QUESTIONS WITH SOLUTIONS
Topic: OOP,OVERLOADING AND C++ BASICS

1. Name the header files to which the following belong:
(i) puts() (ii) isalnum()

1. (i) stdio.h (ii) ctype.h

2. Write the name of the header files to which the following belong:
(i) strcat() (ii) atoi()

2. (i) string.h (ii) stblib.h

3. Name the header files in which the following belong:
(i) pow() (ii) random()

3. (i) math.h (ii) stdlib.h

4. Name the header files in which the following belong:
(i) gets() (ii) open()

4. (i) stdio.h (ii) fstream.h

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, "CBSE");
puts(String);
}
```

5. The header files are : stdio.h, string.h

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

```
void main() {
char word[] = "Exam";
cout<<setw(20)<<word;
}
```

6. The required header files are : iomanip.h and iostream.h

7. What is cascading?

7. The multiple uses of input or output operators in one statement are called cascading of I/O operators.

8. What is the purpose of default clause in a switch statement?

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. What are the applications of void data type in C++?

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. What is the effect of absence of break in switch-case statement?

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. Give the output of the following program segment:

```
int I =200, = 18;
cout << i/j << endl;
```

11. 11

12. When will you make a function inline?

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. What is the significance of any empty parentheses in a function declared?

13. void message() function declare with an empty parentheses, in means that the function does not pass any parameters.

14. How are abstraction and encapsulation inter-related?

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. Write the declaration of inline function named bar() that takes one argument of type float and return type int.

15.
```
inline int bar(float a);
 {
        ……..
 }
```

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 ;
  }
```

16.
```
#include<iostream.h>
 #include<math.h>
```

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;
}
```

17. #inlcude<iostream.h>
#include<maths.h>

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;
}
```

18. #inlcude<iostream.h>
#include<maths.h>
#include<ctype.h>

19. Write the names of the headers files to which the following belong:
(i) puts ( ) (ii) sin ( )

19. puts ( )→<stdio.h
Sin ( ) →<math.h>

20. Write the name of the header file to which the following belong:
(i) puts ( ) (ii) randomize ( )

20. setw ( )→<iomanip.h>
Sqrt ( ) →<math.h>

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;
    }
```

21. cout, cin →#include<iostream.h>
Tolower( ) → #include<ctype.h>

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;          }
```

22. cin, cout →#include<iostream.h>
toupper( ) → #include<ctype.h>

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).

23. #include<iostream.h>→cout
#include<string.h>→strlen( )

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).

Ans 24. #include<iostream.h>→cout
#include<string.h>→strlen( )

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;
                      }
  }
```

25. #include<iostream.h>→cout
#include<ctype.h>→toupper ( )

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;
                      }
  }
```

26. #include<ctype.h>→toupper( )
#include<iostream.h>→cout

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;
       }
```

27. #include<iomanip.h>→setw( )
#include<iostream.h>→cout

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

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

28. #include<iostream.h>→cout
#include<iomanip.h>→setw( )

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);
           }
```

29. iostream.h and ctype.h

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

30. While, Float, 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

31. For, INT, NeW, 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;
        }
```

32. iostream.h and math.h

33. Which of the following are valid identifiers?
Data_rec Data rec, 1 data, data 1, my.file, asm, switch, goto

33. Data_rec asm

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

34. 7

35. What is the significance of preprocessor directive?

35. A preprocessor directive is an instruction to the complier itself. A part of compiler called preprocessor deals with these directives, before real compilation process. # is used as preprocessor directive in C++.

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

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

38. stdlib.h , conio.h

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

39. math.h, stdlib.h

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);
}
```

40. ctype, stdio

2 Marks Questions

1. What is the difference between 'x' and "x" in C++?

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. What is a reference variable? What is its use?

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. 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;
}
```

3. 50

4. Differentiate between break and continue statements.

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. In a control structure switch-case, explain the purpose of using default.

5. The default statement gets executed when no match is found again the values specified in the switch-case statement.

6. Differentiate between getche() and getch()function.

6. In the getche() 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 getch() reads a character from keyboard exactly in the same manner but does not show it on the screen.

7. When are temporary variables created by C++ compiler?

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. How cout and puts() differ from each other?

Ans 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. Give the difference between the type casting and automatic type conversion. Also, give a suitable C++ code to illustrate both.

Ans 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 conversi performed by the compiler. It is als

For example
int A=1, B=2;
float C = (float)A/B; //Type Casting
cout<<C;
Output:
0.5
	For example:
	int N = 65;
	char C = N; // Automatic type conversion
	cout<<C;
	Output:
	A

10. What is the difference between Local Variable and Global Variable? Also, give a suitable C++ code to illustrate both.

Ans **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 ( )
{i
nt 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. What is the purpose of using a typedef command in C++?Explain with suitable example.

Ans 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. What is the difference between #define and const? Explain with suitable example.

Ans #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.

```
cons tint x = 10;
```

13. What is the difference between Actual Parameter and Formal Parameters? Also, give a suitable C++ code to illustrate both.

Ans

Actual Parameter	Fo
Parameters provided at the time of function calling are called actual parameters. These parameters are contain actual values.	Parameters provided at the tim parameters. These parameters are simple v 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 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.

Ans **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. What is benefit of using default parameter/argument in a function? Give a suitable example to ` illustrate it using C++ code.

Ans 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);//nValue(3,4);//override default value for nValue2
return 0; }
```

16. What is the benefit of using function prototype for a function? Give a suitable example to illustrate it using a C++ code.

