

Sample Paper – 2014
Class - XII
Subject: Computer Science (083)

Time Allowed: 3 Hr.

M.M.70

General Instructions:

1. All questions are compulsory.
2. Programming Language- C++.

Q1. Answer the following questions:

- (a) What do you mean by a formal parameter and actual Parameter? 1
- (b) Write the names of the header files to which the following belong:
(a) setw() (d) strcpy () 1
- (c) What is the difference between call by value and call by reference? Give an example in C++ to illustrate both. 2
- (d) What is the difference between Global variable and Local variable? Explain with example. 2
- (e) Rewrite the given snippet after removing the syntactical error(s), if any. Underline each correction 2

```
#include<iostream.h>
void main( )
{
struct movie
{ char movie_name[20];
char movie_type;
int tickets=100;
} MOVIE;
gets(movie_name);
gets(movie_type);
}
```

- (f) Find the output of the following program (Assuming that all required header files are included) 2

```
void main( )
{
char * NAME = "admiNStrAtiOn";
for( int x=0; x<strlen(NAME); x++ )
if(islower(NAME[x])
NAME[x] = toupper(NAME[x]);
else if(isupper (NAME[x]))
    if(x%2==0)
        NAME[x] = NAME[x - 1];
    else
        NAME[x]--;
cout<<NAME <<endl;
}
```

- (g) Find the output of the following program 2

```
#include<iostream.h>
void Modify(int &a, int b=10)
{ if(b%10==0)
a+=5;
for(int i=5;i<=a;i++)
cout<<b++<<":";
cout<<endl;
}
void Disp(int x)
{
if(x%3==0)
Modify(x);
```

```

else
Modify(x,3);
}
void main()
{
    Disp(3);
    Disp(4);
    Modify(2,20);
}

```

(h) What will be the output of following code.

2

```

#include<iostream.h>
#include<ctype.h>
void mycode((char msg[], char ch)
{
    for (int cnt=0;msg[cnt]!='\0';cnt++)
    {
        if(msg[cnt]>='B' && msg[cnt] <='G')
            msg[cnt]=tolower(msg[cnt]);
        else if(msg[cnt]=='A' || msg[cnt]=='a')
            msg[cnt]=ch;
        else
            if(cnt % 2==0)
                msg[cnt]=toupper(msg[cnt]);
            else
                msg[cnt]=msg[cnt-1];
    }
}
void main()
{

```

```
char mytext[]="ApEACeDriVE";
mycode(mytext,'@');
cout<<"New Text : "<<mytext<<endl;
}
```

- (i) Find the output of the following program :

2

```
#include<iostream.h>
void execute(int &X, int Y=200)
{
    int temp=X+Y;
    X+=temp;
    if(Y!=200)
        cout<<temp<<X<<Y<<endl;
}
void main()
{
    int A=50,B=20;
    execute(B);
    cout<<A<<B<<endl;
    execute(A,B);
    cout<<A<<B<<endl;
}
```

- (j) In the following program, find the correct possible output(s) from the options. Justify your answer.

2

```
#include<iostream.h>
#include<stdlib.h>
const int low=25;
void main( )
{
```

```

randomize( );
int point=5, number;
for(int i=1;i<=4;i++)
{
    number=low+random(point);
    cout<<number<<" ";
    point--;
}

```

}

(i) 29:26:25:28: (ii) 24:28:25:26: (iii) 29:26:24:28: (iv) 29:26:25:26:

(k) Study the following program and select the possible output from it:

2

```

#include<iostream.h>
#include<stdlib.h>
const int limit=4;
void main( )
{
    randomize( );
    int points;
    points=100+random(limit);
    for(int p=points;p>=100;p--)
        cout<<p<<"#";
    cout<<endl;
}

```

(i) 103#102#101#100#

(ii) 100#101#102#103#

(iii) 100#101#102#103#104#

(iv) 104#103#102#101#100#

Q2. Answer the following questions:

(a) What do you mean by modular programming? What is its significance?

2

(b) Explain about the Abstract Data type and Data Hiding ?

