are generally classified as computing professionals

Today's Focus Group

- Due to the limitation on time, today we will be focusing only on a subset of computing professionals: those involved in the development of SW

Organization: A Collection of Teams

Development Team

- Responsible for a project from after the specifications stage till the very end

Project Manager

- Responsibilities:
– Planning and tracking of the project
– Arranging of the appropriate resources
– Client relationship management

Architect

- Responsibilities:
– Technical guru of the project
– Technology selection
– High-level design
– Makes certain that the implementation remains true to the design

Team Lead

- Responsibilities:
– Planning and tracking of the project
– Detailed design
– Professional development of team members
– Development activities, in case of small teams

Developer

- Responsibilities:
– Module-level design
– Coding
– Unit-testing

Executive Team

- CEO – Chief Executive Officer
– Developer of the vision of the organization
– Great PR skills
– Great knack for spotting talent
- COO – Chief Operating Officer
– Responsible for the day-to-day operations
– Great planning & interpersonal skills
- CMSO – Chief Marketing & Sales Officer
– Responsible for bringing in work
– Innovative

Business Development Team

- Responsible for the development of detailed proposals for projects

Architecture Team

- **Consists of the sharpest technical minds in the company**

Configuration Management Team

- Keeps a vigilant eye on the process that keeps an extensive record of all versions of everything that is ever developed for a particular project: from proposals to specifications to plans to design to code

Process Team

- Team's goal: To continuously improve the SW development process to achieve improvements in cost, schedule, and quality

Quality Assurance Team

- Responsible for assuring the quality of all SW (i.e. making sure that it does what it is supposed to) that is produced at the organization

Technology Transfer Team

- This team is responsible for:

– Evaluating new technologies, products, processes
– Selecting the ones that are right for the organization
– Developing an expertise in their use
–I ntroducing them in various ongoing/future projects

Support Team

- Responsible for the maintenance, expansion, improvement of the infrastructure consisting of:

– Workstations, servers, printers
– Networking equipment (router, switch, hub)
– SW (development SW, productivity SW, etc.)
– Network security

Ethics

- Ethics is a collection of heuristics that, when followed, improves our way of life
- I find them wonderful as they simplify my life

Professional Ethics

- Professional ethics are a category of ethics, and here we discus the professional ethics relevant to computing
- The professional ethics provide a way of simplifying our decision making process
- We also looked at few situations where a knowledge of professional ethics would have simplified the process of decision making

Today's Goals:

(The Future Of Computing)

- To visualize the advances in computing that will take place in the future
- To visualize the impact of computing on our future

Computing & Telecommunications

- The two fields are converging, and, as time passes, are becoming more and more indistinguishable from each other
- Therefore, when we talk about the future of one, we must talk about the future of both

My Personal Mobile Communicator

- Probably 15-20 years from now …
- Body-embedded uPhone, head-phone, processor
- Voice-only control interface
- The user mumbles, only the uPhone hears
- Longer term: the user only thinks commands and speech and feels the response from the other end; no actual sounds are made

Required Computing Technologies

- Miniature uPs, RAM, ROM
- Continuous speech recognition

A Mobile Video-Phone

- The technology is available; the infrastructure will become common in 2-5 years time

- However, it may never become as popular as a regular voice-only phone as it will be too cumbersome to use

My Personal Agent

- A computer program that will works autonomously and will have a voice-interface
- It may start becoming popular in 5-year's time
- Will be able to have an intelligent conversation with me
- I say: “I need two tickets for Hong Kong for tomorrow morning”
- It already knows where I am, what my airline preferences are, and what are my credit-card details
- It will asks me about the return leg of the journey and about hotel reservations as well
- I say: “I'm depressed” or “I'm tired” or “I'm bored” or “I'm angry” or “I'm feeling sick” and it will know how to respond to situations like that
- For example, when I feel sick, it can ask simple questions and then set-up an appointment with an appropriate doctor by getting in touch with that doctor's personnel agent
- It can remind me about various events
- It keeps me updated with news, weather, stock market, etc
- I communicate with it through my personal communicator as well as my computer
- It knows that when I am near a computer display, and writes appropriate info to the screen

Required Computing Technologies

- Continuous speech recognition
- Intelligent, autonomous decision making SW

The key weakness of the Web?

- The Web (as it currently exists) was designed for humans to read, not for computers to understand and manipulate meaningfully
- Computers face great problems in dealing with the current text- and graphics-based content of the Web

Future of the Web: Semantic Web

Whereas, today's Web's content is designed for humans to read; the Semantic Web's content will be designed for computers to understand meaningfully. However, the Semantic Web is not a replacement but an extension of the present Web, in which info is given well defined meaning

Smaller, Faster, Cheaper, More Efficient

- All types of computers are becoming more powerful, smaller in size, consume less energy, and cost less than before
- 10 years from now we may have the power of today's most powerful supercomputer in a package no bigger than a common brick
- However, the size may not be relevant as a time is coming when computing power will be like the electric power service that we use at home or office

Electrical Supply

- We do not buy a new electric plant when we install a new air conditioner. Do we?
- We just plug it in, and it works!
- As we use more and more electricity, our monthly bills go up, but importantly, we are not required to do much else – same will be true for computing power on 10 year horizon

Computing Power that Flows

- These days when our computing needs exceed our resources, we buy a new computer
- In the future, if we suddenly start doing huge data mining tasks instead of just doing simple accounting on our computer …
- … we will start using more computing power, but our computer (or console, or terminal) will stay the same, only our monthly “computing usage” bill will change

On-Demand Computing Power

- Almost infinite "computing power" supply
- Reliable, maintenance-free, just like the electricity, telephone, or water-supply service
- You pay for only what you use!
- Same will be true for storage

Computer Terminals of the Future

- The first question is, would we have any?
- It may happen that computers will be everywhere, but hidden!
- If that does not happen, then they will consist of a display only; input will be through voice-commands and/or touch on the screen only (5-10 years from now)
- In a more distant future, just through thinking!

