

## CLASSES & OBJECTS WITH OOPS IN C++

### 1 Mark Questions

1. Name the header files to which the following belong:  
(i) strcmp()    (ii) toupper()
2. Write the name of the header files to which the following belong:  
(i) strcmpi()    (ii) clrscr()
3. Name the header files in which the following belong:  
(i) fabs()    (ii) isdigit()
4. Name the header files in which the following belong:  
(i) puts()    (ii) close()
5. Name the header file(s) that shall be needed for successful compilation of the following C++ code:  

```
void main()
{
    char string[20];
    gets(String);
    strcat(String, "APS");
    puts(String);
}
```
6. Name the header files that shall be needed for the following code:  

```
void main()
{
    char word[ ] = "I PRE BOARD-2017-18";
    cout<<setw(20)<<word;
}
```
7. What do you understand by cascading of I/O operators?
8. What is the purpose of default clause in a switch statement?
9. What are the applications of void data type in C++?
10. What is the effect of absence of break in switch-case statement?
11. Give the output of the following program segment:  

```
void main()
{
    int i = 729, j = 9;
    cout<< i/j << endl;
}
```
12. When will you make a function inline?
13. What is the significance of any empty parentheses in a function declared?
14. How are abstraction and encapsulation inter-related?
15. Write the declaration of inline function named bar() that takes one argument of type float and return type int.
16. Observe the following C++ and write the name(s) of the header file(s), which will be essentially required to run it in a C++ compiler.  

```
void main()
{
    float Area, Side;
    cin>>Area;
    Side = sqrt (Area);
    cout<< "One Side of the Square="<<Side<<endl ;
}
```

17. Observe the following C++ code and write the name(s) of the header file(s), which will be essentially required to run it in a C++ compiler.

```
void main ()
{
    Int Number;
    cin>>Number;
    if (abs (Number) == Number);
        cout<< "Positive"<<endl;
}
```

18. Name the header file(s), which are essentially required to run the following program segment.

```
void main ()
{
    char A= 'K',B;
    if (islower (A) )
        B=toupper (A);
    else
        B= '*';
    cout<<A>> "turned to" <<B<<endl;
}
```

19. Write the names of the headers files to which the following belong:

(i) puts ( )      (ii) sin ( )

20. Write the name of the header file to which the following belong:

(i) puts ( )      (ii) randomize ( )

21. Observe the following C++ code and write the name(s) of the header file (s), which will be essentially required to run in a C++ compiler.

```
void main ( )
{
    char Text[20],c;
    cin>>Text;
    C=tolower (Text[0]);
    cout<<C<< "is the first char of"<<Text<<endl;
}
```

22. Observe the following C++ code and write the name(s) of the header file (s), which will be essentially required to run in a C++ compiler.

```
void main ( )
{
    char CH, STR[20];
    cin>>STR;
    CH=toupper(STR[0]);
    cout<<STR<< "starts with"<<CH<<endl; }
```

23. Which C++ header files(s) are essentially required to be included to run/execute the following C++ source code?

```
void main ( )
{char STRING[ ]=Something";
cout<<"Balance Characters :"<<160-
strlen (STRING)<<CH<<endl;
```

```
}
```

(Note Do not include any header file, which is /are not required).

24. Which C++ header file(s) are essentially required to be include to run/execute the following C++ source code?

```
void main ( )  
{  
    char Text[ ]=Something";  
    cout<< "Remaining SMS Chars:"<< "160-  
    strlen (STRING) <<endl;  
}
```

(Note Do not include any header file, which is /are not required).

25. Write the names of the header files, which is/are essentially required to run, execute the following C++ code.

```
void main ( )  
{  
    char C, string[ ]= "Excellence Overload";  
    for (int I=0; string[ I ] != '\0';I++)  
    if(string[ I ]== ' ' )  
        cout<<endl;  
    else  
    {  
        C=toupper (Text[ I ] ) ;  
    }  
    cout<<C;  
}
```

26. Write the names of the header files, which is/are essentially required to run/execute the following C++ code.

```
void main ( )  
{  
    char CH, Text[ ]= "+ve Altitude";  
    for (int I =0; Text [ I ] != '\0';I++)  
    if(Text[ I ]== 1 )  
        cout<<endl;  
    else  
    {  
        CH =toupper (Text[ I ] ) ;  
    }  
    cout<<CH;  
}
```

27. Which C++ header file(s) will be essentially required to be included to run/execute the following C++ code?

```
void main ( )  
{  
    Int Rno=24;  
    char Name[]= "Aman Singhania";  
    cout<<setw(10)<<Rno<<setw(20)  
    <<Name<<endl;  
}
```

28. Name the header files that shall be needed for the following code

```
void main ( )  
{  
    char Text[ ]= "computers";  
    cout<<setw(15)<<Text;  
}
```

29. Name the header files that are essential to run the following code segment successfully

```
void main()  
{ char ch[10];  
    cout<<"enter ur name \n";  cin.getline(ch,10);  
    int l=strlen(ch);  
    cout.write(ch,l);  
}
```

30. Find the correct identifiers out of the following, which can be used for naming variable, constants or functions in a C++ program :

While, for, Float, new, 2ndName, A%B, Amount2, \_Counter

31. Find the correct identifiers out of the following, which can be used for naming Variable, Constants or Functions in a C++ program :

For, while, INT, NeW, delete, 1stName, Add+Subtract, name1

32. Name the header files that are essential to run the following code segment successfully

```
void main()  
{  
    int r;  
    float A,B;  
    cout<<"enter the radius of circle\n";  
    cin>>r;  
    A=3.14*pow(r,2);  
    B=32767;  
    cout<<"area =\t"<<A<<endl;  
    cout<<B;  
}
```

33. Which of the following are valid identifiers?

Data\_rec Data rec, 1 data, data 1, my.file, asm, switch, goto

34. What is the result of following expression? Y= (t=4, t+3);

35. What is the significance of preprocessor directive?

36. Why is char often treated as integer data type in C++ ?

37. Why main function is special in C++ ?

38. Write the names of the headers files to which the following belong:  
randomize(), getch( )

39. Write the names of the headers files to which the following belong:  
fabs(), atoi( )

40. Observe the following program very carefully and write the names of those header file(s), which are essentially needed to compile and execute the following program successfully:

```
typedef char TEXT[80];
void main()
{
    TEXT Str[] = "Peace is supreme";
    int Index=0;
    while (Str[Index]!='\0')
    if (isupper(Str[Index]))
    Str[Index++]='#';
    else
    Str[Index++]='*';
    puts(str);
}
```

41. Observe the program segment given below carefully and fill the blanks marked as Statement 1 and Statement 2 using seekg( ), seekp( ), tellp( ) and tellg( ) functions for performing the required task.

```
#include<fstream.h>
class PRODUCT
{int Pno; char Pname[20]; int Qty;
public :
    void ModifyQty( ) ; // The function is to modify quantity of a
PRODUCT
    } ;
    void PRODUCT: :ModifyQty ( )
{ fstream File ;
    File.open ("PRODUCT.DAT", ios::binary |ios :: in| ios::out) ;
    int MPno;
    cout<<"Product No to modify quantity : ";
    cin>>MPNo;
    while (File.read ((char*) this, sizeof(PRODUCT))
    { if (MPno == Pno)
        { cout<<"Present Quantity:"<<Qty<<endl ;
        cout<<"Changed Quantity:";cin>>Qty ;
        int Position = _____; //Statement 1
        _____; // Statement 2
        File.write ((char*) this, sizeof (PRODUCT)) ; //Re-writing the
record
        } }
    File.close ( );
}
```

42. Find the output of the following C++ code considering that the binary file MEMBER.DAT exists on the hard disk with records of 100 members:

```
class MEMBER
{
int Mno; char Name[20];
public:
void In();void Out();
};
void main()
{ fstream MF;
```

```

MF.open("MEMBER.DAT",ios::binary|ios::in);
MEMBER M;
MF.read((char*)&M,sizeof(M));
MF.read((char*)&M,sizeof(M));
MF.read((char*)&M,sizeof(M));
int POSITION=MF.tellg()/sizeof(M);
cout<<"PRESENT RECORD:"<<POSITION<<endl;
MF.close(); }

```

43. Observe the program segment given below carefully and fill the blanks marked as Statement 1 and Statement 2 using tellg() and seekp() functions for performing the required task.

```

#include <fstream.h>
class Client
{
    long Cno;
    charName[20],Email[30];
public:
    //Function to allow user to enter
    //the Cno, Name,Email
    void Enter();
    //Function to allow user to enter
    //(modify) Email
    void Modify();
    long ReturnCno()
    {
        return Cno;
    }
};

void ChangeEmail()
{
    Client C;
    fstream F;
    F.open("INFO.DAT",ios::binary|ios::in|ios::out);
    long Cnoc; //Client's no. whose
    //Email needs to be changed
    cin>>Cnoc;
    while (F.read((char*)&C,
    sizeof(C)))
    {
        if (Cnoc==C.ReturnCno())
        {
            C.Modify(); //Statement 1
            int Pos = _____
            //To find the current position
            //of file pointer
            _____ // Statement 2
            //To move the file pointer to write the modified record back onto the file for the desired Cno
            F.write((char*)&C, sizeof(C));
        }
    }
    F.close();
}

```

44. Observe the program segment given below carefully and fill in the blanks marked as Line 1 and Line 2 using fstream functions for performing the required task.

```
#include <fstream.h>
class Stock
{
    long Ino ; //Item Number
    char Item [20] ; //Item Name
    int Qty ; //Quantity
public:
    void Get(int);
    //Function to enter the content
    void show( );
    //Function to display the content
    void Purchase (int Tqty)
    {
        Qty + = Tqty ;
    } //Function to increment in Qty
    long KnowIno ( )
    {
        return Ino;
    }
};
void Purchaseitem(long PINo, int PQty)
//PINo -> Ino of the item purchased
//PQty -> Number of item purchased
{f
    stream File;
    File.open("ITEMS.DAT", ios ::binary|ios ::in|ios :: out);
    int Pos = -1 ;
    Stock S ;
    while (Pos == -1 && File.read((char*) &S,sizeof (S)))
    if (S. KnowIno( ) ==PINo)
    {
        S. Purchase (PQty);
        //To update the number of Items
        Pos = File.tellg ( ) -sizeof (S) ;
        _____;
        //Line 1: To place the file
        //pointer to the required position
        _____;
        //Line 2: To write the object s on to
        //the binary file
    }i
    f (Pos == -1)
    cout<<"No updation done as
    required Ino not found.." ;
    File.close ( ) ;

}
```

45. Observe the program segment given below carefully, and answer the question that follows  
class candidate

```
{ long Cid ; // Candidate's Id
char CName[20]; // Candidate's Name
float Marks ; // Candidate's Marks
public ;
void Enter( ) ;
void Display( ) ;
void MarksChange( ) ;
//Function to change marks
long R_Cid( )
{ return Cid;
}
};
void MarksUpdate (long Id)
{
fstream File ;
File.open ("CANDIDATE.DAT", ios ::binary|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.MarksChange( ) ;
_____ //Statement1
_____ //Statement 2
Found = 1 ;
}
Record++ ;
}
if (Found == 1)
cout << " Record Updated" ;
File.close( ) ;
}
```

Write the Statement to position the File Pointer at the beginning of the Record for which the Candidate's Id matches with the argument passed, and Statement 2 to write the updated Record at that position.

46.

Fill in the blanks marked as Statement 1 and Statement 2 in the program segment given below with appropriate functions for the required task.

```
class Club
{
    long int Mno ;           //Member number
    char Mname [20] ;        //Member name
    char Email [30] ;        //Email of member
public:
    void Register ( ) ;      //Function to register member
    void Disp ( ) ;          //Function to display details
    void ChangeEmail ( )     //Function to change Email
    {
        cout<< "Enter Changed Email:";
    }
}
```



```

        long int GetMno ( )
        {
            Return Mno ;
        }
};
void ModifyData ( )
{
    fstream File ;
    File.open ("CLUB DAT".ios : :binary | ios : : out);
    int Modify = 0. Position;
    long int ModiMno;
    cout<< "Mno-whose email required to modified:";
    cin>>ModiMno;
    Club CL;
    while(!Modify&&File.read((char*)&CL,sizeof(CL)))
    {
        If(CL.GetMno( ) == ModiMno)
        {
            CL.ChangeEmail ( );
            Position = File.tellg( )-sizeof(CL);
            //Statement 1: To place file pointer to the required position
            ;
            //Statement 2: To write the object CL onto the binary file
            ;
            Modify++;
        }
    }
    If (Modify)
        cout<< "Email Changed. . ."<<end];
    else
        cout<< "Member not found. . ."<<end];
    File.close( );
}

```

47. Observe the program segment given below carefully and answer the questions that follow:

```

class Stock
{
    int Ino,Qty; char Item[20];
public:
    void Enter() {cin>>Ino;gets(Item); cin>>Qty;}
    void Issue(int Q) { Qty+=Q}
    void Purchase(int Q) { Qty-=Q}
    int GetIno{return Ino;}
};
void Purchaseitem(int Pino, int PQty)
{
    fstream file;
    File.open("STOCK.DAT",ios::binary|ios::in|ios::out);
    Stock S;
    int Success=0;
    while (Success==0 && File.read((char*)&S, sizeof(S)))

```

```

{
if (Pino==S.GetIno())
{
S.Purchase(PQty);
_____//Statement 1
_____//statement 2
Success++;
}
}
if (Success==1)
cout<<"Purchase Updated"<<endl;
else
cout<<"Wrong Item No"<<endl;
File.close( );
}

```

- (i) Write statement 1 to position the file pointer to the appropriate place, so that the data updation is done for the required item.
- (ii) Write statement 2 to perform the write operation so that the updation is done in the binary file.

48.

Observe the program segment given below carefully and fill the blanks marked as Line 1 and Line 2 using fstream functions for performing the required task.

```

#include <fstream.h>
class Library
{
long Ano;           //Ano - Accession Number of the Book
char Title[20];     //Title - Title of the Book
int Qty;           //Qty - Number of Books in Library
public:
void Enter (int);   //Function to enter the content
void Display();    //Function to display the content
void Buy(int Tqty)
{
Qty+=Tqty;
} //Function to increment in Qty
long GetAno( )
{
return Ano;
}
};
void BuyBook(long BANO,int BQty)    //BANO ->Ano of the book purchased
                                   //BQty ->Number of books purchased

{ fstream File;
File.open("STOCK.DAT" ,ios::binary|ios::in|ios::out);
int position=-1;
Library L;
while(Position== -1 &&File.read((char*)&L,sizeof(L)))
if (L.GetAno()==BANO)
{
L.Buy(BQty);
//To update the number of Books
Position = File.tellg()-sizeof(L) ;
_____
};

```

```

//Line 1: To place the file pointer to the required position
_____;
//Line 2: To write the object L on to the binary file
} if (Position==l)
cout<< "No updation done as
required And not found..";
File.close( );
}

```

49. Observe the program segment given below carefully, and answer the question that follows:

```

class Labrecord
{ int Expno;
  char Experiment[20] ;
  char Checked ;
  int Marks ;
public :
  void EnterExp( ) ;
  //function to enter Experiment details
  void ShowExp( ) ;
  //function to display Experiment details
  char RChecked( )
  //function to return Expno
  {
  return Checked;
  }
  void Assignmarks (int M)
  //function to assign Marks
  {
  Marks = M;
  }
};
void ModifyMarks( )
{
  fstream File ;
  File.open ("Marks.Dat", ios :: binary | ios :: in | ios :: out) ;
  Labrecord L ;
  int Rec=0 ;
  while (File.read ( (char*) &L,sizeof (L)))
  {
  if (L.RChecked( ) != 'N')
  L.Assignmarks(0)
  else
  L.Assignmarks (10)
  _____; //Statement 1
  _____; //Statement 2
  Rec++ ;
  }
  File.close( ) ;
}

```

If the function ModifyMarks ( ) is supposed to modify marks for the records in the file MARKS.DAT based on their status of the member Checked (containing value either 'Y' or 'N'). Write C++ statements for the statement 1 and statement 2, where, statement 1 is required to position the file write pointer to an appropriate place in the file and statement 2 is to perform the write operation with the modified record.

50. Observe the program segment given below carefully , and answer the question that follows :

```
class Member
{
int Member_no ;
char Member_name[20] ;
public :
void enterdetails ( ) ; //function to enter Member details
void showdetails ( ) ; //function to display Member details
int RMember_no( )
{
return Member_no;
} //function to return Member_no
};
void Update (Member NEW)
{
fstream File ;
File.open("MEMBER.DAT" , ios ::
binary | ios :: in | ios :: out) ;
Member OM ;
int Recordsread = 0, Found = 0 ;
while (!Found && File.read((char*)& OM, sizeof(OM)))
{
Recordsread++ ;
if (NEW.RMember_no( ) ==OM.RMember_no( ))
{
_____ //Missing Statement
File.write((char*) & NEW ,
sizeof(NEW) ;
Found = 1 ;
}
else
File.write((char*) & OM,
sizeof(OM)) ;
}
if (!Found)
cout<<"Record for modification does
not exist" ;
File.close( ) ;
}
```

If the function Update( ) is supposed to modify a record in file MEMBER.DAT with the values of Member NEW passed to its argument, write the appropriate statement for **Missing statement** using seekp( ) or seekg( ) , whichever needed, in the above code that would write the modified record at its proper place.

## **2 Marks Questions**

1. What is the difference between 'x' and "x" in C++?
2. What is a reference variable? What is its use?
3. What will be the output of the following program segment:  

```
#include <iostream.h>
void main()
{
    int r, x,y;
    x = 50;
    y = 10;
    r = (x<45) ? x : y;
    cout << r;
}
```
4. Differentiate between break and continue statements.
5. In a control structure switch-case, explain the purpose of using default.
6. Differentiate between getch() and getche() function.
7. When are temporary variables created by C++ compiler?
8. How cout and puts() differ from each other?
9. Give the difference between the type casting and automatic type conversion. Also, give a suitable C++ code to illustrate both
10. What is the difference between Local Variable and Global Variable? Also, give a suitable C++ code to illustrate both.
11. What is the purpose of using a typedef command in C++? Explain with suitable example.
12. What is the difference between #define and const? Explain with suitable example.
13. What is the difference between Actual Parameter and Formal Parameters? Also, give a suitable C++ code to illustrate both.
14. What is the difference between call by reference and call by value with respect to memory allocation? Give a suitable example to illustrate C++ code.
15. What is benefit of using default parameter/argument in a function? Give a suitable example to illustrate it using C++ code.
16. What is the benefit of using function prototype for a function? Give a suitable example to illustrate it using a C++ code.
17. What is the significance of classes in OOPs?
  18. What do you understand by polymorphism? Give an example illustrating its use in a C++ program.
  19. Define the term data encapsulation in term of object oriented programming. Give a suitable example using a C++ code to illustrate the same.
  20. Define the term data hiding in the context of object oriented programming give a suitable example using a C++ code to illustrate the same.
  21. Encapsulation is one of the major properties of OOP. How is it implemented in C++?
  22. How are abstraction and encapsulation interrelated?
  23. What is event driven programming?
  24. What do you understand by function overloading? Give an example illustrating its use in a C++ program.
  25. What the operator overloading? Explain with example.
  26. What are the advantages of object oriented programming over procedural programming?

