**jfepWeek 6 – Tutorial Assignment**

**Object-Oriented Programming Paradigm**

1. Write a Java program to create a class called "Person" with a name and age attribute. Create two instances of the "Person" class, set their attributes using the constructor, and print their name and age.

import java.util.Scanner;

class Person{

    private String name;

    private int age;

    public Person(String name,int age) //Constructor

{

        this.name=name;

        this.age=age;

    }

    //Get methods

    public String getName(){

        return name;

    }

    public int getAge(){

        return age;

    }

}

public class Print{

    public static void main(String[]args){

        System.out.println("Trishita Yadav\nRA2211003011325");

        Scanner sc=new Scanner(System.in);

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

        String name=sc.nextLine();

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

        int age=sc.nextInt();

        sc.nextLine(); //Consume the newline

        //Create instances of the Person class using input values

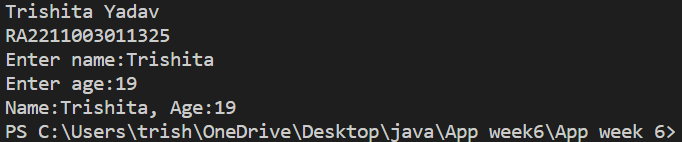
        Person person=new Person(name, age);

        System.out.println("Name:"+person.getName()+", Age:"+person.getAge()); //display name and age

        sc.close();

    }

}



2. Write a Java program to create class called "TrafficLight" with attributes for color and duration, and methods to change the color and check for red or green.

class TrafficLight {

    // Attributes

    private String color; // Stores the current color of the traffic light

    private int duration; // Stores the duration (in seconds) of the current color

    // Constructor

    public TrafficLight(String initialColor, int initialDuration) {

        this.color = initialColor;

        this.duration = initialDuration;

    }

    // Method to change the color of the traffic light

    public void changeColor(String newColor, int newDuration) {

        this.color = newColor;

        this.duration = newDuration;

    }

    // Method to check if the traffic light is currently red

    public boolean isRed() {

        return color.equals("Red");

    }

    // Method to check if the traffic light is currently green

    public boolean isGreen() {

        return color.equals("Green");

    }

    // Method to get the current color of the traffic light

    public String getColor() {

        return color;

    }

    // Method to get the current duration of the color

    public int getDuration() {

        return duration;

    }

    public static void main(String[] args) {

        System.out.println("Trishita Yadav\nRA2211003011325");

        // Create a TrafficLight object with an initial color and duration

        TrafficLight trafficLight = new TrafficLight("Red", 30);

        // Display the current color and duration

        System.out.println("Current color: " + trafficLight.getColor());

        System.out.println("Current duration: " + trafficLight.getDuration() + " seconds");

        // Change the color and duration

        trafficLight.changeColor("Green", 45);

        // Display the updated color and duration

        System.out.println("Updated color: " + trafficLight.getColor());

        System.out.println("Updated duration: " + trafficLight.getDuration() + " seconds");

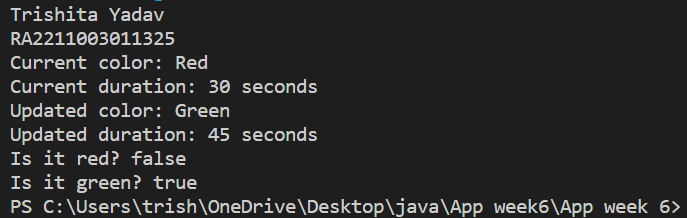
        // Check if the traffic light is currently red or green

        System.out.println("Is it red? " + trafficLight.isRed());

        System.out.println("Is it green? " + trafficLight.isGreen());

    }

}



3. Write a Java program to perform arithmetic operations using method overloading.

import java.util.Scanner;

public class ArithmeticOperations {

    // Method to add two numbers

    public int add(int a, int b) {

        return a + b;

    }

    // Method to add three numbers

    public int add(int a, int b, int c) {

        return a + b + c;

