Lab 26

Self Practice Solution: Write two sample codes on multilevel and hybrid inheritance

Java supports multiple types of inheritance, including multilevel and hybrid inheritance. Here are two sample codes, one demonstrating multilevel inheritance and another demonstrating hybrid inheritance.

Multilevel Inheritance:

In multilevel inheritance, a class inherits from another class, and that subclass can further be inherited by another class, creating a chain of inheritance.

class Grandparent {

void grandparentMethod() {

System.out.println("Grandparent method.");

}

}

class Parent extends Grandparent {

void parentMethod() {

System.out.println("Parent method.");

}

}

class Child extends Parent {

void childMethod() {

System.out.println("Child method.");

}

}

public class MultilevelInheritanceDemo {

public static void main(String[] args) {

Child child = new Child();

child.grandparentMethod(); // Inherited from Grandparent

child.parentMethod(); // Inherited from Parent

child.childMethod(); // Unique to Child

}

}

In this example, `Child` inherits from `Parent`, and `Parent` inherits from `Grandparent`, creating a multilevel inheritance chain.

Hybrid Inheritance:

Hybrid inheritance is a combination of multiple types of inheritance, such as single and multiple inheritance. In this example, we demonstrate a hybrid inheritance scenario involving interfaces and classes.

interface Flyable {

void fly();

}

interface Swimmable {

void swim();

}

class Bird {

void birdMethod() {

System.out.println("Bird method.");

}

}

class Fish {

void fishMethod() {

System.out.println("Fish method.");

}

}

class Duck extends Bird implements Flyable, Swimmable {

@Override

public void fly() {

System.out.println("Duck is flying.");

}

@Override

public void swim() {

System.out.println("Duck is swimming.");

}

}

public class HybridInheritanceDemo {

public static void main(String[] args) {

Duck duck = new Duck();

duck.birdMethod(); // Inherited from Bird

duck.fly(); // Implemented from Flyable

duck.swim(); // Implemented from Swimmable

}

}

In this example, `Duck` inherits from the `Bird` class and implements the `Flyable` and `Swimmable` interfaces. This demonstrates hybrid inheritance, combining class-based inheritance and interface-based inheritance.

These examples illustrate multilevel and hybrid inheritance scenarios in Java, showcasing how classes and interfaces can be used to create complex inheritance hierarchies.