CS201 Subjective:

Question: What is the use of reference data type?

Answer: A reference data type is a variable that can contain an address. The reference data types in Java are arrays, classes and interfaces. You'll hear often say that Java does not have pointers. Yet, you could consider a reference data type to be a pointer

Question: What are the main types of operators in terms of number of arguments they take?

Answer: The difference is in the number of arguments used by the function. In the case of binary operator overloading, when the function is a **member function** then the number of arguments used by the operator member function is **one**. When the function defined for the binary operator overloading is a **friend function**, then it uses **two** arguments.

Question: What is the "this" pointer? Give an example of its use

Answer: In a C++ program, if you create object A of class X, you can then obtain the address of A by using the "this" pointer. The address is available as a local variable in the non-static member functions of X, and its type is **const X*.** The "this" pointer works because C++ creates instances of its data members, and it keeps one copy of each member function.

Question: What are manipulators? Give one example.

Answer:Manipulators are operators used in C++ for formatting output. The data is manipulated by the programmer's choice of displayed.

Endl manipulator: This manipulator has the same functionality as the '\n' newline character.

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

Answer: **malloc** returns a **void pointer** to the allocated space or NULL if there is insufficient memory available. To return a pointer to a type other than void, use a type cast on the return value. The storage space pointed to by the return value is guaranteed to be suitably aligned for storage of any type of object. If size is 0, malloc allocates a zero-length item in the heap and returns a valid pointer to that item.

By default, malloc does not call the new handler routine on failure to allocate memory. You can override this default behavior so that, when malloc fails to allocate memory, malloc calls the new handler routine in the same way that the new operator does when it fails for the same reason.

Question: If we want to send the data by reference and don't want that original data should be affected then what can we do to prevent any change?

Answer: **Call by Reference** shall be used in that case.

Question: Write down the disadvantages of the templates.

Answer: The disadvantages of templates are:

• 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: The following code segment has errors. Locate as many as you can and explain briefly.
Answer:

```
class Circle // no need to enter colon here ,
{
private : //colon missing
double centerX;
double centerY;
double radius;
public: //colon missing
void setCenter(double, double);
void setRadius(int);
};//semi colon missing
```

Question (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 <iostream>
#include <iomanip>

int main ()
{
  double x1 = 12345624.72345
double x2 =  987654.12345
double x3 =  1985.23456
  cout << setprecision (3) << x1<< endl;
  cout << setprecision (4) << x2 << endl;
cout << setprecision (5) << x3<< endl;

  return 0;
}
```

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

Question (Marks: 10)

Write a program which consists of two classes, **Date** and **Person.**
Date class should contain three data members **day**, **month**, **year** and setter and getter function for these data members. Date class should also contain **showdate()** member function to display date.
Person class should contain three data members **Name, Address,** and **Bday**, where Name and Address are char pointer while Bday(Date of birth) is of type Date, Person class should further contain two member functions **Display()** and **setdate().**
In main program Create an object of Class person and call the member functions with it.

Answer:

```
#include <stdio.h>
#include <iostream>
#include <cstring>
using namespace std;

class Date
{
public:
        int day;
        int month;
        int year;

public:
        Date()
                {
                        day=0;
                        month=0;
                        year=0;
                }
        void setDay(int);
        void setMonth (int);
        void setYear(int);

        int getDay();
        int getMonth();
        int getYear();

        void showDate();
};
```

```
void Date: :setDay(int d)
{
        if{d<1 | | d>31)
        cout<<"Invalid month Renter it";
        cin>>d;
}
        day=d;
}

void Date: :setMonth (int m)
{
        if(m<1 | | m>12)
{
        cout<<"Invalid month Renter it";
        cin>>m;
}
        month=m;
}
        void Date: :setYear (int y)
{
        year=y;

        int Date: :getDay()
{
        return day;
}
        int Date: :getMonth()
{
        return month:
}
        int Date: :getYear()
{
        return year;
}
        void Date: :showDate()
{
        cout<<day<<"-"<<month<<"-"<<year<<end1;
}
Class Person
{
public:

        char *Name;
   char *Address
        Date Bday;

public:
        Student()
        {
        Name=new char[20];
```

```
        Address=new char[10];
        cin.getline(Name,20);
        cout<<"Enter Address:";
        cin.getline(Address,10);
        }

