**A)**

**Develop an application with Employee class with Emp\_name, Emp\_id, Address, Mail\_id,Mobile no as members. Inherit the classes, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.**

Program:

package simple;

import java.util.Scanner;

class Employee {

String empName, empId, address, mailId;

long mobileNo;

double basicPay;

void getEmployeeDetails() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Employee Name: ");

empName = scanner.nextLine();

System.out.print("Enter Employee ID: ");

empId = scanner.nextLine();

System.out.print("Enter Address: ");

address = scanner.nextLine();

System.out.print("Enter Mail ID: ");

mailId = scanner.nextLine();

System.out.print("Enter Mobile No: ");

mobileNo = scanner.nextLong();

System.out.print("Enter Basic Pay: ");

basicPay = scanner.nextDouble();

}

void generatePaySlip(String designation) {

double da = 0.97 \* basicPay;

double hra = 0.1 \* basicPay;

double pf = 0.12 \* basicPay;

double staffClubFund = 0.001 \* basicPay;

double grossSalary = basicPay + da + hra;

double netSalary = grossSalary - pf - staffClubFund;

System.out.println("\nPay Slip for " + designation);

System.out.println("Employee Name: " + empName);

System.out.println("Employee ID: " + empId);

System.out.printf("Basic Pay: %.2f\nDA: %.2f\nHRA: %.2f\nPF: %.2f\nStaff Club Fund: %.2f\n", basicPay, da, hra, pf, staffClubFund);

System.out.printf("Gross Salary: %.2f\nNet Salary: %.2f\n", grossSalary, netSalary);

}

}

public class EmployeeMain {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Employee employee = new Employee();

System.out.println("Choose Employee Type:\n1. Assistant Professor\n2. Associate Professor\n3. Professor");

int choice = scanner.nextInt();

scanner.nextLine(); // consume the newline character

employee.getEmployeeDetails();

switch (choice) {

case 1: employee.generatePaySlip("Assistant Professor"); break;

case 2: employee.generatePaySlip("Associate Professor"); break;

case 3: employee.generatePaySlip("Professor"); break;

default: System.out.println("Invalid Choice!"); break;

}

}

}

**B)**

**Create an interface for ‘Playable’ with classes Football, Volleyball and Basketball.**

Program:

interface Playable {

void play();

}

class Football implements Playable {

@Override

public void play() {

System.out.println("Playing football: A team sport played with a spherical ball.");

}

}

class Volleyball implements Playable {

@Override

public void play() {

System.out.println("Playing volleyball: A team sport in which two teams are separated by a net.");

}

}

class Basketball implements Playable {

@Override

public void play() {

System.out.println("Playing basketball: A game played by two teams of five players each on a rectangular court.");

}

}

public class SportsMain {

public static void main(String[] args) {

Playable football = new Football();

Playable volleyball = new Volleyball();

Playable basketball = new Basketball();

football.play();

volleyball.play();

basketball.play();

}

}

C**)**

**Develop a Java program to create a ‘Geometry’ package with relevant classes (For e.g.**

**Circle/ Square/ Polygon and so on.) and export it.**

Program:

package geometry;

public class Circle {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

public double getArea() {

return Math.PI \* radius \* radius;

}

}

package geometry;

public class Square {

private double side;

public Square(double side) {

this.side = side;

}

public double getArea() {

return side \* side;

}

}

package geometry;

public class Polygon {

private int numberOfSides;

private double sideLength;

public Polygon(int numberOfSides, double sideLength) {

this.numberOfSides = numberOfSides;

this.sideLength = sideLength;

}

public double getArea() {

// Formula for regular polygon area: (n \* s^2) / (4 \* tan(π/n))

return (numberOfSides \* sideLength \* sideLength) / (4 \* Math.tan(Math.PI / numberOfSides));

}

}

.

import geometry.Circle;

import geometry.Square;

import geometry.Polygon;

public class Main {

public static void main(String[] args) {

Circle circle = new Circle(5.0);

Square square = new Square(4.0);