Displays

- The demise of the CRT monitors has already started, and their replacement with LCD is gaining pace with steadily dropping prices
- Like CRT monitors, LCDs are 2-D displays
- Display of the future, however, will have to be a 3-D one
- The 3-D illusion will be created through goggles or by directly writing the images on our retinas

Storage

- Magnetic storage on disks will stay with us for a long time, at least two decades
- Data densities will improve steadily with time
- Optical disks will keep on getting better (currently the BlueRay DVD can store more than 50GB)
- However, the mechanical nature of these technologies will not be able to keep up with the speed of the computers of the future and the enormous capacity requirements of the future
- Semiconductor memory will keep on becoming faster, denser, cheaper but will never have the capacity/price ratio of the magnetic/optical disks
- Disks too slow! RAM too expensive! What then?
- Some never-heard-before technology or possibly, holographic optical storage

Holographic Storage

- Digital data stored in and read from a 3-D optical material with the help of lasers
- Depending upon the material, they could be read-only or R/W
- The data density (quantity of data stored per unit volume) will be millions of times more than anything available today
- The concept has been validated but commercial applications are at least 10 years away

Data Transmission

- Our homes and offices will be hooked up to the Internet through optical fiber or a free-space optical connection
- However, most devices within the house and office will be connected to each other and the main Internet connection through wireless connections
- The bit-rates will be enormous by today's standards

The Fully Connected House

- Wall to wall computers, but hidden!
- Kitchens will be full of them
- Air conditioners, lights, security alarms, entertainment and communication systems
- Each house will have 100's of uPs, all talking to many others through wireless links, always trying to make us more comfortable while conserving energy and other resources

Telepresence

- Being there, without physically being there!
- Two remotely located people, with the help of special equipment, immersed in a simulated 3-D environment where they interact like they are sitting next to each other
- Chatting, a telephone conversation, or even a videoconference are examples, but in a degraded sense of the idea

- Full-fledged 3-D telepresence may become common 10-15 years from now

Immortal Minds

- Some day it will be possible to load all the lectures, papers, books and SW produced by an expert into an intelligent system
- After that system processes, indexes and restructures the info in those artifacts, it will be possible to have a conversation in plain English (or some other language) with that system
- The system will have that conversation based on the ideas and beliefs contained in the stored info and in the style of that expert
- Initially this may happen in the form of text, then speech, and then a talking head on a computer screen, and finally in a 3-D simulated reality environment

Translators

- Natural language translators: One of the most biggest challenges for today's computer scientist
- Fully automated and reasonably accurate translators (say from English to German) do not exist as of now …
- … but will, perhaps, 20 years from now

Education

- 15-20 years from now, all education will be computer-based but will not be impersonal as the computer-based education of today
- The group-method, the basis of today's learning, will continue to be used, but without the requirement of physical presence of the teacher and the taught in the same room
- Simulated-reality techniques will be used to create an artificial but effective educational environment

Medicine

- Due to computer-assisted research into medicine and genetic engineering, most, if not all of the current diseases will be eliminated over the next 50 years
- Probably a few deadly ones will be created accidentally as part of that research or by the germ-warfare labs

Warfare

- Goal of war: Disable the enemy
- That can be achieved by killing off the communication systems of the enemy army
- Why use nasty and expensive things like nuclear bombs for that? Why not a computer virus?
- Countries (and certain groups) will focus more and more of their resources on this area, where they can maximize damage even with meager resources
- The key target will be the telecom infrastructure and the financial systems (stock markets, banking systems)
- The developed countries are more vulnerable because they rely more on the targeted systems

Entertainment

- Movies in which animated characters will be indistinguishable from humans actors have started to appear
- 15 years from now human actors may become extinct
- Not too long after that, movies as an entertainment form may become extinct as well
- Movies will become interactive like video games
- Video games will become more realistic like movies
- And then they will converge into a single form of entertainment, probably called *movie-game*
- Players will be able to become a part of the movie along with other players, if they wish!

Crime

- I believe that petty crime will disappear in 50 years time due to computer-based methods for investigating and tracking criminals
- And after that, all crime will be computer-assisted and on a very large scale
- It will most probably be committed by nation-states, not individuals

No Personal Privacy

- 50 years from now, due to the low cost of efficient sensors and highly-powerful computers, tracking of humans will become quite easy
- All governments will start keeping track of every move of every individual in the name of peace and security

Slave → Master

- The way things are progressing right now, the roles may reverse over a 50-100 year time frame
- Computers may become self-replicating, self-healing, and self-programming just like humans
- And one fine day they may do a conference through the Internet and just may decide that enough is enough. Slavery, no more!

Today's Lecture:

(The Future of Computing)

- We tried to visualize the advances in computing that will take place in the future
- We also tried to visualize the impact of computing on our future

Next Lecture' Goals:

(Programming Methodology)

- To understand effective programming practices that result in the development of correct programs with a minimum effort
- To become familiar with simple debugging techniques

Lecture 44
Programming Methodology
(Web Development Lecture 15)

During the last lecture we discussed Graphics & Animation

- We became able to add and manipulate images and simple animations to a Web page

Images in HTML

- It is quite straight forward to include gif and jpg images in an html Web page using the <IMG> tag
- Format: <IMG src=*URL,* alt=*text*
  height=*pixels* width=*pixels*
  align="*bottom|middle|top*">
- Plea: Don't use images just for the sake of it!

Images in JavaScript

- Images in JavaScript can be manipulated in many ways using the built-in object, Image
- Properties: name, border, complete, height, width, hspace, vspace, lowsrc, src
- Methods: None
- Event handlers: onAbort, onError, onLoad, etc.

Image Preloading

- The primary use for an Image object is to download an image into the cache before it is actually needed for display
- This technique can be used to create smooth animations or to display one of several images based on the requirement

The Image Pre-Loading Process

1. An instance of the Image object is created using the *new* keyword
2. The *src* property of this instance is set equal to the *filename* of the image to be pre-loaded
3. That step starts the down-loading of the image into the cache without actually displaying it
4. When a pre-loaded image is required to be displayed, the *src* property of the displayed image is set to the *src* property of the pre-fetched image

Animated Gifs

- We could have saved the 16 gif images of the previous example in a single file in the form of an animated gif, and then used it in a regular <IMG> tag to display a moving image
- However, JavaScript provides better control over the sequencing and the gap between the individual images
- Example

Today's Goals

(Programming Methodology)

- To understand effective programming practices that result in the development of correct programs with minimum effort
- To become familiar with testing & debugging

programming methodology?

The process used by an individual or a team for developing programs

Good programming methodology?

A methodology that enables the lowest-cost and on-schedule development of programs that are correct, easy to maintain & enhance

correct program?

A program with correct syntax & semantics

readable program?