    }

    // Method to add two numbers (double)

    public double add(double a, double b) {

        return a + b;

    }

    // Method to multiply two numbers

    public int multiply(int a, int b) {

        return a \* b;

    }

    // Method to multiply three numbers

    public int multiply(int a, int b, int c) {

        return a \* b \* c;

    }

    // Method to multiply two numbers (double)

    public double multiply(double a, double b) {

        return a \* b;

    }

    public static void main(String[] args) {

        System.out.println("Trishita Yadav\nRA2211003011325");

        Scanner scanner = new Scanner(System.in);

        ArithmeticOperations calculator = new ArithmeticOperations();

        System.out.print("Enter two integers for addition:");

        int num1 = scanner.nextInt();

        int num2 = scanner.nextInt();

        System.out.print("Enter three integers for addition:");

        int num3 = scanner.nextInt();

        int num4 = scanner.nextInt();

        int num5 = scanner.nextInt();

        System.out.print("Enter two doubles for addition:");

        double num6 = scanner.nextDouble();

        double num7 = scanner.nextDouble();

        System.out.print("Enter two integers for multiplication:");

        int num8 = scanner.nextInt();

        int num9 = scanner.nextInt();

        System.out.print("Enter three integers for multiplication:");

        int num10 = scanner.nextInt();

        int num11 = scanner.nextInt();

        int num12 = scanner.nextInt();

        System.out.print("Enter two doubles for multiplication:");

        double num13 = scanner.nextDouble();

        double num14 = scanner.nextDouble();

        System.out.println("\nResults:");

        System.out.println("Addition of integers:" + calculator.add(num1, num2));

        System.out.println("Addition of three integers:" + calculator.add(num3, num4, num5));

        System.out.println("Addition of doubles:" + calculator.add(num6, num7));

        System.out.println("Multiplication of integers:" + calculator.multiply(num8, num9));

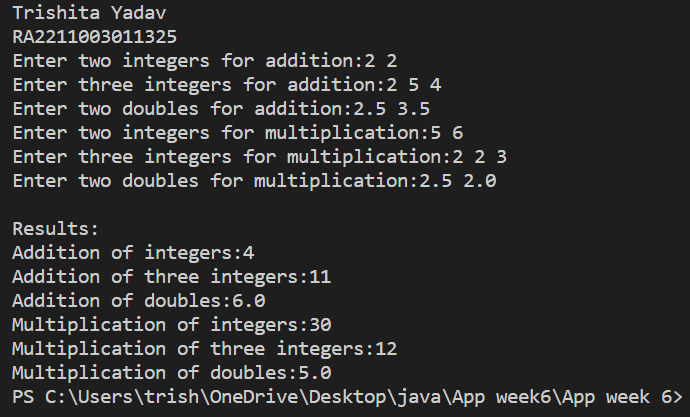
        System.out.println("Multiplication of three integers:" + calculator.multiply(num10, num11, num12));

        System.out.println("Multiplication of doubles:" + calculator.multiply(num13, num14));

        scanner.close();

    }

}



4. Write a Java program to create a class called Employee with methods called work() and getSalary(). Create a subclass called HRManager that overrides the work() method and adds a new method called addEmployee().

class Employee {

    private double salary;

    public Employee(double salary) {

        this.salary = salary;

    }

    public void work() {

        System.out.println("Employee is working");

    }

    public double getSalary() {

        return salary;

    }

}

class HRManager extends Employee {

    public HRManager(double salary) {

        super(salary);

    }

    @Override

    public void work() {

        System.out.println("HR Manager is managing HR tasks");

    }

    public void addEmployee() {

        System.out.println("HR Manager is adding a new employee");

    }

}

public class HR{

    public static void main(String[] args) {

        System.out.println("Trishita Yadav\nRA2211003011325");

        // Create an Employee instance

        Employee employee = new Employee(50000.0);

