Prompt: Write a Java program that takes a number as input and prints its multiplication table upto 10.

Output:

```
import java.util.Scanner;

public class MultiplicationTable {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");

        int num = scanner.nextInt();

        for (int i = 1; i <= 10; i++) {
            System.out.println(num + " x " + i + " = " + (num * i));
        }
    }
}
```

Prompt:
Write a Java program to display the following pattern.

```
*****
****
***
**
*
```

Output:

```
public class Pattern {
    public static void main(String[] args) {
        for (int i = 5; i > 0; i--) {
            for (int j = 0; j < i; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

Prompt:
Write a Java program to print the area and perimeter of a circle.

Output:

```
public class Circle {
    public static void main(String[] args) {
        double radius = 7.5;
        double perimeter = 2 * Math.PI * radius;
        double area = Math.PI * radius * radius;

        System.out.println("Perimeter of the circle is: " + perimeter);
        System.out.println("Area of the circle is: " + area);
    }
}
```

Prompt:
Write a Java program to add two binary numbers.

Output:

```
public class BinaryAddition {
    public static void main(String[] args) {
        String binary1 = "1010";
        String binary2 = "1011";

        int decimal1 = Integer.parseInt(binary1, 2);
        int decimal2 = Integer.parseInt(binary2, 2);

        int sum = decimal1 + decimal2;

        String binarySum = Integer.toBinaryString(sum);

        System.out.println("The sum of the two binary numbers is: " + binarySum);
    }
}
```

Prompt:
Write a Java program to convert a decimal number to binary number and vice versa.

Output:

```
//Convert Decimal to Binary
public class DecimalToBinary {
    public static void main(String[] args) {
        int decimalNumber = 10;
        int[] binaryNumber = new int[40];
        int index = 0;

        while (decimalNumber > 0) {
            binaryNumber[index++] = decimalNumber % 2;
            decimalNumber = decimalNumber / 2;
        }

        for (int i = index - 1; i >= 0; i--) {
            System.out.print(binaryNumber[i]);
        }
    }
}

//Convert Binary to Decimal
public class BinaryToDecimal {
    public static void main(String[] args) {
        int binaryNumber = 1010;
        int decimalNumber = 0;
        int index = 0;

        while (true) {
            if (binaryNumber == 0) {
                break;
            } else {
                int temp = binaryNumber % 10;
                decimalNumber += temp * Math.pow(2, index);
                binaryNumber = binaryNumber / 10;
                index++;
            }
        }

        System.out.println(decimalNumber);

    }
}
```

Prompt:
Write a Java program to reverse a string.

Output:

```
public class ReverseString {
    public static void main(String[] args) {
        String str = "Hello World";
        String reversedStr = "";

        for (int i = str.length() - 1; i >= 0; i--) {
            reversedStr += str.charAt(i);
        }

        System.out.println("The reversed string is: " + reversedStr);
    }
}
```

Prompt:
Write a Java program to count the letters, spaces, numbers and other characters of an input string.

Output:

```
public class CountCharacters {
    public static void main(String[] args) {
        String str = "This is a test string 1!";
        int letter = 0, space = 0, num = 0, other = 0;

        for(int i = 0; i < str.length(); i++)
        {
            char ch = str.charAt(i);
            if(Character.isLetter(ch)) {
                letter ++ ;
            }
            else if(Character.isDigit(ch)) {
                num ++ ;
            }
            else if(Character.isSpaceChar(ch)) {
                space ++ ;
            }
            else {
                other ++;
            }
        }

        System.out.println("Letters: " + letter);
        System.out.println("Spaces: " + space);
        System.out.println("Numbers: " + num);
        System.out.println("Other: " + other);

    }
}
```

Prompt:
Implement a Java function that calculates the sum of digits for a given char array consisting of the digits '0' to '9'. The function should return the digit sum as a long value.

Output:

```
public static long digitSum(char[] digits) {
    long sum = 0;
    for (char c : digits) {
        sum += Character.getNumericValue(c);
    }
    return sum;
}
```

Prompt:
Create a java program that finds the smallest and largest element from the array

Output:

```
public class SmallestLargest {

    public static void main(String[] args) {
        // array of 10 numbers
        int numbers[] = {32, 43, 53, 54, 32, 65, 63, 98, 43, 23};

        // assign first element of an array to largest and smallest
        int smallest = numbers[0];
        int largest = numbers[0];

        for (int i = 1; i < numbers.length; i++) {
            if (numbers[i] > largest) {
                largest = numbers[i];
            } else if (numbers[i] < smallest) {
                smallest = numbers[i];
            }
        }

        System.out.println("Largest Number is : " + largest);
        System.out.println("Smallest Number is : " + smallest);
    }
}
```

Prompt:
Designed a java class SortData that contains the method asec() and desc().

Output:

```
public class SortData {
    public void asec(int[] arr) {
        int temp;
        for (int i = 0; i < arr.length; i++) {
            for (int j = i + 1; j < arr.length; j++) {
                if (arr[i] > arr[j]) {
                    temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
    }

    public void desc(int[] arr) {
        int temp;
        for (int i = 0; i < arr.length; i++) {
            for (int j = i + 1; j < arr.length; j++) {
                if (arr[i] < arr[j]) {
                    temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
    }
}
```

Prompt:
Designed a java class that demonstrates the use of constructor and destructor.

Output:

