Constructor and Destructor

⇒ Constructor :— It is a Special member function of a class which has the Same name as class name. To initialise data automatically into data member of a Class, Constructor member function is used. It is called automatically when an object of Class is Created.

Rules for Creating a Constructor :—

i) The name of the Constructor is the Same as the class name.

ii) Constructor doesnot return any value even though it has not void datatype.

iii) Constructor Can be defined as public member functions. The declaration of Constructor Can be done in the public Section of a class & it Can be defined outside the class by using Scope resolution operator.

Types of Constructor

There are three types of Constructor.

1) Default Constructor.
2) Parameterised Constructor.
(Constructor with Arguments).
3) Copy Constructor.

1) Default Constructor :-

A Constructor which has no arfument Called default Constructor. A Class has only one default Constructor.

Syntax:-

```
Class classname
{
private:
declaration of data member;
public
classname() // default Constructor
{
  body
}
}
```

2) Parametrised Constructor :-

It is also called Constructor with arfuments. A Constructor which has arfuments or parameter Called parametrise Constructor. A Class has one or more parametrised Constructor.

Syntax:-

```
class classname
{
private:
public:
classname(arfuments)
{
  body
}
```

3) Copy Constructor :-

A Constructor which is used to Copy the Contents of an object is Called Copy Constructor.

It is Called in two ways :-

a) Implicit Call.

b) Explicit Call.

Syntax :-

```
Class Classname
{
 private:
 ________
 ________
 public:
 classname (Classname, &obj)
 {
 function of Copying objects;
 ________
 ________
 }
};
```

-> Example of implicit Call :-

```
Classname xobj, yobj(xobj);
      yobj(xobj);
```

-> Example of explicit Call :-

```
Classname xobj, yobj = xobj;
```