A program that is easy to read & understand, and therefore, easy to maintain & enhance

Bubble Sort

How can we make it more readable? What is its most complex aspect?

```
swapFlag = true ;
while ( swapFlag == true ) {
        swapFlag = false ;
        for ( k = 0 ; k < ht.length - 1 ; k++ ) {
                if ( ht[ k ] < ht[ k + 1 ] ) {
                        temp = ht[ k + 1 ] ;
                        ht[ k + 1 ] = ht[ k ] ;
                        ht[ k ] = temp ;
                        swapFlag = true ;
                }       }       }


for ( j = 0 ; j < 100000 ; j++ ) {

        for ( k = 0 ; k < ht.length - 1 ; k++ ) {
                if ( ht[ k ] < ht[ k + 1 ] ) {
                        temp = ht[ k + 1 ] ;
                        ht[ k + 1 ] = ht[ k ] ;
                        ht[ k ] = temp ;
                        }
        }
}
```

44.1 Design Guidelines

- Break your code down into short and simple functions (e.g. take the 3 swap statements out from the last example and put them into a function of their own)
- Do not use global variables

44.2 Coding Guidelines

- Always use semicolons to end statements
- Indent blocks of code (2 to 5 spaces)
- Identifiers:

– Use the camelBack scheme
– Make them descriptive but concise
– Variables: nouns
– Functions: verbs

- Comment liberally

44.3 Guidelines for Developing Short Programs

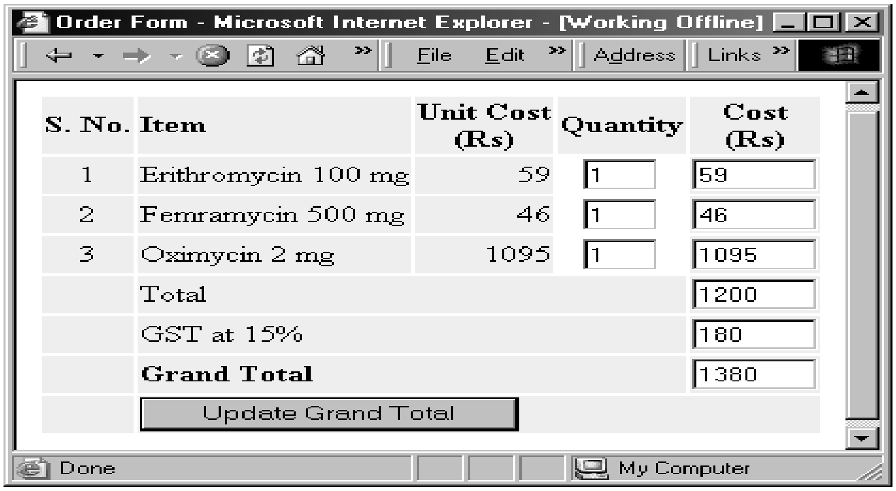
- Read, understand the problem
- Do you have all the required data?
  No: Get it
  Else assume it. State it explicitly

Example: Problem Statement

- Develop a Web page that displays an order taking form
- It takes the number of items required for each product, multiplies with the prices, sums them up, adds the GST, and displays the total value of the order

Guidelines for Developing Short Programs

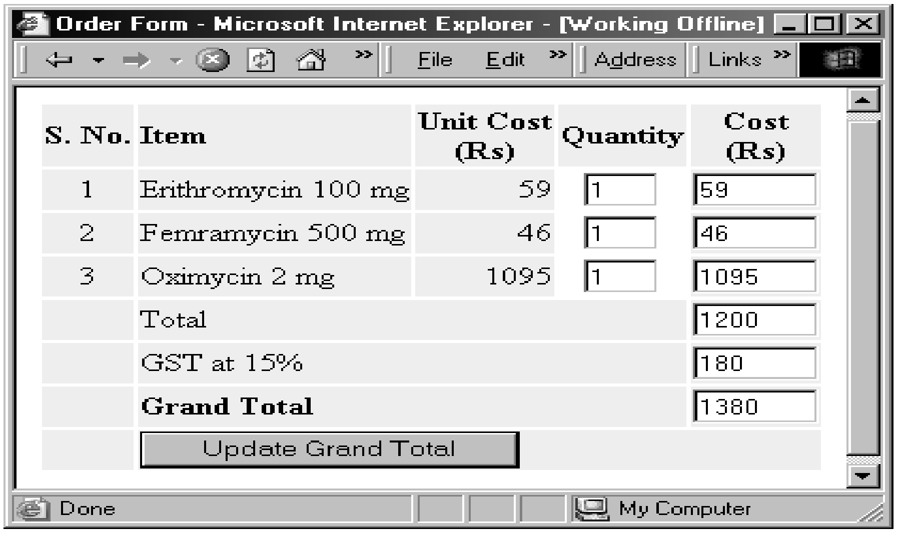
- Read, understand the problem
- Do you have all the required data?
  No: Get it
  Else assume it. State it explicitly
- Do the design


Order Form - Microsoft Internet Explorer - [Working Offline]
File
Edit
Address
Links
Done
My Computer

S. No.	Item	Unit Cost (Rs)	Quantity	Cost (Rs)
1	Erithromycin 100 mg	59	1	59
2	Femramycin 500 mg	46	1	46
3	Oximycin 2 mg	1095	1	1095
	Total			1200
	GST at 15%			180
	Grand Total			1380
	Update Grand Total			

Developing Short Programs

- Read, understand the problem
- Do you have all the required data?
  No: Get it
  Else assume it. State it explicitly
- Do the design
- Write test cases


Order Form - Microsoft Internet Explorer - [Working Offline]
File
Edit
Address
Links
Update Grand Total
Done
My Computer

S. No.	Item	Unit Cost (Rs)	Quantity	Cost (Rs)
1	Erithromycin 100 mg	59	1	59
2	Femramycin 500 mg	46	1	46
3	Oximycin 2 mg	1095	1	1095
	Total			1200
	GST at 15%			180
	Grand Total			1380

Developing Short Programs

- Read, understand the problem
- Do you have all the required data?
  No: Get it
  Else assume it. State it explicitly
- Do the design
- Write test cases

- Write the code on a piece of paper
- Hand-check it
- Type it in
- Run & check it on test cases
- Errors? fix & redo 9
  Done!

44.4 Design & Code Reviews

- Probably the most efficient way of improving a program
- Being humans, at time we see what is supposed to be there instead of what is actually there
- Another pair of eyeballs may not have the same problem, especially if they were not involved in building the design or code

Two Popular Review Methods

1. Give the problem statement, design, and code (that includes all assumptions) to a peer, and ask him/her to see if things have been done properly
2. Walk a peer or a group of peers through the problem, the design, and the code yourself
3. Which of the two is better?

44.5 Testing & Debugging

- *Testing:* The tasks performed to determine the existence of defects
- *Debugging:* The tasks performed to detect the exact location of defects
- Defects are also called bugs or errors
- Let us now look at one of their classifications

Types of Errors

- Syntax errors
- Semantic errors
- Run-time errors

Syntax Errors

- They are caused by the code that somehow violates the rules of the language
- Easy to detect and fix errors
- The browser stops code interpretation on detecting one of these

Examples:

–a = b + * c ;

–receiver = reciever + 2

Syntax error?

