**Assignment-3**

**Code-1:**

import java.io.Serializable;

// Customer class implementing Serializable

Class Customer implements Serializable {

Private static final long serialVersionUID = 1L; // Required for Serializable

Private int id;

Private String name;

Private String contactNo;

Private String address;

Public Customer(int id, String name, String contactNo, String address) {

This.id = id;

This.name = name;

This.contactNo = contactNo;

This.address = address;

}

// Getters and setters (not shown for brevity)

@Override

Public String toString() {

Return “Customer{“ +

“id=” + id +

“, name=’” + name + ‘\’’ +

“, contactNo=’” + contactNo + ‘\’’ +

“, address=’” + address + ‘\’’ +

‘}’;

}

}

**Code-2:**

Public class SynchronizationDemo {

Private static int counter = 0;

Public static void main(String[] args) {

Thread thread1 = new IncrementThread(“Thread 1”);

Thread thread2 = new IncrementThread(“Thread 2”);

Thread1.start();

Thread2.start();

Try {

Thread1.join();

Thread2.join();

} catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println(“Final Counter: “ + counter);

}

Private static synchronized void incrementCounter() {

Counter++;

}

Static class IncrementThread extends Thread {

Public IncrementThread(String name) {

Super(name);

}

@Override

Public void run() {

For (int i= 0; i < 1000; i++) {

incrementCounter();

}

System.out.println(Thread.currentThread().getName() + “ completed.”);

}

}

}

**Code-3:**

Abstract class Shape {

Abstract void draw(); // Abstract method

Void display() {

System.out.println(“Displaying Shape”);

}

}

Class Circle extends Shape {

@Override

Void draw() {

System.out.println(“Drawing Circle”);

}

}

Class Rectangle extends Shape {

@Override

Void draw() {

System.out.println(“Drawing Rectangle”);

}

}

// Interface example

Interface ShapeInterface {

Void draw();

}

Class CircleInterface implements ShapeInterface {

@Override

Public void draw() {

System.out.println(“Drawing Circle (Interface)”);

}

}

Class RectangleInterface implements ShapeInterface {

@Override

Public void draw() {

System.out.println(“Drawing Rectangle (Interface)”);

}

}

Public class AbstractionDemo {

Public static void main(String[] args) {

// Abstract class example

Shape circle = new Circle();

Shape rectangle = new Rectangle();

Circle.draw();

Circle.display();

Rectangle.draw();

Rectangle.display();

// Interface example

ShapeInterface circleInterface = new CircleInterface();

ShapeInterface rectangleInterface = new RectangleInterface();

circleInterface.draw();

rectangleInterface.draw();

}

}