void setDate()
{
        cout<<"Enter Day:";
        cin>>Ad_date.day;
        cout<<"Enter month:";
        cin>>Ad_date.month;
        cout<<"Enter Year:";
        cin>>Ad_date.year;
        }
void Display()
        {
        cout<<"Name: "<<end1;
        cout<<"Address: "<<Address<<end1;
        cout<<"Date of Birth: ";
        Ad-date.showDate();
        }
};
void main()
{
        Person object;
        object.setDate();

        object.Display();

        system("pause");
}
```

Question (Marks: 10)

Write a C++ program that contains a class 'myClass' having two data members of type int.

The class must have

- A default constructor which must initialize all the data members to their meaningful values.
- A destructor with no implementation.
- Setter member functions to set all data members of class
- Getter member functions to get all data members of class

In main function of the program

5. Prompt the user to enter the number of objects to be created.
6. Dynamically allocate memory to objects according to the size entered by user.
7. De-allocate memory that was allocated to objects

Answer:

```
#include <stdio.h>
#include <iostream>
```

```
#include <cstring>
using namespace std;

class myclass
{
public:
int a;
int b;
int *iptr, *sptr;

construct{int,int.int}

        void seta(int);
        void setb(int);
        void setc(int);

        int geta();
        int getb();
        int getc();
};

void Person: :seta(int aa)
{
        a=aa;
}

void Person: :setb (int bb)
{
        b=bb;
}
        void Person: :setc (int cc)
{
        c=cc;
}

main()
{
int num;
cout<<"Enter the number of objects to be created";
cin>>num;
for (int i =1;i==num;i++)
{
Person i_
}
}
```

Question No: 31 (Marks: 1)

How do we provide the default values of function parameters?

Answer:

Question (Marks: 1)
Why do java consider pointer as dangerous
Answer: JAVA, describe pointers as dangerous . if we assign a memory through a pointer where the pointer is destroyed, the memory remains allocated and is wasted. To address these things, there are only references in JAVA instead of pointers. JAVA gives the concept of garbage collection with the use of references. Due to this garbage collection, we are free from the headache of de- allocating the memory. We allocate and use the memory. When it is no longer in use, JAVA automatically deletes (frees) it through garbage collection

Question (Marks: 2)
What is memory leak?
Answer: There is a requirement that if the constructor of a class allocates the memory, it is necessary to write a destructor of that class. We have to provide a destructor for that class, so that when that object ceases to exist, the memory allocated by the constructor, is returned to the free store. It is critically important. Otherwise, when the object is destroyed, there will be an unreferenced block of memory. It cannot be used by our program or by any other program. It's a memory leak that should be avoided.

Question (Marks: 2)
What does optimization the of code means?
Answer: Optimization is the process of transforming a piece of code to make more efficient without changing its output or side-effects. The only difference vuzs visible to the code's user should be that it runs faster and/or consumes less memory.

Question (Marks: 3)
What is the difference between structure and class?
Answer: The ONLY DIFFERENCES between classes and structures are
1) classes DEFAULT to having private members. Structures DEFAULT to having public members. These defaults can be changed so classes can be made to work like structures and vice versa.
2) classes DEFAULT to inheriting privately from base classes. Structures DEFAULT to inheriting public from base classes. These defaults can be changed so classes can be made to work like structures and vice versa.

Question (Marks: 3)
See the following code segment.

