DAY – 9

# JAVA:

****OBJECT-ORIENTED PROGRAMMING:****

It allows developers to structure code using classes and objects, making it easier, reusable, and scalable.

****CLASS:****

A Class is a blueprint that defines properties and behaviors.

Class Student

{

// variables & methods

}

****OBJECT:****

An Object is an instance of a class representing real-world entities.

Student s1=new Student ();

S1.run ();

****INHERITANCE:****

* Accessing properties from one class to another class is called Inheritance.
* It represents the IS-A relationship, which is also known as the parent-child relationship.
* Types:
  + Single
  + Multi-level
  + Hierarchical
  + Multiple
  + Hybrid
* In Java, Multiple inheritance is not supported.

****ABSTRACTION:****

ABSTRACT CLASS:

* It is a class that is declared with abstract keywords and is known as an abstract class.
* It can have abstract or non-abstract methods.

ABSTRACTION:

It is a process of hiding the information details and showing only functionality to the user.

Rules of Abstract class:

* An Abstract class must be declared with an abstract keyword.
* It can have abstract and non-abstract methods.
* It can’t be Instantiated. // Instantiated => creating new instances of objects to be used in a program.
* It can have static methods and constructors.
* It can have final methods.

Abstract class syntax:

Public abstract class Shape

{

public abstract double area ();

public void display ()

{

System.out.println(“Abstract classes”);

}

}

Abstract method syntax:

abstract void printStatus();

****INTERFACE:****

* An Interface in java is a blueprint of a class.
* It has static constants and abstract methods.
* It is a mechanism to achieve abstraction
* Interfaces have abstract methods and variables.
* It can’t have a method body.
* Interface fields are public, static, and final by default

Syntax:

interface <interface\_name>

{

// declare constant fields.

// declare methods that abstract

}

Declaring an Interface:

interface Animal

{

void eat ();

void sleep ();

}