Unit – I

Chapter -1

PROGRAMMING IN C++

Review: C++ covered in C++

Q1. What are the limitations of Procedural Programming ?

Ans. Limitation of Procedural Programming Paradigm

1. Emphasis on algorithm rather than data.
2. Change in a datatype being processed needs to be propagated to all the functions that use the same data type. This is a time consuming process.
3. The procedural programming paradigm does not model real world very well.

Q2. Define Class.

Ans. A Class represents a group of objects that share common properties and relationships.

Exp:-
```
Class ab
{
statements;
}
ab obj;
```

Class name is ab
{ } are used to write statements with in it
; is termination symbol of the statement
obj is the object of a Class to access the data members of the class

Q3. What are the features of OOP ?

Ans. Features of OOP

1. **Data Abstraction**
   Abstraction refers to the act of representing essential features without including the background details or explanations (i.e. Hiding of Data)
2. **Encapsulation**
   The wrapping up of data and functions (that operate on the data) into a single unit (called class) in known as **Encapsulation.** As a base Encapsulation is a way to implement **data abstraction.**
3. **Modularity**
   The act of dividing a complete program into different individual components (functions) is called modularity. **It is a property of a system that has been decomposed into a set of cohesive and loosely coupled modules**
4. **Inheritance**
   Inheritance is the capability of one class of things to inherit the **features or data members or properties from another class.**
   The Class whose properties of data members are **inherited,** is called **Base Class or Super Class** and the class that inherits these properties, is called **Derived Class or Sub Class.**
   Exp:- **If Class A** inherits the data of **Class B** then we can say **A is Sub Class and B is Super Class.**
5. **Polymorphism**
   Polymorphism is the ability for a message or data to be **processed in more than one form**. Polymorphism (overloading) is the property by which the same message can be

sent to objects of several different classes. In which the same operation is performed differently depending upon the data type it is working upon.

Q5. What is the difference between keyword and identifier?

Ans. Keyword is a special word that is reserved in C++ and having special meaning and purpose goto, struct, else, break etc.
Identifier is the user defined name given to a part of a program (variable name)

Q6. Describe different types of operators in C++.

Ans. Operators (to perform some computational operations or to perform some specific actions)

In C++ operators are divided into following categories:-
(Airthmatic, I/O, Increment/ Decrement, Relational & Logical Operators)

(a) I / O operators **(**Input /Output Operators)

Input Operator (>>) is used to read a value from standard input.
cin object is used for taking input from the user

Example :- int a;
cin>>a; (we can input integer value)

Output Operator(<<) is used to direct a value to standard output.
cout object is used for taking output on the display device.

Example:- int a;
cin>>a;
cout<<a;

The multiple use of input or output operators in one statement is called **cascading of I/O** operators

like : cin>>a>>b;
cout<<a<<b;

(b) Arithmetic Operators

+, - , * , / and %

(c) Increment/ Decrement Operators

Increment Operator (++)
Decrement Operator (- -)

We can use both the operators in postfix and prefix mode as per the requirements in the program statement

example:- postfix	**prefix**
int a=10;	int a=10;
a++;	++a;
cout<<a;	cout<<a;
output is a=10	**a=11**

Note:- The postfix form of ++, --, operators follows the rule **use-then change.**
The prefix form follows the rules **change then use**

(d) Relational Operators

< , <= , = = , > , >= and !=

(e) Logical Operators

logical OR (||) , logical AND (&&) , logical NOT(!)

(f) Conditional Operator (? , :)

C++ offers a conditional operator (? :) that stores a value depending upon a condition. This operator is ternary operator.

Syntax: - expression1 ? expression2 : expression3

Exp:-
```
int result;
Result = marks>= 50 ? 'P' : 'F' ;
```

(g) Some other operators (sizeof)

sizeof num // (num is variable name)
sizeof (type) // (c++ data type)

(h) Assignment Operator (=)

Example:

```
int total , item;
total = total + item;
total + =item;
```

Q7. What is the use of main function ?

Ans. Use of Main function in C++

Main ( ) is used to compile and execute the program code of C++ it is used with standard library and header files of the OOPS Program

Example

```
int main ( ) // return values with its parameters
void main ( ) // not return any value not return keyword is used

void main (parameter) // at the end of the only return keyword is
                      // used except parameters
```

Q8. What are the different data types in C++ ?

Ans. C++ Data types

Data types are means to identify the type of data and associated operations for handling it.

C++ data types are of two types

(i) **Fundamental Data Types**

(ii) **Derived Data Types**

(i) Fundamental Data Types

Fundamental Data Types are those that are not composed of other data types. There are **five** fundamental data types in C++: **Char, int, float, double & void** that represents character, integer, float.

(ii) Derived Data Type

(a) **Array:** It is a set of homogeneous values under one name (similar data elements). Array can be one dimensional, two dimensional or multi dimensional

Syntax:- datatype arrname [size] **// Single dimensional**

The data type of array elements is known as the Base Type of the array. **An ARRAY is a collection of variables on the same type that are referenced by a common name.**

Q9. Explain Two dimensional array with an example.
Ans.

Declaration of Two Dimensional Arrary
Syntax : - datatype arrname [size] [size] **// Two dimensional**

Example:- **int num[2][5]** **(array will execute 2 x 5 = 10 times)**

```
int main( )
{
int sales [5][5];
int i,j, total ;
for (i=0; i<5, i++)
{
        total =0;
        cout<< "\n";                    \\(escape sequence)
for (j=0 ; j< 5; j ++)
{
        cin>>sales[i][j];
        total =total + sales [i][j];
}
cout<< " sales is =  "<< total;
}
return 0;
}
```

Q10. What are functions ? Give syntax to define a function.