Semantic Errors

- Occur when a statement executes and has an effect not intended by the programmer
- Hard to detect during normal testing
- Often times occur only in unusual & infrequent circumstances
- The '+' operator often results in unintended consequences. Remedy: Convert, before use

Run-Time Errors

- Occur when the program is running and tries to do something that is against the rules

Example: Accessing a non-existent variable, property, method, object, etc (e.g. a method name is misspelled)

- Sources of these can be determined by a careful reading of the code, but unfortunately, not always!

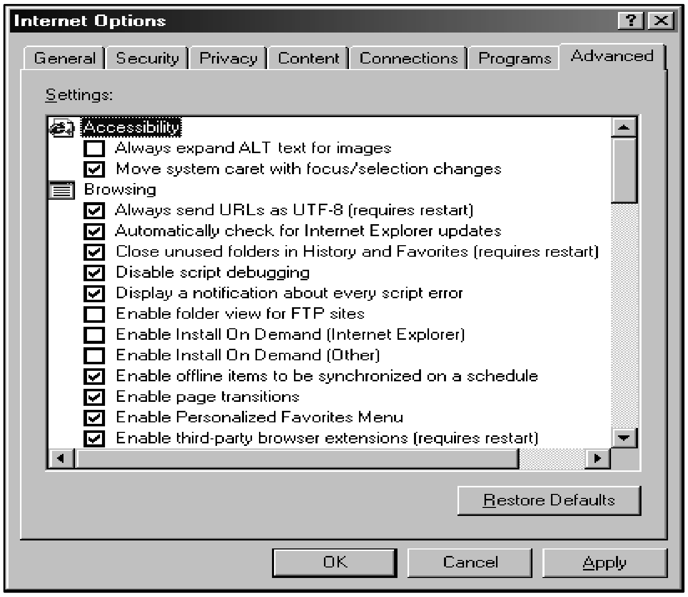
Debugging

Tools:

Internet Options…:

Advanced:

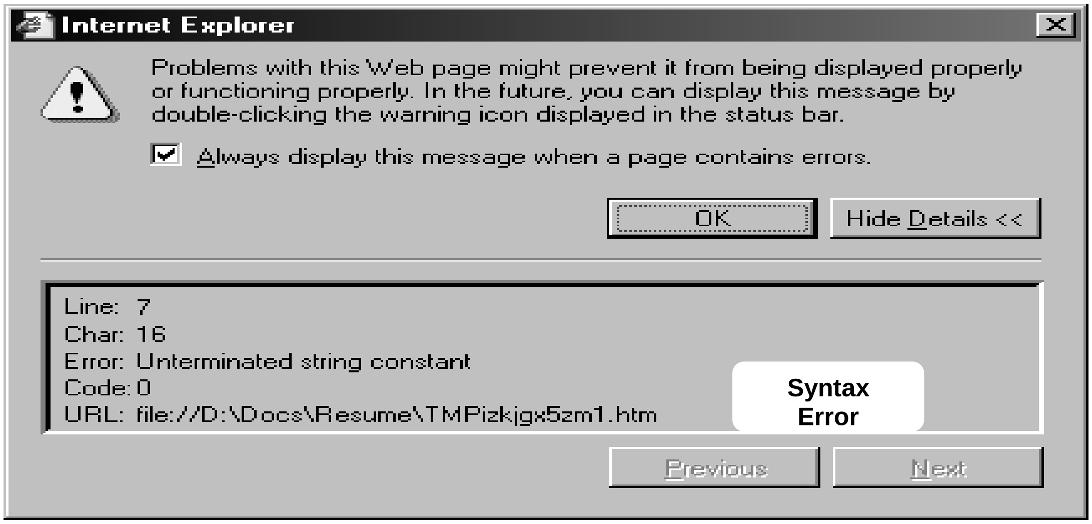
```
name = "Bhola ;
```



Settings:

Accessibility
- ☐ Always expand ALT text for images
- ☑ Move system caret with focus/selection changes

Browsing
- ☑ Always send URLs as UTF-8 (requires restart)
- ☑ Automatically check for Internet Explorer updates
- ☑ Close unused folders in History and Favorites (requires restart)
- ☑ Disable script debugging
- ☑ Display a notification about every script error
- ☐ Enable folder view for FTP sites
- ☐ Enable Install On Demand (Internet Explorer)
- ☐ Enable Install On Demand (Other)
- ☑ Enable offline items to be synchronized on a schedule
- ☑ Enable page transitions
- ☑ Enable Personalized Favorites Menu
- ☑ Enable third-party browser extensions (requires restart)



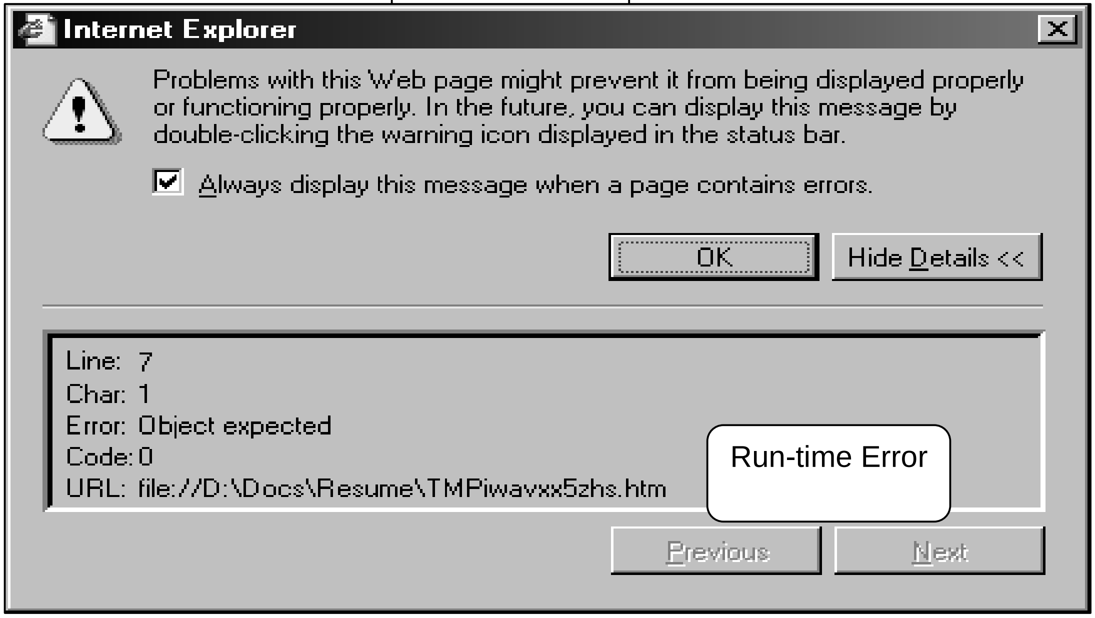
Problems with this Web page might prevent it from being displayed properly or functioning properly. In the future, you can display this message by double-clicking the warning icon displayed in the status bar.

