

Assignment 11

Q1] What are the characteristics of C++ programming language.

- C++ programming language was developed by Bjarne Stroustrup.
 - The C++ programming language release in 1985.
 - C++ influenced by C, Sigma 67, Smalltalk, Algol, etc.
 - C++ is designed and developed in AT & T Bell lab in 1979.
 - Initially name of C++ was 'C with classes' later on it is C++.
- Characteristics of C++.

- 1) It is native programming language.
- 2) It is high level programming language.
- 3) Compiled programming language.
- 4) Object oriented programming language.
- 5) It supports both procedure oriented and object oriented programming language.
- 6) It is block structured programming language.
- 7) It supports static data type concept.

Q2] What is meant by class.

- Object orientation is one of the most important building blocks in C++.
 - To achieve concept of object orientation in C++ provide new datatype class.
 - Class is user defined datatype which contains two things in it.
- 1) Characteristics (data members)
 - 2) Behaviours (Function).

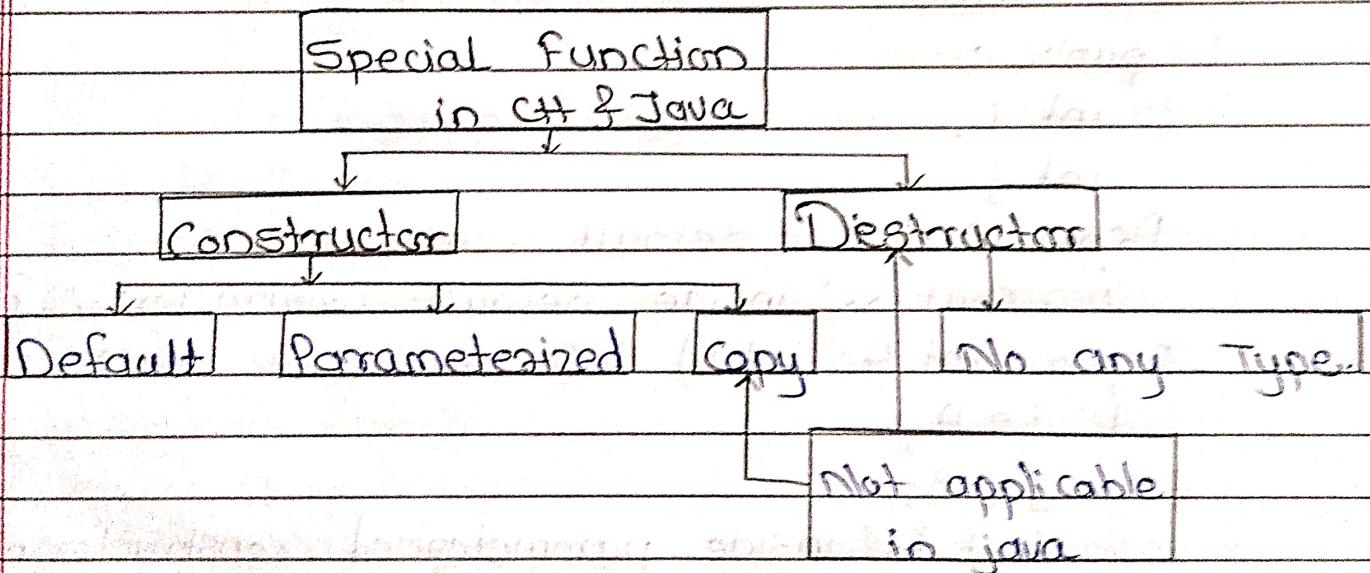
e.g.

```
class Demo
{
    public:
        int i;
        int j;
    void fun()
    {
        cout << "inside
    }
}
```

← class definition

Characteristics of class
(Data members)Behaviours of class (Functions)
fun of Demo class;

(3) Write note on Constructor and Destructor.



Constructor and Destructor are special function.

- Constructor is special function which get automatically call when we create object of class.

- Compiler will internally call constructor before allocating the memory to object.

- Destructor is a function which get automatically call before deallocating memory of a object.

