**Day 14**

**OBJECT ORIENTED PROGRAMMING SYSTEMS**

**Classes and Objects**

**Object-Oriented Programming** is a methodology or paradigm to design a program using classes and objects. It simplifies the software development and maintenance by providing some concepts defined below :

**Class** is a user-defined data type which defines its properties and its functions. Class is the only logical representation of the data. For example, Human being is a class. The body parts of a human being are its properties, and the actions performed by the body parts are known as functions. The class does not occupy any memory space till the time an object is instantiated.

**Object** is a run-time entity. It is an instance of the class. An object can represent a person, place or any other item. An object can operate on both data members and member functions.

Example 1:

class Student {

   String name;

   int age;

   public void getInfo() {

       System.out.println("The name of this Student is " + this.name);

       System.out.println("The age of this Student is " + this.age);

   }

}

public class OOPS {

   public static void main(String args[]) {

       Student s1 = new Student();

       s1.name = "Aman";

       s1.age = 24;

       s1.getInfo();

       Student s2 = new Student();

       s2.name = "Shradha";

       s2.age = 22;

       s2.getInfo();

   }

}

Example 2:

class Pen {

   String color;

   public void printColor() {

       System.out.println("The color of this Pen is " + this.color);

   }

}

public class OOPS {

   public static void main(String args[]) {

       Pen p1 = new Pen();

       p1.color = blue;

       Pen p2 = new Pen();

       p2.color = black;

       Pen p3 = new Pen();

       p3.color = red;

       p1.printColor();

       p2.printColor();

       p3.printColor();

   }

}

**Note** : When an object is created **using** a new keyword, then space is allocated for the variable in a heap, and the starting address is stored in the stack memory.

**‘this’ keyword :** ‘this’ keyword in Java that refers to the current instance of the class. In OOPS it is used to:

1. pass the current object as a parameter to another method
2. refer to the current class instance variable