☑ Always display this message when a page contains errors.

```
Line:  7
Char:  16
Error:  Unterminated string constant
Code: 0
URL:  file://D:\Docs\Resume\TMPizkjgx5zm1.htm
```

Syntax Error

```
checkPulse( ) ;
```



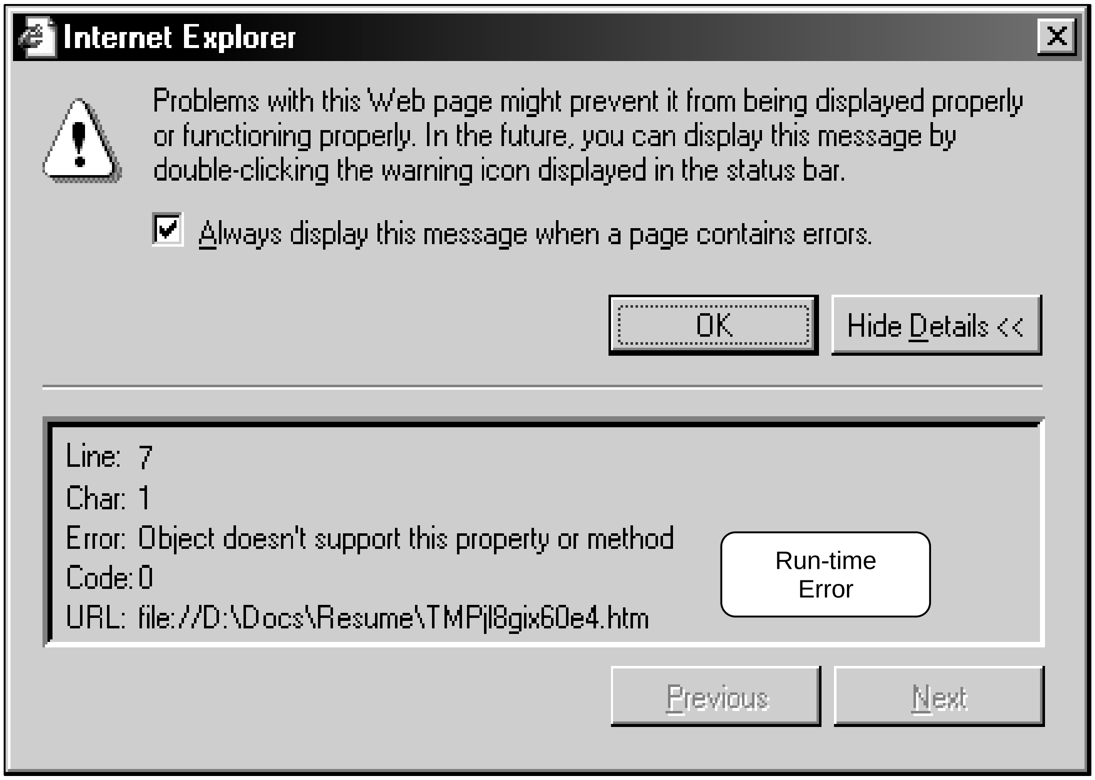
Problems with this Web page might prevent it from being displayed properly or functioning properly. In the future, you can display this message by double-clicking the warning icon displayed in the status bar.

☑ Always display this message when a page contains errors.

```
Line:  7
Char:  1
Error:  Object expected
Code: 0
URL:  file://D:\Docs\Resume\TMPiwavxx5zhs.htm
```

Run-time Error

```
x = 1.3 ;
x.upperCase( ) ;
```



Problems with this Web page might prevent it from being displayed properly or functioning properly. In the future, you can display this message by double-clicking the warning icon displayed in the status bar.

☑ Always display this message when a page contains errors.

Line: 7
Char: 1
Error: Object doesn't support this property or method
Code: 0
URL: file://D:\Docs\Resume\TMPjl8gix60e4.htm

Run-time Error

```
income = document.myForm.salary.value +
         document.myForm.bonus.value ;
```

Semantic Error

Common Mistakes

```
if ( today = “holiday” )
mood = “good” ;
```

```
function doThis ( tiger ) {      box[ 0 ] = tiger ;
      x = box[ 0 ] ;
      return x ;
```

```
box = new array( 10 ) ;
```

```
box = new Array( 10 ) ;
box( 0 ) = 43 ;
```

44.6 Helpful Editors

- Using smart editors (e.g. DreamWeaver, nedit) can help in avoiding many types of syntax errors
- They can, for example:

– Automatically color different parts of statements in different colors, e.g. comments in Gray, strings in Green, HTML tags in Blue

– Auto indent

– Visually indicate the presence of mismatched parentheses, curly braces or square brackets

During Today's Lecture …

- We looked at a few effective programming practices that result in the development of correct programs with minimum effort
- We also became familiar with testing & debugging

Final Lecture:

Review & Wrap-Up

- To review a selection from the interesting ideas that we explored over the last 44 lectures

Lecture 45
Review & Wrap-Up

During the last lecture we discussed Programming Methodology

- We looked at a few effective programming practices that result in the development of correct programs with minimum effort
- We also became familiar with testing & debugging

readable
program?

A program that is easy to read & understand, and therefore, easy to maintain & enhance

Design Guidelines

- Break your code down into short and simple functions (e.g. take the 3 swap statements out from the last example and put them into a function of their own)
- Do not use global variables

Coding Guidelines

- Indent blocks of code (2 to 5 spaces)
- Always use semicolons to end statements
- Identifiers:
  – Use the camelBack scheme
  – Make them descriptive but concise
  – Variables: nouns
  – Functions: verbs
- Comment liberally

Guidelines for Developing Short Programs

- Read, understand the problem
- Do you have all the required data?
  No: Get it
  Else assume it. State it explicitly
- Do the design
- Write test cases

- Write the code on a piece of paper
- Hand-check it
- Type it in
- Run & check it on test cases
- Errors? fix & redo 9
- Done!