There are three constructors in C++.

- 1) Default Constructor - A constructor with no parameters.
- 2) Parameterized Constructor - A constructor that takes arguments to initialize data members.
- 3) Copy Constructor - A constructor that creates a new object as a copy of an existing object.

e.g.

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

```
#include < iostream >
```

```
using namespace std;
```

```
class Demo :
```

```
{ public :
```

```
    int i;
```

```
    int j;
```

```
Demo () // default constructor
```

```
{ cout << "inside default constructor\n"; }
```

```
Demo (int A, int B) // parameterized constructor
```

```
{ i = A;
```

```
    j = B;
```

```
    cout << "inside parameterized constructor\n"; }
```

```
Demo (Demo &ref) // copy constructor
```

```
{ i = ref.i;
```

```
    j = ref.j;
```

```
    cout << "inside copy constructor\n"; }
```

```
~Demo () // Destructor
```

```
{ cout << "inside destructor\n"; }
```

```
}
```

```
int main()
```

```
{
```

```

Demo obj1;
Demo obj2 (11,22);
Demo obj3 (obj2);
return 0;
}

```

(Q4) What are the data types in C++? list them with their size.

Data Types in C++

Primitive Data Type	Derived Data Types	User Defined Data Types
character (1 byte)	Array	Structure
integer (4 bytes)	Pointer	Union
float (4 bytes)	Function	Enumeration
double (8 bytes)	Reference	class
void (1 byte)		
boolean (1 byte)		

• Derived Data Types

- 1) Array - Collection of elements of the same type.
 - 2) Pointer - Stores the memory address of a variable.
 - 3) Reference - Alias for another variable.
 - 4) Function - Represents a function.
- User Define Data Type.
- 1) class - Blueprint for object
 - 2) Structure - Group of variables of different types
 - 3) Union - Similar to Structure but shares memory
 - 4) Enumeration - Set of named integral Constants.

- Q5) Explain the tool chain of C++.
- Toolchain is considered as set of software which are used to convert human understandable program into machine understandable program.
 - Built in process of C++ programming language is as same as C programming.
 - It contains Editor, preprocessor, compiler, assembler, linker, loader.

Step I

Programmer uses editor to write a program. After writing get completed the file get save inside the hdd as Demo.cpp. The content in Demo.cpp is human readable and understandable.

Step II -

Preprocessor accept the input in Demo.cpp format it and generate the file which is expanded version of .cpp file.

File created by preprocessor is Demo.i. Content of .i file are human readable and understandable.

Step III

Output of preprocessor get input of compiler.

Compiler is a software which convert program one language to another.

In our case compiler convert program human understandable to machine dependent i.e. assembly language.

file created by compiler having extension

as Demo.asm or Demo.s.

Step IV

Output of compiler get pass as input of assembler.

Assembler is software which is used to convert machine dependent to binary file which is not executable.

Output of assembler is Demo.obj file.

Demo.obj contain the code in binary format but it is not directly executable.

Step V

Linker responsible to link Demo.obj file to executable format.

Linker will generate output as Demo.exe

Step VI

Demo.exe file is currently stored in Hdd.

To execute any application or program which has to be present in RAM.

Loader is responsible to load Demo.exe file to Hdd to RAM.

After loading Demo.exe file into RAM it considered as process and it get executed with help of operating system.

Q6) What is mean by Abstraction.

→ Abstraction in C++ is an object oriented programming concept that focuses on hiding the internal implementation details of a system and showing only the necessary functionality to the users.

If we want to hide something from outsiders then we use private specifier. Data which is return under private specifier which only access by private specifiers. By using concept of private specifier we achieve the object oriented paradigm which is Abstraction.

Q7] What is mean by Access Specifier? explain each with an example.

→ Concept of access specifier is used to specify which part of class can be accessed by outsiders.

Access specifiers are keywords used to define the accessibility or visibility of class member.

C++ provides three main access specifiers.

1) public : Member declared as public are accessible from anywhere in the program.