27. What are the major differences between Object Oriented Programming and Procedural Programming?
28. What are syntax, run-time errors and logical errors?
29. What is the difference between while and do-while loop?
30. How is OOP implemented in C++?
31. What is escape sequence? Name any two escape sequences.
32. What are type modifiers? Give examples.
33. Define abstract class and concrete class.
34. What is a constructor? Explain its use with a suitable example.
35. Differentiate between Default Constructor and parameterized Constructor.
36. Differentiate between constructor and destructor.
37. What is copy constructor? Give an example in C++ to illustrate copy constructor.
38. Give any four properties of a constructor.
39. Give any four properties of a destructor.
40. What is constructor overloading? Explain default arguments and overloading.
41. How are binary files different from text files in C++?
42. What is a stream? Name the streams generally used for file I/O.
43. Difference between `get()` and `getline()`.
44. Difference between `ios::app` and `ios::out`.
45. What is pointer?
46. Differentiate between Protected and Private members of a class in context of inheritance using C++.
47. Define Multilevel and Multiple inheritance in context of Object Oriented Programming. Give suitable example to illustrate the same.
48. Differentiate between public and protected visibility in context of Object Oriented Programming giving suitable examples for each.
49. How does the invocation of constructor differ in derivation of class and nesting of classes?
50. What is the difference between the constructor and normal function?

51. Observe the following C++ code very carefully and rewrite it after removing any/all syntactical errors with each correction underlined.

Note: Assume all required header files are already being included in the program.

```
#Define float Max=70.0;
Void main()
{
int Speed
char Stop='N';
cin>>Speed;
if Speed>Max
Stop='Y';
cout<<Stop<<end;
}
```

52.

Observe the following C++ code very carefully and rewrite it after removing any/all syntactical errors with each correction underlined.

Note: Assume all required header files are already being included in the program.

```
const Max=70.0;
void main[]
{
int Speed;
char Stop='N';
cin<<Speed;
if (Speed>Max)
Stop='Y';
cout<<Stop<<endl;
}
```

53.

Rewrite the following program after removing the syntactical error(s) if any. Underline each correction.

```
#include<iostream.h>
void main( )
{ First = 10, Second = 20;
Jumpto(First;Second);
Jumpto(Second);
}
void Jumpto(int N1, int N2 = 20)
{ N1=N1+N2;
count<<N1>>N2;
}
```

54.

Rewrite the following program after removing the syntactical error(s) if any. Underline each correction.

```
#include<iostream.h>
const int Max 10;
void main()
{ int Numbers[Max];
Numbers = {20,50,10,30,40};
for(Loc=Max-1;Loc>=10;Loc--)
cout>>Numbers[Loc];
}
```

55.

Rewrite the following program after removing the syntactical error(s), if any. Underline each correction.

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

56.

Rewrite the following program after removing all the syntax error(s), if any.

```
#include<iostream.h>
void main() {
int X[]={60, 50, 30, 40},Y;Count=4;
cin>>Y;
for(I=Count-1;I>=0,I--)
switch(I)
{ case 0:
case 2:cout<<Y*X[I]<<endl;break;
case1:
case 3:cout>>Y+X[I];
}}
}
```

57.

Rewrite the following program after removing all the syntax error(s), if any.

```
#include<iostream.h>
void main() {
int P[]={90, 10, 24, 15},Q;Number=4;
Q=9;
for(int I=Number-1;I>=0,I--)
switch(I)
{ case 0:
case 2:cout>>P[I]*Q<<endl;
break;
case 1:
case 3:cout<<P[I]+Q;
}
}
```



58.

Rewrite the following program after removing all the syntactical error(s), if any. Underline each correction.

```
#include<iostream.h>
void main() {
    Present=25,Past=35;
    Assign(Present;Past);
    Assign(Past);
}
void Assign(int Default1,Default2=30)
{
    Default1=Default1+Default2;
    cout<<Default1>>Default2;
}
```

59.

Rewrite the following program after removing all the syntactical error(s), if any. Underline each correction.

```
#include<iostream.h>
void main() {
    One=10,Two=20;
    Callme(One;Two);
    Callme(Two);
}
void Callme(int Arg1,int Arg2=20)
{
    Arg1=Arg1+Arg2
    cout<<Arg1>>Arg2;
}
```

60.

Rewrite the following program after removing all the syntactical error(s), if any. Underline each correction.

```
#include<iostream.h>
typedef char[80];
void main() {
    String S="Peace";
    int L=strlen(S);
    cout<<S<<'has '<<L<<'characters '<<endl;
}
```

61.

What are programming paradigms? Give names of some popular programming paradigms.

62. Reusability of classes is one of the major properties of OOP. How is it implemented in C++?

63. Write a short note on OO programming.

64.

Rewrite the following program after removing the syntactical errors (if any). Underline each correction.

```
#include[iostream.h]
typedef char Text(80) ;
void main ( )
{
    Text T= "Indian";
    int Count=strlen(T) ;
    cout<<T<<'has '<<Count<<'characters '<<endl;
```

65. }  
Observe the following C++ code very carefully and rewrite it after removing any/all syntactical errors with each correction underlined.

Note: Assume all required header files are already being included in the program.

```
typedef char[50] STRING
void main()
{
    City STRING;
    gets(City);
    cout<<City[0]<<"\t,<<City[2];
    cout<<City<<endl;
}
```

66.

Study the following program and select the possible output(s) from the option (i) to (iv) following it. Also write the maximum and the minimum values that can be assigned to the variable NUM.

Note:- Assume all required header files are already being included in the program.

- random(n) function generates an integer between 0 and n-1.

```
void main()
{
    randomize();
    int NUM;
    NUM=random(3)+2;
    char TEXT[]="ABCDEFGHGIJK";
    for (int I=1;I<=NUM; I++)
    {
        for (int J=NUM;J<=7;J++)
            cout<<TEXT[J];
        cout<<endl;
    }
}
```

(i) FGHI	(ii) BCDEFGH	(iii) EFGH	(iv) CDEFGH
FGHI	BCDEFGH	EFGH	CDEFGH
FGHI		EFGH	
FGHI		EFGH	

67.

Go through the C++ code shown below, and find out the possible output or outputs from the suggested Output Options (i) to (iv). Also, write the least value and highest value, which can be assigned to the variable Guess.

```
#include<iostream.h>
#include<stdlib.h>
void main ( )
{
    randomize ( ) ;int Guess, High=4;Guess=random(High)+ 50 ;
    for(int C=Guess ; C<=55 ; C++)
        cout<<C<<"#" ;}
```

i) 50 # 51 # 52 # 53 # 54 # 55 #  
(iii) 53 # 54 #

(ii) 52 # 53 # 54 # 55  
(iv) 51 # 52 # 53 # 54 # 55

68. Go through the C++ code shown below, and find out the possible output or outputs from the suggested Output Options (i) to (iv). Also find out the minimum and maximum value that can be assigned to Guess at the time when value of Turn is 3.

```
#include<iostream.h>
#include<stdlib.h>
void main()
{
    char Result[][10]={"GOLD","SILVER","BRONZE"};
    int Gt=9,Guess;
    for(int Turn=1;Turn<4;Turn++)
    {
        Guess=random(Turn);
        cout<<(Gt-Guess)<<Result[Guess]<<"*";
    }
}
```

(i) 9GOLD\*9GOLD\*8SILVER\*  
(ii) 9GOLD\*7BRONZE\*8GOLD\*  
(iii) 9GOLD\*8SILVER\*9GOLD\*  
(iv) 9GOLD\*8SILVER\*8GOLD\*

69. Observe the code carefully and find which output(s) will be expected from the program? and justify your answer:

```
#include <iostream.h>
#include <stdlib.h>
const int K=2;
void main()
{
    randomize();
    int A;
    A=random(K)+2;
    for(int i=A;i<3;i++)
        cout<<i<<" ";
    cout<<endl;
}
```

(i) 1,2,                      (ii) 0,1,2                      (iii) 2,                      (iv) 0,1,2,

70.

Observe the code carefully and find which output(s) will be expected from the program?

Justify your answer:

```
#include <iostream.h>
#include <stdlib.h>
void main()
{
    randomize(); int A; A=2+random(3);
    for(int i=A;i<5;i++)
        cout<<'# '<<i;
}
```

#1#2#3                      (ii) #2#3#4                      (iii) #4#3#2                      (iv) None of these

71.

Observe the code carefully and find which output(s) will not be expected from the program?

Justify your answer:

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

```
#include <stdlib.h>
```

```
const int K=4;
```

```
void main()
```

```
{  randomize();
```

```
  int A;
```

```
  A=2+random(K);
```

```
    for(int i=0;i<A;i++)
```

```
      cout<<i<<",";
```

```
}
```

(i) 0,1, (ii) 0,1,2, (iii) 0,2,4, (iv) 0,1,2,3,4,5,

72.

Observe the code carefully and select most possible answer from the choices given below and justify your

answer:

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

```
#include <stdlib.h>
```

```
#define K 4
```

```
void main()
```

```
{  randomize();
```

```
  int A;
```

```
  A=20+random(K);
```

```
    for(int i=A;i>=20;i--)
```

```
      cout<<i<<"^";
```

```
  cout<<endl;
```

```
}
```

(i)  $22^{21}20^{19}$

(ii)  $24^{23}22^{21}20^{\wedge}$

(iii)  $20^{21}22^{\wedge}$

(iv)  $20^{\wedge}$

Observe the following code carefully, if the value of num entered by user is 4, choose the correct possible output(s) from the options from (i) to (iv) and justify your option.

Output options: (i) 1 2 3                  (ii) 5 4 3 2 1  
                    (iiii) 1 2 3 4 5 6         (iv) 1 2 3 4

Read the following C++ code carefully and find out, which out of the given options (i) to (iv) are the expected correct output(s) of it. Also, write the maximum and minimum value that can be assigned to the variable Start used in the code :

(i) 200#150# (ii) 150#20# (iii) 150#20#250# (iv) 20#250#

Read the following C++ code carefully and find out, which out of the given options (i) to (iv) are the expected correct output(s) of it. Also, write the maximum and minimum value that can be assigned to the variable **Taker** used in the code :

(i) 100#  
(ii) 50#200#  
(iii) 100#50#200#  
(iv) 100#50

76.

Write the output of the following C++ program code:

Note: Assume all required header files are already being included in the program .

```
void Position (int &C1, int C2=3)
{
    C1+=2;
    C2+=Y;
}
void main()
{
    int P1=20, P2=4;
    Position(P1);
    cout<<P1<<" , "<<P2<<endl;
    Position(P2,P1);
    cout<<P1<<" , "<<P2<<endl;
}
```

77. Write the output of the following C++ program code:

Note: Assume all required header files are already being included in the program.

```
void Location(int &X,int Y=4)
{
    Y+=2;
    X+=Y;
}
void main()
{
    int PX=10,PY=2;
    Location(PY);
    cout<<PX<<" , "<<PY<<endl;
    Location(PX,PY);
    cout<<PX<<" , "<<PY<<endl;
}
```

78. Write the output of the following C++ program code:

```
#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]=='N'||Msg[cnt]=='n'||Msg[cnt]==' ')
                Msg[cnt]=CH;
            else
                if(cnt%2==0)
```

```

    Msg[cnt]=toupper(Msg[cnt]);
else
    Msg[cnt]=Msg[cnt-1];
} }
void main()
{ char MyText[]="Input Raw";
  Mycode(MyText,'@');
  cout<<"NEW TEXT:"<<MyText<<endl;
}

```

79.

Obtain the output from the following C++ program as expected to appear on the screen after its execution.

Important Note : - All the desired header files are already included in the code, which are required to run the code.

```

void main()
{ char *Text="AJANTA";
  int *P, Num[]={1,5,7,9}; P=Num;
  cout<<*P<<Text<<endl; Text++;
  P++;
  cout<<*P<<Text<<endl;
}

```

80 . Obtain the output from the following C++ program as expected to appear on the screen after its execution.

```

#include<iostream.h>
void SwitchOver(int A[],int N,int split)
{for(int K=0;K<N;K++)
  if(K<split)
    A[K]+=K;    else  A[K]*=K;
}
void Display(int A[],int N)
{ for(int K=0;K<N;K++)
  (K%2==0)?cout<<A[K]<<"%":cout<<A[K]<<endl;
}
void main(){int H[]={30,40,50,20,10,5};
  SwitchOver(H,6,3);
  Display(H,6);
}

```

```
}
```

81.

Find the output of the following program :

```
#include <iostream.h>
#include <ctype.h>
void Changelt(char Text[ ], char C)
{ for (int K=0;Text[K]!='\0';K++)
{ if (Text[K]>='F' && Text[K]<='L')
Text[K]=tolower (Text[K]);
else
if (Text[K]=='E' || Text[K]=='e')
Text[K]=C;
else
if (K%2==0)
Text[K]=toupper(Text[K]);
else
Text[K]=Text[K-1];
}
}
void main ( )
{
char OldText[ ]="pOwERALone";
Changelt(OldText,'%');
cout<<"New TEXT:"<<OldText<<endl;
}
```

82.

Find. the output of the following program:

```
#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;}
```

83. Find the output of the code segment given below:

```
#include<iostream.h>
void main( )
{ int A=5,B=10;
  for(int l=1;l<=2;l++)
  { cout<<"Line1"<<A++<<"&"<<B-2 <<endl;
    cout<<"Line2"<<++B<<"&"<<A+3 <<endl;
  }
}
```

84. Find the putput.

```
#include<iostream.h>
void main( )
{ long int NUM=1234543;
  int F=0,S=0;
  do
  { int R=NUM % 10;
    if (R %2!= 0)
    F+= R;
    else
    S+= R;
    NUM/= 10;
  } while (NUM>0);
  cout<<F-S;
}
```

85.Find the output of the following program:

```
#include<iostream.h>
void main( )
{ int var1=5,var2=10;
  for(int i=1;i<=2;i++)
  { cout<<var1++<<"\t"<<--var2<<endl;
    cout<<var2--<<"\t"<<++var1<<endl;
  }
}
```

86.

What will be the result of following code in C++?

```
#include<iostream.h>
void main()
{ int a=4,b=2,c=6,d=1;
  cout<<(a+6)>=9+b || d*b<=10 && a+b+c/d)<<endl;
  cout<<(a--+2*b+++a/d);
}
```

87.

Find the output of code given below:

```
#include <iostream.h>
int main() { int i=0,a=0,b=0,c=0;
while(i<=4)
{ switch(i++)
{ case 1: ++a;break;
case 2:
case 3: ++b;
case 4: ++c;
default: break; } }
cout<<"a="<<a<<"b="<<b<<endl; cout<<"c="<<c;
return 0;}
```

88.

What will be the output of the following code fragment?

```
#include<iostream.h>
int x=10;
void test( int a,int &b,int c=5)
{ a+=x+c; b=x*c; c=x/c;
  cout<<a<<" "<<b<<" "<<c<<endl; }
void main( )
{ int x=10,y=20;
  test(x,y,::x);
cout<<x<<" "<<y<<" "<<::x<<endl;
  test(::x,x,y); cout<<::x<<endl; }
```

89.

Find the output of the following program:2

```
#include <iostream.h>
#include <ctype.h>
void Encode (char Info [ ], int N) ;
void main ( )
{ char Memo[ ]= "Justnow" ;
  Encode (Memo,2) ;
  cout<<Memo<<endl ;
}
void Encode (char Info [ ], int N)
{ for (int I = 0;Info[I] !='\0';I++)
  if (I%2==0)
    Info[I] = Info[I] -N ;
  else if (islower(Info[I]))
    Info[I] = toupper(Info[I]) ;
  else
    Info[I] = Info[I] +N ;
}
```

90.

Find the output of the following program:2

```
#include <iostream.h>
#include <ctype.h>
void Secret (char Mig[ ], int N);
void main ( )
{ char SMS[ ] = "rEPorTmE" ;
  Secret(SMS,2);
  cout<<SMS<<endl;
}
void Secret(char Msg[ ], int N)
{ for (int C=0; Msg[C] !=' \0' ;C++)
  if (C%2==0)
    Msg[C] = Msg[C]+N;
  else if (isupper(Msg[C]))
    Msg[C] = tolower(Msg[C]);
  else
    Msg[C] = Msg[C]-N;
}
```

91.