```
template
class myclass {
private:
T x;
public:
myclass (T a) {
 x = a;
}
};
```

Write the main function which creates two objects of class for int and double data types.
Answer:

Question (Marks: 3)
Is it possible to define two functions as given below? Justify your answer.
func(int x, int y)
func(int &x, int &y)
Answer:

Yes, function can be defined in both given ways, as
in function no.1 , function is calling the variable by its value (which is default function call method).In the second function , function call is done by using call by reference methos.

Question (Marks: 5)
Write a program using **getline()** member function to inputs a string up to delimiter character comma (,) and then display the string on the screen.
Answer:

Question (Marks: 5)
Do you think that friend functions violate encapsulation? Justify your answer.
Answer: The friend functions of a class have access to the private data members of class. Despite being a good thing, there is possibility of vulnerability. We are opening our thoughts, inside view for somebody else. Without having 100% trust, it will be risky to make our thoughts and feelings public. We want that our private data is accessible to someone outside, not public for everybody. Otherwise, the data encapsulation and data-hiding concept will be violated. We keep the data members private and declare some specific functions that are not member of the class but friend of the class. As friends, they have access to the inside data structure of the class despite not being members.

Question (Marks: 10)
Write a simple program using the **get()** member function of **cin** object reading a text of **30** characters from the keyboard, store them in an array and then using **put()** member function of **cout** object to display them on the screen.
Answer:

Question (Marks: 10)
Write a small program which defines two user-defined manipulators named **octal** and **hexadecimal**. These manipulators should display the decimal numbers into octal and hexadecimal.
In the main function, input a decimal number from the user and then display this decimal number into octal and hexadecimal using user-define manipulators named **octal** and **hexadecimal**.
Answer:

Question: What are User defined Manipulators?
Answer: Parameterized manipulators require one or more arguments. setfill (near the bottom of the iomanip.h header file) is an example of a parameterized manipulator. You can create your own parameterized manipulators and your own simple manipulators.

Question: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: 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: What is principle of friendship in the context of functions and classes?
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: What are the limitations of the friendship relation between classes?
Answer: Class can declare a friend class from inside and someone from outside the class cannot declare itself friend of a class.

Question: 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: 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:
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: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

Question: What is an assignment operator?
Answer :Assignment Operator
At first, we ascertain whether there is need of an assignment operator or not? It is needed when we are going to assign one object to the other, that means when we want to have expression like a = b. C++ provides a default assignment operator. This operator does a member-wise assignment.

Question: What is drawback of writing the definitions of all the functions before main function?
Answer

Question (Marks: 1)
How do we provide the default values of function parameters?
Answer
Question (Marks: 2)
What is difference between endl and \n?
Answer

Question (Marks: 2)
When does an object get destroyed?
Answer

Question (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;
}
```

2)

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

Question (Marks: 3)
Why stream insertion and stream extraction operators cannot be overloaded as member functions?

Question No: 38(Marks: 5)
What is difference between Unary and binary operators and how they can be overloaded?

Question No: 39(Marks: 5)
What steps we must follow to design good program?

Question No: 40(Marks: 10)
Write the program that inputs an **octal** number from the user and then display the entered octal number into **hexadecimal** number using **manipulators** (parameter-less, parameterized) and **member function** of input/output streams.

Question No: 41(Marks: 10)
Develop a class **Vector** having two data members; x and y.
The class should also provide the following Overloaded operator capabilities.
a) Overload the addition operator(+) to add two **Vectors**
b**)** Overload the assignment operator(=) to assign **Resultant Vector**
c) Write function **Display()** to display x, y coordinates
Note: Addition of vector Let suppose there are two vectors A and B with their x, y coordinates.

Question: How do we provide the default values of function parameters?
Answer: The default value of a parameter is provided inside the function prototype or function definition. For example, we could declare the default function arguments for a function while declaring or defining it.

Question No: 27 (Marks: 2)
What is the difference between **switch** statement and **if** statement.
Question No: 28 (Marks: 2)
How can we initialize data members of contained object at construction time?
Question No: 29 (Marks: 2)
How the data members of a class are initialized with meaningful values?
Question No: 30 (Marks: 2)
Can we overload ***new*** and ***delete*** operators?
Question No: 31 (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;
}
```

2)

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

Question No: 32 (Marks: 3)

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

Question No: 33 (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?
(ii) The destructor of which class will be called first?

Question No: 34 (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.

Question No: 35 (Marks: 5)

Write down the disadvantages of the templates.

Question No: 36 (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
xxxxx20000
xxxxx30000
xxxxx60000

Answer

```
#include <stdio.h>
#include <iostream>
#include <cstring>
using namespace std;

class myclass
{
public:
int a;
int b;
int *iptr, *sptr;

construct{int,int.int}

        void seta(int);
        void setb(int);
        void setc(int);

        int geta();
        int getb();
        int getc();

};

void Person: :seta(int aa)
{

        a=aa;
```

```
}

void Person: :setb (int bb)
{

        b=bb;
}
        void Person: :setc (int cc)
{
        c=cc;
}

main()