Polygon polygon = new Polygon(5, 6.0);

System.out.println("Circle Area: " + circle.getArea());

System.out.println("Square Area: " + square.getArea());

System.out.println("Polygon Area: " + polygon.getArea());

}

}

D**)**

**Implement an arithmetic calculator using JAVA. The program should get input from User**

Program:

import java.util.Scanner;

public class SimpleCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Get user input

System.out.print("Enter first number: ");

double num1 = scanner.nextDouble();

System.out.print("Enter second number: ");

double num2 = scanner.nextDouble();

System.out.print("Enter an operator (+, -, \*, /): ");

char operator = scanner.next().charAt(0);

double result = 0;

if (operator == '+') {

result = num1 + num2;

} else if (operator == '-') {

result = num1 - num2;

} else if (operator == '\*') {

result = num1 \* num2;

} else if (operator == '/') {

if (num2 != 0) {

result = num1 / num2;

} else {

System.out.println("Error: Division by zero is not allowed.");

scanner.close();

return;

}

} else {

System.out.println("Invalid operator.");

scanner.close();

return;

}

System.out.println("Result: " + result);

scanner.close();

}

}

**E)**

**Implement an employee package in Java**

**containing classes such as Employee, Manager,**

**and Engineer. Each class should store relevant details (e.g., name, ID, and department),**

**and include methods to calculate and display salary and position-specific allowances.**

**Finally, create a main program to demonstrate the functionality of each class in the**

**employee package."**

Program:

import java.util.Scanner;

class Employee {

protected int id;

protected String name;

protected int age;

protected double basicSalary;

public void getData() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Employee ID: ");

id = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter Employee Name: ");

name = scanner.nextLine();

System.out.print("Enter Employee Age: ");

age = scanner.nextInt();

System.out.print("Enter Basic Salary: ");

basicSalary = scanner.nextDouble();

}

public void displayData() {

System.out.println("\n--- Employee Details ---");

System.out.println("ID: " + id);

System.out.println("Name: " + name);

System.out.println("Age: " + age);

System.out.println("Basic Salary: $" + basicSalary);

}

public double calculateSalary() {

return basicSalary;

}

}

class Programmer extends Employee {

@Override

public void getData() {

super.getData();

}

@Override

public void displayData() {

super.displayData();

}

@Override

public double calculateSalary() {

return basicSalary;

}

}

class AssistantProfessor extends Employee {

@Override

public void getData() {

super.getData();

}

@Override

public void displayData() {

super.displayData();

}

@Override

public double calculateSalary() {

return basicSalary;

}

}

class Professor extends Employee {

@Override

public void getData() {

super.getData();

}

@Override

public void displayData() {

super.displayData();

}

@Override

public double calculateSalary() {

return basicSalary;

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int choice;

System.out.println("=== Employee Management System ===");

System.out.println("Select Employee Type:");

System.out.println("1. Programmer");

System.out.println("2. Assistant Professor");

System.out.println("3. Professor");

System.out.print("Enter your choice (1-3): ");

choice = scanner.nextInt();

scanner.nextLine();

Employee emp = null;

switch(choice) {

case 1:

emp = new Programmer();

break;

case 2:

emp = new AssistantProfessor();

break;

case 3:

emp = new Professor();

break;

default:

System.out.println("Invalid choice!");

System.exit(0);

}

emp.getData();

emp.displayData();

double salary = emp.calculateSalary();

System.out.println("Total Salary: $" + salary);

scanner.close();

}

}

**F)**

**Create an interface Shape that contains two methods: calculateArea() and display().Create two classes Circle and Rectangle that implement the Shape interface.Write a**

**MainApp class that demonstrates the use of these two classes. The program should**

**calculate and display the area of a circle and a rectangle.**

Program:

// Shape.java

interface Shape {

double calculateArea();

void display();

}

// Circle class implementing Shape interface

class Circle implements Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override

public double calculateArea() {

return Math.PI \* radius \* radius;

}

@Override

public void display() {

System.out.println("Circle with radius: " + radius);