        System.out.println("Employee Salary: " + employee.getSalary());

        employee.work(); // Call the work() method of Employee class

        System.out.println();

        // Create an HRManager instance

        HRManager hrManager = new HRManager(80000.0);

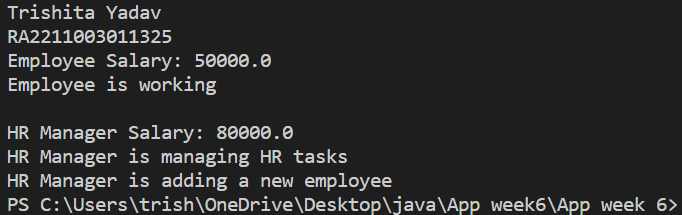
        System.out.println("HR Manager Salary: " + hrManager.getSalary());

        hrManager.work(); //Call the overridden work() method of HRManager class

        hrManager.addEmployee(); //Call the specific method of HRManager class

    }

}



5. Write a Java program to create a class called Shape with methods called getPerimeter() and getArea(). Create a subclass called Circle that overrides the getPerimeter() and getArea() methods to calculate the area and perimeter of a circle.

import java.util.Scanner;

class Shape{

    // Placeholder methods

    public double getPerimeter() {

        return 0.0;

    }

    public double getArea() {

        return 0.0;

    }

}

class Circle extends Shape {

    private double radius;

    // Constructor

    public Circle(double radius) {

        this.radius = radius;

    }

    // Override getPerimeter() for circle

    @Override

    public double getPerimeter() {

        return 2 \* Math.PI \* radius;

    }

    // Override getArea() for circle

    @Override

    public double getArea() {

        return Math.PI \* radius \* radius;

    }

}

public class Calculate{

    public static void main(String[] args) {

        System.out.println("Trishita Yadav\nRA2211003011325");

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the radius of the circle:");

        double radius = scanner.nextDouble();

        Circle circle = new Circle(radius);

        double perimeter = circle.getPerimeter();

        double area = circle.getArea();

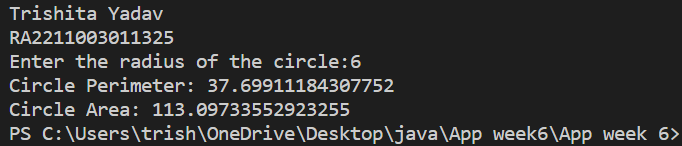
        System.out.println("Circle Perimeter: " + perimeter);

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

        scanner.close();

    }

}



6. Write a Java program to create an interface Sortable with a method sort() that sorts an array of integers in ascending order. Create two classes BubbleSort and SelectionSort that implement the Sortable interface and provide their own implementations of the sort() method.

interface Sortable {

    void sort(int[] arr);

}

class BubbleSort implements Sortable {

    @Override

    public void sort(int[] arr) {

        int n = arr.length;

        for (int i = 0; i < n - 1; i++) {

            for (int j = 0; j < n - i - 1; j++) {

                if (arr[j] > arr[j + 1]) {

                    // Swap arr[j] and arr[j+1]

                    int temp = arr[j];

                    arr[j] = arr[j + 1];

                    arr[j + 1] = temp;

                }

            }

        }

    }

}

class SelectionSort implements Sortable {

    @Override

    public void sort(int[] arr) {

        int n = arr.length;

        for (int i = 0; i < n - 1; i++) {

            int minIndex = i;

            for (int j = i + 1; j < n; j++) {

                if (arr[j] < arr[minIndex]) {

                    minIndex = j;

                }

            }

            // Swap arr[i] and arr[minIndex]

            int temp = arr[i];

            arr[i] = arr[minIndex];

            arr[minIndex] = temp;

        }

    }

}

public class SortImplement {

    public static void main(String[] args) {

        System.out.println("Trishita Yadav\nRA2211003011325");

        int[] arr = { 64, 34, 25, 12, 22, 11, 90 };