Obtain the output from the following c++ program as expected to appear on the screen after its execution.  
Important Note :

- All the desired header files are already included in the code, which are required to run the code.

```
#include <iostream.h>
void main ( )
{
    char *String="SARGAM";
    int *Ptr, A[]={1,5,7,9};
    Ptr=A;
    cout <<*Ptr<<String<<endl;
    String++;
    Ptr+=3;
    cout<<*Ptr<<String<<endl ;
}
```

92. Find the output of following program:

```
#include<iostream.h>
#include<conio.h>
#include<ctype.h>
typedef char str80[80];
void main()
{
    char *notes;
    str80 str="vR.zGood";
    int L=6;
    notes=str;
    while(L>=3)
    {
        str[L]=isupper(str[L])?tolower(str[L]):toupper(str[L]);
        cout<<notes<<endl;
        L--;
        notes++;
    }
    getch();
}
```

93. Find the output of following program:

```
#include<Iostream.H>
void main()
{
    float *p,val[]={30,40,30,60,10};
    p=val;
    cout<<*p<<endl;
    p+=2; val[2]+=4.5;
    cout<<*p<<endl;
```

```
p++;
(*p)+=4.5;
cout<<val[3]<<endl; }
```

94. Find the output of following program:

```
#include<IOSTREAM.H>
void main()
{
    char *str="Bhakti";
    int *p,val[]={90,115,70,19};
    p=val;
    cout<<*p<<str<<endl;
    str+=2;
    p+=2;
    cout<<*p<<str;
}
```

95. Find the output of following program:

```
#include<iostream.h>
void main()
{
    int a=32,*X=&a;
    char ch=66,&eco=ch;
    eco+=a;
    *X+=ch;
    cout<<a<<','<<ch<<endl;
}
```

96. Find the output of following program:

```
#include<iostream.h>
#include<string.h>
void main()
{
    char *a="Banka";
    for(int x=strlen(a)-1;x>=0;x--)
    { for( int y=0;y<=x;y++)
      cout<<a[y];
      cout<<endl;
    }
}
```

97. Find the output of following program:

```
#include<iostream.h>
void main( )
{ int Numbers[]={2,4,8,10};
  int *ptr=Numbers;
  for(int C=1;C<3;C++)
  { cout<<*ptr<<"@";
    ptr++;
  }
  cout<<endl;
  for(C=0;C<4;C++)
  { (*ptr)*=2;
    --ptr;
  }
  for(C=0;C<4;C++)
  cout<<Numbers[C]<<"#";
  cout<<endl;
}
```

98. Find the output of following program:

```
#include<iostream.h>
void main()
{int *Queen,Moves []={11,22,33,44};
  Queen=Moves;
  Moves [2 ]+=22;
  cout<<"Queen@"<<*Queen<<endl;
  *Queen-=11;
  Queen+=2;
  cout<<"Now@"<<*Queen<<endl;
```

```
}
```

99. Find the output of following program:

```
#include<iostream.h>
void main( )
{
  int Array[ ]={4,6,10,12};
  int *pointer=Array;
  for(int l=1;l<=2;l++)
  {
    cout<<*pointer<<"#";
    pointer++;
  }
  cout<<endl;
  for(l=1;l<=3;l++)
  { (*pointer)*=3;
    --pointer;
  }
  for(l=1;l<3;l++)
  cout<<Array[l-1]<<"@";
  cout<<endl;
}
```

100. Find the output of following program:

```
#include<iostream.h>
void main( )
{ char *p="Difficult";
  char c;
  c=*p++;
  cout<<c<<endl;
  cout<<*p;
}
```

101. Observe the following C++ code and answer the questions (i) and (ii) :

```
class Traveller
{
  long PNR;
  char TName[20];
public :
  Traveller() //Function 1
  {cout<<"Ready"<<endl;}
  void Book(long P,char N[]) //Function 2
  {PNR = P; strcpy(TName, N);}
  void Print() //Function 3
  {cout<<PNR << TName <<endl;}
  ~Traveller() //Function 4
  {cout<<"Booking cancelled!"<<endl;}
};
```

(i) Fill in the blank statements in Line 1 and Line 2 to execute Function 2 and Function 3 respectively in the following code:

```
void main()
{
  Traveller T;
  _____ //Line 1
  _____ //Line 2
} //Stops here
```

(ii) Which function will be executed at } //Stops here? What is this function referred as ?

102. Observe the following C++ code and answer the questions (i) and (ii) :

```
class Passenger
{
  long PNR;
  char Name [20] ;
public:
  Passenger() //Function 1
  { cout<<"Ready"<<endl; }
  void Book(long P,char N[]) //Function 2
  { PNR = P; strcpy(Name, N); }
  void Print() //Function 3
  { cout<<PNR << Name <<endl; }
  ~Passenger() //Function 4
  { cout<<"Booking cancelled!"<<endl; }
};
```

(i) Fill in the blank statements in Line 1 and Line 2 to execute Function 2 and Function 3 respectively in the following code:

```
void main()  
{  
Passenger P;  
____ //Line 1  
____ //Line 2  
} //Ends here
```

(ii) Which function will be executed at } //Ends here? What is this function referred as?

103. Answer the questions (i) and (ii) after going through the following class :

```
class Health  
{  
int PId, DId;  
public:  
Health (int PPId) ; // Function 1  
Health ( ) ; // Function 2  
Health (Health &H); //Function 3  
void Entry ( ) ; // Function 4  
void Display( ) ; // Function 5  
};
```

```
void main ( )  
{  
Health H (20) ; //statement 1  
}
```

(i) Which of the function out of Function 1,2,3, 4 or 5 will get executed when the Statement I is executed in the above code ?

(ii) Write a statement to declare a new object G with reference to already existing object H using Function 3.

104.

Answer the questions (i) and (ii) after going through the following class:

```
class TEST  
{  
int Regno, Max, Min, Score;  
public:  
TEST() //Function 1  
{  
Regno= 101;  
Max=100;  
Min=40;  
Score=75;  
}  
TEST(int Pregno,int Pscore) //Function 2  
{  
Regno=Pregno;  
Max=100;  
Min=40;  
Score=Pscore;  
}
```



```

~TEST() //Function 3
{
cout<<"TEST Over"<<endl;
}
void Display() //Function 4
{
cout<<Regno<<":"<<Max<<":"<<Min<<endl;
cout<<"[Score]"<<Score<<endl;
};

```

**(i)** As per Object Oriented Programming, which concept is illustrated by **Function 1** and **Function 2** together?

**(i)** What is **Function 3** specifically referred as? When do you think, **Function 3** will be invoked/called?

105.

**Answer the questions (i) and (ii) after going through the following class:**

```

class Race
{
    int CarNo, Track;
public:
Race(); // Function 1
Race(int CN); // Function 2
Race(Race &R); // Function 3
void Register(); // Function 4
void Drive(); // Function 5
};
void main()
{
    Race R;
}

```

**(i)** Out of the following, which of the option is correct for calling Function 2?

Option 1- Race T(30);

Option 2- Race U(R);

**(ii)** Name the feature of Object Oriented Programming, which is illustrated by Function 1, Function 2 and Function 3 combined together.

106. Answer the questions after going through the following class.

```

class Exam
{
    char Subject[20] ;
    int Marks ;
public :
    Exam() // Function 1
    {
        strcpy(Subject, "Computer" ) ; Marks = 0 ;}

    Exam(char P[ ]) // Function 2
    {
        strcpy(Subject, P) ;
        Marks=0 ;
    }

    Exam(int M) // Function 3
    {
        strcpy(Subject, "Computer") ; Marks = M ;
    }

    Exam(char P[ ], int M) // Function 4
    {
        strcpy(Subject, P) ; Marks = M ;    } };

```

- (i) Which feature of the Object Oriented Programming is demonstrated using Function 1, Function2, Function 3 and Function 4 in the above class Exam?
- (ii) Write statements in C++ that would execute Function 3 and Function 4 of class Exam.

107. Answer the questions (i) and (ii) after going through the following class:

```
class Exam
{
    int Rno,MaxMarks,MinMarks,Marks;
public:
    Exam ( ) //Module 1
    {
        Rno=101;
        MaxMarks=100;
        MinMarks=40;
        Marks=75;
    }
    Exam (int Prno, int Pmarks) //Module 2
    { Rno=Prno;
      MaxMarks=100;
      MinMarks=40;
      Marks=Pmarks;
    }
    ~Exam () //Module 3
    {
        cout<<"Exam Over"<<endl;
    }
    void Show () //Module 4
    {
        cout<<Rno<<":"<<MaxMarks<<":"<<MinMarks<<endl;
        ;
        cout<<"[Marks Got]"<<Marks<<endl;
    }
};
```

- (i) As per Object Oriented Programming, which concept is illustrated by **Module 1** and **Module2** together?
- (ii) What is **Module 3** referred as ? When do you think, **Module 3** will be invoked/called?

108.

Consider the following declaration:

```
class welcome
{
public:
    welcome (int x, char ch); // constructor with parameter
    welcome(); // constructor without parameter
    void compute();
private:
    int x; char ch;
};
```

Which of the following are valid statements?

```
welcome obj (33, 'a9');
welcome obj1(50, '9');
```

```
welcome obj3();  
obj1= welcome (45, 'T');
```

109.

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

```
class TestMeOut  
{ public:  
~TestMeOut( ) //Function 1  
{  
cout<<"Leaving the examination hall"<<endl;  
}  
TestMeOut( ) //Function 2  
{  
cout<<"Appearing for examination"<<endl;  
}  
void MyWork( )  
{  
cout<<"Attempting Questions"<<endl;  
}  
};
```

- (i) In Object Oriented programming what is Function 1 referred as and when does it get invoked/called?
- (ii) In Object Oriented Programming, what is Function 2 referred as and when does it get invoked/called?

110.

Answer the following questions (i) and (ii) after going through the following class.

```
class Test  
{ char Paper[20];  
int Marks  
public:  
Test() //Function 1  
{ strcpy(Paper, "Computer");  
Marks=0;  
}  
//Function 2  
Test(char P[])  
{ strcpy(Paper,P);  
Marks=0;  
}  
//Function 3  
Test(int M)  
{ strcpy(Paper,"Computer");  
Marks=M;  
}  
Test(char P[],int M)  
//Function 4  
{ strcpy(Paper,P);  
Marks=M;  
}  
};
```

- (i) Which feature Object Oriented programming is demonstrated using Function 1, Function 2, Function 3 and Function 4 in the above class text?
- (ii) Write statements in C++ that would execute Function 2 and Function 4 of class Text.

111.

Write function definition for SUCCESS () in C++ to read the content of a text file STORY.TXT count the presence of word STORY and display the number of occurrence of this word.

Note : The word STORY should be an independent word.

Ignore type cases (i.e. lower/upper case)

112.

Write function definition for TOWER() in C++ to read the content of a text file WRITEUP.TXT, count the presence of word **TOWER** and display the number of occurrences of this word.

Note :- The word TOWER should be an independent word

- Ignore type cases (i.e. lower/upper case)

Example:

If the content of the file WRITEUP.TXT is as follows:

Tower of hanoi is an interesting problem. Mobile phone tower is away from here. Views from EIFFEL TOWER are amazing.

The function TOWER () should display the following:3

113.

Write a function EUCount() in C++, which should read each character of a text file IMP.TXT, should count and display the occurrence of alphabets E and U (including small cases e and u too).

Example :

If the file content is as follows:

Updated information is simplified by official websites.

The EUCount() function should display the output as:

E:4

U:1

114.

Write a function AECCount() in C++, which should read each character of a text file **NOTES.TXT**, should count and display the occurrence of alphabets **A** and **E** (including small cases **a** and **e** too).

Example :

If the file content is as follows :

CBSE enhanced its

CCE guidelines further.

The AECCount() function should display the output as

A:1

E:7

115.

Write a function CountYouMe() in C++ which reads the contents of a text file story.txt and counts the words You and Me (not case sensitive).

For example, if the file contains:

You are my best friend.

You and me make a good team.

The function should display the output as

Count for You: 2

Count for Me: 1

116.

Write a function TotalDigits() in C++ , which reads the contents of the Text File "child.txt" and display the number of digits in it.

If the file contains:

All in all, I would like to be 500 note  
and my elders would be 1000 note.

Then the output should be: 7

117.

Assuming the class WORKER as declared below, write a function in c++ to read the objects of WORKER from binary file name "WORKER.DAT" and display those records of workers whose wage is less than 300.

```
class WORKER
{ int WNO;char WName[50]; float Wage;
  public :
  void enter() { cin>>WNO; gets(WName); cin>>Wage;}
  void display() { cout<<WNO<<"*"<<WName<<"*"<<Wage<<endl;}
  float GetWage() { return Wage;}
};
```

118.

Write a function in a C++ to read the content of a text file "UNO.TXT" and display all those lines on screen, which are either starting with 'D' or starting with 'M'.

119.

Write a function in C++ to count the words "this" and "these" present in a text file "ARTICLE.TXT".

[Note that the words "this" and "these" are complete words]

120.

Write a function in C++ to count the words "to" and "the" present in a text file "POEM.TXT".

[Note that the words "to" and "the" are complete words]

121.

Write a function in a C++ to count the number of lowercase alphabets present in a text file "BOOK.txt".

122.

Assume a text file "coordinate.txt" is already created. Using this file create a C++ function to count the number of words having first character capital.

123.

Write a function to count the number of blanks present in a text file named "PR.TXT" .

### 3 Mark Questions

1. Write the output of the following C++ program code:

Note: Assume all the required header files are already being included in the program.

```
class Calc
{
    char Grade;
    int Bonus;
public:
    Calc(){Grade='E' ; Bonus=0;}
    void Down(int G)
    { Grade= G; }
    void Up(int G)
    { Grade+=G; Bonus++; }
    void Show()
    { cout<<Grade<<"#"<<Bonus<<endl; }
};

void main()
{
    Calc c;
    C.Down(97); C.Show();
    C.Up(3);      C.Show();
    C.Up(-32); C.Show();
}
```

2. Write the output of the following C++ program code:

Note: Assume all the required header files are already being included in the program.

```
class Eval
{
    char Level;
    int Point;
public:
    Eval() {Level='E';Point=0;}
    void Sink(int L)
    { Level-= L; }
    void Float(int L)
    { Level+= L; Point++; }
    void Show()
    { cout<<Level<<"#"<<Point<<endl; }
};

void main()
{
    Eval E;
    E.Sink(3);
    E.Show();
    E.Float(7);
    E.Show();
    E.Sink(2);
    E.Show();
}
```

3. Obtain the output of the following C++ program code:

Note: Assume all the required header files are already being included in the program.

```
class Player
{
    int Score,Level;char Game;
public :
    Player(char GGame='A')
    {Score=0;Level=1;Game=GGame;}
    void Start(int SC) ;
        void Next();
        void Disp()
        {   cout<<Game<<"@"<<Level<<endl;
            cout<<Score<<endl;
        }
};

void main()
{   Player P,Q('B');
    P.Disp();
    Q.Start(75);
    Q.Next();
    P.Start(120);
    Q.Disp();
    P.Disp();
}

void Player::Next()
{   Game=((Game=='A')?'B':'A'); }

void Player::Start(int SC)
{   Score+=SC;
    if (Score >=100 )
        Level=3;
    else if (Score>=50 )
        Level=2;
    else
        Level=1;
}
```

4.

Obtain the output of the following C++ program code:

Note: Assume all the required header files are already being included in the program.

```
void in(int x,int y,int &z)
{ x+=y;  y--;  z*=(x-y); }
void out(int z,int y, int &x)
{ x*=y;  y++;z/=(x+y); }

void main()
{ int a=20, b=30, c=10;
  out(a,c,b);
  cout<<a<<"#"<<b<<"#"<<c<<"#"<<endl;
  in(b,c,a);
  cout<<a<<"@"<<b<<"@"<<c<<"@"<<endl;
  out(a,b,c);
  cout<<a<<"$"<<b<<"$"<<c<<"$"<<endl;
}
```

5. Obtain the output of the following C++ program code:

```
#include<iostream.h>
class TQ
{
    int r; float s;
public:
    TQ() { r=1;s=5;}
    TQ(TQ &Q)
    { r=Q.r++; s=Q.s+=5; }
    void Bonus(float B=5)
    { s+=B; }
    void Res() { cout<<r<<', '<<s<<endl;}
};
void main()
{    TQ A;
        A.Res();
        A.Bonus(10);
        A.Res();
        TQ B(A);
        B.Bonus();
        B.Res();
}
```

6. Obtain the output of the following C++ program code:

```
#include<iostream.h>
class env
{
    char pl; int humd,temp;
public:
    env()
    { pl='B';humd=100;temp=40; }
    void hot(int t=5)
    { temp+=t;}
    void humid(int h=10)
    { humd+=h;}
    void forecast()
    { cout<<pl<<"++"<<temp<<"&"<<humd<<"%"<<endl; }
};
void main()
{
    env A;
    A.hot(10);
    A.forecast();
    A.humid(5);
    A.forecast();
        A.hot(); A.humid();
        A.forecast();
}
```



7. Obtain the output of the following C++ program code:

```
#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=10, b=20;
        execute(b);
        cout<<a<<','<<b<<endl;
        execute(a,b);
        cout<<a<<','<<b<<endl;
}
```

8. Obtain the output of the following C++ program code:

```
#include<iostream.h>
int a=4;
void func(int x, int &y)
{
    y=x+10;
    x=x+y;
}
void main( )
{
    int a=7;
    func(a, ::a);
        cout<<a<< " "<<::a<<endl;
    func(a, ::a);
        cout<<a<< " "<<::a<<endl;
    { int a=6; cout<< - - a; }
}
```

9. Obtain the output of the following C++ program code:

```
#include <iostream.h>
struct PLAY
{ int Score, Bonus;};
void Calculate(PLAY &P, int N=10)
{
    P.Score++;P.Bonus+=N;
}
void main()
{
    PLAY PL={10,15};
    Calculate(PL,5);
    cout<<PL.Score<<": "<<PL.Bonus<<endl;
    Calculate(PL);
    cout<<PL.Score<<": "<<PL.Bonus<<endl;
    Calculate(PL,15);
    cout<<PL.Score<<": "<<PL.Bonus<<endl;
}
```

10. Obtain the output of the following C++ program code:

```
#include<iostream.h>
struct Package
{ int Length,Breadth,Height;
};
void Occupies(Package M)
{ cout<<M.Length<<"x"<<M.Breadth<<"x";
cout<<M.Height<<endl;
}
void main( )
{ Package P1={10,20,30},P2,P3;
++P1.Height;
Occupies(P1);
P3=P1;
++P3.Length;
P3.Breadth++;
Occupies(P3);
P2=P3;
P2.Height+=50;
P2.Length--;
Occupies(P2);
}
```

11. Obtain the output of the following C++ program code:

```
#include<iostream.h>
void Indirect(int Temp=25)
{ for(int I=15;I<=Temp;I+=5)
cout<<I<<' ';
cout<<endl;
}
void Direct(int &Num)
{ Num+=10;
Indirect(Num);
}
void main( )
{ int Number=10;
Direct(Number);
Indirect( );
cout<<"Number ="<<Number<<endl;
}
```

12. Obtain the output of the following C++ program code:

```
#include<iostream.h>
#include<ctype.h>
void main( )
{ char STR[]="WhatOutPut!";
for(int I=0; STR[I]!='\0';I++)
{ if(!isalpha(STR[I]))
```

```

STR[I]='*';
else if(isupper(STR[I]))
STR[I]=STR[I]+1;
else
STR[I] =STR[I+1];
}
cout<<STR;}

```

13. Find the output of the following C++ program code:

```

#include <iostream.h>
struct POINT
{ int X, Y, Z;
};
void StepIn(POINT & P, int Step=1)
{ P.X+=Step;
P.Y-=Step;
P.Z+=Step;
}
void StepOut(POINT & P, int Step=1)
{ P.X-=Step;
P.Y+=Step;
P.Z-=Step;
}
void main ( )
{ POINT P1={15, 25, 5}, P2={10, 30, 20};
StepIn(P1);
StepOut(P2,4);
cout<<P1.X<<" "<<P1.Y<<" "<<P1.Z<<endl;
cout<<P2.X<<" "<<P2.Y<<" "<<P2.Z<<endl;
StepIn(P2,12);
cout<<P2.X<<" "<<P2.Y<<" "<<P2.Z<<endl;
}

```

14. Find the output of the following C++ program code:

```

#include<iostream.h>
void SwitchOver(int A[], int N, int Split)
{for (int K=0 ; K<N; K++)
if (K<Split)
A[K]+=K;
else
A[K]*=K;
}
void Display (int A[], int N)
{for (int K=0 ; K<N ; K++)
(K%2==0)?cout<<A[K]<<"%":cout<<A[K]<<endl;
}
void main ()
{ int H[]= {30,40,50,20,10,5};
SwitchOver (H,6,3);
Display(H,6);
}

```

15. Write a definition for function Economic() in C++ to read each record of a binary file ITEMS.DAT, find and display those items, which costs less than 2500. Assume that the file ITEMS.DAT is created with the help of objects of class ITEMS, which is defined below:

```
class ITEMS
{
    int ID; char GIFT[20]; float Cost;
public :
    void Get()
    {
        cin >> CODE; gets(GIFT); cin >> Cost;
    }
    void See()
    {
        cout << ID << " : " << GIFT << " : " << Cost << endl;
    }
    float GetCost() {return Cost;}.
};
```

16. Write a definition for function COSTLY() in C++ to read each record of a binary file GIFTS.DAT, find and display those items, which are priced more than 2000. Assume that the file GIFTS.DAT is created with the help of objects of class GIFTS, which is defined below:

```
class GIFTS
{
    int CODE; char ITEM[20]; float PRICE;
public:
    void Procure()
    {
        cin >> CODE; gets(ITEM); cin >> PRICE;
    }
    void View()
    {
        cout << CODE << " : " << ITEM << " : " << PRICE << endl;
    }
    float GetPrice() {return PRICE;}
};
```

17. Assuming the class GAMES as declared below, write a functions in C++ to read the objects of GAMES from binary file GAMES.DAT and display the details of those GAMES, which are meant for children of Age Range "8 to 13".

```
class GAMES
{
    int GameCode;
    char GameName [10] ;
    char *AgeRange;
```

```

public :
void Enter ()
{
cin>>GameCode;
gets (GameName);
gets (AgeRange);}
void Display()
{
cout <<Gamecode<<": "<<GameName<<endl;
cout<<AgeRange<<endl ;
}
Char*AgeR () {return AgeRange; }
};

```

18.