System.out.println("Area: " + calculateArea());

}

}

// Rectangle class implementing Shape interface

class Rectangle implements Shape {

private double length;

private double width;

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

@Override

public double calculateArea() {

return length \* width;

}

@Override

public void display() {

System.out.println("Rectangle with length: " + length + " and width: " + width);

System.out.println("Area: " + calculateArea());

}

}

// MainApp class to demonstrate the functionality

public class MainApp {

public static void main(String[] args) {

Shape circle = new Circle(5.0);

Shape rectangle = new Rectangle(4.0, 6.0);

System.out.println("=== Shape Area Calculation ===");

circle.display();

System.out.println();

rectangle.display();

}

}

**G)**

**Write a Java program to demonstrate thread synchronization.**

Program:

class Table {

void printTable() { // Removed synchronized

for (int i = 1; i <= 5; i++) {

System.out.println("Value: " + i);

try {

// Sleep for a while to simulate a time-consuming task

Thread.sleep(100);

} catch (InterruptedException e) {

System.out.println(e);

}

}

System.out.println();

}

}

class Thread1 extends Thread {

Table table;

Thread1(Table table) {

this.table = table;

}

public void run() {

table.printTable(); // Print simple table of values

}

}

class Thread2 extends Thread {

Table table;

Thread2(Table table) {

this.table = table;

}

public void run() {

table.printTable(); // Print simple table of values

}

}

public class ThreadSynchronizationExample {

public static void main(String[] args) {

Table table = new Table(); // Create a single Table object

Thread1 thread1 = new Thread1(table);

Thread2 thread2 = new Thread2(table);

thread1.start(); // Start thread1

thread2.start(); // Start thread2

try {

thread1.join(); // Wait for thread1 to finish

thread2.join(); // Wait for thread2 to finish

} catch (InterruptedException e) {

System.out.println(e);

}

System.out.println("tables printed.");

}

}

**H)**

**Implement JAVA package for Library and User. Create suitable classes and methods**

Program:

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

class Book {

String title;

boolean isBorrowed;

Book(String title) {

this.title = title;

this.isBorrowed = false;

}

@Override

public String toString() {

return title + (isBorrowed ? " (Borrowed)" : "");

}

}

class Library {

List<Book> books;

Library() {

books = new ArrayList<>();

}

void addBook(String title) {

books.add(new Book(title));

}

void displayBooks() {

System.out.println("\nAvailable Books:");

for (Book book : books) {

System.out.println(book);

}

}

boolean borrowBook(String title) {

for (Book book : books) {

if (book.title.equalsIgnoreCase(title) && !book.isBorrowed) {

book.isBorrowed = true;

return true;

}

}

return false;

}

}

public class LibraryManagementSystem {

public static void main(String[] args) {

Library library = new Library();

library.addBook("1984");

library.addBook("To Kill a Mockingbird");

library.addBook("The Great Gatsby");

Scanner scanner = new Scanner(System.in);

System.out.println("=== Library Management System ===");

library.displayBooks();

System.out.print("\nEnter the title of the book to borrow: ");

String borrowTitle = scanner.nextLine();

if (library.borrowBook(borrowTitle)) {

System.out.println("You've borrowed: " + borrowTitle);

} else {

System.out.println("Sorry, the book is not available or already borrowed.");

}

scanner.close();

}

}

**I)**

**Develop a online course registration application using JAVA Fx.**

Program:

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

public class SimpleCourseRegistrationApp extends Application {

@Override

public void start(Stage primaryStage) {

primaryStage.setTitle("Course Registration");

// Create a TextField for name input

TextField nameField = new TextField();

nameField.setPromptText("Enter your name");

// Create a Button for registration

Button registerButton = new Button("Register");

// Create a Label for output

Label outputLabel = new Label();

// Set button action

registerButton.setOnAction(e -> {

String name = nameField.getText();

if (name.isEmpty()) {

outputLabel.setText("Please enter your name.");

} else {

outputLabel.setText("Registered: " + name);

nameField.clear(); // Clear the input field

}

});