        Sortable bubbleSort = new BubbleSort();

        bubbleSort.sort(arr);

        System.out.println("Sorted using Bubble Sort:");

        printArray(arr);

        int[] arr2 = { 64, 25, 12, 22, 11 };

        Sortable selectionSort = new SelectionSort();

        selectionSort.sort(arr2);

        System.out.println("\nSorted using Selection Sort:");

        printArray(arr2);

    }

    static void printArray(int[] arr) {

        for (int value : arr) {

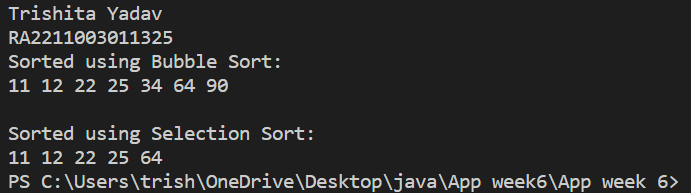
            System.out.print(value + " ");

        }

        System.out.println();

    }

}



7. Write a Java program to create an interface Resizable with methods resizeWidth(int width) and resizeHeight(int height) that allow an object to be resized. Create a class Rectangle that implements the Resizable interface and implements the resize methods.

import java.util.Scanner;

// Define an interface Resizable with methods for resizing width and height

interface Resizable {

    void resizeWidth(int width);

    void resizeHeight(int height);

}

// Create a Rectangle class that implements the Resizable interface

class Rectangle implements Resizable {

    private int width;

    private int height;

    // Constructor to initialize the initial dimensions

    public Rectangle(int width, int height) {

        this.width = width;

        this.height = height;

    }

    // Implementation of the resizeWidth method

    @Override

    public void resizeWidth(int newWidth) {

        this.width = newWidth;

    }

    // Implementation of the resizeHeight method

    @Override

    public void resizeHeight(int newHeight) {

        this.height = newHeight;

    }

    // Method to display the current dimensions of the rectangle

    public void displayDimensions() {

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

        System.out.println("Height:" + height);

    }

}

public class Resize{

    public static void main(String[] args) {

        System.out.println("Trishita Yadav\nRA2211003011325");

        Scanner scanner = new Scanner(System.in);

        // Get initial dimensions from the user

        System.out.print("Enter initial width of the rectangle:");

        int initialWidth = scanner.nextInt();

        System.out.print("Enter initial height of the rectangle:");

        int initialHeight = scanner.nextInt();

        // Create a Rectangle object with the initial dimensions

        Rectangle rectangle = new Rectangle(initialWidth, initialHeight);

        // Display the initial dimensions

        System.out.println("Initial Rectangle Dimensions:");

        rectangle.displayDimensions();

        // Get resizing values from the user

        System.out.print("Enter new width for resizing:");

        int newWidth = scanner.nextInt();

        System.out.print("Enter new height for resizing:");

        int newHeight = scanner.nextInt();

        // Resize the rectangle

        rectangle.resizeWidth(newWidth);

        rectangle.resizeHeight(newHeight);

        // Display the resized dimensions

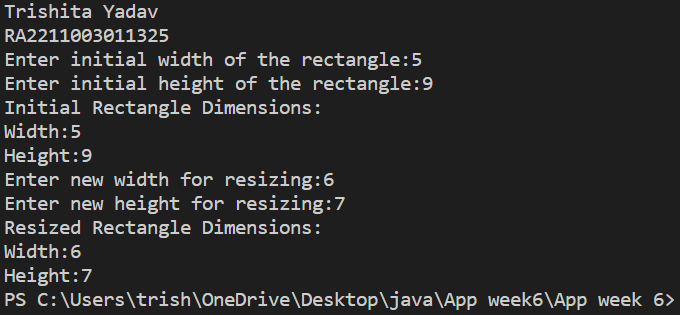
        System.out.println("Resized Rectangle Dimensions:");

        rectangle.displayDimensions();

        scanner.close();

    }

}



8. Write a Java program to create an interface Flyable with a method called fly\_obj(). Create three classes Spacecraft, Airplane, and Helicopter that implement the Flyable interface. Implement the fly\_obj() method for each of the three classes. Hint :- fly\_obj definition – prints the particular object is flying.

