

CS201 – Introduction to Programming
Solved Subjective Questions
From spring 2010 Final Term Papers
By *vuZs Team*

Question No: 1 (Marks: 2)

Write a declaration statement for an array of 10 elements of type float. Include an initialization statement of the first four elements to 1.0, 2.0, 3.0 and 4.0.

Answer:

```
float floatArray[10] = {1.0,2.0,3.0,4.0};
```

Question No: 2 (Marks: 2)

Write the general syntax for the declaration of pre-increment and post-increment member operator function.

Answer:

```
Classname operator ++(); ---- pre increment  
Classname operator ++(int) ---- post increment
```

Question No: 3 (Marks: 2)

Give the general syntax of class template.

Answer:

```
template  
class myclass { --- } ;
```

Question No: 4 (Marks: 2)

What is a truth Table?

Answer:

There are some areas where the decision structures become very complicated. Sometimes, we find it difficult to evaluate a complicated logical expression. Sometimes the logic becomes extremely

complicated so that even writing it as a simple syntax statement in any language. It becomes complicated to determine what will be evaluated in what way. We know the concept of truth table. The truth tables are very important. These are still a tool available for analyzing logical expressions. We will read logic design in future, which is actually to do with chips and gates. How we put these things together.

Question No: 5 (Marks: 2)

Online Study
Group

What will be the output of following code, if user input a number 123?

```
int input ;
cin >> oct >> input;
cout << hex << input ;
```

Answer:

53

Rational: it will take 123 as octal and print it in hex form which is 53.

Question No: 6 (Marks: 2)

What is principle of friendship in the context of functions and classes?

Online Study
Group

Answer:

Class can declare a friend function and someone from outside the class cannot declare itself friend of a class.

A friend function can access the private variables of class just like a member function

Question No: 7 (Marks: 2)

How many arguments a Unary Operator take? Can we make a binary operator as unary operator?

Answer:

Unary operator takes only one argument like $i++$ or $i--$ (Post increment or post decrement operators for integers) or $++i, --i$ (Pre increment or pre decrement operators for integers) ,we can not make Unary operator as binary or binary as Unary operator.

Question No: 8 (Marks: 2)

Which arithmetic operators cannot have a floating point operand?

Answer:

Modulus operator:

This operator can only be used with integer operands ONLY

Question No: 9 (Marks: 2)

What are manipulators? Give one example.

Answer:

The manipulators are like something that can be inserted into stream, effecting a change in the behavior. For example, if we have a floating point number, say pi (π), and have written it as float pi = 3.1415926 ; Now there is need of printing the value of pi up to two decimal places i.e. 3.14. This is a formatting functionality. For this, we have a manipulator that tells about width and number of decimal points of a number being printed.

Question No: 10 (Marks: 2)

Write down piece of code that will declare a matrix of 3x3. And initialize all its locations with 0;

Answer:

```
int matrix [3] [3] ;
```

```
include<iostream.h>
```

```
main () {
    int matrix [3][3];
    int inivalue = 0;

    for (int a=0;a<3;a++)
    { for (int b = 0;b<3;b++)
    { matrix[a][b]= inivalue;
    cout<<matrix[a][b]<<endl;}}
}
```

Question No: 11 (Marks: 2)

What is the difference between switch statement and if statement.

Answer:

The "If" statement is used to select among two alternatives. It uses a Boolean expression to decide which alternative should be executed. The switch statement is used to select among multiple alternatives. It uses an int expression to determine which alternative should be executed.

Question No: 12 (Marks: 2)

How can we initialize data members of contained object at construction time?

Answer:

Initializer list is used to initialize the contained objects at the construction time.

Question No: 13 (Marks: 2)

Can we overload new and delete operators?

Answer:

Yes, it is possible to overload new and delete operators to customize memory management. These operators can be overloaded in global (non-member) scope and in class scope as member operators.

Question No: 14 (Marks: 2)

Suppose there is a template function 'func' having argument of type U and return type T. What will be the C++ syntax to call this function, passing a variable 'x' of type double and returning an int type?

Answer:

```
template <class T, class U>
T func (T a, U b) {
    return (a<b?a:b);
}
calling
int i;
double x;
x = func<int,double> (j,l);
```

Question No: 15 (Marks: 2)

Which variable will be used in inner code block if we have the same names of variable at outer code block and inner code block?

Answer:

Simply: variable of the inner code is use in the inner code block.

Question No: 16 (Marks: 2)

What is the benefit of reference and where can we use it?

Answer:

In references we give the memory address of the object, due to references we pass values without making the copy. Hence, when we have many values & we want efficiency we use references to avoid copy.

Question No: 17 (Marks: 2)

Write the C++ code for the declaration of overloaded stream insertion and stream extraction operator for the object d of type Date.