Ans 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. What is the significance of classes in OOPs?

Ans 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. What do you understand by polymorphism? Give an example illustrating its use in a C++ program.

Ans 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. Define the term data encapsulation in term of object oriented programming. Give a suitable example using a C++ code to illustrate the same.

Ans 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. Define the term data hiding in the context of object oriented programming give a suitable example using a C++ code to illustrate the same.

Ans 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. Encapsulation is one of the major properties of OOP. How is it implemented in C++?

Ans 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. How are abstraction and encapsulation interrelated?

Ans 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. What is event driven programming?

Ans 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. What do you understand by function overloading? Give an example illustrating its use in a C++ program.

Ans 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. What the operator overloading? Explain with example.

Ans 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. What are the advantages of object oriented programming over procedural programming?

Ans 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. What are the major differences between Object Oriented Programming and Procedural Programming?

Ans

Object Oriented Programming	*Procedural Programming*
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.	Polymorphism, inheritance, Data Encapsulation not available.

28. What are syntax, run-time errors and logical errors?

Ans **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. What is the difference between while and do-while loop?

Ans 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. How is OOP implement in C++?

Ans 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. What is escape sequence? Name any two escape sequences.

Ans 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. What are type modifiers? Give examples.

Ans 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. Define abstract class and concrete class.

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

C**oncrete class:** A class having objects is known as concrete class.

34. 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;
}
```

Ans `#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
}
```

35.

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<<endline;
}
```

Ans .

```
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
        }
```

36.

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;
```

```
}
```

Ans
```
#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) //  dafault value  redeclared
            { N1=N1+N2;
            cout<<N1<<N2;   // cout and << operator required
            }
```

37.

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];
}
```

Ans
```
#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 >>
            }
```

38.

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);
}
```

Ans
```
#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
}
```

39.

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];
}}
```

Ans

```
#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
}
}
```

40.

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;
```

```
}
```

```
Ans #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
```

41.

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;
}
```

```
 Ans #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;
}
```

42.

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;
}
```

Ans

```
#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;
}
```

43.

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;
}
```

Ans

```
#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;
}
```

44.

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

Ans .

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

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

Ans

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 donewith the following syntax:

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

45. Write a short note on OO programming.

Ans 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

46.

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'<<end1;
}
```

Ans

```
#include<iostream.h>
#include<string.h>
typedef char Text [80];
void main ( )
{
Text T= "Indian";
int Count=str1en(T);
cout<<T<< "has" <<Count<< "cbaracters"<<end1;
```

```
}
```

47. 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<<endline;
 }
```

Ans

```
 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
 }
```

48.
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[]="ABCDEFGHIJK";
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

Ans (iii) and (iv)

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

49.

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 # (ii) 52 # 53 # 54 # 55

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

68.

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

Least value 50

Highest value 53

50. 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*

Ans Correct answer is 9GOLD*9GOLD*8SILVER*
Minimum value of Guess is 0 and Maximum is 2

51. 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,

Ans (iii) 2,
Minimum value of A is 2 and Maximum is 3

52.
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;
           }
```

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

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

53.

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,

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

54.

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^

(iii) 20^21^22^ (iv) 20^

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

55.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.

```
#include<iostream.h>
#include "stdlib.h"
 void main()
  { randomize();
    int num, rn;
    cin>>num;
    rn=random(num)+5;
      for(int n=1;n<=rn;n++)
       cout<<n<<" ";
   }
```

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

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

56.
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 cod e :

```
#include<iostream.h>
#include <stdlib.h>
void main ()
{
int guess[4]={200,150,20,250};
int Start=random(2)+2;
for(int C1=Start; C1<4; C1++)
cout<<guess[C1]<<'#';
}
```

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

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

57
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 :

```
#include<iostream.h>
#include <stdlib.h>
void main()
{ int GuessMe[4]={100,50,200,20};
int Taker=random(2)+2;
 for (int Chance=0;Chance<Taker;Chance++)
cout<<GuessMe[Chance]<<"#"; }
```

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

58. 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;
}
```

Ans Error: Undefined symbol y in function definition,
if Y is declared with some value **then output will be**:
22,4
22,6

59. 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 ;
}
```

Ans

10, 8
20, 8

60. 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;
         }
```

Ans .

```
New Text:I@PPT@RRW
```

61.

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;
}
```

Ans

```
1AJANTA
5JANTA
```

62. 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);
}
```

Ans

```
30%41
52%60
40%25
```

63.

Find the output of the following program :

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

```
void ChangeIt(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";
ChangeIt(OldText,'%');
cout<<"New TEXT:"<<OldText<<endl;
}
```

Ans `New Text : PPW%RRllN%`

64.

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; }
```

Ans .
NEW TEXT :@@e@ccddIIe

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

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

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

66. 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;
}
```

Ans Output: **2**

67. 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;
}
}
```

Ans .

Output:

```
5       9
9       7
7       7
7       9
```

68.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);
}
```

Ans

```
1
12
```

69.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; }
```

Ans.

```
a=1b=2
c=3
```

70.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;    }
```

Ans

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

71.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 ;
}
```

Ans .

```
HuqTlOu
```

72.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<<end1;
}
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;
}
```

Ans teRmttoe