// Define an interface called Flyable with a method fly\_obj()

interface Flyable {

    void fly\_obj();

}

// Create a class Spacecraft that implements the Flyable interface

class Spacecraft implements Flyable {

    @Override

    public void fly\_obj() {

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

    }

}

// Create a class Airplane that implements the Flyable interface

class Airplane implements Flyable {

    @Override

    public void fly\_obj() {

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

    }

}

// Create a class Helicopter that implements the Flyable interface

class Helicopter implements Flyable {

    @Override

    public void fly\_obj() {

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

    }

}

// Main class to demonstrate flying objects

public class fly{

    public static void main(String[] args) {

        System.out.println("Trishita Yadav\nRA2211003011325");

        // Create instances of the Flyable objects

        Flyable spacecraft = new Spacecraft();

        Flyable airplane = new Airplane();

        Flyable helicopter = new Helicopter();

        // Display the flying objects

        System.out.println("Flying Objects:");

        System.out.print("1. ");

        spacecraft.fly\_obj();

        System.out.print("2. ");

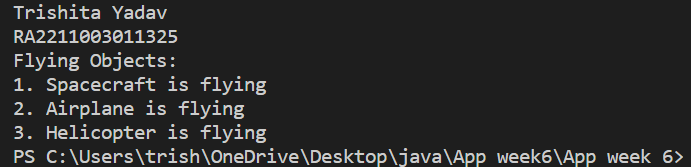
        airplane.fly\_obj();

        System.out.print("3. ");

        helicopter.fly\_obj();

    }

}



9. Write a Java program to have the arithmetic functions defined in different user-defined packages and incorporate all the packages and perform the function in a single class.

javac calculation/Calculation.java

java calculation.Calculation

package calculation;

import addition.Addition;

import subtraction.Subtraction;

public class Calculation {

public static void main(String[] args) {

int x = 10, y = 5;

int sum = Addition.add(x, y);

int difference = Subtraction.subtract(x, y);

System.out.println("Sum: " + sum);

System.out.println("Difference: " + difference);

}

}

package subtraction;

public class Subtraction {

public static int subtract(int a, int b) {

return a - b;

}

}

package addition;