// Layout

VBox layout = new VBox(10);

layout.getChildren().addAll(nameField, registerButton, outputLabel);

// Create scene and set it on the stage

Scene scene = new Scene(layout, 300, 150);

primaryStage.setScene(scene);

primaryStage.show();

}

public static void main(String[] args) {

launch(args);

}

}

**J)**

**Create a JAVA program to demonstrate single inheritance by implementing a base class Person with attributes such as name and age. Derive a class Studentfrom Person, adding attributes specific to Student (e.g., studentID and major).Include methods to display the details of a student. Write a main program to create an instance of Student and display its details to demonstrate single inheritance.**

Program:

// Base class

class Person {

String name; // Person's name

int age; // Person's age

// Constructor to initialize Person

Person(String name, int age) {

this.name = name;

this.age = age;

}

}

// Derived class

class Student extends Person {

String studentID; // Student's ID

// Constructor to initialize Student

Student(String name, int age, String studentID) {

super(name, age); // Call to the base class constructor

this.studentID = studentID;

}

// Method to display Student details

void displayDetails() {

System.out.println("Name: " + name);

System.out.println("Age: " + age);

System.out.println("Student ID: " + studentID);

}

}

// Main program

public class InheritanceDemo {

public static void main(String[] args) {

// Create a Student object

Student student = new Student("Alice", 20, "S12345");

// Display the student's details

student.displayDetails();

}

}

**K)**

**Develop a JAVA program to implement inheritance using “super” keyword . Create a base class “Person” and derived class “Employee” with necessary attributes.**

Program:

// Base class

class Person {

protected String name; // Person's name

// Constructor for Person

public Person(String name) {

this.name = name; // Initialize name

}

// Method to display person's name

public void display() {

System.out.println("Name: " + name);

}

}

// Derived class

class Employee extends Person {

private int employeeId; // Employee ID

// Constructor for Employee

public Employee(String name, int employeeId) {

super(name); // Call the constructor of Person

this.employeeId = employeeId; // Initialize employee ID

}

// Method to display employee details

public void display() {

super.display(); // Call the display method from Person

System.out.println("Employee ID: " + employeeId);

}

}

// Main class to run the program

public class InheritanceExample {

public static void main(String[] args) {

// Create an Employee object

Employee emp = new Employee("John Doe", 101);

// Display employee details

emp.display();

}

}

**L)**

**Write a Java program to demonstrate thread synchronization for printing any two**

**multiplication table .**

Program:

class MultiplicationTable {

// Synchronized method to print the multiplication table

public synchronized void printTable(int number) {

for (int i = 1; i <= 10; i++) {

System.out.println(number + " \* " + i + " = " + (number \* i));

}

System.out.println(); // Blank line between tables

}

}

// Thread class to print a specific multiplication table

class TableThread extends Thread {

private final MultiplicationTable table;

private final int number;

public TableThread(MultiplicationTable table, int number) {

this.table = table;

this.number = number;

}

public void run() {

table.printTable(number);

}

}

// Main class

public class Main {

public static void main(String[] args) {

MultiplicationTable multiplicationTable = new MultiplicationTable();

// Create threads for multiplication tables of 2 and 3

Thread thread1 = new TableThread(multiplicationTable, 2);

Thread thread2 = new TableThread(multiplicationTable, 3);

// Start the threads

thread1.start();

thread2.start();

// Wait for threads to finish

try {

thread1.join();

thread2.join();

} catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println("Multiplication tables printed.");

}

}

**M)**

**Develop an application with Employee class with Emp\_name, Emp\_id, Address, Mail\_id,Mobile no as members. Inherit the classProfessor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.**

Program:

Program:

package simple;

import java.util.Scanner;

class Employee {

String empName, empId, address, mailId;

long mobileNo;

double basicPay;

void getEmployeeDetails() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Employee Name: ");

empName = scanner.nextLine();

System.out.print("Enter Employee ID: ");

empId = scanner.nextLine();

