Name – Harshit

MIS – 112316018

Java Lab

Assignment 2

**Write a program to convert uppercase to lowercase and lower case to upper case in java.**

**Code:**

import java.util.Scanner;

public class CaseConverter {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");

        String input = scanner.nextLine();

        String convertedString = convertCase(input);

        System.out.println("Converted string: " + convertedString);

        scanner.close();

    }

    public static String convertCase(String str) {

        StringBuilder result = new StringBuilder();

        for (char ch : str.toCharArray()) {

            if (Character.isUpperCase(ch)) {

                result.append(Character.toLowerCase(ch));

            } else if (Character.isLowerCase(ch)) {

                result.append(Character.toUpperCase(ch));

            } else {

                result.append(ch);

            }

        }

        return result.toString();

    }

}

**Output:**

****

**Write a Program to print the area of different triangles.**

**Code:**

import java.util.Scanner;

public class TriangleAreaCalculator {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Select the type of triangle:");

        System.out.println("1. Right-angled triangle");

        System.out.println("2. Equilateral triangle");

        System.out.println("3. Scalene triangle");

        int choice = scanner.nextInt();

        switch (choice) {

            case 1:

                System.out.print("Enter the base of the triangle: ");

                double base = scanner.nextDouble();

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

                double height = scanner.nextDouble();

                System.out.println("Area of the right-angled triangle: " + calculateRightTriangleArea(base, height));

                break;

            case 2:

                System.out.print("Enter the side length of the equilateral triangle: ");

                double side = scanner.nextDouble();

                System.out.println("Area of the equilateral triangle: " + calculateEquilateralTriangleArea(side));

                break;

            case 3:

                System.out.print("Enter the first side of the triangle: ");

                double a = scanner.nextDouble();

                System.out.print("Enter the second side of the triangle: ");

                double b = scanner.nextDouble();

                System.out.print("Enter the third side of the triangle: ");

                double c = scanner.nextDouble();

                System.out.println("Area of the scalene triangle: " + calculateScaleneTriangleArea(a, b, c));

                break;

            default:

                System.out.println("Invalid choice. Please select a valid triangle type.");

        }

        scanner.close();

    }

    public static double calculateRightTriangleArea(double base, double height) {

        return 0.5 \* base \* height;

    }

    public static double calculateEquilateralTriangleArea(double side) {

        return (Math.sqrt(3) / 4) \* side \* side;

    }

    public static double calculateScaleneTriangleArea(double a, double b, double c) {

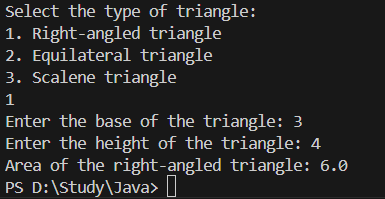
        double s = (a + b + c) / 2;

        return Math.sqrt(s \* (s - a) \* (s - b) \* (s - c));

    }

}

**Output:**

****

**Write a java Program to check the number is Prime or not.**

**Code:**

import java.util.Scanner;

public class PrimeChecker {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

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

        int number = scanner.nextInt();

        if (isPrime(number)) {

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

        } else {

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

        }

        scanner.close();

    }

    public static boolean isPrime(int num) {

        if (num <= 1) {

            return false;

        }

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

            if (num % i == 0) {

                return false;

            }

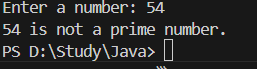
        }

        return true;

    }

}

**Output:**

****

**Write a java Program to generate a Ladder of number.**

**Code :**

import java.util.Scanner;

public class NumberLadder {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

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

        int rows = scanner.nextInt();

        generateLadder(rows);

        scanner.close();

    }

    public static void generateLadder(int rows) {

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

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

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

            }

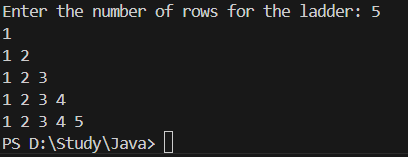
            System.out.println();

        }

    }

}

**Output:**

****

**Write a program to return a new string for given a string where the first and last chars have been exchanged.**

**Code:**

import java.util.Scanner;

public class SwapFirstLast {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");

        String input = scanner.nextLine();

        String result = swapFirstLast(input);

        System.out.println("Modified string: " + result);

        scanner.close();

    }

    public static String swapFirstLast(String str) {

        if (str.length() <= 1) {

            return str;

        }

        char first = str.charAt(0);

        char last = str.charAt(str.length() - 1);

        return last + str.substring(1, str.length() - 1) + first;

    }

}

**Output:**

****

**Explore Classes and Objects, Constructors**.

**1. Classes in Java:**

A **class** is a blueprint for creating objects. It defines a data structure that contains fields (attributes) and methods (functions) that operate on the data. A class encapsulates data for the object and methods to manipulate that data.

**2. Objects in Java:**

An **object** is an instance of a class. When you create an object, the class blueprint is used to allocate memory and initialize its attributes. Each object has its own copy of instance variables, which can be different from other objects.

**3. Constructors in Java:**

A **constructor** is a special method that is called when an object is instantiated. It is used to initialize the newly created object. Constructors have the same name as the class and do not have a return type.

There are two types of constructors:

* **Default Constructor**: A constructor with no parameters. If no constructor is defined, Java provides a default constructor.
* **Parameterized Constructor**: A constructor that takes parameters to initialize the object with specific values.