2

Q3. Answer the following questions:

- (a) What is function over loading? Explain it. Illustrate with a proper example? 2
- (b) What is the role of default value in the function? Explain with suitable example 2

Q4. Answer the following questions:

- (a) Define a class 'Flight' in C++ with following description: 4

Private Members

- A data member Flight number of type integer
- A data member Destination of type string
- A data member Distance of type float
- A data member Fuel of type float
- A member function CALFUEL() to calculate the value of Fuel as per the following criteria

Distance	Fuel
<=1000	500
more than 1000 and <=2000	1100
more than 2000	2200

Public Members

- A function FEEDINFO() to allow user to enter values for Flight Number, Destination, Distance & call function CALFUEL() to calculate the quantity of Fuel
- A function SHOWINFO() to allow user to view the content of all the data members

- (b) Define a class Housing in C++ with the following specification: 3

Private Members:

- REG_No an integer (Ranges 10- 1000)
- Name an array of char of size[20]
- type Character
- Cost float

Public Members:

- Function Read_Data () to read an object of Housing type.

- Function Display to display the details of an object.
- Function Draw_Nos() to choose and display the details of 2 house selected randomly from an array of 10 objects of type Housing. Use random function to generate the registration **nos** to match with REG_NO from the array.

(c) What do you mean by function? How it is differ from a constructor? 2

(d) What is Static Data member ? Give one characteristic? 1

Q5. Answer the following questions:

(a) What is the difference between constructor and destructor? Give a suitable example. 2

(b) Answer the questions (i) and (ii) after going through the following program: 2

```
class Match
{
    int Time;
public:
    Match() //Function 1
    {
        Time=0;
        cout<<"Match commences"<<endl;
    }
    void Details() //Function 2
    {
        cout<<"Inter Section Basketball Match"<<endl;
    }
    Match(int Duration) //Function 3
    {
        Time=Duration;
        cout<<"Another Match begins now"<<endl;
    }
    Match(Match &M) //Function 4
```

```
{  
    Time=M.Duration;  
    cout<<"Like Previous Match "<<endl;  
}  
};
```

1. Which category of constructor - Function 4 belongs to and what is the purpose of using it?
 2. Write statements that would call the member Functions 1 and 3
- (c) What is copy constructor. Give the suitable example to demonstrate it? 2
- Q6. (a) Answer the questions (i) to (iv) based on the following: 4

```
class CUSTOMER  
{  
    int Cust_no;  
    char Cust_Name[20];  
  
protected:  
    void Register();  
  
public:  
    CUSTOMER();  
    void Status();  
};  
  
class SALESMAN  
{  
    int Salesman_no;  
    char Salesman_Name[20];  
  
protected:  
    float Salary;  
  
public:
```

```
SALESMAN();  
void Enter();  
void Show();  
};  
class SHOP : private CUSTOMER , public SALESMAN  
{  
    char Voucher_No[10];  
    char Sales_Date[8];  
public:  
    SHOP();  
    void Sales_Entry();  
    void Sales_Detail();  
};
```

- (i) Write the names of data members which are accessible from objects belonging to class CUSTOMER. 1
- (ii) Write the names of all the member functions which are accessible from objects belonging to class SALESMAN. 1
- (iii) Write the names of all the members which are accessible from member functions of class SHOP. 1
- (iv) How many bytes will be required by an object belonging to class SHOP? 1
- (b) Consider the following declarations and answer the questions given below 4

```
class Goods  
{  
int id;  
protected :  
char name[20];  
long qty;
```

```
void Incr(int n);
public :
Goods();
~Goods();
void in();
};
class Food_products : protected Goods
{
    char exp_dt[10];
    protected :
    int id;
    int qty;
    public :
        void getd();
        void showd();
};
class Cosmetics : private Goods
{
    int qty;
    char exp_date[10];
    protected :
    int id;
    public :
        ~Cosmetics();
        Cosmetics();
        void show();
};
```

(i) Name the all protected members of class Food_products.

1

- (ii) Name the member functions accessible through the object of class Food_products. 1
- (iii) From the following, Identify the member function(s) that cannot be called directly from the object of class Cosmetics 1
show() , getd() , in()
- iv) If the class cosmetics inherit the properties of food_products class also, then name the type of inheritance. 1
- (c) How does the **visibility mode** control the access of members in the derived class in terms of inheritance? Explain with example. 2
- (d) What is difference between the multiple and multilevel inheritance ? 1
- Q7. Observe the program segment given below carefully and answer the question that follows

```
#include<fstream.h>
class candidate
{
    long cid,marks;
    char cname[20];
    public:
        void enter();
        void display();
        void markschange();
        long r_cid() { return cid;}
};
void markschange(long id)
{
    ifstream file;
    file.open("candidate.dat",ios::in | ios::out);
    candidate C;
    int record=0, found=0;
    while(!found && file.read((char *) &C,sizeof(C))
```

```

{
    if(id==C.r_cid())
    {
        cout<<"Enter new marks ";
        C.markchange();
        _____ //statement 1
        _____ //statement 2
        found=1;
    }
    Record++;
}
if(found==1)
    cout<<"Record Updated";
file.close();
}

