Name : Prathik Balaji N
Date : 09-08-2024

JAVA

1. Implement Abstract class with overloading and overriding.

```
package Samp;

public abstract class Vehicle {
    abstract void drive();

    void startEngine() {
        System.out.println("Engine started");
    }
}

class Car extends Vehicle {
    private String model;

    Car(String model) {
        this.model = model;
    }

    void drive() {
        System.out.println("Driving " + model);
    }

    void startEngine(boolean pushButtonStart) {
        String method = pushButtonStart ? "push button" : "key";
        System.out.println("Starting engine with " + method);
    }
}

class Bike extends Vehicle {
    private String type;
```

```
    Bike(String type) {
        this.type = type;
    }

    void drive() {
        System.out.println("Riding " + type);
    }
}
public class Main {
    public static void main(String[] args) {
        Vehicle myCar = new Car("Sedan");
        Vehicle myBike = new Bike("Sports");

        myCar.drive();
        myBike.drive();

        myCar.startEngine();

        Car anotherCar = new Car("SUV");
        anotherCar.startEngine(true);
    }
}
```

OUTPUT :

```
Driving Sedan
Riding Sports
Engine started
Starting engine with push button
```

2. Implement Multiple inheritance with Interface.

```
interface Printable {
    void print();
}
```

```
interface Showable {
    void show();
}

class TestInterface implements Printable, Showable {
    public void print() {
        System.out.println("Printing...");
    }

    public void show() {
        System.out.println("Showing...");
    }
}

public class Main {
    public static void main(String[] args) {
        TestInterface obj = new TestInterface();
        obj.print();
        obj.show();
    }
}
```

OUTPUT :

```
Printing…
Showing…
```

3. Show final methods in the class that can't be overridden.

```
class Vehicle {
    final void run() {
        System.out.println("Vehicle is running");
    }

    void speedUp() {
        System.out.println("Vehicle is speeding up");
    }
```

```
}

class Bike extends Vehicle {
    void speedUp() {
        System.out.println("Bike is speeding up");
    }
}

public class FinalProp{
    public static void main(String[] args) {
        Bike bike = new Bike();
        bike.run();
        bike.speedUp();
    }
}
```

OUTPUT :

```
Vehicle is running
Bike is speeding up
```