System.out.print("Enter Address: ");

address = scanner.nextLine();

System.out.print("Enter Mail ID: ");

mailId = scanner.nextLine();

System.out.print("Enter Mobile No: ");

mobileNo = scanner.nextLong();

System.out.print("Enter Basic Pay: ");

basicPay = scanner.nextDouble();

}

void generatePaySlip(String designation) {

double da = 0.97 \* basicPay;

double hra = 0.1 \* basicPay;

double pf = 0.12 \* basicPay;

double staffClubFund = 0.001 \* basicPay;

double grossSalary = basicPay + da + hra;

double netSalary = grossSalary - pf - staffClubFund;

System.out.println("\nPay Slip for " + designation);

System.out.println("Employee Name: " + empName);

System.out.println("Employee ID: " + empId);

System.out.printf("Basic Pay: %.2f\nDA: %.2f\nHRA: %.2f\nPF: %.2f\nStaff Club Fund: %.2f\n", basicPay, da, hra, pf, staffClubFund);

System.out.printf("Gross Salary: %.2f\nNet Salary: %.2f\n", grossSalary, netSalary);

}

}

public class EmployeeMain {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Employee employee = new Employee();

System.out.println("Choose Employee Type:\n1. Assistant Professor\n2. Associate Professor\n3. Professor");

int choice = scanner.nextInt();

scanner.nextLine(); // consume the newline character

employee.getEmployeeDetails();

switch (choice) {

case 1: employee.generatePaySlip("Assistant Professor"); break;

case 2: employee.generatePaySlip("Associate Professor"); break;

case 3: employee.generatePaySlip("Professor"); break;

default: System.out.println("Invalid Choice!"); break;

}

}

}

**N)**

**Develop an application for Train Ticket booking using JAVA Fx.**

Program:

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

public class SimpleTrainBooking extends Application {

@Override

public void start(Stage primaryStage) {

primaryStage.setTitle("Train Booking");

// Input field for name

TextField nameInput = new TextField();

nameInput.setPromptText("Enter your name");

// Button to book ticket

Button bookButton = new Button("Book Ticket");

Label messageLabel = new Label(); // Label to show messages

// Action for button click

bookButton.setOnAction(e -> {

String name = nameInput.getText(); // Get text from input

if (name.isEmpty()) {

messageLabel.setText("Please enter your name."); // Show error message

} else {

messageLabel.setText("Ticket booked for " + name + "!"); // Show success message

}

});

// Layout

VBox layout = new VBox(10);

layout.getChildren().addAll(nameInput, bookButton, messageLabel);

// Scene setup

Scene scene = new Scene(layout, 300, 150);

primaryStage.setScene(scene);

primaryStage.show(); // Show the window

}

public static void main(String[] args) {

launch(args); // Launch the application

}

}

**O)**

**Develop a JAVA application with University class . Inherit classes for department and**

**student with required details.**

Program:

// Base class

class University {

String name;

public University(String name) {

this.name = name;

}

public void display() {

System.out.println("University Name: " + name);

}

}

// Derived class for Department

class Department extends University {

String departmentName;

public Department(String name, String departmentName) {

super(name);

this.departmentName = departmentName;

}

public void display() {

super.display();

System.out.println("Department Name: " + departmentName);

}

}

// Derived class for Student

class Student extends Department {

String studentName;

public Student(String name, String departmentName, String studentName) {

super(name, departmentName);

this.studentName = studentName;

}

public void display() {

super.display();

System.out.println("Student Name: " + studentName);

}

}

// Main class to demonstrate functionality

public class UniversityApp {

public static void main(String[] args) {

// Create an instance of Student

Student student = new Student("ABC University", "Computer Science", "John Doe");

// Display details

student.display();

}

}

**P)**

**Develop a Java program to print and return the current class instance using ‘this’keyword.**

Program:

package geometry;

public class Rectangle {

private double length;

private double width;

// Constructor

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

// Method to calculate area

public double area() {

return length \* width;

}

// Method to calculate perimeter

public double perimeter() {

return 2 \* (length + width);

