/\*

Task 1: Data Types/Variables

Write a program that declares two integer variables, swaps their values without using a third variable, and prints the result.

\*/

import java.util.Scanner;

public class DataTypes {

public static void main(String[] args) {

Scanner sn = new Scanner(System.in);

int a = sn.nextInt();

int b = sn.nextInt();

// Display the initial values

System.out.println("Before swapping:");

System.out.println("a = " + a);

System.out.println("b = " + b);

// Swap the values without using a third variable

a = a + b;

b = a - b;

a = a - b;

// Display the swapped values

System.out.println("After swapping:");

System.out.println("a = " + a);

System.out.println("b = " + b);

sn.close();

}

}

/\*

Task 2: Operators

Create a program that simulates a simple calculator using command-line arguments to perform and print the result of addition,

subtraction, multiplication, and division..

\*/

public class Operaters {

public static void main(String[] args) {

if (args.length != 3) {

System.out.println("Usage: java SimpleCalculator <operand1> <operator> <operand2>");

System.exit(1);

}

double operand1 = Double.parseDouble(args[0]);

char operator = args[1].charAt(0);

double operand2 = Double.parseDouble(args[2]);

// Perform the calculation based on the operator

double result = 0.0;

switch (operator) {

case '+':

result = operand1 + operand2;

break;

case '-':

result = operand1 - operand2;

break;

case '\*':

result = operand1 \* operand2;

break;

case '/':

if (operand2 != 0) {

result = operand1 / operand2;

} else {

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

System.exit(1);

}

break;

default:

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

System.exit(1);

}

// Print the result

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

}

}

import java.util.Scanner;

public class ControlFlow {

/\*

Task 3: Control Flow

Write a Java program that reads an integer and prints whether it is a prime number using a for loop and if statements.

\*/

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter an integer: ");

int number = scanner.nextInt();

scanner.close();

boolean isPrime = true;

if (number < 2) {

isPrime = false;

} else {

for (int i = 2; i <= Math.sqrt(number); i++) {

if (number % i == 0) {

isPrime = false;

break;

}

}

}

// Print whether the number is prime or not

if (isPrime) {

System.out.println(number + " is a prime number.");

} else {

System.out.println(number + " is not a prime number.");

}

}

}

import java.util.Scanner;

/\*

Task 4: Constructors

Implement a Matrix class that has a constructor which initializes the dimensions of a matrix and

a method to fill the matrix with values.

\*/

public class Matrix {

private int[][] matrix;

// Constructor to initialize the dimensions of the matrix

public Matrix(int rows, int cols) {

matrix = new int[rows][cols];

}

// Method to fill the matrix with values

public void fillMatrixUsing() {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter values for the matrix:");

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[0].length; j++) {

System.out.print("Enter value at position (" + (i + 1) + ", " + (j + 1) + "): ");

matrix[i][j] = scanner.nextInt();

}

}

scanner.close();

}

// Method to print the matrix

public void printMatrix() {

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[0].length; j++) {

System.out.print(matrix[i][j] + " ");

}

System.out.println();

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter number of rows: ");

int rows = scanner.nextInt();

System.out.print("Enter number of columns: ");

int cols = scanner.nextInt();

// Create a Matrix object with user-specified dimensions

Matrix myMatrix = new Matrix(rows, cols);

myMatrix.fillMatrixUsing();

// Print the matrix

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

myMatrix.printMatrix();

scanner.close();

}

}

Task 5: Inheritance

Create a Shape class with a method area() and extend it with Circle and Rectangle classes overriding the area() method appropriately.

public class Shape {

public double area() {

return 0.0;

}

}

public class Circle extends Shape{

private double radius;

// Constructor for Circle class

public Circle(double radius) {

this.radius = radius;

}

@Override

public double area() {

return Math.PI \* radius \* radius;

}

}

public class Rectangle extends Shape{

private double length;

private double width;

// Constructor for Rectangle class

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

@Override

public double area() {

return length \* width;

}

}

import java.util.Scanner;

public class Inheritance {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Get user input for circle radius

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

double radius = scanner.nextDouble();

// Create a Circle object using user input

Circle circle = new Circle(radius);

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

// Get user input for rectangle dimensions

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

double length = scanner.nextDouble();

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

double width = scanner.nextDouble();

// Create a Rectangle object using user input

Rectangle rectangle = new Rectangle(length, width);

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

scanner.close();

}

}