C++ notes

Constructor&Destructor

Constructor is a member function of a class.

The name of the constructor is same as the name of the class.

It has no return types so cant use return keyword.

It must be an instance member function that is it can never be static.

Class complex

{

Private ;

Int a,b;

Public;

Complex()

{

}

};

How to call constructor

Constructor is implicity invoked when an object is created.

Constructor is used to solve problem of initialization.

#include<conio.h>

Class complex

{

Private:

Int a,b;

Public:

Complex()

{

Cout<<”hello constructor”;

};

Void main()

{

Clrscr();

Complex c1(3,4) , c2;

Getch();

}

Compiler makes two constructor

1 default constructor

2. copy constructor

In copy constructor we need to create

Reference variable

Complex(complex &c)

{

a=c.a;

b=c.a;

}

Destr uctor

Destructor is an instance member function of a class

The name of the destructor is same as the name of a class but preceded by tilde (~) symbol

Destructor can never be static

Destructor has no return type

Destructor takes no argument (no overloading is possible)

It is invoked implicity when object is going to destroy

Why Destructor ?

It should be defined to release resources allocated to an object.

#include<conio.h>

#include<iostream.h>

Class complex

{

Private:

Int a,b;

Public:

~complex()

{

Cout<<”destructor “;

}

Void fun()

{

Complex obj;

}

Void main()

{

Clrscr();

Fun();

Getch();

}