Assuming the class TOYS as declared below, write a functions in C++ to read the objects of TOYS from binary file "TOYS.DAT" and display the details of those TOYS, which are meant for children of Age Range "5 to 8".

```

class TOYS
{
int ToyCode;
char ToyName [10] ;
char *AgeRange;
public :
void Enter ()
{
cin>>ToyCode;
gets (ToyName);
gets (AgeRange);}
void Display()
{
cout <<Toycode<<": "<<ToyName<<endl;
cout<<AgeRange<<endl ;
}
Char *WhatAge () {return AgeRange; }
};

```

19.

Given a binary file GAME.DAT, containing records of the following structure type

```

struct Game
{
char GameName[20];
char Participant[10][30];
};

```

Write a function in C++ that would read the contents from the file GAME.DAT and creates a file named BASKET.DAT copying only those records from GAME.DAT where the gamename is "Basket Ball".

20.

```

class book
{
int book_no;

```

```

char book_name[20];
float price;
public:
void enter_book_Details( )
{
cin>> book_no>> price; gets(book_name);}
void show_book_Details( );
int checkbookno(int bookno)
{
If(book_no==bookno)
Return(0);
Else
Return (1);
};

```

Write a function deleteBook() in C++ that deletes the required book record from the binary file BOOKS.DAT based on book\_no.

21. Assuming the class VINTAGE as declared below, write a function in C++ to read the objects of VINTAGE from binary file "VINTAGE.DAT" and display those vintage vehicles, which are priced between 20000 and 250000.

```

class VINTAGE
{
int VNO; Char VDesc[10]; float Price;
public:
void GET()
{ cin>>VNO; gets(VDesc);cin>>Price;}
void VIEW()
{
cout<<VNO<<endl;
cout<<VDesc<<endl;
cout<<Price<<endl;}
float ReturnPrice()
{ return Price;}
};

```

22. Assuming the class NETBOOK as declared below, write a function in C++ to read the objects of NETBOOK from binary file "NETBOOK.DAT" and display those netbooks, which are priced between 25000 and 55000.

```

class NETBOOK
{
int NBID; Char NBDesc[10]; float Price;
public:
void GET()
{ cin>>NETBID; gets(NBDesc);cin>>Price;}
void VIEW()
{
cout<<NBID<<endl;
cout<<NBDesc<<endl;
cout<<Price<<endl;}
float ReturnPrice()
{ return Price;}
};

```

23. Write a function in C++ to search for the details (Number& calls) of those mobile phones, which have more than 1000 calls from a binary file "mobile.dat". Assuming that this binary file contains records/objects of class mobile, which is defines below:

```
class mobile
{
    char number[10]; int calls;
public:
    void Enter() { gets(number);cin>>calls;}
    void Billing(){cout<<number<<"#"<<calls<<endl;

    int GetCalls() { return calls;}
};
```

24. Write a function in C++ to search for a tablet from a binary file "tablet.dat" containing objects of class tablet (defined below). The user should enter the ModelNo and function should search and display the detail of tablet.

```
class tablet
{
    long ModelNo; float RAM, HD;
    char Details[20];
public:
    void Enter(){ cin>>ModelNo>>RAM>>HD; gets(Details);}
    void Disp(){ cout<<ModelNo<<RAM<<HD<<Details<<endl;}
    long Model(){ return ModelNo;}
};
```

25. Write a function in C++ to read and display the detail of all the members whose membership type is 'L' or 'M' from a binary file "CLUB.DAT". Assume the binary file "CLUB.DAT" contains objects of class CLUB, which is defined as follows:

```
class CLUB
{
    int Mno;           //Member Number
    char Mname [20]; //Member Name
    char Type; //Member Type: L Life Member M Monthly Member
public:
    void Register (); //Function to enter the content
    void Display (); // Function to display all data members
    char WhatType () {return Type;}.
};
```

26.

Write a function in C++ to search and display details of all flights, whose destination is "Mumbai" from a binary file "FLIGHT.DAT". Assuming the binary file is containing the objects of the following class.

```
class FLIGHT
{ int Fno; //Flight Number
  char From[20]; //Flight Starting Point
  char To[20]; //Flight Destination
public:
  void Enter( )
  {
    cin>>Fno;
    gets(From);
    gets(To);
  }
  char* GetFrom( )
  {return From;}
  char* GetTo( )
  {return To;}

  void Display( )
  {
    cout<<Fno<<": "<<From<<": "<<To<<endl;
  }
};
```

27. Write a function in C++ to search and display details of all trains, whose destination is "Delhi" from a binary file "TRAIN.DAT". Assuming the binary file is containing the objects of the following class.

```
class TRAIN
{ int Tno; // Train Number
  char From[20]; // Train Starting Point
  char To [20]; // Train Destination
public:
  char* GetFrom( )
  {return From;}
  char* GetTo( )
  {return To;}
  void Input( )
  {cin>>Tno; gets(From); gets(To);
  }
```



```

void Show( )
{
cout<<Tno<<": "<<From<<": "<<To<<endl;
}
};

```

## 4 Marks Questions

1. Write the definition of a class PIC in C++ with following description:

### Private Members

Pno //Data member for Picture Number (an integer)  
Category//Data member for Picture Category (a string)  
Location//Data member for Exhibition Location (a string)  
FixLocation /\* A member function to assignExhibition Location as per categoryas shown in the following table \*/

Category	Location
Classic	Amina
Modern	Jim Plaq
Antique	Ustad Khan

### Public Members

Enter()// A function to allow user to enter valuesPno,category and call  
//FixLocation() function  
SeeAll()// A function to display all the data members

2. Write the definition of a class Photo in C++ with following description:

### Private Members

Pno //Data member for Photo Number (an integer)  
Category //Data member for Photo Category (a string)  
Exhibit //Data member for Exhibition Gallery (a string)  
FixExhibit// A member function to assign Exhibition Gallery as per Category  
//as shown in the following table

Category	Exhibit
Antique	Zaveri
Modern	Johnsen
Classic	Terenida

### Public Members

Register()// A function to allow user to enter values  
//Pno,Category and call FixExhibit()function  
ViewAll()// A function to display all the data members

3. Define a class CONTEST in C++ with the following description:

### Private Data Members

Eventno integer  
Description char(30)  
Score integer  
qualified char

Public Member functions

- A constructor to assign initial values Eventno as 11, Description as 'School level', Score as 100, qualified as 'N'.
- Input() - To take the input for Eventno, description and score.
- Award (int cutoffscore) - To assign qualified as 'Y', if score is more than the cutoffscore that is passed as argument to the function, else assign qualified as 'N'.
- Displaydata() - to display all data members.

4. **Define a class Tourist** in C++ with the following specification:

Data Members

- CNo - to store Cab No
- CType - to store a character 'A', 'B', or 'C' as City Type
- PerKM - to store per Kilo Meter charges
- Distance - to store Distance travelled (in KM)

Member Functions

- A constructor function to initialize CType as 'A' and CNo as '0000'
- A function CityCharges( ) to assign PerKM as per the following table :
 

Ctype	PerKM
A	20
B	18
C	15
- A function RegisterCab() to allow administrator to enter the values for CNo and CType. Also, this function should call CityCharges() to assign PerKM Charges.
- A function Display() to allow user to enter the value of Distance and display CNo, CType, PerKM, PerKM\*Distance (as Amount) on screen.

5. **Define a class ITEM** in C++ withfollowing description:

**Private Members**

- \_ Code of type integer (Item Code)
- \_ Iname of type string (Item Name)
- \_ Price of type float (Price of each item)
- \_ Qty of type integer (Quantity of item in stock)
- \_ Offer of type float (Offer percentage on the item)
- \_ A member function GetOffer() to calculate

Offer percentage as per the following rule:

If Qty<=50 Offer is 0

If 50<Qty<=100 Offer is 5

If Qty>100 Offer is 10

**Public Members**

- A function GetStock() to allow user to enter values for Code, Iname, Price, Qty and call function GetOffer() to calculate the offer
- A function ShowItem() to allow user to view the content of all the data members

6. **Define a class STOCK** in C++ with following description:

Private Members

- \_ ICode of type integer (Item Code)
- \_ Item of type string (Item Name)
- \_ Price of type float (Price of each item)
- \_ Qty of type integer (Quantity in stock)
- \_ Discount of type float (Discount percentage on theitem)

\_ A member function FindDisc() to calculate

discount as per the following rule:

If Qty<=50 Discount is 0, If 50<Qty<=100 Discount is 5

If Qty>100 Discount is 10

**Public Members**\_ A function Buy() to allow user to enter values for ICode, Item, Price, Qty and call functionFindDisc() to calculate the Discount.

\_ A function ShowAll() to allow user to viewthe content of all the data members.

7. Define a class **RESORT** in C++ with following description:

**Private Members**

\_ Rno //Data member to store Room No

\_ Name //Data member to store customer name

\_ Charges //Data member to store per day charges

\_ Days //Data member to store number of days of stay

\_ COMPUTE( )//A function to calculate andreturn Amount as Days\*Charges & if thevalue of Days\*Charges is more than 11000then as 1.02\*Days\*Charges

**Public Members**

\_ Getinfo ( ) //A function to enter the contentRno,Name,Charges & Days

\_ Dispinfo ( ) //A function to display Rno,Name, Charges,Days and Amount (Amount tobe displayed by calling function COMPUTE( ) )

8. Define a class **HOTEL** in C++ with thefollowing description:

**Private Members:**

\_ Rno //Data member to store Room No

\_ Name //Data member to store customer name

\_ Tariff //Data member to store per day charges

\_ NOD //Data member to store number of days of stay

\_ CALC( ) /\*A function to calculate and returnAmount as NOD\*Tariff and if the value ofNOD\*Tariff is more than 10000 then as1.05\*NOD\*Tariff \*/

**Public Members**

\_ Checkin ( ) / / A function to enter the contentRno, Name, Tariff and NOD

\_ Checkout( ) / / A function to display Rno,Name, Tariff,NOD and Amount (Amount tobe displayed by calling function CALC( ))

9. Define a class TEST in C++ with following description:

**Private Members**

• TestCode of type integer

• Description of type string

• NoCandidate of type integer

• CenterReqd (number of centers required) of type integer

• A member function CALCNTR() to calculate and return the number of centers as(NoCandidates/100+1)

**Public Members**

• A function SCHEDULE() to allow user to enter values for TestCode, Description,NoCandidate & call function CALCNTR() to calculate the number of Centres

• A function DISPTTEST() to allow user to view the content of all the data members

10. Define a class in C++ with following description:

**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

11 Define a class Clothing in C++ with the following descriptions:

**Private Members:**

Code of type string

Type of type string

Size of type integer

Material of type string

Price of type float

A function Calc\_Price() which calculates and assigns the value of Price as follows:

For the value of Material as "COTTON"	
Type	Price (Rs.)
TROUSER	1500
SHIRT	1200

For Material other than "COTTON" the above mentioned Price gets reduced by 25%.

**Public Members:**

A constructor to assign initial values of Code, Type and Material with the word "NOT ASSIGNED" and Size and Price with 0.

A function Enter () to input the values of the data members Code, Type, Size and Material and invoke the CalcPrice() function.

A function Show () which displays the content of all the data members for a Clothing.

12. Define a class Travel in C++ with the description given below:

**Private Members:**

T\_Code of type string  
No\_of\_Adults of type integer  
No\_of\_Children of type integer  
Distance of type integer  
TotalFare of type float

**Public Members:**

A constructor to assign initial values as follows :

T\_Code with the word "NULL"

No\_of\_Adults as 0

No\_of\_Children as 0

Distance as 0

TotalFare as 0

A function AssignFare( ) which calculates and assigns the value of the data member TotalFare as follows :

For each Adult

Fare (Rs)	Distance (Km)
500	>=1000
300	<1000 & >=500
200	<500

For each Child the above Fare will be 50% of the Fare mentioned in the above table.

For example :

If Distance is 750, No\_of\_Adults = 3 and No\_of\_Children = 2

Then TotalFare should be calculated as No\_of\_Adults \* 300 +

No\_of\_Children \* 150 i.e.  $3 * 300 + 2 * 150 = 1200$

- A function EnterTraveK ) to input the values of the data members

T\_Code, No\_of\_Adults, No\_of\_Children and Distance; and invoke the AssignFare( ) function.

- A function ShowTraveK) which displays the content of all the data members for a Travel.

13.

Define a class Candidate in C++ with following description:

**Private Members**

\_ A data member RNo (Registration Number) of type long

\_ A data member Name of type string

\_ A data member Score of type float

\_ A data member Remarks of type string

A member function AssignRem( ) to assign Remarks as per the Score obtained by a candidate. Score range and the respective Remarks are

shown as follows:

Score	Remarks
>=50	Selected
less than 50	Not selected

Public Members

\_ A function ENTER ( ) to allow user to enter values for RNo, Name, Score & call function AssignRem( ) to assign the remarks.

\_ A function DISPLAY ( ) to allow user to view the content of all the datamembers.

14. Define a class TAXPAYER in C++ with following description:

Private members :

- Name of type string
- PanNo of type string
- Taxabincm (Taxable income) of type float
- TotTax of type double
- A function CompTax( ) to calculate tax according to the following slab:

Taxable Income	Tax%
Up to 160000	0
>160000 and <=300000	5
>300000 and <=500000	10
>500000	15

Public members :

- A parameterized constructor to initialize all the members
- A function INTAX( ) to enter data for the tax payer and call function CompTax( ) to assign TotTax.

A function OUTAX( ) to allow user to view the content of all the data members.

15.

Define a class Applicant in C++ with following description:

Private Members

- A data member ANo ( Admission Number) of type long
- A data member Name of type string
- A data member Agg(Aggregate Marks) of type float
- A data member Grade of type char
- A member function GradeMe( ) to find the Grade as per the Aggregate Marks obtained by a student. Equivalent Aggregate marks range and the respective Grades are shown as follows

Aggregate Marks	Grade
>= 80	A
Less than 80 and >= 65	B
Less than 65 and >= 50	C
Less than 50	D

Public Members

- A function Enter( ) to allow user to enter values for ANo, Name, Agg & call function GradeMe( ) to find the Grade

- A function Result ( ) to allow user to view the content of all the data members.

16.

Define a class ORDER in C++ with following description:

Private Members

- ICode of type integer (Item Code)
- Item of type string (Item Name)
- Price of type float (Price of each item)
- Qty of type integer (Quantity in stock)
- Discount of type float (Discount percentage on the item)
- A member function FindDisc() to calculate discount percentage as per the following rule:  

If Qty <=50	Discount is 0
If 50 < Qty <=100	Discount is 5
If Qty > 100	Discount is 10

Public Members

- A function Buy ( ) to allow user to enter values for ICode, Item, Price, Qty and call function FindDisc() to calculate Discount.
- A Function ShowAll() to allow user to view the content of all the data members.

17. Define a class TravelPlan in C++ with the following descriptions:

Private Members:

- |                      |                                  |
|----------------------|----------------------------------|
| Plan Code            | of type long                     |
| Place                | of type character array (string) |
| Number_of_travellers | of type integer                  |
| Number_of_buses      | of type integer                  |

Public Members:

A constructor to assign initial values of PlanCode as 1001, Place as "Agra", Number\_of\_travellers as 5, Number\_of\_buses as 1

A function NewPlan() which allows user to enter PlanCode, Place and Number\_of\_travellers. Also, assign the value of Number\_of\_buses as per the following conditions:

Number_of_travellers	Number_of_buses
Less than 20	1
Equal to or more than 20 and less than 40	2
Equal to 40 or more than 40	3

A function ShowPlan() to display the content of all the data members on screen

18. A dining hall can accommodate only 50 guests. Define a class to store seat number and name of the guests who are seated on first come first seated basis. Define functions to display details of any seat number and to display the current seating situation. Write a program to show the working of this class.

19.

Define a class named Tour in C++ with following description?

Private members:

tcode	integer (Ranges 6 - 10)
adults, children, distance	integer
totalfare	float
AssignFare( )	A function which calculates and assign the value to data member totalfare as follows:-

- <b>For adults</b>	Fare	Distance
	Rs. 500	>=1500

And fare get reduced by 25% if distance is < 1500.

- **For Children**

For every child a fixed Rs. 50 is charged as fare.

Public members:

- A constructor which initialized initialize all data members with 0
- Function EnterTour() to input the values of the data members tcode, adults, children and call to AssignFare function.
- Function ShowTour() to print all the details of object of Travel type.

20.

Define a class named Admission in C++ with following description?

4

**Private members:**

admno	integer (Ranges 10-1500)
name	string of 20 characters
cls	integer
fees	float

**Public members:**

A constructor which initialized admno with 10, name with "NULL", cls with 0 & fees with 0

Function getdata() to read the object of Admission type.

Function putdata() to print the details of object of admission type.

Function draw\_nos() to generate the admission no. randomly to match with admno and display the detail of object.