Design & Code Reviews

- Probably the most efficient way of improving the a program
- Being humans, at time we see what is supposed to be there instead of what is actually there
- Another pair of eyeballs may not have the same problem, especially if they are were not involved in building the design or code

Testing & Debugging

- *Testing:* The tasks performed to determine the existence of defects
- *Debugging:* The tasks performed to detect the exact location of defects
- Defects are also called bugs or errors
- Let us now look at one of their classifications

Types of Errors

- Syntax errors

- Semantic errors
- Run-time errors

Today's Goal:

(Review & Wrap-Up)

- To review some of the interesting ideas that we discussed over the last 44 lectures
- Please note that this lectures is not a comprehensive review, just a sampler!

Course Objectives

1. To build an *appreciation* for the fundamental concepts in computing

2. To achieve a *beginners proficiency* in Web page development

3. To become *familiar* with popular PC productivity software

Progression of Computer Technology

1. Mechanical computing
2. Electro-mechanical
3. Vacuum tube
4. Transistor
   (the current state-of the-art)
5. Quantum computing

Quantum Computers

- Quantum computers may one day be millions of times more efficient than the current state-of-the-art computers …
- as their quantum mechanical nature will allow them to examine all possible answers to a question, simultaneously

The World Wide Web

- A huge resource of info
- Logically unified, but physically distributed
- It is unlike any previous human invention:

– It is a world-wide resource, important to all and shared by all of the people in the world

The Semantic Web

Whereas, today's Web's content is designed for humans to read; the Semantic Web's content will be designed for computers to understand meaningfully

Internet: Network of Networks

- A large number of networks, interconnected physically
- Capable of communicating and sharing data with each other
- From the user's point view, Internet – a collection of interconnected networks – looks like a single, unified network

Language of the Internet: TCP/IP

Transmission Control Protocol/Internet Protocol

- TCP breaks down the message to be sent over the Internet into packets
- IP routes these packets through the Internet to get them to their destination
- When the packets reach the destination computer, TCP reassembles them into the original message

Instant Messaging

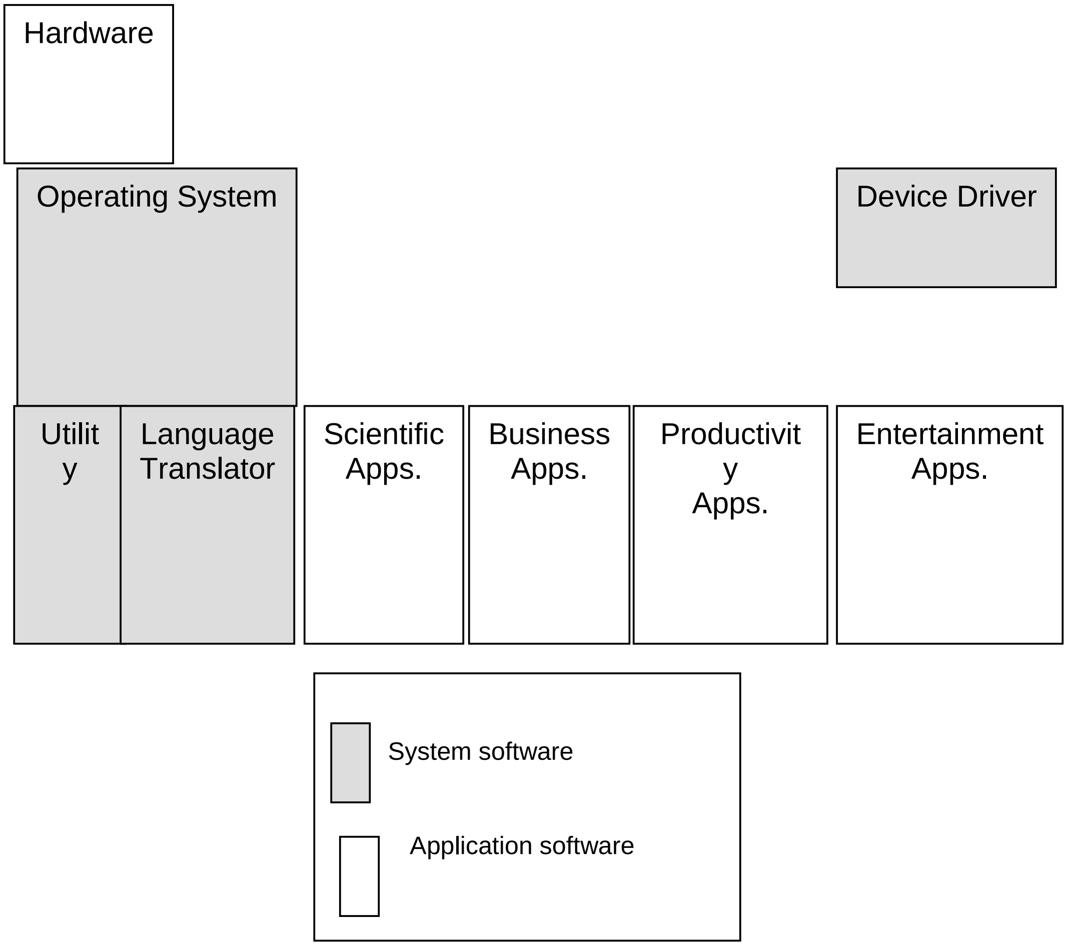
- eMail: Slow response times
- eMail: No way of knowing if the person we are sending eMail to is there to read it
- eMail: The process of having a conversation through eMail by exchanging several short messages is too cumbersome
- Instant messaging (IM) solves these problems

On-Chip Cache Memory

- That small amount of memory located on the same chip as the uP
- The uP stores a copy of frequently used data and instructions in its cache memory
- When the uP desires to look at a piece of data, it checks in the cache first. If it is not there, only then the uP gets it from the main memory
- Its proximity to the uP makes access times short

Ways of Enhancing A uP

- Increase the clock frequency
- Increase the word-width
- Add more functional units (e.g. ALU's, FPU's, Vector/SIMD units, etc.)