{

int num;

cout<<"Enter the number of objects to be created";

cin>>num;

for (int i =1;i==num;i++)
{

Person i_

}

}
```

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  floatArry[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: 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: 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

```
 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<
}
```

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
T func (T a, U b) {
  return (a
}
calling
int  i;
double x;
x = func
```

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<<<"-"<<<"-"<

}

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 in 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

xxxxx20000

xxxxx30000

xxxxx60000

Answer:

```
#include
#include
main(){
    int sal1 =1000;
    int sal2 =1500;
    int sal3 =20000;
    int sal4 =30000;
    int sal5 =60000;

cout << setfill ('x') << setw (10);
cout<< sal1<
cout << setfill ('x') << setw (10);
cout<< sal2<
cout << setfill ('x') << setw (10);
```

```
cout<< sal3<
cout << setfill ('x') << setw (10);
cout<< sal4<
cout << setfill ('x') << setw (10);
cout<< sal5<
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<
cout<
cout<
}
```

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: 39 (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

Question No: 27 (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

Question No: 28 (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

Question No: 29 (Marks: 2)

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

Answer

Question No: 30 (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

Question No: 33 (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?

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

Question No: 34 (Marks: 5)

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

Question No: 35 (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.

Question No: 36 (Marks: 5)

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

Answer

Question: What are the advantages of STL ?
Answer: **Advantages of (STL)**

- Templates are easier to write then 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 can be easier to understand, since they can provide a straightforward way of abstraction type information.
- 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 help in utilizing compiler optimizations to the extreme.

Question: what is meant by default constructor
Answer:

Question : what is conversion constructor
Answer:

Question: Y we keep utility functions as private in class?
Answer

Question: Write a program using get() function of cin n puts function of cout n take 30 array characters from keyboard n display on screen using these functions?

Question: write a program define a line (-) operator n manipulate it
For example the output of program should beVirtual--------university
2nd: write a program to overload a unary operator(++)n make a date class. Day, year, month n lastday(last day of month) should be its data members, operator as a function member,n display function to display date.
Now increment the day by using operator function n month should also b incremented after 30 or 31 days,n year should also incremented after the 12 month

Question:
What is self assignment mean, why should it be avoided? [3 marks]

Question:
Is it possible to overload operators for primitive data type? Justify [3 marks]

Question:
What are the limitations of the friendship relations between classes?[3 marks]
Answer

Question:
C and C++ are free format languages. What does mean by free format? [2 marks]
Answer

Question:
What are the two types of conversions for user defined data type? [1 marks]
Answer

Question:
Describe the three important constructs to solve a given problem. [5 marks]
Answer

Question:
How can we increase the size of dynamic allocated memory in C? [5 marks]
Answer

Question:
Does the following statement create new variable? If not, why? [1 marks]
int &ref = val
Answer

Question: [10 marks]
Write a program which contains a class Account. The class should contains three data members Name, AcNo. and Balance. The class shouls further contains two constructors i.e. default constructor and parameterized constructor. Overload the stream insertion operator (<<) for this class.
Answer

Question: [10 marks]
Write a program that reads a number that says how many integer numbers to be stored in an array. Create an array to for the exact size of data and read in that many numbers into the array.
Answer

Question No: 36 (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  ;
```

Ans:

```
#include <iostream.h>
math
{
mth operator + (obj1,obj2)
mth operator + (obj1,obj2)
{
 mth operator + (obj1,obj2)
mth operator + (obj1,obj2)