21. Answer the question (i) to (iv) based on the following:

```
class Exterior
{
int OrderId;
char Address[20];
protected:
float Advance;
public:
Exterior();
void Book();
void View();
};
class Paint:public Exterior
{
int WallArea,ColorCode;
protected:
char Type;
public:
Paint() ;
void PBook();
void PView();
};
class Bill:public Paint
{
float Charges;
void Calculate();
public:
Bill() ;
void Billing() ;
void Print() ;
};
```

(i) Which type of Inheritance out of the following is illustrated in the above example?

- Single Level Inheritance
- Multi Level Inheritance
- Multiple Inheritance

(ii) Write the names of all the data members, which are directly accessible from the member functions of class Paint.

(iii) Write the names of all the member functions, which are directly accessible from an object of class Bill.

iv) What will be the order of execution of the constructors, when an object of class Bill is declared?

22. Answer the questions (i) to (iv) based on the following:

```
class Interior { int OrderId; char Address[20];
protected: float Advance;
public: Interior(); void Book(); void View();
};
class Painting:public Interior
{ int WallArea,ColorCode;
protected:
char Type; public:
Painting();
void PBook();
void PView(); };
class Billing : public Painting
{ float Charges;
void Calculate();
public:
Billing();
void Bill();
void BillPrint(); };
```

(i) Which type of Inheritance out of the following is illustrated in the above example ?

- Single Level Inheritance
- Multi Level Inheritance
- Multiple Inheritance

(ii) Write the names of all the data members, which are directly accessible from the member functions of class Painting.

(iii) Write the names of all the member functions, which are directly accessible from an object of class Billing.

(iv) What will be the order of execution of the constructors, when an object of class Billing is declared ?

23. Consider the following C++ code and answer the questions from (i) to (iv) :

```
class University { long Id; char City[20];
protected: char Country[20]; public: University(); void Register( ); void
Display( ); };
class Department: private University { long DCode[10];char HOD[20];
protected:
double Budget; public: Department();
void Enter();
void Show(); };
class Student: public Department { long RollNo; char Name[20];
public:
Student();
void Enroll();
void View(); };
```

(i) Which type of Inheritance is shown in the above example ?

(ii) Write the names of those member functions, which are directly accessed from the objects of class Student.

(iii) Write the names of those data members, which can be directly accessible from the member functions of class Student.

(iv) Is it possible to directly call function Display( ) of class University from an object of class Department ? (Answer as Yes or No).

24. Consider the following C++ code and answer the questions from (i) to (iv):

```
class Campus
{
long Id;
char City[20];
protected:
char Country [20] ;
public :
Campus();
void Register();
void Display() ;
};

class Dept : private Campus
{
long DCode [10] ;
char HOD [20] ;
protected :
double Budget;
public:
Dept() ;
void Enter();
void Show();
};

class Applicant: public Dept
{ long RegNo;
char Name [20] ;
public:
Applicant() ;
void Enroll();
void View();
};
```

- (i) Which type of Inheritance is shown in the above example ?
- (ii) Write the names of those member functions, which are directly accessed from the objects of class Applicant.
- (iii) Write the names of those data members, which can be directly accessed from the member functions of class Applicant.
- (iv) Is it possible to directly call function Display() of class Campus from an object of class Dept ? (Answer as Yes or No).

25. Consider the following C++ code and answer the questions from (i) to (iv):

```
class Student
{
    int Class, Rno;

    char Section;
protected :
    char SName[20];
public:
    Student() ;
    void Stentry();
    void Stdisplay();
};

class Score: private Student
{
    float Marks[S];
protected:
    char Grade[S];
public:
    Score() ;
    void Sentry() ;
    void Sdisplay() ;
};

class Report: public Score
{
    float Total, Avg;
public:
    char OverallGrade, Remarks[20];
    Report();
    void REvaluate();
    void RPrint();
};
```

- (i) Which type of Inheritance is shown in the above example?
- (ii) Write the names of those data members, which can be directly accessed from the objects of class Report.
- (iii) Write the names of those member functions, which can be directly accessed from the objects of class Report.
- (iv) Write the names of those data members, which can be directly accessed from the Sentry() function of class Score.

26. **Consider the following c++ code and answer the questions from (i) to (iv):**

```
class Personal
{
int Class,Rno;
char Section;
protected:
char Name[20];
public:
personal();
void pentry();
void Pdisplay();
};
class Marks: private Personal
{ float M[5];
    protected:
        char Grade[5];
    public:
        Marks(); void Mentry (); void Mdisplay();
};
class Result: public Marks
{ float Total, Agg;
    public:
        char FinalGrade, Comments[20];
        Result();
        void Rcalculate();
        void Rdisplay();
};
```

- (i) Which type of Inheritance is shown in the above example?
- (ii) Write the names of those data members, which can be directly accessed from the objects of class Result.
- (iii) Write the names of those member functions, which can be directly accessed from the objects of class Result.
- (iv) Write the names of those data members, which can be directly accessed from the Mentry() function of class Marks.

27. Answer the questions (i) to (iv) based on the following: .

```
class COMPANY
{
    char Location[20];
    double Budget, Income;
protected:
    void Accounts();
public:
    COMPANY();
    void Register();
    void Show();
};

class FACTORY : public COMPANY
{
    char Location[20];
    int Workers;
protected:
    double Salary;
    void Computer();
public:
    FACTORY();
    void Enter();
    void Show();
};

class SHOP : private COMPANY
{
    char Location[20];
    float Area;
    double Sale;

public:
    SHOP();
    void Input();
    void Output();
};
```

- (i) Name the type of inheritance illustrated in the above C++ code.
- (ii) Write the name of data members, which are accessible from member functions of class SHOP.
- (iii) Write the names of all the member functions, which are accessible from objects belonging to class FACTORY.
- (iv) Write the names of all the members, which are accessible from objects of class SHOP.

28. Answer the questions (i) to (iv) based on the following:

```
class Chairperson
```

```
{
long CID; //Chairperson Identification
Number
char CName[20];
protected:
char Description [40];
void Allocate();
public:
Chairperson();
void Assign();
void Show();
};
```

```
class Director
```

```
{
int DID; //Director ID
char Dname[20];
protected:
char Profile[30];
public:
Director();
void Input();
void output();
};
```

```
class Company:private Chairperson, public Director
```

```
{
int CID; //Company ID
char City[20], Country[20];
public:
Company();
void Enter();
void Display();
};
```

(i) Which type of inheritance out of the following is specifically illustrated in the above C++ code?

- (a) Single Level Inheritance
- (b) Multi Level Inheritance
- (c) Multiple Inheritance

(ii) Write the names of data members, which are accessible by objects of class type Company.

(iii) Write the names of all member functions, which are accessible by objects of class type Company.

(iv) Write the names of all members, which are accessible from member functions of class Director.

29. Answer the questions (i) to (iv) based on the following:

```
class Director
{
long DID; //Director Identification Number
char Name[20];
protected:
char Description[40];
void Allocate () ;
public:
Director() ;
void Assign () ;
void Show () ;
} ;
```

```
class Ractory:public Director
{
int FID; //Factory ID
char Address[20];
protected:
int NOE; //No. of Employees
public:
Factory();
void Input ();
void Output ();
};
class ShowRoom:private Factory
{
int SID; //Showroom ID
char City[20];
public:
ShowRoom();
void Enter ();
void Display ();
};
```

(i) Which type of inheritance out of the following is illustrated in the above C++ code?

- (a) Single Level Inheritance
- (b) Multi Level Inheritance
- (c) Multiple Inheritance

(ii) Write the names of data members, which are accessible by objects of class type ShowRoom.

(iii) Write the names of all member functions which are accessible by objects of class type ShowRoom.

(iv) Write the names of all members, which are accessible from member functions of class Factory.



30. Answer the questions (i) to (iv) based on the following: class FaceToFace

```
{
char CenterCode [10] ;
public:
void Input ( ) ;
void Output ( ) ;
} ;
class Online
{
char website [50] ;
public:
void SiteIn ( ) ;
void SiteOut ( ) ;
} ;
class Training: public FaceToFace, private Online
{
long Tcode ;
float charge;
int period;
public:
void Register ( ) ;
void Show ( ) ;
} ;
```

- (i) Which type of Inheritance is shown in the above example?
- ii) Write names of all the member functions accessible from Show( ) function of class Training.
- iii) Write the name of all the members accessible through an object of class Training.
- iv) Is the function Output( ) accessible inside the function SiteOut( )? Justify your answer.

31. Answer the questions (i) to (iv) based on the following:

```
class PUBLISHER
{
    char Pub[12];
double Turnover;
protected:
void Register();
public:
PUBLISHER();
void Enter();
void Display();
};
class BRANCH
{
    char CITY[20];
protected:
float Employees
public:
BRANCH();
void Haveit();
void Giveit();};
```

```

class AUTHOR : private BRANCH , public PUBLISHER
{
    int Acode;
    char Aname[20];
    float Amount;
public:
    AUTHOR();
    void Start();
    void Show();
};

```

(i) Write the names of data members, which are accessible from objects belonging to class AUTHOR.

(ii) Write the names of all the member functions which are accessible from objects belonging to class BRANCH.

(iii) Write the names of all the members which are accessible from member functions of class AUTHOR.

(iv) How many bytes will be required by an object belonging to class AUTHOR?

32. Answer the questions (i) to (iv) based on the following code:

```

class Dolls
{
    char DCode[5];
protected:
    float Price ;
    void CalcPrice(float);
public:
    Dolls( );
    void DInput( );
    void DShow( );
};
class SoftDolls: public Dolls
{
    char SDName[20];
    float Weight;
public:
    SoftDolls( );
    void SDInput( );
    void SDSHOW( );
};
class ElectronicDolls: public Dolls
{
    char EDName[20];
    char BatteryType[10];
    int Battieries;
public:
    ElectronicDolls ( );
    void EDInput( );
    void EDSHOW( );
};

```

(i) Which type of Inheritance is shown in the above example?

(ii) How many bytes will be required by an object of the class ElectronicDolls?

(iii) Write name of all the data members accessible from member functions of the class SoftDolls.

(iv) Write name of all the member functions accessible by an object.

33. consider the following class declaration and answer the question below :

```
class university
{
    int noc;
protected:
    char uname[25];
public:
    university();
    char state[25];
    void enterdata();
    void displaydata();
};

class college : public university
{
    int nod;
    char cname[25];
protected:
    void affiliation();
public:
    college();
    void enrol(int ,int);
    void show();
};

class department : public college
{
    char dname[25];
    int nof;
public:
    department();
    void display();
    void input();
};
```

(i) Which class's constructor will be called first at the time of declaration of an object of class department?

(ii) How many bytes does an object belonging to class department require?

(iii) Name the member function(s), which are accessed from the object of class department.

(iv) Name the data member, which are accessible from the object of class college.

34. Answer the questions(i) to (iv) based on the following :

```
class cloth
{
    char category[5];
    char description[25];
protected:
    float price;
public:
    void Entercloth( );
    void dispcloth( );
};

class Design : protected cloth
{
    char design[21];
protected:
    float cost_of_cloth;
public:
    int design_code;
    Design( );
    void Enterdesign( );
    void dispdesign( );
};

class costing : public cloth
{
    float designfee;
    float stitching;
    float cal_cp( );
protected:
    float costprice;
    float sellprice;
public:
    void Entercost( );
    void dispcost( );
    costing ( ) { };
};
```

(i) Write the names of data members which are accessible from objects belonging to class cloth.

(ii) Write the names of all the members which are accessible from objects belonging to class Design.

(iii) Write the names of all the data members which are accessible from member functions of class costing.

(iv) How many bytes will be required by an object belonging to class Design?

35. Answer the questions(i) to (iv) based on the following :

```
class Regular
{
    char SchoolCode[10];
public:
void InRegular( );
void OutRegular( );
};
class Distance
{
    char StudyCentreCode[5];
public:
void InDistance( );
void OutDistance( );
};
class Course : public Regular, private Distance
{
    char Code[5];
float Fees;
int Duration;
public:
void InCourse( );
void OutCourse( );
};
```

(i) Which type of Inheritance is shown in the above example?

(ii) Write names of all the member functions accessible from OutCourse function of class Course.

(iii) Write name of all the members accessible through an object of the Class Course.

(iv) Is the function InRegular( ) accessible inside the function InDistance ( )? Justify your answer.

36. Consider the following declarations and answer the questions given below :

```
class living_being
{
    char name[20];
protected:
    int jaws;
public:
    void inputdata(char, int);
    void outputdata();
}
class animal : protected living_being
{
    int tail;
protected:
    int legs;
public:
    void readdata(int, int);
    void writedata();
};
```

```

class cow : private animal
{
    char horn_size;
public:
    void fetchdata(char);
    void displaydata();
};

```

- (i) Name the base class and derived class of the class animal.
- (ii) Name the data member(s) that can be accessed from function displaydata.
- (iii) Name the data member(s) that can be accessed by an object of cow class.
- (iv) Is the member function outputdata accessible to the objects of animal class.

37. Consider the following and answer the questions given below:

```

class MNC
{
    char Cname[25]; // Company name
protected :
    char Hoffice[25]; // Head office
public :
    MNC( );
    char Country[25];
    void EnterDate( );
    void DisplayData( );
};
class Branch : public MNC
{
    long NOE; // Number of employees
    char Ctry[25]; // Country
protected:
    void Association( );
public :
    Branch( );
    void Add( );
    void Show( );
};
class Outlet : public Branch
{
    char State[25];
public :
    Outlet();
    void Enter();
    void Output();
};

```

- (i) Which class's constructor will be called first at the time of declaration of an object of class Outlet?
- (ii) How many bytes an object belonging to class Outlet require ?
- (iii) Name the member function(s), which are accessed from the object(s) of class Outlet.
- (iv) Name the data member(s), which are accessible from the object(s) of class Branch.

## SOLUTION: OBJECT ORIENTED PROGRAMMING IN C++

### Answers to 1 Mark Questions

1. (i) `stdio.h` (ii) `ctype.h`
2. (i) `string.h` (ii) `stdlib.h`
3. (i) `math.h` (ii) `stdlib.h`
4. (i) `stdio.h` (ii) `fstream.h`
5. The header files are : `stdio.h`, `string.h`
6. The required header files are : `iomanip.h` and `iostream.h`
7. The multiple uses of input or output operators in one statement are called cascading of I/O operators.
8. The default statement gives the switch construct away to take action if the value of the switch variable does not match any of the case constant.
9. The void type specifies an empty set of values. It is used as the return type for function that do not return any value. No object of void type may be declared because it depicts a nil parameter list for a function.
10. In **switch-case** statement, when a match is found, the statement sequence associated with that case is executed until a break statement or the end of **switch** statement is reached. So if break statement is missing, then the statement sequence is executed until the end of the switch-case statement is reached.
11. 11
12. When the size of the code of a function is small that the overhead of the function call becomes prominent then the function should be declared as inline.
13. `void message()` function declare with an empty parentheses, in means that the function does not pass any parameters.
14. Encapsulation means wrapping up of data and functions which operate the data into a single unit and ensures only essential features get represented without representing the detail background. i.e., called Abstraction. Therefore, both are inter-related.
15. 

```
inline int bar(float a);  
{  
    .....  
}
```
16. `#include<iostream.h>`  
`#include<math.h>`
17. `#include<iostream.h>`  
`#include<maths.h>`
18. `#include<iostream.h>`  
`#include<maths.h>`  
`#include<ctype.h>`
19. `puts ( ) → <stdio.h>`  
`Sin ( ) → <math.h>`
20. `setw ( ) → <iomanip.h>`  
`Sqrt ( ) → <math.h>`
21. `cout, cin → #include<iostream.h>`  
`Tolower( ) → #include<ctype.h>`
22. `cin, cout → #include<iostream.h>`  
`toupper( ) → #include<ctype.h>`
23. `#include<iostream.h> → cout`  
`#include<string.h> → strlen( )`

24. `#include<iostream.h>→cout`  
`#include<string.h>→strlen( )`
25. `#include<iostream.h>→cout`  
`#include<ctype.h>→toupper( )`
26. `#include<ctype.h>→toupper( )`  
`#include<iostream.h>→cout`
27. `#include<iomanip.h>→setw( )`  
`#include<iostream.h>→cout`
28. `#include<iostream.h>→cout`  
`#include<iomanip.h>→setw( )`
29. `iostream.h` and `ctype.h`
30. `While, Float, Amount2, _Counter`
31. `For, INT, NeW, name1`
32. `iostream.h` and `math.h`
33. `Data_rec asm`
34. `7`
35. A preprocessor directive is an instruction to the compiler itself. A part of compiler called preprocessor deals with these directives, before real compilation process. # is used as preprocessor directive in C++.
36. The memory implementation of char data type is in terms of the number code. Therefore, it is said to be another integer data type.
37. Whenever a C++ program is executed, execution of the program starts and ends at main(). The main is the driver function of the program. If it is not present in a program, no execution can take place.
38. `stdlib.h` , `conio.h`
39. `math.h`, `stdlib.h`
40. `ctype`, `stdio`
41. **Statement 1:** `File.tellg ( ) ;`  
**Statement 2:** `File.seekg (-sizeof (PRODUCT), ios::cur);`
42. `PRESENT RECORD: 3`
43. **Statement 1:** `F. tellg ( ) ;`  
**Statement 2:** `F. seekp(Pos-sizeof(C)) ;`  
`OR`  
`F.seekp(-sizeof(C), ios::cur);`
44. **Statement 1:**  
`File.seekp(Pos);`  
**OR**  
`File.seekp(-sizeof(A), ios:: cur);`  
**Statement 2:**  
`File.write((char*)&S, sizeof(S));`  
**OR**  
`File.write((char*)&S, sizeof(Stock));`
45. **Statement 1**  
`File.seekp(File.tellp( )-sizeof(C));`  
`Or`  
`File.seekp(Record*sizeof(C));`  
**Statement 2**  
`File.write((char*)&C,sizeof(C));`  
`Or`  
`File.write((char*)&C,sizeof(Candidate));`



46.     Statement 1  
          File.seekp(Position)  
          Statement 2  
          File.write((char\*)&CL,sizeof(CL)
47.     Statement 1 -   File.seekp(Success);  
          Statement 2 -   File.write((char\*) &S, sizeof(S)
48.     **Statement 1**  
          File.seekp(Position);  
          **OR**  
          File. seekp (-sizeof (L), ios::cur);
- Statement 2**  
          File.write((char\*)&L, sizeof(L));  
          **OR**  
          File.write((char\*)&L,sizeof(Library));
49.     **Statement 1**  
          File.seekp(File.tellp( )-sizeof(L));  
          or  
          File.seekp(Rec\*sizeof(L));
- Statement 2**  
          File.write((char\*)&L,sizeof(L));  
          or  
          File.write((char\*)&L,sizeof(Labrecord));
50.     File.seekp((Recordsread-1)\*sizeof(OM));  
          **OR**  
          File.seekp(Recordsread\*sizeof(OM));  
          **OR**  
          File.seekp(-l\*sizeof(OM),ios::curr);  
          **OR**  
          File.seekp(file.tellg()-sizeof(OM));

## Answers to 2 Mark Questions

1. 'x' is a character constant and its size is 1 character whereas "x" is a string constant and its size is 2 character because compiler automatically assign a null character ('\0') at the end of a string constant i.e. "\0".
2. A reference variable is an alias name for a previously defined variable. The usage of it is that the same data object can be referred to by two names and these names can be usually interchangeably.
3. 50
4. The break statement provides immediate termination of the entire loop body whereas continue statement terminates forces the next iteration of the loop to take place, skipping any code following continue statement in the loop body.
5. The default statement gets executed when no match is found again the values specified in the switch-case statement.
6. In the getch() function, it directly reads a character from the console as soon as it is typed without waiting for the enter key to be pressed, whereas getche() reads a character from keyboard exactly in the same manner but does not show it on the screen.
7. Provided that function parameter is a "const reference", compilers generates temporary variable in following 2 ways:

(a) The actual argument is the correct type, but isn't Lvalue

```
double Cube(const double & num) {  
    num = num * num * num;  
    return num;  
}
```

```
double temp = 2.0;
```

```
double value = cube(3.0 + temp); //argument is an expression and not a Lvalue;
```

(b) The actual argument is of the wrong type, but of a type that can be converted to the correct type

```
long temp = 3L;
```

```
Double value = cuberoot (temp); //long to double conversion
```

8. As we know the last character of the string is NULL. The puts() convert NULL value into the newline character while cout does not do so. For example, If the input is Rahul Sharma.