Answer:

```
Date operator >> (date & d1){  
cout<<d1.day<<"-"<<d1.month<<"-"<<d1.year;  
}
```

Question No: 18 (Marks: 2)

What is difference between endl and \n?

Answer:

Endl is manipulator and it inserts new line character and flushes the stream.

\n is control character which is used to insert line break.

Question No: 19 (Marks: 2)

What does code optimization mean?

Answer:

It is process by which we make our code such a way that it improves the speed of program. By use of optimization we refine program codes in such a way that it run faster and consume less memory. We do it in such a way that output quality is not compromised.

Question No: 20 (Marks: 3)

How is the following cout statement interpreted by compiler?

cout << a << b << c ;

Answer:

It will give a compiler error because a,b,c are not declared.

Question No: 21 (Marks: 3)

Suppose an object of class A is declared as data member of class B.

(i) The constructor of which class will be called first?

Answer: A

(ii) The destructor of which class will be called first?

Answer: B

Question No: 22 (Marks: 3)

What will be the output of following functions if we call these functions three times?

```
1)
void func1(){
int x = 0;
x++;
cout << x << endl;
}
```

Answer:

```
1  
1  
1  
2)  
void func2(){  
static int x = 0 ;  
x++;  
cout << x << endl ;  
}
```

Answer:

1
2
3

Question No: 23 (Marks: 3)

If is not available in the system then what does calloc/malloc and new operator return?

Answer:

calloc/malloc and new operator return returns a null pointer to indicate that no memory is available

Question No: 24 (Marks: 3)

What is the keyword 'this' and what are the uses of 'this' pointer?

Answer:

'this' is use to refer the current class member without using the name of the class.

Question No: 25 (Marks: 3)

Which one (copy constructor or assignment operator) will be called in each of the following code segment?

- 1) Matrix m1 (m2);
- 2) Matrix m1, m2;
m1 = m2;
- 3) Matrix m1 = m2;

Answer:

- 1) Matrix m1 (m2); copy constructor
- 2) Matrix m1, m2;
m1 = m2; assignment operator
- 3) Matrix m1 = m2; assignment operator

Question No: 26 (Marks: 3)

What will be the output of following function if we call this function by passing int 5?

template T reciprocal(T x) {return (1/x); }

Answer:

0
The output will zero as $1/5$ and its .05 but conversion to int make it zero
Above is prototype of template class so assume passing an int and returning an int

Question No: 27 (Marks: 3)

Identify the errors in the following member operator function and also correct them.

```
math * operator(math m);
math * operator (math m)
{
    math temp;
    temp.number= number * number;
    return number;
}
```

Answer:

The errors are in the arguments of the member operation function and also in the body of operator member function.

Correct function should be
math *operator (math *m)

```
{
    math temp;
```



```

temp = m;
temp.number= number * number;
return temp.number;

}

```

Question No: 28 (Marks: 3)

What are the limitations of the friendship relation between classes?

Answer:

friendship relation between classes is a one way relation that is if one class declare friend another class then the another class is the friend of first class but not the first class if the friend of another class.

Question No: 29 (Marks: 3)

Define static variable. Also explain life time of static variable?

Answer:

When you declare a static variable (native data type or object) inside a function, it is created and initialized only once during the lifetime of the program.

Question No: 30 (Marks: 5)

What is difference between Unary and binary operators and how they can be overloaded?

Answer:

Unary operator takes one argument.
a ++ is an example of unary operator

Binary take two operators
+, -, * are example of binary operators
Overloaded binary operator may return any type

Here is general syntax of overloading
Return-type operator symbol (parameters);

Operator is keyword

Question No: 31 (Marks: 5)

What steps we must follow to design good program?

Answer 1:

Hence to design a program properly, we must:

- Analyze a problem statement, typically expressed as a word problem.
- Express its essence, abstractly and with examples.
- Formulate statements and comments in a precise language.
- Evaluate and revise the activities in light of checks and tests and
- Pay attention to detail.

Answer 2:

Details: we must check very details of any program. It is very important aspect of any program. We must pay complete attention to calculation.

We must give attention to logic and its flow should be smooth.

Reusable: We must write program in such a way that we can reuse them in other program. Like we define function in such a way that in future if we need any similar kind of function is requires in that case we can easily modify or reuse it.

Comments: we write the details of important steps in the form of comments. We should use comments in such a way if any body else wanted to reuse or debug or codes he can easily understand it.

Code readability: We should use Tab and spaces so codes are easily readable.

User interface: we make interface user friendly. Use polite prompts for user while take input.

Question No: 32 (Marks: 5)

Write a program which defines five variables which store the salaries of five employees, using setw and setfill manipulators to display all these salaries in a column.