}
}
```

Question No: 36 (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 this code the matrix function is defined, it get the number of rows from the user and create the row of matrix and then get the columns from the user and create the columns. The New is showing for creating more array space for the data which user enters. The elements [i][j] will print the data in matrix form.

Question No: 35 (Marks: 5)

Why the first parameter of operator function for << operator must be passed by reference?

Answer:

Operator<<'s first parameter must be an ostream passed by reference. Its second parameter, the IntList that is printed, does not have to be passed as a const-reference parameter; however it is more efficient to pass it by reference than by value (since that avoids a call to the copy constructor), and it should not be modified by operator<<, so it should be a const reference parameter

Question No: 34 (Marks: 5)

What is the difference between function overloading and operator overloading?

Answer:

Difference b/w function overloading and operator overloading is:
In function overloading, the functions have the same name but differ either by the number of arguments or the type of the arguments.
Operator overloading is to allow the same operator to be bound to more than one implementation, depending on the types of the operands.

Question No: 33 (Marks: 3)

What happens when we use new and delete operator?

Answer: When we use ***new*** operator to create objects the memory space is allocated for the object and then its constructor is called. Similarly, when we use ***delete*** operator with our objects, the destructor is called for the object before deallocating the storage to the object.

Question No: 32 (Marks: 3)

Is it possible to define two functions as given below? Justify your answer.
func(int x, int y)
func(int &x, int &y)

Answer:

No, we cannot define two functions as func(intx, inty) func(int &x, int&y) because it's give an error function not initializing.

Question No: 31 (Marks: 3)

Consider the following code segment. What will be the output of the following code segment?

```
class class1{
public:
class class2{
public:
class2(){
cout << "Calling default constructor of class2\n" ;
}
};
class1(){
cout << "Calling default constructor of class1\n" ;
}
} ;
main(){
class1::class2 obj1;
class1 obj2 ;
}
```

Question No: 27 (Marks: 2)

Give the general syntax of class template.

Answer:

Syntax of class template:

```
template <class T>
 class class-name()
```

{
definition of class
};

Question No: 28 (Marks: 2)
What is difference between endl and \n?
Answer:
The difference between endl and \n *is that* endl is use to start a new line for the next row
And \n is a new line character.

Question No: 29 (Marks: 2)
What is the **this** pointer? Give an example of its use.
Answer:
This pointer is use to points to the current object in programming.

Question No: 30 (Marks: 2)
Identify each of the following as function call, function definition and function declaration.
int func(int num1, int num2);
Function call:
Function ; Function definition: Integer; Function declaration: Num1 and Num2

int func(int, int);
Function call:
Function ; Function definition: Integer; Function declaration: integers

func(5, 6) ;

Function call:
Function ; Function definition: numbers; Function declaration: 5&6

int func(int num1, int num2){}
Function call:
Function ; Function definition: Integer; Function declaration: Num1 and Num2 from user

Question No: 27 (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?

Question No: 28 (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?

Question No: 29 (Marks: 2)
What is the benefit of reference and where can we use it?

Question No: 30 (Marks: 2)
Write the C++ code for the declaration of overloaded stream insertion and stream extraction operator for the object **d** of type **Date**.

Question No: 31 (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;
}
```
2)
```
void func2(){
static int x = 0 ;
x++;
cout << x << endl ;
}
```

Question No: 32 (Marks: 3)
If the requested memory is not available in the system then what does **calloc/malloc** and **new** operator return?

Question No: 33 (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?
(ii) The destructor of which class will be called first?

Question No: 34 (Marks: 5)
What is difference between Unary and binary operators and how they can be overloaded?

Question No: 35 (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.

Question No: 36 (Marks: 5)

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

}

2)

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

Question No: 37(Marks: 3)

Why stream insertion and stream extraction operators cannot be overloaded as member functions?

Question No: 38(Marks: 5)

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

Question No: 39(Marks: 5)

What steps we must follow to design good program?

Question No: 40(Marks: 10)

Write the program that inputs an octal number from the user and then display the entered octal number into hexadecimal number using manipulators (parameter-less, parameterized) and member function of input/output streams.

Question No: 41(Marks: 10)

Develop a class Vector having two data members; x and y.

The class should also provide the following Overloaded operator capabilities.

a) Overload the addition operator(+) to add two Vectors

b) Overload the assignment operator(=) to assign Resultant Vector

c) Write function Display() to display x, y coordinates

Note:Addition of vector Let suppose there are two vectors A and B with their x, y coQuestion No: 31(Marks: 1)

What is drawback of writing the definitions of all the functions before main function?

Question No: 32(Marks: 1)

How do we provide the default values of function parameters?

Question No: 33(Marks: 2)

What is difference between endl and \n?

Question No: 34(Marks: 2)

When does an object get destroyed?

Question No: 35(Marks: 3)

What is the difference between structure and class?

Question No: 36(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;
```