Name – Harshit

MIS – 112316018

Java Lab

Assignment 6

1. Study about the Inheritance and its types in java.

**Inheritance in Java**

Inheritance is one of the fundamental principles of Object-Oriented Programming (OOP). It allows one class (subclass or child class) to acquire the properties and behaviors (methods and attributes) of another class (superclass or parent class). This promotes code reuse, modularity, and a hierarchical structure.

**Types of Inheritance in Java**

Java supports different types of inheritance, but **multiple inheritance through classes** is **not supported** due to ambiguity issues. However, it can be achieved using **interfaces**.

**1. Single Inheritance**

* In this type, a subclass inherits from a single superclass.
* It allows code reuse and extension of functionalities.

**2. Multilevel Inheritance**

* A class inherits from another class, which in turn inherits from another class, forming a chain.

**3. Hierarchical Inheritance**

* A single superclass is inherited by multiple subclasses.

**4. Multiple Inheritance (Through Interfaces)**

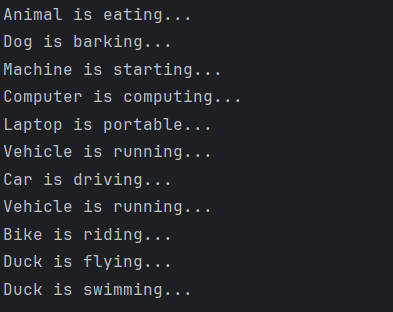
* Java does **not** support multiple inheritance with classes to avoid ambiguity issues.
* However, it **supports multiple inheritance using interfaces**.

1. Implement the types of inheritance using separate example.

Code:

// Implementing different types of inheritance in Java  
  
// 1. Single Inheritance Example  
class Animal {  
 void eat() {  
 System.*out*.println("Animal is eating...");  
 }  
}  
class Dog extends Animal {  
 void bark() {  
 System.*out*.println("Dog is barking...");  
 }  
}  
  
// 2. Multilevel Inheritance Example  
class Machine {  
 void start() {  
 System.*out*.println("Machine is starting...");  
 }  
}  
class Computer extends Machine {  
 void compute() {  
 System.*out*.println("Computer is computing...");  
 }  
}  
class Laptop extends Computer {  
 void portable() {  
 System.*out*.println("Laptop is portable...");  
 }  
}  
  
// 3. Hierarchical Inheritance Example  
class Vehicle {  
 void run() {  
 System.*out*.println("Vehicle is running...");  
 }  
}  
class Car extends Vehicle {  
 void drive() {  
 System.*out*.println("Car is driving...");  
 }  
}  
class Bike extends Vehicle {  
 void ride() {  
 System.*out*.println("Bike is riding...");  
 }  
}  
  
// 4. Multiple Inheritance using Interfaces  
interface Flyer {  
 void fly();  
}  
interface Swimmer {  
 void swim();  
}  
class Duck implements Flyer, Swimmer {  
 public void fly() {  
 System.*out*.println("Duck is flying...");  
 }  
 public void swim() {  
 System.*out*.println("Duck is swimming...");  
 }  
}  
  
public class InheritanceExample {  
 public static void main(String[] args) {  
 // Testing Single Inheritance  
 Dog myDog = new Dog();  
 myDog.eat();  
 myDog.bark();  
  
 // Testing Multilevel Inheritance  
 Laptop myLaptop = new Laptop();  
 myLaptop.start();  
 myLaptop.compute();  
 myLaptop.portable();  
  
 // Testing Hierarchical Inheritance  
 Car myCar = new Car();  
 myCar.run();  
 myCar.drive();  
  
 Bike myBike = new Bike();  
 myBike.run();  
 myBike.ride();  
  
 // Testing Multiple Inheritance  
 Duck myDuck = new Duck();  
 myDuck.fly();  
 myDuck.swim();  
 }  
}

Output:



Code:  
// Base class  
class VideoTape {  
 protected String title;  
 protected int duration;  
 protected boolean availability;  
  
 public VideoTape(String title, int duration, boolean availability) {  
 this.title = title;  
 this.duration = duration;  
 this.availability = availability;  
 }  
  
 public void display() {  
 System.*out*.println("\nVideo Details:");  
 System.*out*.println("Title: " + title);  
 System.*out*.println("Duration: " + duration + " minutes");  
 System.*out*.println("Available for rent: " + (availability ? "Yes" : "No"));  
 }  
}  
  
// First derived class  
class Film extends VideoTape {  
 private String producer;  
 private double boxOfficeCollection;  
  
 public Film(String title, int duration, boolean availability,  
 String producer, double boxOfficeCollection) {  
 super(title, duration, availability);  
 this.producer = producer;  
 this.boxOfficeCollection = boxOfficeCollection;  
 }  
  
 @Override  
 public void display() {  
 super.display();  
 System.*out*.println("Producer: " + producer);  
 System.*out*.println("Box Office Collection: $" + boxOfficeCollection + " million");  
 }  
}  
  
// Second derived class  
class MusicClip extends VideoTape {  
 private String bandName;  
 private String genre;  
  
 public MusicClip(String title, int duration, boolean availability,  
 String bandName, String genre) {  
 super(title, duration, availability);  
 this.bandName = bandName;  
 this.genre = genre;  
 }  
  
 @Override  
 public void display() {  
 super.display();  
 System.*out*.println("Band/Artist: " + bandName);  
 System.*out*.println("Genre: " + genre);  
 }  
}  
  
// Main class to test the implementation  
public class Main {  
 public static void main(String[] args) {  
 // Create instances of each class  
 VideoTape video = new VideoTape("Generic Video", 90, true);  
 Film film = new Film("The Matrix", 136, false, "Joel Silver", 463.5);  
 MusicClip musicClip = new MusicClip("Bohemian Rhapsody", 6, true, "Queen", "Rock");  
  
 // Display using all three versions of display()  
 System.*out*.println("1. Calling display() on VideoTape object:");  
 video.display();  
  
 System.*out*.println("\n2. Calling display() on Film object:");  
 film.display();  
  
 System.*out*.println("\n3. Calling display() on MusicClip object:");  
 musicClip.display();  
 }  
}

Output:  