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

```
main() {
```

```
    char name[20];
```

```
    gets(name);
```

```
    cout << "The name is";
```

```
    cout << name;
```

```
    cout<< "End of the output";
```

```
    getch(); }
```

The output is :

The name is Rahul Sharma End of the output

Now consider the program using puts()

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

```
main() {
```

```
    char name[20];    gets(name);
```

```
    cout<< "The name is";    puts(name);
```

```
    cout<< "End of the output"; }
```

The output is:

The name is Rahul Sharma

End of the output

9. Difference between Type Casting and Automatic type conversion

Type Casting	Automatic Type Conversion
It is an explicit process of conversion of a data from one type to another. (It is performed by the programmer.)	It is an implicit process of conversion of a data from one data type to another. It is performed by the compiler. It is also called as type promotion.
<b>For example</b> int A=1, B=2; float C = (float)A/B; //Type Casting cout<<C; <b>Output:</b> <b>0.5</b>	<b>For example:</b> int N = 65; char C = N; // Automatic type conversion cout<<C; <b>Output:</b> <b>A</b>

10. **Local Variables:** Local variables are those variables which are declared within a function or a compound statement and these variables can only be used within that function/scope.

**Global Variables:** Global variables are those variables which are not declared within any function or scope. So, these variables can be accessed by any function of the program.

**Example:**

```
#include<iostream.h>
#include<conio.h>
int G; // Global variable declared
void Fun ( )
{
    int L=25; // Local variable of function Fun ( ) assigned value 25
    G=5; // Global Variable is accessed and assigned value 5
    cout<<G<<endl; //Value of global variable is displayed as 5
    cout<<L<<endl; // Value of local variable is displayed as 25
}
void main ( )
{
    Fun ( ) ; // Function call
    G = G + 5; // Global variable is incremented by 5
    cout<<G<<endl; // Global variable is displayed as 10
}
```

11. C++ allows you to define explicitly new data type names by using the keyword typedef. Using typedef does not actually create a new data class, rather it defines a new name for an existing type.

The syntax of the typedef statement is :

typedef type name;

Where type is any C++ data type and name is the new name for this type. This defines another name for the standard type of C++. For example, you could create a new name for float values by using the following statement:

```
typedef float amount;
amount x=10.5; // amount is alternative name to float
```

12. **#define** preprocessor directive is used to define a macro with some value/expression, which is substituted during compilation of program. Unlike variable, it does not occupy memory.

e.g.

```
#define Max 10
void main( )
{
    Int arr[Max];
}
```

**const** It is used in declaration of a variable, it occupy memory to store a constant value, which once initialized cannot be changed.

e.g.

```
const int x = 10;
```

13.

Actual Parameter	Formal Parameter
Parameters provided at the time of function calling are called actual parameters. These parameters are contain actual values.	Parameters provided at the time of function definition are called formal parameters. These parameters are simple variable declarations,i.e. they do not contain actual values.

e.g.

```
#include <iostream.h>
void Calc(int T) //T is formal parameter
{
    cout<<5*T;
}
void main()
{ int A=45;
  Calc(A); //A is actual parameter
}
```

14. **Call by value:** The formal parameter makes a copy of actual parameter. It does not make the changes In actual parameter If the changes are done In formal parameters.

**Call by reference:** The formal parameter is an alias of actual parameter. The changes made In the formal parameter are reflected In actual parameter. It is preceded by &.void Calculator (int A,int & B )

```
{
A++;
a+=A;
}
```

Here A is called by value and B is called by reference.

15. 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 supply a value for the default parameter, the user supplied value is used.

Consider the following program:

```
void PrintValues ( int nValue1,
int nValue2=10)
{
    cout<< "1st value:"<<nValue1<<endl;
    cout<< "2st value:"<<nValue2<<endl;
}
int main ( )
{
    PrintValues(1); //nValue2 will use default parameter of 10
    PrintValues(1,4); //override default value for nValue2
    return 0; }
```

16. The function prototype serve to ensure that calls to the function are made with proper number and types of arguments. In the case of function overloading, the different prototype serve to distinguish which version of the function to call. The computer will complain with an error, if no function prototype is found for any particular call to function.

e.g.

```
#include <iostream.h>
int square (int); //function prototype
int main( )
{
    for( int x = 1; x<=10; x++)
        cout<<square( x )<< " ";
    cout<<endl;
    return 0;
}
//Function definition
int square(int y)
{
    return y*y;
}
```

17. The classes are the manufacturing units of the objects of their type, i.e. It is the class that can be used to create an object. Since encapsulation and also abstraction are done at the class level, it is the class that can be model the objects from the real world problem.

18. Polymorphism means processing of data or messages in more than one form. C++ implements polymorphism through overloaded functions and overloaded operators.

e.g.

```
float computer(float a)
{
    return a*a;
}
float computer(float a, float b)
{
    return (a*b);
}
```

19. The wrapping up of data and function into a single unit is called data encapsulation. That single unit is known as class.

e.g.

```
class person
{
    char name[ 30 ];
    int age;
public:
    void getdata(void);
    void display(void);
};
```

The above program implements data hiding as data can't be accessed directly from outside.

20. Data hiding is a property, where internal data structure of an object is hidden from the outside world. Data hiding helps to secure the data. It is implemented with private and protected keywords.

e.g.

```
class item
{
    private:
        int item_no;
        float item_cost;
    public:
        void getdata( );
        void putdata( );
};
```

21. The wrapping up of data and function into a single unit is called data encapsulation. That single unit is known as class.

e.g.

```
class person
{
    char name[ 30 ];
    int age;
    public:
        void getdata(void);
        void display(void);
};
```

The above program implements data hiding as data can't be accessed directly from outside.

22. Abstraction refers to the representation of only the essential features of the real world object in the program object. This process does not include the background details and explanations. This concept of abstraction is used in classes, whereas data encapsulation is the most significant characteristic of the class. By this term, we mean the wrapping up of data and function which operate on the data, into a single unit called the class. This encapsulation prevents free access to the data within an object.

23. In the event driven programming, the user indicates the order of program execution not the programmer. Instead of, the program 'driving' the user 'drives' the program. Programming, the code that responds to the event is called event driven programming.

24. When several functions have same name but performing different tasks, then it is known as function overloading. The definitions of these functions are differentiable by the number or types of their arguments.

e.g.

```
float compute(float radius)
{
    return(3.14*radius*radius);
}
float compute(float l, float b)
{
    return(l*b);
}
float compute(int b, float h)
```

```

{
    return(0.5*b*h);
}

```

25. The process of making an operator to exhibit or show different behavior in different situations is called as operator overloading.

e.g. consider the operation of (+) operator, Operation is sum, if operands are integer type and the operation is concatenation if operands are strings.

26. Object oriented programming focus on objects. It gives the benefits of security of data, reusability of previously created functions. It is based on the principle of data hiding, abstraction, inheritance and polymorphism. But procedural programming emphasizes on doing things. It revolves around functions and execution of these functions.

27.

<i>Object Oriented Programming</i>	<i>Procedural Programming</i>
Emphasis on data	Emphasis on doing things (function)
Follow bottom up approach in program design.	Follow top-down approach in program design
Concept of Data hiding prevents accidental change in the data.	Due to presence of global variables, there is a possibility of accidental change in data.
Polymorphism, inheritance, Data Encapsulation possible.	Not applicable

28.

**Syntax error** - The errors which are traced by the compiler during compilation, due to wrong grammar for the language used in the program, are called syntax errors.

For example, cin<<a; // instead of extraction operator insertion operator is used.

**Run time Error** - The errors encountered during execution of the program, due to unexpected input or output are called run-time error.

For example - a=n/0; // division by zero

**Logical Error** - These errors are encountered when the program does not give the desired output, due to wrong logic of the program.

For example : remainder = a+b // instead of using % operator + operator is used.

29. While is an Entry Controlled Loop, the body of the loop may not execute even once if the test expression evaluates to be false the first time, whereas in do..while, the loop is executed at least once whether the condition holds true the first time or not.

30. A class binds together data and its associated function under one unit thereby enforcing encapsulation. The private and protected member remain hidden from outside world. Thus a class enforces data hiding. The outside world is given only the essential information through public members, thereby enforcing abstraction.

31. These characters are interpreted at run time. Represented by backslash(\) followed by a character. Two characters together in a escape sequence are treated as single character.

e.g.    '\n'     Newline  
         '\t'     Horizontal tab

32. In C++ basic data types (except void ) can be modified according our needs using certain keyword known as type modifiers. e.g. signed, unsigned and long etc.

33.    **Abstract Class:**           A class with no instances (no objects) is known as abstract class.

**Concrete class:**       A class having objects is known as concrete class.

34. It is a member function having same name as it's class and which is used to initialize the objects of that class type with a legal initial value. Constructor is automatically called when object is created.

class Exam

```
{ int Marks; char Subject[20];
```

```
public: Exam () //constructor for class Exam
```

```
{ Marks = 0;
```

```
strcpy (Subject,"Computer"); }
```

```
};
```

35.    **Default Constructor:-** A constructor that accepts no parameters is known as default constructor. If no constructor is defined then the compiler supplies a default constructor.

```
student :: student ()
```

```
{ rollno=0; }
```

**Parameterized Constructor -:** A constructor that receives arguments/parameters, is called parameterized constructor.

```
student :: student(int r)
```

```
{ rollno=r; }
```

36.

Constructor	Destructor
Its name is same as the class name	Its name is same as the class name preceded by the tilde(~) sign.
Called automatically whenever object is created.	Called automatically whenever object goes out of scope.
e.g.                   class Flight { public: Flight(); // constructor for class Flight ~ Flight(); // Destructor for class Flight };	

37.    **Copy Constructor:-** A constructor that initializes an object using values of another object passed to it as parameter, is called copy constructor. It creates the copy of the passed object.

```
class Sample
```

```
{ int i,j;
```

```
public:
```

```
Sample (int a, int b) //Constructor
```

```
{ i = a; j = b; }
```

```
Sample (Sample &s) //Copy Constructor
```

```
{ i=s.i; j=s.j;
```

```
cout<<"Copy constructor Working\n";
```



```

}
void print( )
{
cout<<i<<"\t"<<j<<"\n";
}
};
void main( )
{
Sample S1(4,9);    //S1 initialized first constructor used
Sample S2(S1);     //S1 copied to S2. Copy constructor called.
Sample S3=S1;      //S1 copied to S3. Copy constructor called again.
-----
-----
}

```

38.

- a. Constructor functions are invoked automatically when the objects are created.
- b. No return type (not even void) can be specified for a constructor.
- c. They cannot be inherited, though a derived class can call the base class constructor.
- d. A constructor may not be static.
- e. It is not possible to take the address of a constructor.

39.

- a. Destructor functions are invoked automatically when the objects are destroyed.
- b. No argument can be provided to a destructor, neither does it return any value.
- c. They cannot be inherited.
- d. A destructor may not be static.
- e. It is not possible to take the address of a destructor.

40.

#### **Constructor Overloading:**

The constructor of a class may also be overloaded so that even with different number and types of initial values, an object may still be initialized.

#### **Default Arguments Versus Overloading:**

Using default arguments gives the appearance of overloading, because the function may be called with an optional number of arguments.

**e.g:**

#### **Prototype :**

float amount (float principal, int time=2, float rate=0.08);

Can be called as

Amount(2000.0,4,0.10);

Amount(3520.5,3);

Amount(5500.0);

41. A **text file** store information in ASCII characters. In text files, each line of text is terminated, with a special character known as EOL character.

A **binary file** store information in the same format in which the information is held in memory. In binary file, there is no delimiter for a line.

42. A stream is a sequence of byte.

**ofstream:** Stream class to write on files

**ifstream:** Stream class to read from files

**fstream:** Stream class to both read and write from/to files.

43. `get()` does not extract the delimiter newline character from input stream.

On the other hand `getline()` does extract the delimiter newline character from the input stream so that the stream is empty after `getline()` is over.

44. The `ios::out` is the default mode of `ofstream`. With the mode of the file does not exist, it gets created but if the file exists then its existing contents get deleted.

The `ios::app` is output mode of `ofstream`. With the mode of the file does not exist, it gets created but if the file exists then its existing contents are retained and new information is appended to it.

45.

Pointer is an address of a memory location. A variable, which holds an address of a memory location, is known as a Pointer variable (or Simply Pointer). For example `int *P;`

46. Protected members will be inherited into the derived class (they are accessible from the derived class). But Private members cannot be accessed from the derived class.

```
e.g. class S { private : int x;
              protected : int y;
              public :
              void show() { cout<<x<<y;}
            };
```

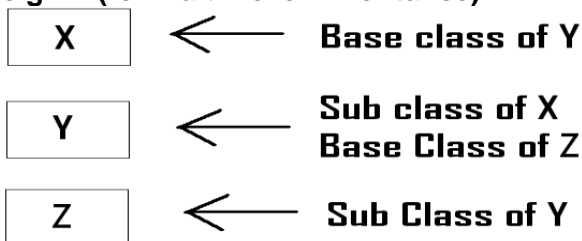
```
class S1      :      public S
{
  private: int z;
  public :
    void display() { cout<<x <<y<<z;}
};
```

**// here y and z can be accessed but x cannot be as it is private member of S**

(Remember that the memory will be reserved for private as well as protected members for the derived class object)

47. **Multilevel Inheritance:** When a subclass inherits from a class that itself inherits from another class, it is known as multilevel inheritance.

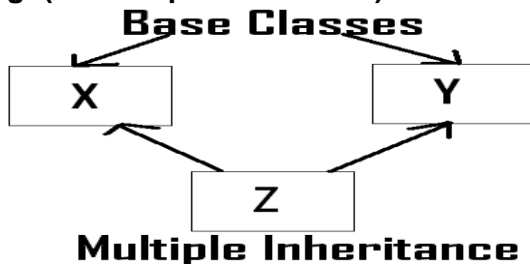
**e.g: (for Multi Level Inheritance)**



```
class X
{ ----- }
class Y:public class X
{ ----- }
class Z:protected Y
{
  -----
  -----
}
```

}  
**Multiple Inheritance:** When a sub class inherits from multiple base classes, it is known as multiple inheritance.

eg: (for Multiple Inheritance)



```
class X
{
-----
-----
}
class Y
{
-----
-----
}
class Z:public X, protected Y
{
-----
-----
}
```

48.

Public visibility	Protected visibility
The public derivation means that the derived class can access the public and protected members of the base class but not the private members of the base class.	The protected derivation means that the derived class can access the public and private members of the base class protectedly.
With publicly derived class, the public members of the base class become the public members of the derived class, and the protected members of the base class become the protected members of the derived class.	With protectedly derived class, the public and protected members of the base class become protected members of the derived class.
Example: <pre>class super { private: int x; void get(); public: int y; void put(); protected: int z; void disp(); }; class sub:public super { private: int a; void init(); public: int b; void readit(); protected: int c; void writeit(); };</pre>	Example: <pre>class super { private: int x; void get(); public: int y; void put(); protected: int z; void disp(); }; class sub:protected super { private: int a; void init(); public: int b; void readit(); protected: int c; void writeit(); };</pre>

49. In derivation of class first the base class constructor is invoked, followed by the derived class constructor, whereas in nested classes constructors of all the member objects are called before the constructors of the object enclosing other objects.

50.

Constructor	Normal Function
1. Constructor has same name as class name.	1. A normal function can have any legal name but not class name.
2. Constructor cannot have any return type value not even void.	2. A function should have any return type value.
3. Constructor is automatically called.	3. A function is explicitly called.
4. Constructor cannot be static.	4. A Function can be static.