}

// Method to display dimensions

public void display() {

System.out.println("Length: " + length);

System.out.println("Width: " + width);

System.out.println("Area: " + area());

System.out.println("Perimeter: " + perimeter());

}

}

import geometry.\*;

public class Main {

public static void main(String[] args) {

// Create a rectangle object

Rectangle rect = new Rectangle(5.0, 3.0);

// Display the rectangle's details

rect.display();

}

}

**Q)**

**Create an interface for ‘Animal ’ with**

**classes Lion, Tiger and Leopard. Create**

**suitable methods .**

Program:

interface Animal {

void sound();

void habitat();

}

class Lion implements Animal {

public void sound() {

System.out.println("Lion roars");

}

public void habitat() {

System.out.println("Lions live in savannas and grasslands.");

}

}

class Tiger implements Animal {

public void sound() {

System.out.println("Tiger growls");

}

public void habitat() {

System.out.println("Tigers live in forests and grasslands.");

}

}

class Leopard implements Animal {

public void sound() {

System.out.println("Leopard snarls");

}

public void habitat() {

System.out.println("Leopards live in forests, mountains, and grasslands.");

}

}

public class Main {

public static void main(String[] args) {

Animal lion = new Lion();

Animal tiger = new Tiger();

Animal leopard = new Leopard();

lion.sound();

lion.habitat();

tiger.sound();

tiger.habitat();

leopard.sound();

leopard.habitat();

}

}

Output:

Lion roars

Lions live in savannas and grasslands.

Tiger growls

Tigers live in forests and grasslands.

Leopard snarls

Leopards live in forests, mountains, and grasslands.

**R)**

**Implement a Java code to print the sum, multiply, subtract, divide and remainder**

**of two numbers. Get the input from the user.**

Program:

import java.util.Scanner;

public class SimpleArithmetic {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int a = sc.nextInt(), b = sc.nextInt();

System.out.println("Sum: " + (a + b));

System.out.println("Difference: " + (a - b));

System.out.println("Product: " + (a \* b));

System.out.println("Quotient: " + (a / b));

System.out.println("Remainder: " + (a % b));

sc.close();

}

}

**S)**

**Develop a Java program to print and return the current class instance using this**

**keyword.**

Program:

package geometry;

public class Rectangle {

private double length;

private double width;

// Constructor

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

// Method to calculate area

public double area() {

return length \* width;

}

// Method to calculate perimeter

public double perimeter() {

return 2 \* (length + width);

}

// Method to display dimensions

public void display() {

System.out.println("Length: " + length);

System.out.println("Width: " + width);

System.out.println("Area: " + area());

System.out.println("Perimeter: " + perimeter());

}

}

MAIN:

import geometry.\*;

public class Main {

public static void main(String[] args) {

// Create a rectangle object

Rectangle rect = new Rectangle(5.0, 3.0);

// Display the rectangle's details

rect.display();

}

}

**T)**

**Write a Java program to demonstrate thread synchronization for printing any**

**two addition table.**

Program:

class AdditionTable {

// Synchronized method to print addition table

public synchronized void printTable(int number) {

System.out.println("Addition Table for " + number + ":");

for (int i = 1; i <= 10; i++) {

System.out.println(number + " + " + i + " = " + (number + i));

}

System.out.println();

}

}

// Thread class to handle table printing

class TableThread extends Thread {

private AdditionTable additionTable;

private int number;

public TableThread(AdditionTable additionTable, int number) {

this.additionTable = additionTable;

this.number = number;

}

public void run() {

additionTable.printTable(number);

}

}

// Main class

public class Main {

public static void main(String[] args) {

AdditionTable additionTable = new AdditionTable();

// Create and start threads for two addition tables

TableThread thread1 = new TableThread(additionTable, 5);

TableThread thread2 = new TableThread(additionTable, 7);

thread1.start();

thread2.start();

// Wait for both threads to finish

try {

thread1.join();

thread2.join();

} catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println("Addition tables printed.");

}

}