```

(a) Write the statement1 to position the file pointer at the beginning of the record for which the candidate's Id matches with the argument passed, statement 2 to write the updated record at that position. 2

(b) Write a function to count the words and characters from a text file named as 'matter.txt'. e.g.
 No Smoking i.e. characters -10 and words -2 2

(c) Given a binary file "BUS.DAT", containing records of the following class 3

```

class bus
{
    int bus_no;
    char desc[40];
    int distance; //in km
public:
    void read();
    void display();
}

```

```
int retdist()
{
    return distance;
};
```

Write a function in C++ that would read contents of file “BUS.DAT” and display and transfer the details of those buses which travels distance more than 100 km to other file “Temp.DAT”.

- (d) Write a function in C++ to add a new record of the student at the end of a binary file “Student.dat”, assuming the binary file is containing the records of the class ‘student type’:- 1

```
class student
{
    int rno;
    char name[25];
public:
    void enter( )
    {
        cin>>rno;
        cin>>name;
    }
    void display( )
    {
        cout<<rno;
        cout<<name;
    }
};
```

Q8. Answer the following questions:

- (a) If two – dimensional array C[5.....9, -4.....10] is stored using column major representation, then calculate the address of C[8,6], if the base address is 100 and each element requires 4 bytes of memory. 3

- (b) Suppose A,B,C are arrays of integers of size M, N and M + N respectively. The numbers in array A appear in ascending order and B appear in descending order. Give the necessary declaration for array A,B and C in C++. Write a user define function in C++ to produce third array C by merging arrays A an B in ascending order 3
- (c) Write the user defined function which accept an integer 1D array and its size and convert it into 2D Array as follow **2 D array should be**

if array is 1 2 3 then

1 2 3

1 2 0

1 0 0

and if 1 2 3 4 5 then

1 2 3 4 5

1 2 3 4 0

1 2 3 0 0

1 2 0 0 0

1 0 0 0 0

Answer Key 12th C.Sc. Term test-1

1.

- a)** The formal parameter is the parameter as defined on the function signature.

E.g.,

```
int myFunction(int parm1, double parm2);
```

The actual parameter is when you actually pass real values in calling that function.

E.g.

```
int myAge = 99;
```

```
double myPie = 3.14;
```

```
int myAnswer = myFunction (myAge, myPie);
```

b) iomanip.h

string.h

- c)** Call by value:- Passing only the values so that only copy of the value is sent to functions

Eg:

```
int main()
```

```
{
```

```
int a=10,b=20,c;
```

```
c=add(a,b);
```

```
cout<<c;
```

```
return 0;
```

```
}
```

```
int add(int x,int y) // copy of values ie a,b is received here
```

```
{
```

```
int z;
```

```
z=x+y;
```

```
return x;
```

```
}
```

call by Reference:

the address of the value is passed and since address is passed the original value will change since address is modified

Eg:

```
int main()
```

```
{
```

```
int a=10,b=20,c;
```

```
c=add(&a,&b); // address is passed
```

```
cout<<c;
```

```
return 0;
```

```
}
```

```
int add(int *x,int *y) // address locatin is received
{
int z;
z=x+y;
return x;
}
```

- d) A global variable is accessible to any function. A variable in main acts exactly like any other local variable, and is only accessible to code in main. Also, if main (or another function) calls itself, the local variables will be copied, but global variables will not.

```
int x = 0;
void myfunction()
{
x++;
printf("%i ",x);
}
int main()
{
myfunction();
myfunction();
}
```

This example will output: 1 2

- e) **Rewrite the following code:**

```
#include<iostream.h>
void main( )
{
struct movie
{ char movie_name[20];
char movie_type;
int tickets // Cant be assign here
} MOVIE;
Void input() // outside inside the structure function definition
{
must be there //
gets(MOVIE.movie_name); // data member must be call with structure
variable //
gets(MOIVE.movie_type); // data member must be call with structure
variable
```

}

f). ADMIIRTRRTINN

g) 10:11:12:13:

20:21:22:

h) New Text : @@e@ccddlle

I). 50 240

290 340 240

340 240

J). Option (IV) 29:28:25:26:

k) option (I) 103#102#101#100#

2.