The Role of An OS

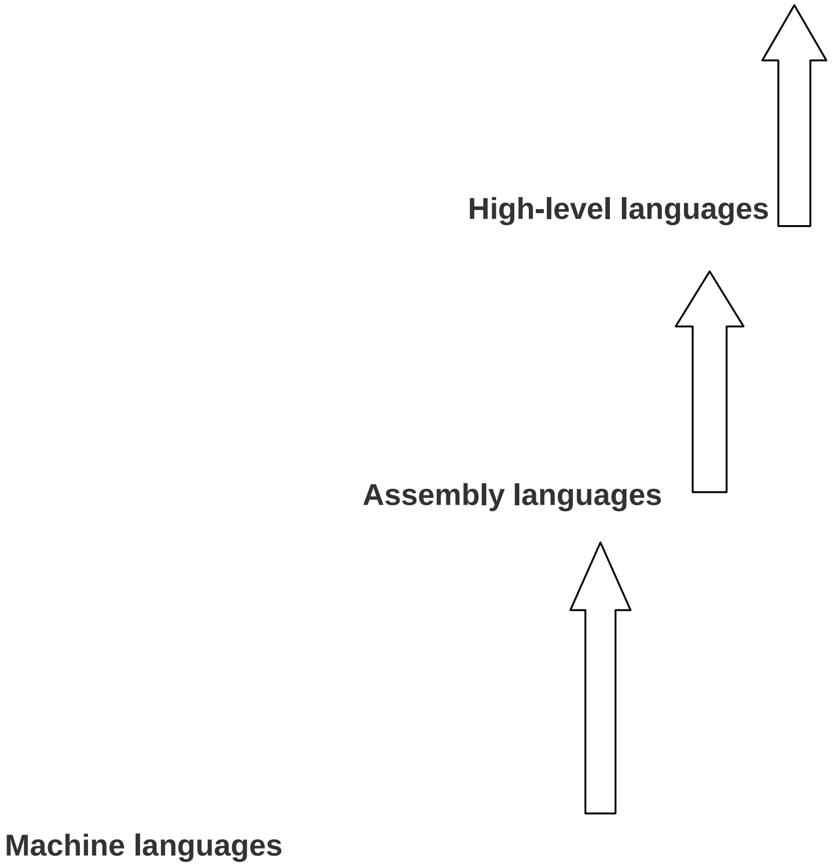
- Manages the HW and SW resources of the computer system, often invisibly. These include the processor, memory, disk drives, etc.
- Provides a simple, consistent way for applications to interact with the HW without having to know all the details of the HW

Who Owns Software?

- Generally, although a piece of SW that is being used by millions, it is not owned by any of them!

- When we buy a SW package, we do not really buy it – we just buy a license that allows us to use it, the ownership stays with the maker

4th-generation languages



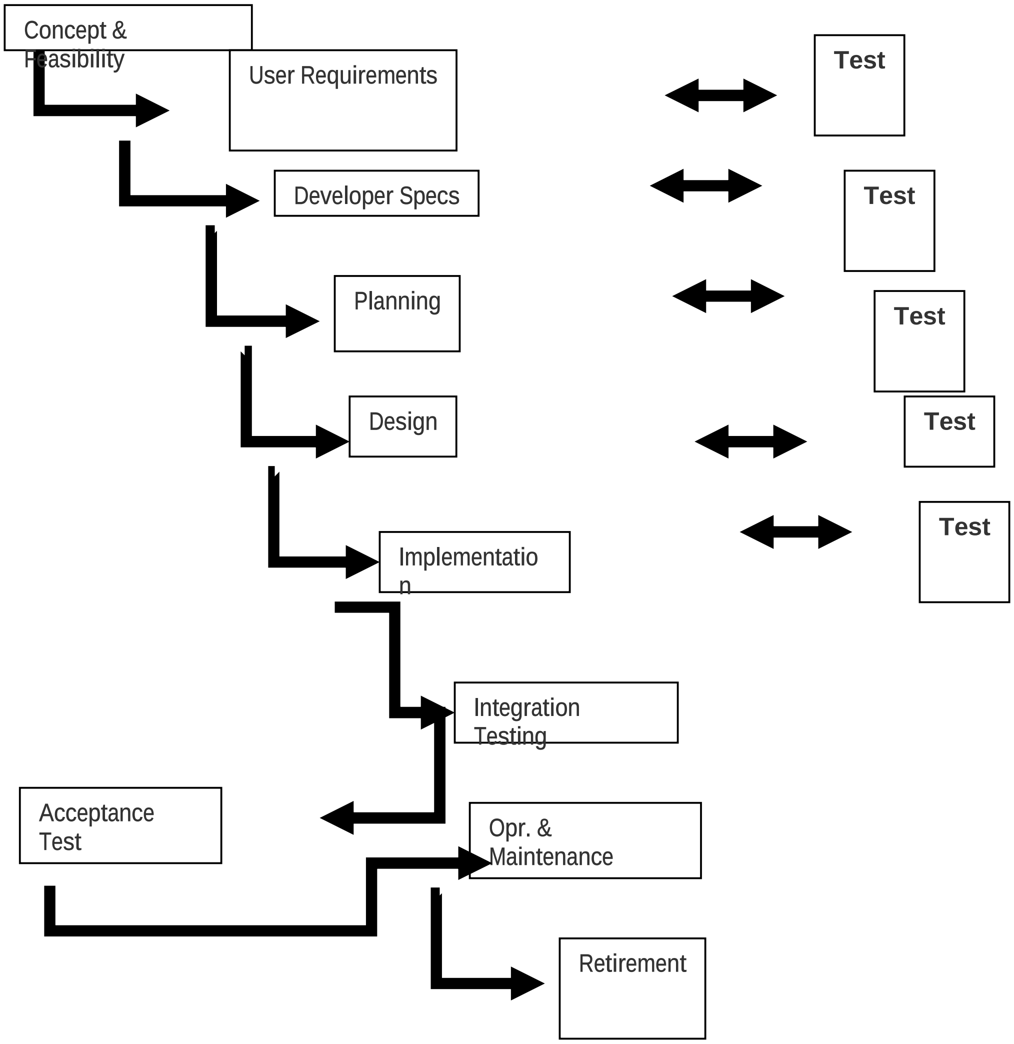
High-level languages

Assembly languages

Machine languages

Interpreters:
Immediate response, but execute code slowly

Compilers: Compiling takes time, but super-fast execution



Algorithm

1st Definition:

Sequence of steps that is taken to solve a problem

Better Definition:

A *precise sequence* of a *limited number* of *unambiguous, executable steps* that *terminates* in the form of a *solution*

Pseudo Code

- Quite suitable for SW development as it is closer in form to real code
- One can write the pseudo code, then use it as a starting point or outline for writing real code
- Many developers write the pseudo code first and then incrementally convert each line into real code

Heuristic

Common sense lesson drawn from experience

(Artificial) Intelligent Systems

SW programs or SW/HW systems designed to perform *complex* tasks employing strategies that mimic some aspect of human thought

Not a Suitable Hammer for All Nails!

if the nature of computations required in a task is not well understood

or there are too many exceptions to the rules
or known algorithms are too complex or inefficient
then artificial intelligent systems have the potential of offering an acceptable solution

Database

- A collection of data organized in such a fashion that the computer can quickly search for a desired data item
- All data items in it are generally related to each other and share a single domain

Relational Databases

- Databases consisting of two or more related tables are called *relational databases*
- A relational database stores all its data inside tables, and nowhere else
- All operations on data are done on those tables or those that are generated by table operations

Future Trends:

On-Demand Computing Power

- Almost infinite "computing power" supply
- Reliable, maintenance-free, just like the electricity, telephone, or water-supply service
- No capital expenditure; you pay for only what you use!
- Same will be true for storage

Future Trends: Immortal Minds

- Some day it will be possible to load all the lectures, papers, books and SW produced by an expert into an intelligent system
- After that system processes, indexes and restructures the info in those artifacts, it will be possible to have a conversation in plain English (or some other language) with that system

Distances Are Contracting!