51.

```
#define Max 70.0 //Error 1,2,3
void main() //Error 4
{
    int Speed ; //Error 5
    char Stop='N' ;
    cin>>Speed;
    if (Speed>Max) //Error 6
        Stop='Y' ;
    cout<<Stop<< endl ; //Error 7
}
```

52.

```
const float Max=70.0; //error1
void main() // error2
{
    int Speed;
    char Stop='N';
    cin>>Speed; // error3
    if (Speed>Max)
        Stop='Y';
    cout<<Stop<<endl; // error4
}
```

53.

```
#include<iostream.h>
void main( )
{ void Jumpto(int, int x=20); // prototype missing with default value
    int First = 10, Second = 20; //data type missing
    Jumpto(First, Second); //comma instead of ;
    Jumpto(Second);
}
void Jumpto(int N1, int N2) // default value redeclared
{ N1=N1+N2;
    cout<<N1<<N2; // cout and << operator required
}
```

54.

```
#include<iostream.h>
const int Max=10; // constant must be initialized
void main()
{ int Numbers[Max]= {20,50,10,30,40}; // expression syntax error
for(int Loc=Max-1;Loc>=10;Loc--) // Loc to be declared
cout<<Numbers[Loc]; // << operator required instead of >>
}
```

55.

```
#include<iostream.h>
#include<stdio.h> // error1
void main( )
{ struct movie
    { char movie_name[20];
      char movie_type;
      int ticket_cost; // error2
    }M;
gets(M.movie_name); // error3
cin>>(M.movie_type); // error4
}
```

56.

```
#include<iostream.h>
void main()
{
int X[]={60, 50, 30, 40},Y,Count=4; // error 1
cin>>Y;
for(int I=Count-1;I>=0;I--) // error 2,3
switch(I)
{ case 0:
case 2:cout<<Y*X[I]<<endl;break;
case 1: // error 4
case 3:cout<<(Y+X[I]);break; // error 5
}
}
```

57.

```
#include<iostream.h>
void main(){
int P[]={90, 10, 24, 15},Q,Number=4; //error 1
Q=9;
for(int I=Number-1;I>=0;I--) // error 2
switch(I)
{
case 0:
case 1 : // error 3
case 2:cout<<P[I]*Q<<endl; // error 4
break;

case 3:cout<<(P[I]+Q);
}
} //error 5
```

58.

```
#include<iostream.h>
void Assign(int Default1,int Default2=30);
void main()
{
int Present=25,Past=35;
Assign(Present,Past);
Assign(Past);

}
void Assign(int Default1,int Default2)
{
Default1=Default1+Default2;
cout<<Default1<<Default2;
}
```

59.

```
#include<iostream.h>
void Callme (int Arg1,int Arg2=20);
void main(){
int One=10,Two=20;
Callme(One;Two);
Callme(Two);
}
void Callme( int Arg1,int Arg2)
{
Arg1=Arg1+Arg2;
cout<<Arg1<<Arg2;
}
```

60.

```
#include<iostream.h>
#include<string.h>
typedef char String[80];
void main(){
String S="Peace";
int L=strlen(S);
cout<<S<<"has"<<L<<"characters"<<endl;
}
```

61.

Programming Paradigm: A Programming Paradigm defines the methodology of designing and implementing programs using the key features and building blocks of a programming language.

Following are the different programming paradigms:

- (i) Procedural Programming
- (ii) Object Based Programming
- (iii) Object Oriented Programming

62.

Reusability of classes is implemented through inheritance in C++.

Inheritance is implemented by specifying the name of the (base) class from which the class being defined (the derived class) has to inherit from.

It is done with the following syntax:

```
class<derived class name> : <base class name>
{
    <- derived class own features.
}
```

63.

OOP stands for Object Oriented Programming. In, Object-Oriented Programming (OOP), the program is organized around the data being operated upon rather than the operations performed. The basic idea behind OOP is to combine both, data and its functions that operate on the data into a single unit called object.

Following are the basic OOP concepts:

- 1. Data Abstraction 2. Data Encapsulation 3. Modularity
- 4. Inheritance 5. Polymorphism

64.

```
#include<iostream.h>
#include<string.h>
typedef char Text [80];
void main ( )
{
    Text T= "Indian";
    int Count=strlen(T);
    cout<<T<< "has" <<Count<< "characters"<<endl;
}
```

65.

```
typedef char STRING[50]; // error 1

void main()
{
    STRING City; // error 2
    gets(City);
    cout<<City[0]<<'\t'<<City[2]; // error 3
    cout<<City<<endl; // error 4
}
```

66.

(iii) and (iv)

Minimum value of NUM = 2 and Maximum value of NUM = 4

67.

(i) 50 # 51 # 52 # 53 # 54 # 55 #

Least value 50

Highest value 53

68.

Correct answer is 9GOLD\*9GOLD\*8SILVER\*

Minimum value of Guess is 0 and Maximum is 2

69

(iii) 2,

Minimum value of A is 2 and Maximum is 3

70

(ii) #2#3#4 because Minimum value of A=2 and Maximum is 4 and i is increasing (i++)

71.

(iii) **0,2,4,** because minimum value of A=2 and Maximum value is 5

72.

(iv) 20^ because Minimum value of A=20 and Maximum value is 23

73.

(iii) 1 2 3 4 5 6 because minimum value of rn is 5 and maximum is 8

74.

(iv) 20#250# because Start has minimum value as 2 and maxim as 3

75

(iii) 100#50#200# because minimum value of Tanker is 2 and Maximum value is 3

76.

Error: Undefined symbol y in function definition, if Y is declared with some value **then output will be:**

22,4

22,6

77.

10,8

20,8

78.

New Text:I@PPT@RRW

79.

1AJANTA

5JANTA



80.

30%41  
52%60  
40%25

81.

New Text : PPW%RR11N%

82.

NEW TEXT : @@e@ccddlle

83.

Line15&8  
Line211&9  
Line16&9  
Line212&10

84.

Output: 2

85.

Output:

5	9
9	7
7	7
7	9

86.

1  
12

87.

a=1b=2  
c=3

88.

30,100,1  
10,100,10  
120,1000,0  
10

89.

HuqTlOu

90.

teRmttoe

91.

1SARGAM  
9ARGAM

92.

vR.zGoOD  
R.zGOOD  
.zgOOD  
ZgOOD

93.

30  
34.5  
64.5

94.

90Bhakti  
70akti

95.

130,b

96.

Banka  
Bank  
Ban  
Ba  
B

97.

2@4@  
4#8#16#10#

98.

Queen@11  
Now@55

99.

4#6#  
12@18@

100.

D  
i

101.

(i) T.Book(1234567,"Ravi"); //Line 1  
T.Print(); //Line 2

(ii)

Function 4

OR

~Traveller()

**It is a Destructor function.**

102.

(i) P.Book(1234567,"Ravi"); //Line 1  
P.Print(); //Line 2

(ii) Function 4

**OR**

~Passenger()

**It is a Destructor function.**

103.

- (i) Function1 will be executed.
- (ii) Health G(H)

104.

- (i) Polymorphism (OR) Function Overloading (OR) Constructor Overloading
- (ii) Destructor, invoked or called when Object goes out of a scope.

105.

- (i) Option 1- Race T(30); is correct
- (ii) Constructor overloading.

106.

- (i) Function Overloading (Constructor overloading)
- (ii) Exam A(10) and Exam B("Comp", 10)

107.

- (i) Polymorphism (OR) Constructor Overloading (OR) Function Overloading.
- (ii) Destructor. It is invoked as soon as the scope of the object gets over.

108.

Valid and invalid statements are

welcome obj (33, 'a9');	<b>valid</b>
welcome obj1(50, '9');	<b>valid</b>
welcome obj3();	<b>invalid</b>
obj1= welcome (45, 'T');	<b>valid</b>

109.

- (i) Function 1 is called as Destructor, It will automatically executed at the time of destruction of the object of class TestMeOut.
- (ii) Function 2 is called as constructor (Non-parameterized or default constructor) ,it will automatically executed at the time of creation of the object of class TestMeOut.

110.

- (i) Function overloading (here it is constructor overloading).
- (ii) (**let** char name[20]; int X=30; strcpy(name, "SCIENCE"); are declared in the program)  
Test A(name); //Will execute Function 2  
Test B(name,X); //Will execute Function 4

111.

```
void SUCCESS()
{
    int count=0;
    ifstream f("STORY.TXT");
    char s[20];
    while (!f.eof())
    {
        f>>s;
        if(strcmpi(s,"STORY")==0)
            count++;
    }
    cout<<count;
    f.close();
}
```

112.

```
void TOWER()
{
    int count=0;
    ifstream f("WRITEUP.TXT");
    char s[20];
    while (!f.eof())
    {
        f>>s;
        if (strcmpi(s,"TOWER")==0)
            count++;
    }
    cout<<count;
    f.close();
}
```

113.

```
void EUCount()
{
    ifstream fi("IMP.TXT")
    char ch;
    int CountE=0,CountU=0;
    while(fi)
    {
        fi.get(ch);
        if(ch=='e' || ch=='E') CountE++;
        else if(ch=='u' || ch=='U') CountU++;
    }
    cout<<"E:"<<CountE<<endl;
    cout<<"U:"<<CountU;
    fi.close();
}
```

114.

```
void AECOUNT()
{
    ifstream fi("NOTES.TXT")
        char ch;
        int CountA=0,CountE=0;
    while(fi)
    {
        fi.get(ch);
        if(ch=='A' || ch=='a') CountA++;
        else if(ch=='E' || ch=='e') CountE++;
    }
    cout<<"A:"<<CountA<<endl;
    cout<<"E:"<<CountE;
    fi.close();
}
```

115.

```
void CountYouMe ()
{
    ifstream Fil;
    Fil.open("STORY.TXT",ios::in);
    char Word [80];
    int C1=0, C2=0;
    while (!Fil.eof())
    {
        Fil>>Word;
        if (strcmp(Word, "You")==0)
            C1++;
        else if (strcmp(Word, "Me") ==0)
            C2++;
    }
    cout<<"Count for You:"<<C1<<endl;
    cout<<"Count for Me:"<<C2<<endl;
    fil.close();
}
```

116.

```
void TotalDigits()
{
    ifstream chi("child.txt");
    int c1=0;
    char ch;
    while(chi)
    { chi.get(ch);
        if(ch>='0' && ch<='9') c1++;
    }
    cout<< "no of total digits in file:="<<c1;
    chi.close();
}
```

117.

```
void show()
{   ifstream fin("WORKER.DAT", ios::in|ios::binary);
    WORKER W;
    while(fin)
        { fin.read((char*)&W, sizeof(W));

            if( W.GetWage()<300)
                { W.display();}

        }
    Fin.close();
}
```

118.

```
void SHOW()
{
    ifstream File("UNO.TXT")
    char str[80];
    File.getline(str, 80);
    while(File)
    {
        if(str[0] == 'D' || str[0] == 'M')
            { cout<<str<<endl; }
        File.getline(str, 80);
    }
    File.close(); //Ignore
}
```

119.

```
void COUNT ( )
{ifstream Fil; // ifstream Fil("ARTICLE.TXT");
Fil.open("ARTICLE.TXT");
char Word[80] ,Ch;
int C1 =0, C2 = 0, I=0;
while(Fil.get(Ch))
{   if (Ch!= ' ')
    Word[I++] = Ch;
    else
    {
        Word[I] = '\0';
        if (strcmp(Word, "this")==0)
            C1++;
        else if (strcmp(Word, "these")==0)
            C2++;
        I=0;
    }
}   cout<<"Count of -this- in file:"<<C1;
    cout<<"Count of -these- in file:"<<C2;
Fil.close( );
}
```

120.

```
void WCount ()
{
    ifstream fil("POEM.TXT")
    char word[80];
    int WC=0;
    while( !fil.eof())
    {
        if( (strcmp(word, 'to')==0 ) || (strcmp(word, 'the')==0))
            WC++;
    }
    cout<< WC;
    fil.close();
}
```

121.

```
int countalpha()
{
    ifstream Fin("BOOK.txt");
    char ch;
    int count=0;
    while(!Fin.eof())
    {
        Fin.get(ch);
        if (islower(ch))
            count++;
    }
    Fin.close();
    return count;
}
```

122.

```
int countword()
{
    ifstream Fin("BOOK.txt");
    char ch[25];
    int count=0;
    while(!Fin.eof())
    {
        Fin>>ch;
        if (isupper(ch[0]))
            count++;
    }
    Fin.close();
    return count;
}
```

123.

```
void BlanksCount( )
{
ifstream fin("PR.TXT",ios::in);
char ch;
int B=0;
if(!fin)
{ cout<<"No words at all in the file.
So no blank spaces";
exit(0);
}
while(fin)
{
fin.get(ch);
if(ch== ' ')
B++;
}
cout<<"\nTotal number of Blank
Spaces in the file = "<<B;
Fin.close();
}
```



## Answers to 3 Mark Questions

1.

Undefined symbol C in function main()  
If error is removed then output will be:

a#0

d#0

D#0

2.

B#0

I#1

G#1

3.

A@1

0

A@2

75

A@3

120

4.

20#300#10#

6020@300@10@

6020\$300\$3000\$

5.

1,5

1,15

1,25

6.

B++50&100%

B++50&105%

B++55&115%

7.

10,240  
250,260,240  
260,240

8.

7,17  
7,17  
6

9.

11:20  
12:30  
13:45

10.

Error: length is not member of 'Package' in function main().  
If 'length' replaced with 'Length' then output will be:  
10x20x31  
11x21x31  
10x21x81

11.

15 20  
15 20 25  
Number=20

12.

XatOPtPQt!\*

13.

16 24 6  
6 34 16  
18 22 28

14.

30%41  
52%60

40%25

15.

```
void Economic()
{
    ITEMS I;
    ifstream fin("ITEMS.DAT",ios::binary);
    while (fin.read((char *)&I,sizeof(I)))
    {
        if(I.GetCost()<2500)
        I.See();
    }
    fin.close();
}
```

16.

```
void COSTLY()
{
    GIFTS G;
    ifstream fin("GIFTS.DAT",ios::binary);
    while (fin.read((char *)&G,sizeof(G)))
    {
        if(G.GetPrice()>2000)
        G.View();
    }
    fin.close();
}
```

17.

```
void READGAMES()
{
    GAMES obj;
    ifstream infile("GAMES.DAT");
    while(infile)
    {
        infile.read((char*)&obj,sizeof(obj));
        if(strcmp(obj.AgeR(),"8 to 13")==0)
        obj.Display();
    }
    infile.close();
}
```

18.

```
void READTOYS()
{
    TOYS obj;
    ifstream infile("TOYS.DAT");
    while(infile)
    {
        infile.read((char*)&obj,sizeof(obj));
        if(strcmp(obj.WhatAge(),"5 to 8")==0)
        obj.Display();
    }
    infile.close();
}
```

19.

```
void CopyBasket( )
{ Game G;
  ifstream fin;
  fin.open("GAME.DAT",ios::binary);
  ofstream fout;
  fout.open("BASKET.DAT",ios::binary);
  while(fin.read((char *)&G, Sizeof(G)))
  { if(strcmp(G.GameName,"Basket Ball")==0)
    fout.write((char *)&G, sizeof(G));
  }
```

20

```
void deleteBook(int bookNoToDelete)
{
  ifstream fin;
  ofstream fout;
  book ob;
  fin.open("BOOKS.DAT",ios::in|ios::binary);
  fout.open("temp.dat",ios::out|ios::binary);
  if(!fin||!fout)
  {
    cout<<"File not opened.";
    getch();
    return;
  }
  while(fin.read((char*)&ob,sizeof(ob)))
  {
    if(ob.checkbookno(bookNoToDelete)==1)
      fout.write((char*)&ob,sizeof(ob));
  }
  fin.close();
  fout.close();
  remove("BOOKS.DAT");
  rename("temp.dat","BOOKS.DAT");
}
```

21.

```
void show()
{ ifstream fcin("VINTAGE.DAT",ios::in|ios::binary);
  VINTAGE V;
  float prc;
  while ( fcin)
  { fcin.read((char*)&V, sizeof(V));
    prc=V.ReturnPrice();
    if( prc>=20000 & prc<=250000)
      V.VIEW();
  }fcin.close();
```

```

}
22.
void show()
{ ifstream fcin("NETBOOK.DAT",ios::in|ios::binary);
NETBOOK N;
float prc;
while ( fcin)
{ fcin.read((char*)&N, sizeof(N));
prc=N.ReturnPrice();
if( prc>=25000 & prc<=55000)
N.VIEW();
}fcin.close();
}

```

```

23.
void display()
{ mobile M;
ifstream fin("mobile.dat",ios::in|ios::binary);
fin.read((char*)&M, sizeof(M));
while(fin)
{ if(M.GetCalls()>1000) M.Billing();
fin.read((char*)&M, sizeof(M));
}fin.close();
}

```

```

24.
void Search()
{ tablet t;
long modeln; ifstream fin;
cout<<"enter the model no. of tablet : ";
cin>>modeln;
fin.open("tablet.dat",ios::binary);
while(fin.read((char*)&t,sizeof(t))
{ if(t.Model()==modeln)
t.Disp();
}
fin.close();
}

```

25.

```
void DisplayDemo ()
{
    CLUB CBJ;
    ifstream fin;
    fin.open ("CLUB.DAT", ios::binary);
    while ( fin.read (char*) &CBJ, sizeof(CBJ) )
    {
        if(CBJ.WhatType()=='L' || 'M' )
            CBJ.Display();
    }
    fin.close();
}
```

26.

```
void Read ( )
{
    FLIGHT F;
    ifstream fin;
    fin.open("FLIGHT.DAT",ios::binary);
    //OR ifstream fin ("FLIGHT. DAT", ios: :binary) ;
    while(fin.read((char*)&F,sizeof(F)))
    {
        if (strcmp(F. GetTo(),"Mumbai"))
            F.Display( ) ;
    } fin.close(); //
}
```

27.

```
void Read ( )
{
    TRAIN T;
    ifstream fin;
    fin.open("TRAIN.DAT",ios::binary);
    //OR ifstream fin ("TRAIN.DAT", ios::binary);
    while(fin.read((char*)&T, sizeof(T)))
    {
        if(strcmp(T.GetTo(),"Delhi")==0)
            T.Show( ) ;
    } fin.close( );
}
```

## Answers to 4 Mark Questions

```
1.  class PIC
    { int Pno; char Category[20]; char Location[20];
    void FixLocation();
    public:
    void Enter();
    void SeeAll();
    };
    void PIC::FixLocation()
    {
    if(strcmpi(Category,"Classic")==0)
    strcpy(Location,"Amina");
    else if(strcmpi(Category,"Modern")==0)
    strcpy(Location," Jim Plaq" );
    else if strcmpi(Category,"Antique")==0)
    strcpy(Location," Ustad Khan" );
    }
    void PIC::Enter()
    {
    cin>>Pno;gets(Category);
    FixLocation();
    }
    void PIC:: SeeAll()
    {
    cout<<Pno<<Category<<Location<<endl;
    }
```

```
2.  class Photo
    { int Pno; char Category[20]; char Exhibit[20];
    void FixExhibit();
    public:
    void Register();
    void ViewAll();
    };
    void Photo::FixExhibit()
    {
    if(strcmpi(Category,"Antique")==0)
    strcpy(Exhibit,"Zaveri");
    else if(strcmpi(Category,"Modern")==0)
    strcpy(Exhibit,"Johnsen");
    else if strcmpi(Category,"Classic")==0)
    strcpy(Exhibit,"Terenida");
    }
    void Photo::Register()
    { cin>>Pno;gets(Category);
    FixExhibit();
    }
    void Photo:: ViewAll()
    {
    cout<<Pno<<Category<<Exhibit<<endl;
```

```

    }
3.  class CONTEST
    { int Eventno; char Description[30]; int Score; char qualified;
    public:
    CONTEST(){ Eventno=11; strcpy(Description,"School level");
              Score=100; qualified='N'; }
    void Input()
    { cin>>Eventno; gets(Description);cin>>score; }
    void Award( int Cutoffscore)
    {
        if(Score>Cutoffscore)
            qualified='Y';
        else
            qualified='N';
    }
    void Displaydata()
    {cout<<"Event Number is"<<Eventno<<endl;
      cout<<"description is"<<Description<<endl;
      cout<<"Score is"<<Score<<endl;
      cout<<"Qualified is"<<qualified;
    }
};