(a) modular programming

- A programming style in which a program is divided into separately compiled files.
- In C++, modular programming uses separate implementation and interface files.
- Interface files contain declarations, but no executable code.
- Implementation files use the declarations which are in the interface files, by including them with preprocessor *#include* statements, and contain executable code.
- Interface files have the extension *.h*.
- Implementation files have the extension *.cpp*.

(b) An ADT is a collection of data and associated operations for manipulating that data . ADTs support abstraction, encapsulation, and information hiding They provide equal attention to data and operations

- Data Hiding is also known as Encapsulation.
- Encapsulation is the process of combining data and function into a single unit called class.
- Data Hiding is the mechanism where the details of the class are hidden from the user.
- The user can perform only a restricted set of operations in the hidden member of the class.

3.

- (a) C++ permits the use of two function with the same name. However such functions essentially have different argument list. The difference can be in terms of number or type of arguments or both.

This process of using two or more functions with the same name but differing in the signature is called function overloading.

But overloading of functions with different return types are not allowed.

In overloaded functions , the function call determines which function definition will be executed.

The biggest advantage of overloading is that it helps us to perform same operations on different datatypes without having the need to use separate names for each version.

Example

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

```
int abslt(int );
long abslt(long );
float abslt(float );
double abslt(double );
```

```
int main()
{
```

```
int intgr=-5;
long lng=34225;
float flt=-5.56;
double dbl=-45.6768;
cout<<\ " absolte value of \ "<<intgr<<\ " = \ "<<abslt(intgr)<<endl;
cout<<\ " absolte value of \ "<<lng<<\ " = \ "<<abslt(lng)<<endl;
cout<<\ " absolte value of \ "<<flt<<\ " = \ "<<abslt(flt)<<endl;
```

```

cout<<\" absolute value of \"<<dbl<<\" = \"<<abslt(dbl)<<endl;
}
int abslt(int num)
{
if(num>=0)
return num;
else
return (-num);
}
long abslt(long num)
{
if(num>=0)
return num;
else return (-num);
}
float abslt(float num)
{
if(num>=0)
return num;
else return (-num);
}
double abslt(double num)
{
if(num>=0)
return num;
else return (-num);
}

```

OUTPUT

absolute value of -5 = 5
absolute value of 34225 = 34225
absolute value of -5.56 = 5.56
absolute value of -45.6768 = 45.6768

- (b). A **default parameter** is a function parameter that has a default value provided to it. If the user does not supply a value for this parameter, the default value will be used. If the user does supply a value for the default parameter, the user-supplied value is used.

Consider the following program:

```
void PrintValues(int nValue1, int nValue2=10)
{
    using namespace std;
    cout << "1st value: " << nValue1 << endl;
    cout << "2nd value: " << nValue2 << endl;
}

int main()
{
    PrintValues(1); // nValue2 will use default parameter of 10
    PrintValues(3, 4); // override default value for nValue2
}
```

This program produces the following output:

```
1st value: 1
2nd value: 10
1st value: 3
2nd value: 4
```

4.A.

```
#include<iostream.h>
#include<conio.h>
class flight
{
    int fnumber;
    char dest[10];
    float dist,fuel;
    void calfuel()
    {
        if(dist<=1000)
            fuel=500;
        else if(dist<=2000)
            fuel=1100;
        else
            fuel=2200;
    }
public:
    void feedinfo()
    {
        cout<<"\n Enter the values ";
        cin>>fnumber>>dest>>dist;
```

```

        calfuel();
    }
    void showinfo()
    {
        cout<<fnumber<<"\t"<<dest<<"\t"<<dist<<"\t"<<fuel;
    }
};
void main()
{
    flight f1;
    f1.feedinfo();
    f1.showinfo();
    getch();
}

```

```

(B).    #include<stdio.h>
#include<conio.h>
#include<stdlib.h>
class Housing
{
    int reg_no;
    char name[20];
    char type;
    float cost;
public:
    void Read_data()
    {
        cout<<"\n\n\t enter the values";
        cin>>reg_no>>name>>type>>cost;
    }
    void Display()
    {
        cout<<reg_no<<"\t"<<name<<"\t"<<type<<"\t"<<cost;
    }
    void Draw_nos()
    {
        randomize();
        int n1=10+random(991);
        int n2=10+random(991);
        for(int i=0;i<100;i++)
        {

```

```

        if((reg_no==n1) || (reg_no==n2))
            Display();
    }
};

```

(c). A function is a collection of some statements which is used for particular operation. It is a part of a program. The name of function is different from its class. A function is explicitly called by the object.

