Date No.

Object Construction and Destruction

(X) Class Constructor:

A class constructor is a special member function of a class that is executed whenever we create new objects of that class.

The constructor is called everytime the instance of a class is created i.e, object is created.

→ Constructor is very useful for setting initial values for certain membes variables.

→ Constructor can be defined either inside the class definition

OR,

outside class definition using class name and scope resolution (::) operator.

→ Constructor is always public and it cannot be private.

(X) Rules:

(i): Constructor must have same name as the class.

(ii) It doesn't have any datatype (not even void).

Date No.

(x) Types of Constructor:

These are three types of constructors used in class. They are as follows:

(i) Default constructor
(ii) Parametrized constructor
(iii) Copy constructor.

(i): Default constructor:

→ Default constructor is constructor not taking any argument and has no parametes.

→ When the object is created, the constructor initializes the data membeu.

→ Default constructor is provided by compiles in default.

(ii): Parameterized Constructor:

→ Parametrized constructor are the constructors that take arguments and contain parametes.

→ Parametrized constructor can be useful to set values to our data members.

(iii): Copy constructor:

→ Copy constructor are special constructors that take object as an argument and copies value of one object to another another.

Syntax: class_name constructor_name (& object_name).

Date No.

(*) Note:

i) Constructor overloading is possible and executable.
ii) If constructor is explicitly generated, the implicitly defined generated constructor is not provided.
(iii) To gain access to private members, use of getter and setter is done (ie, encapsulation process).
(iv) In copy constructor, we pass reference to the address.

(x) Destructor:

A destructor is a special member function of a class executed whenever the object goes out of scope or whenever delete expression is applied to a pointer to the object of that class.

Syntax: ~class_name();

→ It can neither return a value nor can take any parameters.

→ Destructors are very useful for releasing resources before coming out of program.

(*) Note:

→ Destructor is used to avoid memory leaks.
→ Implicitly defined destructor clears memory but if we have used DMA, we need to * uses destructor.