

## Assignment no - 11

- a) what are the characteristics of C++ programming language
- following are characteristics of C++ programming language
- 1) It is the negative programming language
  - 2) It is high level programming language
  - 3) It is compiled programming language
  - 4) It is an object oriented programming language.
  - 5) It's support procedural as well as object oriented
  - 6) It is block structured language
  - 7) It support static type of variable.

Q2] what is mean by class?

- - class is the userdefined datatype in C++
  - to Achieve the concept of object orientation in C++ provide the new datatype i.e class.
  - class is almost similar as the structure of in C programming
  - class contain 2 units

1) characteristics (Data members)

- characteristics of class which contain the values.

2) Behaviours (functions)

- Behaviours of the class which contain the functions.

Q3) write note on constructor & Destructor

→ constructor.

- constructor is a special function which gets automatically call when create the object of class, we can create multiple constructors of one class

- the name of constructor will be same as class name
- constructor should be inside public access specifier.
- there should be no any return type or datatype to constructor
- there are 3 types of constructor in C++
  - 1) Default constructor
  - 2) Parametrized constructor
  - 3) Copy constructor

### Destructor:

- Destructor is a special function which gets automatically call before deAllocating the memory of an object.
- Destructor get automatically call before deAllocating memory of an object
- we can create the destructor at once
- there is no any type of constructor

Q4) what are the data types in C++?  
list them with their size

→ There are 3 datatypes in C++

1) Primitive Datatype

1) char - 1 byte

2) int - 4 byte

3) float - 4 byte

4) double - 8 byte

5) boolean - 1 or 0 bit

6) void

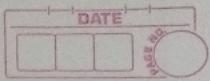
2) Derived Datatypes

1) Array

2) pointer - 8 byte

3) function

4) Reference



3) userdefined datatype

1) structure

2) union

3) Enumeration

4) class

Q5) Explain the tool chain of C++ program.



Step 1:-

- programmer uses editor to write the program.
- After writing gets completed, the file gets save with inside harddisk demo.C++
- the content of doc.C++ are human readable / understandable

Step 2:-

- Pre-processor Accept input, in demo.C++ format & it generate file which is the expanded version of .C++ file

- file created by `demo.i` stands for intermediate code

- Contents of `.i` file are human readable & understandable

Step 3:-

- output of pre-processor gets provided as input to the compiler

- Compiler is software which converts program from one language to another

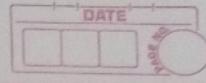
- In our case the compiler converts the program human undestable to machine dependant i.e Assembly lang

- file created by compiler having extension `.asm`

- Newly created file by compiler is `.asm` or `.s`

Step 4:-

- output of compiler gets pass to the Assembler



- Assembler is software which convert program machine dependant into machine understandable format

- output of Assembly having file extension : obj

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

Step 5:-

- linker is responsible to link .obj file generated by the Assembler & its dependant obj file

- linker generate output with extension : .exe

- in our case demo.exe is output of linker.

Step 6:-

- .exe generate by linker currently stored inside harddisk

- to execute any appn or program is has to be present or located inside RAM

- After loading the .exe file into RAM it get considered as process & it get executed with operating system

Q6) what is mean by Abstraction?

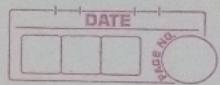
- Abstraction is a fundamental concept in object oriented programming (oop) in c++ that involves simplifying complex systems by breaking them down into smaller parts & hiding unnecessary details.
- To Achieve Abstraction we use Access specifier like private Access specifier.
- Abstraction is the process of hiding the internal details of an appn from the outer world.

Q7) what is mean by Access specifier? explain each with an example.

- The concept of Access specifier is used to specify which part of a class can be accessed by outside us & which part cannot be accessed by outside.

- there are 3 different types of Access specifier in c++

- 1) Public
- 2) Private
- 3) protected



## U Public Access Specifier

If we want to allow everyone to access without any restriction then it should be under the public access specifier.

e.g:-

```
class Demo {
```

```
    public:
```

```
        int x;
```

```
        int y;
```

```
    };
```

```
int main ()
```

```
{
```

```
    Demo obj;
```

```
    obj.x = 25;
```

```
    obj.y = 50;
```

```
    return 0;
```

```
}
```

## 2) Private Access Specifier

- If we want to a private members can only be accessed within the class.

e.g.:

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

```
class Demo {
```

```
private:
```

```
int age;
```

```
public:
```

```
void display ( int a )
```

```
{ age = a;  
cout << "Age = " << age << endl;
```

```
};
```

### 3) Protected Access Specifier-

- protected Access specifier is used in inheritance.

- the protected members can be accessed within the class and from the derived class

e.g. -

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

```
class Sample {
```

```
protected:
```

```
int age;
```

class derived : public sample {

public :

void displayAge (int a) {

age = a;

cout << "Age = " << age << endl;

}

}

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

# include <stdio.h>

int main ()

{

int a, b;

scanf ("%d %d", &a, &b);

if (a > b)

{

printf ("a is greater of two numbers");

else

{

printf ("b is greater of two numbers");

}

return 0;

}

Q9) Explain the object oriented programming paradigm (OOP concepts)

→ following are the object oriented programming paradigms

- 1) Encapsulation
- 2) Abstraction
- 3) Inheritance
- 4) Polymorphism

#### 1) Encapsulation

- Encapsulation means Binding the characteristics and behaviours in the class.

#### 2) Abstraction

- Abstraction means Hiding Something from outside the world.

- To Achieve Abstraction we use Access Specifier i.e

- 1) Public
- 2) Private
- 3) Protected

#### 3) Inheritance

- By using the inheritance we Achieve Reusability.

there are 4 type of inheritance

- 1) Single level inheritance
- 2) Multi level inheritance
- 3) Multiple inheritance
- 4) Hierarchical inheritance

In the inheritance we inherit the properties of parent class

Inheritance parent class & child class are present.

#### 4) Polymorphism

single name and multiple behaviour is considered as polymorphism

there are 2 type of polymorphism

##### 1) Compile time polymorphism

in compile time polymorphism the overloading concept is present.

##### 2) Run-time polymorphism

in run-time polymorphism the overriding concept is present.