Ans. Functions

A function is a named unit of a group of program statements. This unit can be invoked from other parts of the program. **A function return values and numbers and arguments as per instructions stored in a function**

Syntax

```
type function-name (parameter list)
{
        body of the function
}
```

if type of a function is declared then it return values

Q11. Define Pointer, Reference and Constant .
Ans.

Pointer: A pointer is a variable that holds a memory address. This address is usually the location of another variable in memory.

Reference: A reference is an alternative name for an object. A reference variable provides an **alias** for a previously **defined variable.**

Constant: to declare constant value

Syntax :- const datatype var_name= value;

const int num =10;

Q12. Explain any two user defined data types.

Ans. User defined derived data types

Class: A class represents a group of similar objects. A class describes all the properties of a data type and an object is an entity created according to that description

```
class cls_name
{
        statement
};
```

Structure: A structure is a collection of variables of different data types referenced under one name. **Variables** defined under structure called with the help of structure object. **struct keyword is used to define structure**

```
struct stru_name
        {
                type varname;
                type varname;
        };
stru_name obj_name;

cin>>obj_name.varname;
cout<<obj_name.varname;
```

Q13. What do you mean by variable ?

i) **Ans. Variables:** Variables represent named storage locations whose values can be manipulated

Q14. What are the different types of errors in C++.

Ans. Types of errors in C++

Errors may be made during program creation even by experienced programmers. Such types of error are detected by the compiler. Debugging means removing the errors.

The errors are categorized in four types:-

(i) Syntax errors
(ii) Linking errors
(iii) Execution –time errors (Run Time errors)
(iv) Logical errors

Q15. Write the C++ equivalent expressions for the following.
Volume = 3 .1459r^2 h/3

Ans. Volume = 3.1459*r*r*h/3;

Q16. Find the syntax error (s) if if any, in the following program:

```
#include<iostream.h>
int main()
int x;
cin<< x;
for(int y = 0; y < 10; y++)
cout >> x+y;
```

Ans The syntax error are:

1. illegal`<<` operator in cin statement.
2. illegal`>>` operator in cout statement., return value function.

Q17. differentiate between a run-time error and syntax error. Give one example of each.

Ans. While execution, a compiled program may not behave properly because of some errors called run time errors. For example, divide by zero is a run time error. The following program segment will generate such an error.The following program segment will generate such an error

```
while flag
    {
        .
        b = b-1 ;
        term = a/b;
        .
     }
```

Array indicates out of bound, and range errors are other examples of run time errors. A syntax error, on the other hand, is because of misuse of a programming language. Any violation of a grammatical rule of a programming language will produce a syntax error. Such errors are caught by language compiler. The following statement is not systactically correct as far as c++ is concerned .

X= y + z** E;

Q18. What is the difference between an object and a class?

Ans. An object is an identifiable entity with some characteristics and behaviour. It represents an entity that can store data and its associated functions.
A class is a group of objects that share common properties and relationships

Q19. Write a program to input any number and print in reverse order

Ans in this program the number is input by user

```
#include<iostream.h>
#include<conio.h>
void main()
{
 clrscr();
 int n,s=0;
```

```
cout<<" enter the number:-  ";
cin>>n;
while(n>0)
{
 s=s*10+n%10;
 n=n/10;
}
cout<<"revese="<<s;
getch();
}
```

Q20. Write a program for decalring and calling of function inside <u>main</u> and defining outside the main ()

Ans

```
#include<iostream.h>
#include<conio.h>
void main()
{
 void fact();
 fact();
 getch();          // for freeze the montior
}
void fact()
{
 int i=1,n,fact=1;
 cout<<"enter the number=";
 cin>>n;
 while(i<=n)
 {
   fact=fact*i
   i++;
  }
  cout<<"fact="<<fact;
  getch();
}
```

Q21. Swapping of two numbers using function call by value & call by reference.

Ans

Call by value

```
#include<iostream.h>
#include<conio.h>
main()
{
 clrscr();
  int a,b;
  cout<<"enter the number=";
  cin>>a>>b;
  void swap(int,int);
  swap(a,b);
  cout<<"A="<<a<<endl;
  cout<<"B="<<b;
```

```
    getch();
}
void swap(int a,int b)
{
 int t=a;
 a=b;
 b=t;
 cout<<a<<b;
}
```

Call by reference

```
#include<iostream.h>
#include<conio.h>
void main()
{
void swap( int &a, int &b);              //prototype of a function
int num1,num2;
clrscr();
cout << " enter both numbers:   num1 & num2:";
cin>>num1;
cout<< "\n";
cin>>num2;
cout<< "\n  Before swapping numbers are \n";
cout<< " num1=  "<<num1;
cout<< "\n";
cout<< " num2=  "<<num2;
cout<< "\n";
swap(num1,num2);              //calling of function
cout<< "\n  After swapping numbers are \n";
cout<< " num1=  "<<num1;
cout<< "\n";
cout<< " num2=  "<<num2;
cout<< "\n";
getch( );                     //for freeze the monitor
}
```

```
        void swap(int &a, int &b)          //function definition
{
int temp=a;
a=b;
b=temp;
}
```

Q22. Name the header files of following built in functions :

Strcpy(), strcat(),log(), clrscr(),setw(),fabs(),isalnum(),isupper()

Ans.

Strcpy()	**string.h**
Strcat()	**string.h**
log()	**math.h**
clrscr()	**conio.h**
setw	**iomanip.h**
fabs()	**math.h**
isalpnum()	**ctype.h**
isupper()	**ctype.h**