OOP

Lab Manual (Lab 5)

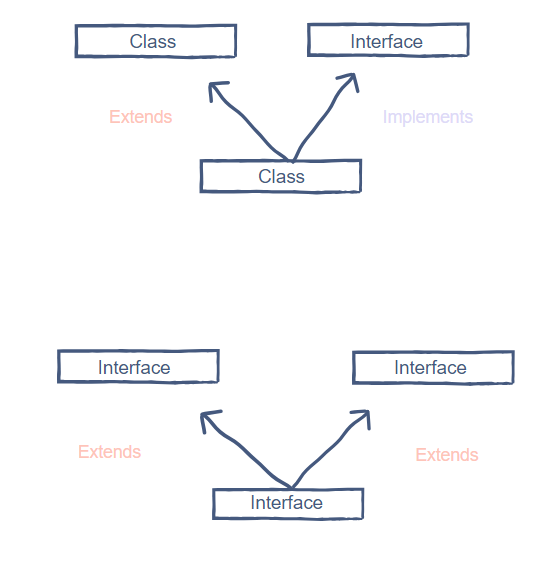
****

Session: Spring 2024

LAB INSTRUCTOR: AYESHA MAJID ALI

**Differences between an interface and a class:**

* An interface cannot be instantiated ie., you cannot create an object of an interface. However, classes may have an object.
* An interface is implemented without any constructors.
* Each field in an interface must be declared final and static.
* An interface is implemented by another class. It is never extended by a class.
* An interface, however, may extend multiple interfaces.



**Why do we need interfaces?**

* Data abstraction
* To achieve multiple inheritances
* Allow different objects to interact easily
* Facilitate reuse of the software

**Example 1**

This example shows how an interface is **implemented** by a class. (Pig implements Animal)

*// Interface*

interface Animal {

  public void animalSound(); *// interface method (does not have a body)*

  public void sleep(); *// interface method (does not have a body)*

}

*// Pig "implements" the Animal interface*

class Pig implements Animal {

  public void animalSound() {

*// The body of animalSound() is provided here*

    System.out.println("The pig is calling you: WEE WEEE WEEEEE");

  }

  public void sleep() {

*// The body of sleep() is provided here*

    System.out.println("Sleep Sound: ZzzzzzzZZzzzzZZzzz");

  }

}

class main {

  public static void main(String[] args) {

    Pig p = new Pig();  *// Create a Pig object*

    p.animalSound();

    p.sleep();

  }

}

**Example 2**

This example runs for **multiple interfaces** problem. There are two interfaces which are implemented by a single class.

interface InterfaceOne {

  public void Method(); *// interface method*

}

interface InterfaceTwo {

  public void Method2(); *// interface method*

}

*// Demo "implements" InterfaceOne and InterfaceTwo*

class Demo implements InterfaceOne, InterfaceTwo {

  public void Method() {

    System.out.println("Text One");

  }

  public void Method2() {

    System.out.println("Text two");

  }

}

class main {

  public static void main(String[] args) {

    Demo Obj = new Demo();

    Obj.Method();

    Obj.Method2();

  }

}

TASKS:

TASK 1: Write a Java program to create an interface Shape with the getArea() method. Create three classes Rectangle, Circle, and Triangle that implement the Shape interface. Implement the getArea() method for each of the three classes.

TASK 2: Write a Java program to create an interface Shape1 with the getArea() method and interface Shape2 for getPerimeter method. Create three classes Rectangle, Circle, and Triangle that implement the Shape and Shape1 interface. Implement the getArea() and getPerimeter() method for each of the three classes.

TASK 3: Write a Java program to create an abstract class Shape3D with abstract methods calculateVolume() and calculateSurfaceArea(). Create subclasses Sphere and Cube that extend the Shape3D class and implement the respective methods to calculate the volume and surface area of each shape.