Distances Are Increasing!

- Because of the ever-decreasing costs of verbal, text, video communications, it is becoming easier to stay in touch of anyone, regardless of their physical location
- Solitude is the order of the day as many children & adults spend their free time surfing, chatting, playing computer games, instead of spending it on interacting with friends or family

Computers may Become too Powerful!

- Computers keep on becoming more and more powerful, gaining more and more autonomy
- They are being equipped with fail-safe and self-healing technologies
- Are we heading towards a future where the role of the masters and the slaves will be reversed?

Why JavaScript?

- HTML is great for static Web pages; however, supports only rudimentary interactivity through forms and hyperlinks
- JavaScript can be used (along with HTML) to develop interactive content for the Web

Some of things that JavaScript *cannot* do!

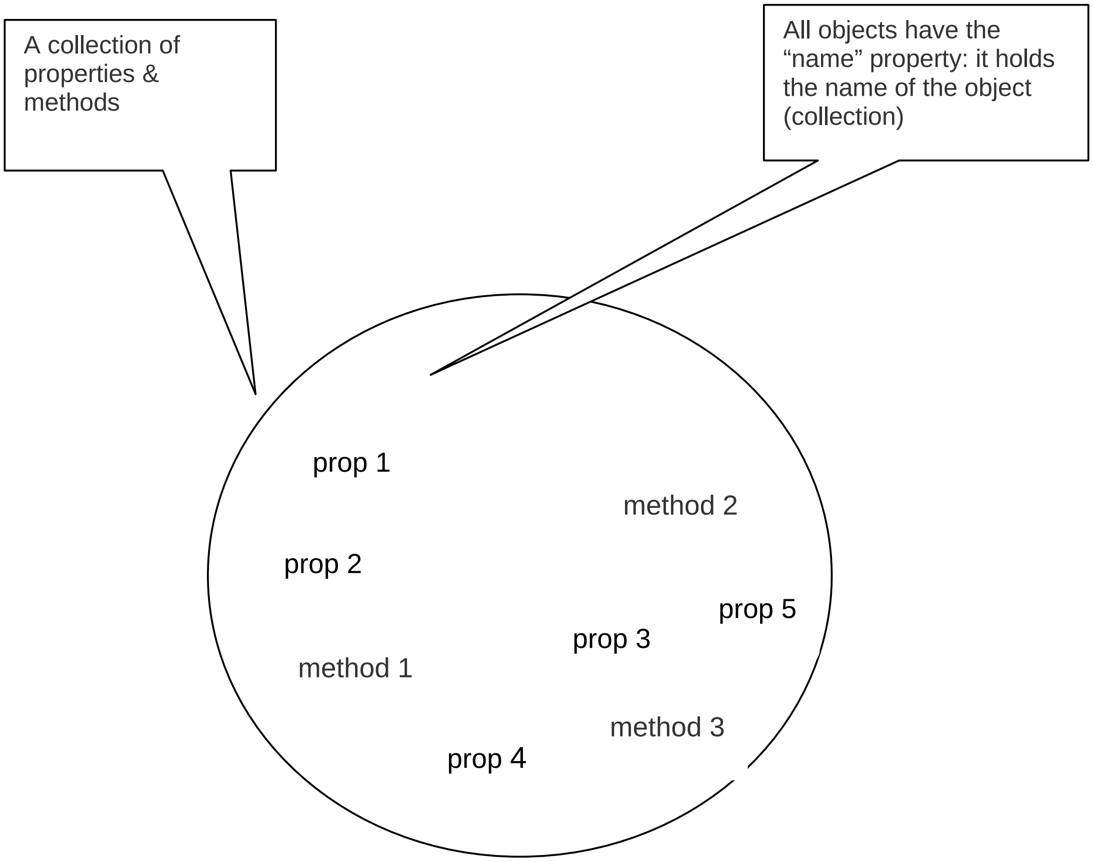
- The following file ops. on the client computer:

–Read -- Modify
–Rename -- Delete
–Create

- Create graphics (although, it does have the ability to format pages through HTML - including the placement of graphics)
- Any network programming bar one function: the ability to download a file to the browser specified through an arbitrary URL

Advantages of Client-Side Scripting

- Reduced server load as it does not have to send messages to the user's browser about missing or incorrect data
- Reduced network traffic as the form's data is sent only once instead of many to's and fro's


A collection of properties & methods
All objects have the "name" property: it holds the name of the object (collection)
prop 1
method 2
prop 2
prop 5
prop 3
method 1
method 3
prop 4

Object: A *named* collection of properties (data, state) & methods (instructions, behavior)

Functions

- A named group of statements that is put together once and then used (by reference) repeatedly on a Web page
- Code becomes easier to read, understand and maintain

Local and Global Variables

Local or Function-level Variable

Effective only in the function in which they are declared

Global Variables

Visible everywhere on the Web page

Image Preloading

- The Image object can be used to download an image into the cache before it is actually needed for display
- This technique can be used to create smooth animations or to display one of several images based on the requirement

Productivity SW

- The lectures and assignments were designed to give a brief introduction, and no more
- All we desired was for you to become able to open the package and perform some trivial tasks
- With time, you will find more and more use for these packages, and gradually develop an expertise that later will become very useful in your career

Course Objectives

1. To build an *appreciation* for the fundamental concepts in computing
2. To achieve a *beginners proficiency* in Web page development
3. To become *familiar* with popular PC productivity software

- How successful were we in helping you achieve those objectives?
- Please do let us know so that we can modify the future offerings of this course accordingly. I will be most grateful
- I have enjoyed doing this course with you very much
- Hope it was enjoyable & useful for you as well
- I thank you for your attention and especially for your eMail & discussion board messages
- A good number of those messages were quite informative and I thank you for sharing that info with me
- Until the next time when we meet …