Note: Display all data with in a particular width and the empty space should be filled with character x

Output should be displayed as given below:

xxxxxx1000

xxxxxx1500

xxxxxx20000

xxxxxx30000

xxxxxx60000

Answer:

```
#include <iostream.h>
#include <iomanip.h>
main(){
    int sal1 =1000;
    int sal2 =1500;
    int sal3 =20000;
    int sal4 =30000;
    int sal5 =60000;

    cout << setfill ('x') << setw (10);
    cout<< sal1<<endl;
    cout << setfill ('x') << setw (10);
    cout<< sal2<<endl;
    cout << setfill ('x') << setw (10);
    cout<< sal3<<endl;
    cout << setfill ('x') << setw (10);
    cout<< sal4<<endl;
    cout << setfill ('x') << setw (10);
    cout<< sal5<<endl;
    int i=0;
    cin>>i; // to stop the screen to show the output
}
```

Question No: 33 (Marks: 5)

Suppose we have the following class.

```
class Matrix
{
private:
```

```
int Elements[3][3];
};
```

Write the operator function of stream extraction operator (>>) for this class.

Answer:

```
Element operator >> (Element &element){
cout<<element[0][0]<<element[0][1]<<element[0][2];
cout<<element[1][0]<<element[1][1]<<element[1][2];
cout<<element[2][0]<<element[2][1]<<element[2][2];
}
```

Question No: 34 (Marks: 5)

**What is meant by user interface and class interface in C++ ?
And what role a class interfaces can play in user interface
[Marks 5]**

Answer:

Question No: 35 (Marks: 5)

Write the general syntax of a class that has one function as a friend of a class along with definition of friend function.

Answer:

```
class frinedclass{
public:
friend int compute(exforsys e1)
};
Int compute(exforsys e1)
{
//Friend Function Definition which has access to private data
return int(e1.a+e2.b)-5;
}
```

Question No: 36 (Marks: 5)

What are the advantages and disadvantages of using templates?

Answer:

Many things can be possible without using templates but it does offer several clear advantages not offered by any other techniques:

Advantages:

- Templates are easier to write than writing several versions of your similar code for different types. You create only one generic version of your class or function instead of manually creating specializations.
- Templates are type-safe. This is because the types that templates act upon are known at compile time, so the compiler can perform type checking before errors occur.
- Templates can be easier to understand, since they can provide a straightforward way of abstracting type information.
- It helps in utilizing compiler optimizations to the extreme. Then of course there is room for misuse of the templates. On one hand they provide an excellent mechanism to create specific type-safe classes from a generic definition with little overhead.

Disadvantages:

On the other hand, if misused

- Templates can make code difficult to read and follow depending upon coding style.
- They can present seriously confusing syntactical problems esp. when the code is large and spread over several header and source files.
- Then, there are times, when templates can "excellently" produce nearly meaningless compiler errors thus requiring extra care to enforce syntactical and other design constraints. A common mistake is the angle bracket problem.

Question No: 37 (Marks: 5)

Suppose a program has a math class having only one data member number.

Write the declaration and definition of operator function to overload + operator for the statements of main function.

```
math obj1, obj2;  
obj2= 10 + obj1 ;
```

Answer:

```
#include  
math  
{  
mth operator + (obj1,int x)
```

```

{
number temp;
temp=obj1.number+x;
return temp.number;
}
}

```

Question No: 38 (Marks: 5)

Write a program which defines three variables of type double which store three different values including decimal points, using setprecision manipulators to print all these values with different number of digits after the decimal number.

Answer:

```

#include
#include
main () {
double a = 12.12345;
double b = 13.123456;
double c = 14.1234567;
cout << setprecision (5) << a << endl;
cout << setprecision (2) << a << endl;
cout << setprecision (3) << a << endl;
}

```

Question No: 95 (Marks: 5)

Let we have a class,
class String

```

{
private:
char buf[25];
};

```

Write code for assignment (=) operator function which assign one String object to other object. Your code should also avoid self assignment

Answer:

```

void String::operator = ( const String &other )
{ int length ;
length = other.length();
delete buf;
}

```

```
buf = new char [length + 1];
strcpy( buf, other.buf ); }
```

Question No: 40 (Marks: 5)

Read the given below code and explain what task is being performed by this function

Matrix :: Matrix (int row , int col)

```
{
  numRows = row ;
  numCols = col ;
  elements = new ( double * ) [ numRows ] ;
  for ( int i = 0 ; i < numRows ; i ++ )
  {
    elements [ i ] = new double [ numCols ] ;
    for ( int j = 0 ; j < numCols ; j ++ )
      elements [ i ] [ j ] = 0.0 ;
  }
}
```

Hint : This function belong to a matrix class, having

Number of Rows = numRows

Number of Columns = numCols

Answer:

In the above mentioned code, first of all programmer call the constructor who have two parameters for the number of rows & columns in the matrix. Then this constructor also dynamically allocates the memory for the elements of the matrix & also initializes the value of the all elements of matrix with 0.0