```

```

4.  class Tourist
    { int CNo; char CType; int PerKM; int Distance;
    public :
        Tourist() { CNo=0000; CType='A'; }
    void CityCharges()
    {   if (CType=='A') PerKM=20;
        else if (CType=='B') PerKM=18;
        else if ( CType=='C')PerKM=15;
    }
    void Register (abc)
    { cout<<"enter cab no. "; cin>>CNo;
      cout<<"enter cab type"; cin>>CType;
      CityCharges();
    }
    void Display()
    { cout<<"enter distance";cin>>Distance;
      cout<<"cab no is"<<CNo<<endl;
      cout<<"cab type is"<<CType<<endl;
      cout<<"PerKm"<<PerKM<<endl;
      cout<<"Amount is"<<PerKM*Distance<<endl;
    }
};

```



5.

```
class ITEM
{
int Code;char Iname[20];float Price; int Qty;
float Offer;
void GetOffer() ;
public:
void GetStock ()
{ cin>>Code; gets (Iname) ;
  cin>>Price>>Qty;
  GetOffer() ;
}
void ShowItem ( )
{
cout<<Code<<Iname<<Price<<Qty<<Offer;
};
void ITEM: : GetOffer ()
{
if (Qty<=50)
Offer = 0;
else if (Qty <=100)
Offer = 5;
else
Offer = 10;
}
```

6.

```
class STOCK
{ int ICode,Qty; char Item[20]; float Price,Discount;
void FindDisc();
public:
void Buy();
void ShowAll();
} ;
void STOCK::Buy()
{ cin>>ICode;
gets(Item);
cin>>Price;
cin>>Qty;
FindDisc();
}
void STOCK::FindDisc()
{ if (Qty<=50)
Discount=0;
else if (Qty<=100)
Discount=5;
Else
Discount=10;
}
void STOCK::ShowAll()
```

```

{ cout<<ICode<<'\\t'<<Item<<'\\t'<<Price<<'\\t'<<Qty
<<'\\t'<<Discount<<endl;
}

```

7.

```

class RESORT
{ int Rno;
char Name [20];
float Charges;
int Days;
float COMPUTE();
public:
void Getinfo() ;
void Dispinfo();
};
void RESORT::Getinfo()
{ cin>>Rno;
gets (Name);
cin>>Charges;
cin>>Days;
}
void RESORT::Dispinfo()
{ cout<<Rno<<" "<<Name<<" "<<Charges<<"
"<<Days<< COMPUTE()<<endl;
} float RESORT::COMPUTE()
{
float Amount = Charges*Days;
if (Amount>11000)
Amount = 1.02*Days*Charges;
return Amount;
}

```

8.

```

class HOTEL
{ int Rno; char Name[20];float Tariff; int NOD;
float CALC() ;
public:
void Checkin() ;
void Checkout() ;
} ;
float HOTEL::CALC()
{ float Amount = Tariff*NOD;
if (Amount>10000)
Amount = 1.05*NOD*Tariff;
return Amount;
}
void HOTEL::Checkin()
{
cin>>Rno;
gets (Name);
cin>>Tariff;
cin>>NOD;
}
void HOTEL::Checkout()

```

```

{
cout<<Rno<<" "<<Name<<" "<<Tariff<<"
"<<NOD<<CALC ()<<endl;
}
9.. class TEST
{
    int TestCode; char Description[20];
    int NoCandidate,CenterReqd;
    void CALCNTR();
public:
    void SCHEDULE();
    void DISPTEST();
};
void TEST::CALCNTR()
{
    CenterReqd=(NoCandidate/100 + 1);
}
void TEST::SCHEDULE()
{
    cout<<"Test Code :";cin>>TestCode;
    cout<<"Description :";gets(Description);
    cout<<"Number :";cin>>NoCandidate;
    CALCNTR();
}
void TEST::DISPTEST()
{
    cout<<"Test Code :"<<TestCode<<endl;
    cout<<"Description :"<<Description<<endl;
    cout<<"Number :"<<NoCandidate<<endl;;
    cout<<"Centres :"<<CenterReqd<<endl;;
}

10. class FLIGHT
{
    int Fno; char Destination[20];
    float Distance, Fuel;
    void CALFUEL();
public:
    void FEEDINFO();
    void FEEDINFO();
    void SHOWINFO();
};
void FLIGHT::CALFUEL()
{
    if (Distance<=1000)
        Fuel=500;
    else
        if (Distance<=2000)
            Fuel=1100;
        else
            Fuel=2200;
}
void FLIGHT::FEEDINFO()
{
    cout<<"Flight No :"; cin>>Fno;
    cout<<"Destination :"; gets(Destination);
    cout<<"Distance :"; cin>>Distance;
    CALFUEL();
}

```

```

void FLIGHT::SHOWINFO()
{
    cout<<"Flight No : "<<Fno<<endl;
    cout<<"Destination : "<<Destination<<endl;
    cout<<"Distance : "<<Distance<<endl;;
    cout<<"Fuel : "<<Fuel<<endl; }

```

```

11. class Clothing
{
    char Code[25], Type[25];
    int Size ; char Material[30];
    float Price;
public:
    Clothing();
    void Calc_Price();
    void Enter();
    void Show();
};

Clothing::Clothing()
{
    strcpy(Code,"NOT ASSIGNED");
    strcpy(Type,"NOT ASSIGNED");
    Size=0;
    strcpy(Material,"NOT ASSIGNED");
    Price=0;
}

void Clothing::Calc_Price() or void Clothing::CalcPrice()
{if(strcmp(Type,"TROUSER") == 0 && strcmp(Material,"COTTON") == 0)
    Price=1500;
else if (strcmp(Type,"SHIRT") == 0 && strcmp(Material,"COTTON") == 0)
    Price=1200;
else if (strcmp(Type,"TROUSER")==0 && strcmp(Material,"COTTON")!=0)
    Price=1500*0.75;
else if (strcmp(Type,"SHIRT")==0)&& strcmp(Material,"COTTON")!=0 )
    Price=1200*0.75;
}

void Clothing::Enter()
{
    gets(Code); // or cin >> Code;
    gets(Type); // or cin >> Type;
    cin>>Size;
    gets(Material); // or cin >> Material;
    Calc_Price(); OR CalcPrice();
}

void Clothing::Show()
{
    cout<<Code<<Type<<Size<<Material<<Price<<endl;
}

12. class Travel
{
    char TCode[5]; //OR char *Tcode;
    int No_of_Adults;
    int No_of_Children;
    int Distance;
    float TotalFare;
public:

```

```

        Travel();
        void AssignFare();
        void EnterTravel();
        void ShowTravel();
};
Travel::Travel() //Constructor
{
    strcpy(TCode,"NULL");
    // OR TCode[0]='\0' OR strcpy(TCode,"\0")
// OR TCode=NULL if TCode is declared as char pointer
    No_of_Adults = 0;
    No_of_Children = 0;
    Distance = 0;
    TotalFare = 0;
}
void Travel::AssignFare()
{
    if(Distance>=1000)
        TotalFare = 500*No_of_Adults+250*No_of_Children;
    else
        if (Distance >= 500)
            TotalFare = 300*No_of_Adults+150*No_of_Children;
        else
            TotalFare = 200*No_of_Adults+100*No_of_Children;
}
void Travel::EnterTravel()
{
    gets(TCode); // or cin >> TCode;
    cin>>No_of_Adults>>No_of_Children>>Distance;
    AssignFare();
}
void Travel::ShowTravel()
{
    cout<<TCode<<No_of_Adults<<No_of_Children<<Distance<<TotalFare<<endl;
}

```

13.

```

class Candidate
{ long RNo; char Name[20]; float Score; char Remarks[20];
void AssignRem( );
public:
void Enter( );
void Display( );
};
void Candidate::AssignRem( )
{
    if (Score>=50)
        strcpy (Remarks,"Selected");
    else
        strcpy(Remarks,"Not Selected");
}
void Candidate: : Enter ( )
{
    cin>>RNo ;
    gets (Name) ; cin>>Score;
    AssignRem( );
}

```

```

    }
    void Candidate: :Display()
    {
        cout<<RNo<<Name<<Score<<Remarks<<endl;
    }

```

14.

class TAXPAYER

```

{
    char Name[30],PanNo[30]; float Taxabincm; double TotTax;
    void CompTax()
    {
        if(Taxabincm >500000)
            TotTax= Taxabincm*0.15;
        else if(Taxabincm>300000)
            TotTax= Taxabincm*0.1;
        else if(Taxabincm>160000)
            TotTax= Taxabincm*0.05;
        else
            TotTax=0.0;
    }
public:
    TAXPAYER(char nm[ ], char pan[ ], float tax, double tax) //parameterized constructor
    {
        strcpy(Name,nm);
        strcpy(PanNo,pan);
        Taxabincm=tax;
        TotTax=ttax;
    }

    void INTAX()
    {
        gets(Name);
        cin>>PanNo>>Taxabincm;
        CompTax();
    }
    void OUTAX()
    { cout<<Name<<'\\n'<<PanNo<<'\\n'<<Taxabincm<<'\\n'<<TotTax<<endl; }
};

```

15.

class Applicant

```

{
    long ANo; char Name[25]; float Agg; char Grade;
    void GradeMe( )
    {
        if (Agg >= 80)
            Grade = 'A';
        else if (Agg >= 65 && Agg < 80 )
            Grade = 'B';
        else if (Agg >= 50 && Agg < 65 )
            Grade = 'C';
        else
            Grade = 'D';
    }
}

```

```

    }
public:
    void Enter ( )
    { cout <<"\n Enter Admission No. "; cin>>ANo;
      cout <<"\n Enter Name of the Applicant "; cin.getline(Name,25);
      cout <<"\n Enter Aggregate Marks obtained by the Candidate :"; cin>>Agg;
      GradeMe( );
    }
    void Result( )
    {
      cout <<"\n Admission No. "<<ANo;
      cout <<"\n Name of the Applicant "<<Name;
      cout<<"\n Aggregate Marks obtained by the Candidate. " << Agg;
      cout<<"\n Grade Obtained is " << Grade ;
    }
};

16.
class ORDER
{ int ICode; char Item [ 20 ] ; float Price;  int Qty; float Discount;
void FindDisc();
public:
void Buy()
{ cin >> ICode;
  gets(Item);
  cin >> Price >> Qty;
  FindDisc();
}
void ShowAll()
{  cout << ICode << Item << Price<< Qty << Discount; }
};

void STOCK::FindDisc()
{
if (Qty < =50)
Discount=0;
else if (Qty < =100)
Discount=5;
else
Discount=10;
}

17.
class TravelPlan
{
    long PlaceCode;
    char Place[30];
    int Number_of_travellers;
    int Number_of_buses;
public:
    TravelPlan()
    {

```

```

        PlanCode=1001;
        Place="Agra";
        Number_of_travellers=5;
        Number_of_buses=1;
    }
    void NewPlan()
    {
        cout << "Enter Plan Code:";
        cin >> PlanCode;
        cout << "\n Enter Place:";
        gets(Place);
        cout << "\n Enter Plan:";
        cin >> Plan;
        cout << "\n Enter the number of travellers:";
        cin >> Number_of_travellers;
        if(Number_of_travellers >= 40)
            Number_of_buses=3;
        else if(Number_of_travellers >=20)
            Number_of_buses=2;
        else
            Number_of_buses=1;
    }
    void ShowPlan()
    {
        cout << "\n Plan Code: " << PlanCode << endl;
        cout << "\n Place Name: " << Place << endl;
        cout << "\n Number_of_travellers:" << Number_of_travellers << endl;
        cout << "\n Number_of_buses: " << Number_of_buses;
    }
};

```

18.

```

#include <iostream.h>
#include <ctype.h>
class DH{int seat;
    char name[20];
    static int taken;
public:
    void getdata(){
        taken++;
        cout<<"\nSeat Number = "<< taken;
        cout<<"\nName = ";
        cin.getline(name,20);
        seat=taken;
    }
    static void status(){
        cout<<"\nTotal Seats = 50";
        cout<<"\nSeats Taken = "<< taken;
        cout<<"\n Available = "<< 50-taken;
    }
};

```



```

    }
    void disp( ){
        cout<<"\nSeat No. = "<< seat;
        cout<<"\nName = "<< name;
    }
};
DH ob[50];
int DH::taken;

```

```

void main()
{
    int i,s;
    char mo='Y',c;
    for(i=0;mo=='Y' && i<50;i++){
        ob[i].getdata();
        DH::status();
        cout<<"\nMore? (Y/N) ";
        cin>>mo;
        mo=toupper(mo);
        c=cin.get();
    }
    cout<<"Enter Seat No. to display";
    cin>>s;
    ob[s-1].disp();
}

```

19.

```

class tour
{
    int tcode,adults,children,distance; float totalfare;
    void assignfare()
    {
        float cfare=50, afare=1500;
        if(distance<1500)
            afare=afare-(afare*25/100);
        totalfare=(children*cfare)+(adults*afare);
    }
public:
    travel()
    {
        tcode=adults=children=distance=totalfare=0; }
    void entertour()
    {
        do
        {
            cout<<"Enter tcode between 6-10 ";
            cin>>tcode;
            if (tcode<6 || tcode>10)
                cout<<"Invalid tcode "<<endl;
        }while(tcode<6 || tcode>10);
        cout<<"Enter children, adults, distance";
        cin>>children>>adults>>distance;
        assignfare();
    }
    void showtour()

```

```

{
    cout<<"tcode:"<<tcode<<endl;
    cout<<"children:"<<children<<endl;
    cout<<"adults :"<<adults<<endl;
    cout<<"distance:"<<distance<<endl;
    cout<<"total fare:"<<totalfare<<endl;
};

```

20.

class admission

```

{
    int admno;
    char name[20];
    int cls;
    float fees;
public:
    admission()
{
    admno=10;
    strcpy(name,"NULL");
    cls=0;
    fees=0;
}
void getdata()
{
    do
    {
        cout<<"Enter admno between 10-1500 ";
        cin>>admn
        if (admno<10 || admno>1500)
            cout<<"Invalid admission no !"<<endl;
    }while(admno<10 || admno>1500);
    cout<<"Enter name ";
    gets(name);
    cout<<"Enter class and fees ";
    cin>>cls>>fees;
}
void putdata()
{
    cout<<"Admno : "<<admno<<endl;
    cout<<"Name : "<<name<<endl;
    cout<<"Class : "<<cls<<endl;
    cout<<"Fees : "<<fees<<endl;
}
void draw_nos()
{
    int num;
    randomize();
    num=random(1491)+10;
    if (num==admno)
        putdata();
}
};

```

21.

Ans.

- (i) Multi Level Inheritance
- (ii) WallArea, ColorCode, Type, Advance
- (iii) Billing(), Print(), PBook(), PView(), Book(), View()
- (iv) Exterior(), Paint(), Bill()

22.

Ans.

- (i) Multi Level Inheritance
- (ii) WallArea, ColorCode, Type, Advance
- (iii) Bill(), BillPrint(), PBook(), PView(), Book(), View()
- (iv) Interior, Painting, Billing

23.

Ans.

- (i) Multi Level Inheritance
- (ii) Enroll(), View(), Enter(), Show()
- (iii) RollNo, Budget
- (iv) No, it is not possible because class Department is inheriting from class University

privately. So all public and protected members of class University will become private and hence cannot be accessed by objects.

24.

Ans.

- (i) Multi Level Inheritance
- (ii) Enroll(), View(), Enter(), Show()
- (iii) RollNo, Name, Budget
- (iv) Yes, it is possible as it is inherited publically.

25.

Ans.

- (i) Multi Level Inheritance
- (ii) OverallGrade, Remarks[20];
- (iii) Revaluate(), RPrint(), Sentry(), Sdisplay();
- (iv) Marks[5], Grade[5], SName[20];

26.

Ans.

- (i) Multi Level Inheritance
- (ii) FinalGrade, Comments[20];
- (iii) Rcalculate(), Rdisplay(), Mentry(), Mdisplay();
- (iv) M[5], Grade[5], Name[20];

27.

Ans.

- (i) Hierarchical Inheritance
- (ii) location, area, sale
- (iii) Enter(), Show(), Register (), Show();
- (iv) No Data Member, but Member functions are: Input(), Output()

28.

Ans.

- (i) Multiple Inheritance
- (ii) None
- (iii) Enter(), Display(), Input(), output()
- (iv) Input(), output(), Profile, Dname, DID

29.

Ans.

- (i) Multilevel Inheritance
- (ii) None
- (iii) Enter(), Display()
- (iv) FID, Address, NOE, Description, Input(), Output(), Assign(), Show(), Allocate()

30.

Ans.

- (i) Multiple Inheritance
- (ii) Register( ), SiteIn( ), SiteOut( ), Input( ), Output( )
- (iii) Register( ), Show( ), Input( ), Output( )
- (iv) No, function Output( ) is not accessible inside the function SiteOut( ), because Output( ) is a member of class FaceToFace and SiteOut( ) is a member of class Online and the classes FaceToFace and Online are two independent classes.

31.

Ans.

- (i) None of data members are accessible from objects belonging to class AUTHOR.
- (ii) Haveit(), Giveit()
- (iii) Data members : Employees, Acode, Aname, Amount  
Member function : Register(), Enter(), Display(), Haveit(), Giveit(), Start(), Show(),
- (iv) 70

32.

Ans.

- (i) Hierarchical Inheritance OR Single Level Inheritance
- (ii) 41 bytes
- (iii) SDName, Weight, Price
- (iv) EDInput(), EDSHOW(), DInput(), DSHOW()

33.

Ans.

- (i) Constructor of University Class ( Top most Base class)
- (ii) 106 bytes
- (iii) display(), input(), enrol(int,int), show(), enterdata(), displaydata()
- (iv) state

34.

- (i) None of the data members
- (ii) void Enterdesign( ), void dispdesign( )
- (iii) price, cost\_of\_cloth, design\_code, designfee, stitching, costprice, sellprice;
- (iv) 61 bytes

35.

- (i) Multiple Inheritance
- (ii) InCourse( ), InDistance( ), OutDistance( ), InRegular( ), OutRegular( )
- (iii) InCourse( ), OutCourse( ), InRegular( ), OutRegular( )
- (iv) Yes, It can be accessed by using the object because it is a public member.

36.

- (i) Base class : living\_being      Derived class : cow
- (ii) horn\_size, legs, jaws
- (iii) etchdata() and displaydata()
- (iv) No

37.

- (i) class MNC
- (ii) 129
- (iii) void Enter(), void Output(), void Add(), void Show(), void EnterData(), void DisplayData().
- (iv) char country[25]