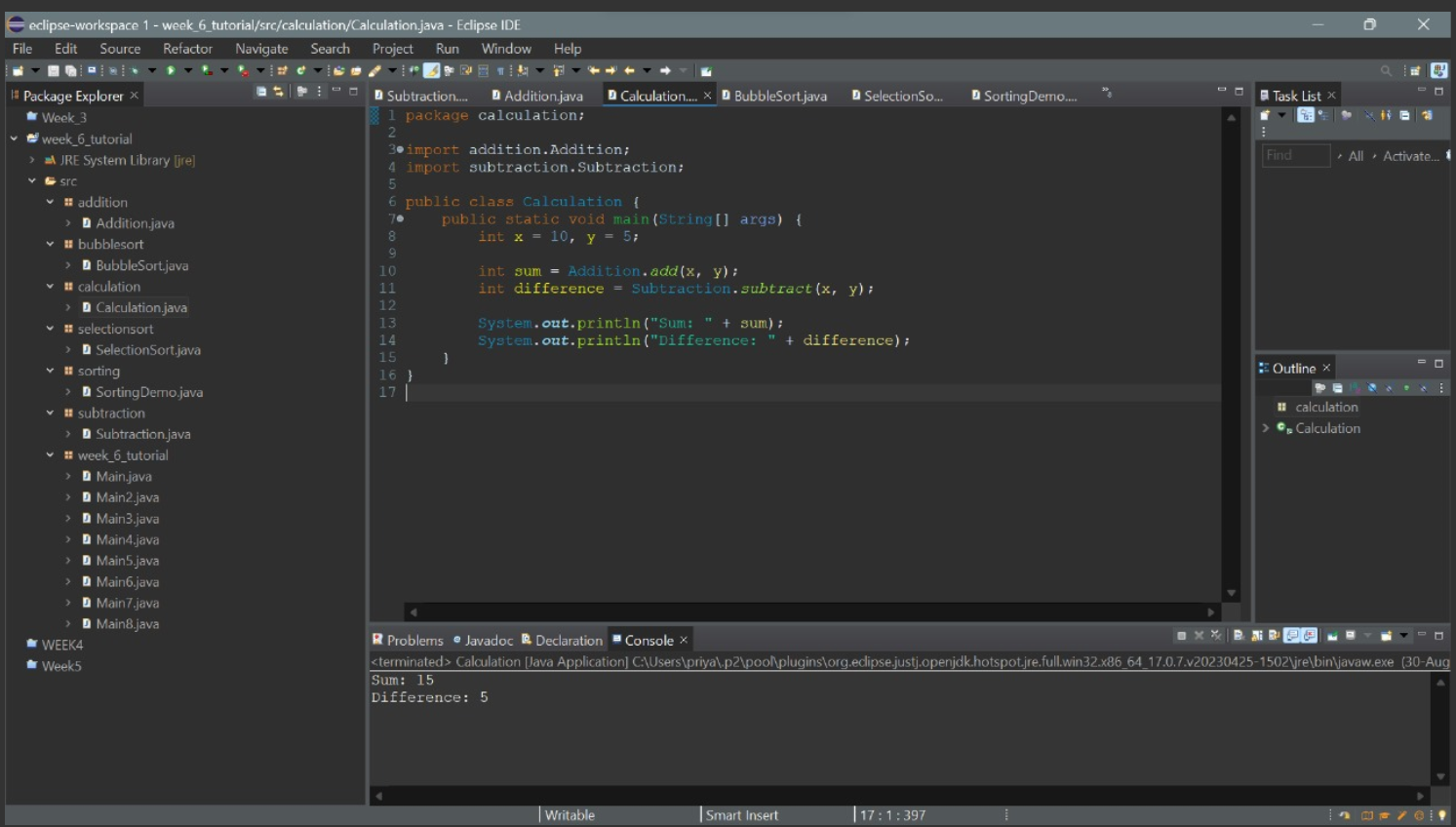
public class Addition {

public static int add(int a, int b) {

return a + b;

}

}



10. Create two different packages to compute bubblesort and selection sort. Write a Java program to implement sorting functions in a single class.

package bubblesort;

public class BubbleSort {

public static void sort(int[] arr) {

int n = arr.length;

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

// Swap arr[j] and arr[j+1]

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

}

package selectionsort;

public class SelectionSort {

public static void sort(int[] arr) {

int n = arr.length;

for (int i = 0; i < n - 1; i++) {

int minIndex = i;

for (int j = i + 1; j < n; j++) {

if (arr[j] < arr[minIndex]) {

minIndex = j;

}

}

// Swap arr[i] and arr[minIndex]

int temp = arr[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;

}

}

}

package sorting;

import bubblesort.BubbleSort;

import selectionsort.SelectionSort;

public class SortingDemo {

public static void main(String[] args) {

int[] arr1 = { 64, 34, 25, 12, 22, 11, 90 };

int[] arr2 = { 64, 25, 12, 22, 11 };

BubbleSort.sort(arr1);

SelectionSort.sort(arr2);

System.out.println("Sorted using Bubble Sort:");

printArray(arr1);

System.out.println("\nSorted using Selection Sort:");

printArray(arr2);

}

static void printArray(int[] arr) {

for (int value : arr) {

System.out.print(value + " ");

}

System.out.println();

}

}

javac sorting/SortingDemo.java

java sorting.SortingDemo