But a constructor is a member function of a class having same name as its class. A constructor is implicitly called. It should be always defined in public section.

d). The declaration of a static data member in the member list of a class is not a definition. You must define the static member outside of the class declaration, For example:

```

class X
{
public:
static int i;
};
int X::i = 0; // definition outside class declaration

```

5.

(i) A constructor is a function that is automatically called when an object of a class is created; a destructor is a function that is automatically called when an object of a class is deleted.

(ii). (1) copy constructor.

```

(2) Match M1 // Function -1
    Match M2(40); // Function-3

```

(iii) A copy constructor is a constructor that copy the value of an existing object. E.g.

```

Sample(Sample &s)
{
    J=s.J;
    I=s.I;
    cout<<"Copy constructor";
}

```

6. (a) (i) None
 (ii) Enter(), Show();
 (iii) Data members: salary
 Member Functions: Register(), Status(), Enter(), Show(), Sales_Entry(), Sales_Detail()
 (iv) 66 bytes
- (b) (i) protected data members: name, qty, id, qty
 Protected function members: Incr(int n), in()
 (ii) getd(), showd()
 (iii) getd(), in()
 (iv) Hybrid Inheritance
- c) The visibility mode controls the accessibility of the members of base class. It also provides the security to all the members of the base class. There are basically 3 types of accessibility modes:
 (i) private: These members are accessible only inside the class.
 (ii) protected: These members can be accessed within the sub-class only.
 (iii) public: These members can be accessed anywhere in the program by the object of the class.
- d) In multilevel inheritance, one class inherits the properties of more than one class.

In Multilevel inheritance, the sub-class of a class is the base class of another class.

7. (a) Statement 1: `file.seekp(-1* sizeof(C), ios::cur);`

Statement 2: `file.write((char *)&C, sizeof(C));`

(b) void count()

```
{
    ifstream fin("matter.txt");
    char ch, word[10];
    int chr=0, w=0;
    while(!fin.eof())
    {
        fin.get(ch);
        chr++;
    }
    fin.seekg(0);
    while(!fin.eof())
```

```

    {
        fin>>word;
        w++;
    }
    cout<<"Total no. of characters "<<<chr<<<"\nTotal no of words "<<<w;
}

```

c) void transfer()

```

{
    ifstream fin("BUS.DAT");
    ofstream fout("Temp.DAT");
    bus b1;
    while(!fin.eof())
    {
        fin.read((char *)&b1, sizeof(b1));
        if(fin.eof()) break;
        if(b1.retdist(>100)
        {
            b1.display();
            fout.write((char *)&b1,sizeof(b1));
        }
    }
}

```

d) void insert()

```

{
    ofstream f1("student.dat",ios::app|ios binary);
    student s1;
    s1.enter();
    f1.write((char *)&s1,sizeof(s1));
    f1.close();
}

```

8. a) Address of C[8,6] =

$$\begin{aligned}
 N &= 5; \\
 B &= 100, W = 4, Lr = 5, Lc = -4 \\
 &= B + W ((I - L r) + (J-Lc) * N) \\
 &= 100 + 4 ((8-5) + (6-(-4) * 5)) \\
 &= 312
 \end{aligned}$$

```

b) void merge()
{ int a[5],b[5],c[10],k=0,m=5,n=5;
cout<<"\n\n\t Enter first array in asc. order";
for(int i=0;i<m;i++)
    cin>>a[i];
cout<<"\n\n\t Enter second array in desc. order";
for(i=0;i<n;i++)
    cin>>b[i];
i=0;
int j=4;
while(i<5 && j>=0)
{    if(a[i] <b[j])
    {    c[k]=a[i];
        i++;
    }
    else
    {    c[k]=b[j];
        j--;
    }
    k++;
}
if(i<5)
{    while(i<5)
    c[k++]=a[i++];
}

```

```
if(j>=0)
{
    while(j>=0)
        c[k]=b[j--];
}
for(i=0;i<m+n;i++)
    cout<<c[i];
getch();
}
c) void fun( int arr[], int size)
{
    int a2[20][20], i,j;
    for(i=0;i<size;i++)
    {
        for(j=0;j<size;j++)
        {
            if((i+j)>=size)
                a2[i][j]=0;
            else
                a2[i][j]=arr[j];
        }
        cout<<a2[i][j];
    }
    cout<<endl;
}
}
```

Paper Submitted By:

Name Anil Khatri

Email anil_2279@rediffmail.com

Phone No. 09812835914