These members can be accessed both inside and outside the class.

2) private : Members declared as private are accessible only within the class in which they are defined.

These members cannot be accessed directly from outside the class.

3) protected : If class want to provide access its child class (Derived class) then that class should be under protected access specifier. Members declared as protected are accessible within class in which they are defined. In derived (child) classes.

Q8) Write a program to find out maximum of two numbers using procedural (C) and Object oriented approach (C++).

→ Using oriented approach (C).

```
#include <stdio.h>
int max(int a, int b)
{ if (a>b)
    { return a;
    } else
    { return b;
    }
}
int main()
{ int Num1, Num2, maximum;
printf("Enter two number :\n");
scanf ("%d %d", &Num1, &Num2);
maximum = max(Num1, Num2);
printf ("The maximum of %d and %d is %d\n", Num1, Num2, maximum);
return 0;
}
```

Using object oriented approach (C++)

```
#include <iostream>
#include <algorithm>
using namespace std;
int main()
{
    int Num1, Num2;
```

```

cout << "Enter two integers : ";
cin >> num1 >> num2;
int maximum = max(num1, num2);

if (num1 == num2)
    cout << "Both numbers are equal." << endl;
else if
    cout << "The maximum of " << num1 <<
    " and " << num2 << " is " << maximum <<
    endl;
}

return 0;
}

```

Q9) Explain the object Oriented programming paradigm (oop concepts).

oop

Encapsulation

Abstraction

Inheritance

Polymorphism

① Encapsulation :-

Binding of characteristics and behaviours that operate on data into a single unit or class.

Encapsulation restricts direct access to some of an object's components, which can prevent the accidental modification of data.

② Abstraction -

The concept of hiding the complex implementation details and showing only the necessary

Features of an object

- Abstraction helps in reducing programming complexity and effort.

③ Inheritance - In simple term inheritance is reusability. A mechanism by which one class can inherit attributes and methods from another class.

There are main three types of Inheritance

1) Single level Inheritance

2) Multi level Inheritance

3) Multiple Inheritance

4) Polymorphism - The term define as single name and multiple behaviour.

The ability of different classes to be treated as instances of the same class through a common interface.

Polymorphism allows methods to do different things based on the object it is acting upon, even though they share the same name.

Q10] Write down the difference between C and C++

C - C was developed by Dennis Ritchie between 1969 to 1973 at AT&T Bell Labs.

C++ - C++ was developed by Bjarne strausup in 1979 at AT&T Bell lab

C - C not support object oriented programming

C++ C++ support object oriented programming.

C - for the development of code, C support

features of an object.

- Abstraction helps in reducing programming complexity and effort.

③ Inheritance - In simple term inheritance is reusability. A mechanism by which one class can inherit attributes and methods from another class.

There are main three types of Inheritance.

1) Single level Inheritance.

2) Multi level Inheritance.

3) Multiple Inheritance.

④ Polymorphism - The term define as single name and multiple behaviour.

The ability of different classes to be treated as instances of the same class through a common interface.

Polymorphism allows methods to do different things based on the object it is acting upon, even though they share the same name.

Q10) Write down the difference between C and C++.

C - C was developed by Dennis Ritchie between 1969 to 1973 at AT&T Bell Labs.

C++ - C++ was developed by Bjarne stroustrup in 1979 at AT&T Bell lab

C - C not support object oriented programming

C++ C++ support object oriented programming.

C - for the development of code, C support

procedural programming.

C++ - Also known as hybrid language because C++ supports both procedural and object-oriented programming paradigm.

C - Data and functions are separated in C because it is a procedural programming language.

C++ - Data and functions are encapsulated together in form of an object in C++.

C - Function and operator overloading is not supported in C.

C++ - Function and operator overloading is supported by C++.

C - Standard I/O header is stdio.h and uses scanf() and printf(), functions are used for input/output in C.

C++ - Standard I/O header is iostream.h and uses cin and cout used for input/output in C++.

C - Virtual and friend functions are not supported by C.

C++ - Virtual and friend functions are supported by C++.