```
public class ConstructorAndDestructor {
   private int value;

   // Constructor
   public ConstructorAndDestructor(int value) {
      this.value = value;
      System.out.println("Constructor called with value: " + value);
   }

   // Destructor
   @Override
   protected void finalize() throws Throwable {
      System.out.println("Destructor called with value: " + value);
      super.finalize();
   }

   public static void main(String[] args) {
      ConstructorAndDestructor obj = new ConstructorAndDestructor(10);
      obj = null;

      System.gc(); // Invoke garbage collector to call destructor
   }
}
```

Prompt:
Write a java program to demonstrate the implementation of abstract class.

Output:

```
public abstract class Animal {
   public abstract void makeSound();
}

public class Dog extends Animal {
   @Override
   public void makeSound() {
      System.out.println("Woof!");
   }
}

public class Cat extends Animal {
   @Override
   public void makeSound() {
      System.out.println("Meow!");
   }
}

public class Main {

    public static void main(String[] args) {

        Dog dog = new Dog();
        dog.makeSound();

        Cat cat = new Cat();
        cat.makeSound();

    }
}
```

Prompt:
Write a java program to implement single level inheritance.

Output:

```
public class Animal {
    public void eat() {
        System.out.println("Eating...");
    }
}

public class Dog extends Animal {
    public void bark() {
        System.out.println("Barking...");
    }
}

public class TestInheritance {

    public static void main(String args[]) {
        Dog d = new Dog();
        d.bark();
        d.eat();
    }
}
```

Prompt:
Write a java program to implement method overriding

Output:

```
public class MethodOverriding {
   public static void main(String[] args) {
      Animal animal = new Animal();
      animal.move();
      Bird bird = new Bird();
      bird.move();
   }
}

class Animal {
   public void move() {
      System.out.println("Animals can move");
   }
}

class Bird extends Animal {
   public void move() {
      System.out.println("Birds can fly");
   }
}
```

Prompt:
Write a java program to implement multiple inheritance.

Output:

```
public class MultipleInheritance {

    public static void main(String[] args)
    {
        // creating object of sub class B
        B obj = new B();

        // calling the display method of sub class B
        obj.display();
    }
}

// superclass A
class A {
    public void print()
    {
        System.out.println("Hello from A");
    }
}

// superclass B
class B extends A {
    public void display()
    {
        // calling the print method of superclass A
        super.print();

        System.out.println("Hello from B");
    }
}
```

Prompt:
Create a package, Add the necessary classes and import the package in java class.

Output:

```
// Package name
package com.example;

// Class 1
public class Class1 {
    public void printMessage() {
        System.out.println("Hello from Class1!");
    }
}

// Class 2
public class Class2 {
    public void printMessage() {
        System.out.println("Hello from Class2!");
    }
}

// Java class
import com.example.*;  // Import the package
public class MyClass {

    public static void main(String[] args) {

        // Create objects of the classes
        Class1 obj1 = new Class1();
        Class2 obj2 = new Class2();

        // Call the methods of the classes
        obj1.printMessage();
        obj2.printMessage();

    }
}
```

Prompt:
Write a java program to add two matrices and print the resultant matrix.

Output:

```
public class MatrixAddition {

    public static void main(String[] args) {

        int rows, columns;

        // Initializing two matrices
        int[][] a = { { 1, 3, 4 }, { 2, 4, 3 }, { 3, 4, 5 } };
        int[][] b = { { 1, 3, 4 }, { 2, 4, 3 }, { 1, 2, 4 } };

        // Finding the number of rows and columns in matrix
        rows = a.length;
        columns = a[0].length;

        // Adding corresponding elements of two matrices
        int[][] c = new int[rows][columns];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                c[i][j] = a[i][j] + b[i][j];
            }
        }

        // Displaying the resultant matrix after addition
        System.out.println("Resultant Matrix:");
        for (int[] row : c) {
            for (int column : row) {
                System.out.print(column + "    ");
            }
            System.out.println();
        }
    }
}
```

Prompt:
Write a java program for multiplying two matrices and print the product for the same.

Output:

```
public class MatrixMultiplication {

    public static void main(String[] args) {
        int rowsInA = 3;
        int columnsInA = 3;
        int rowsInB = 3;
        int columnsInB = 3;

        // Matrix A
        int[][] a = { {1, 2, 3},
                      {4, 5, 6},
                      {7, 8, 9} };

        // Matrix B
        int[][] b = { {1, 2, 3},
                      {4, 5, 6},
                      {7, 8, 9} };

        // To store result
        int[][] c = new int[rowsInA][columnsInB];

        // Multiplying matrix A and B and storing in C.
        for (int i = 0; i < rowsInA; i++) {
            for (int j = 0; j < columnsInB; j++) {
                for (int k = 0; k < columnsInA; k++) {
                    c[i][j] += a[i][k] * b[k][j];
                }
            }
        }

        // Displaying the result.
        System.out.println("Product of two matrices is: ");
        for (int[] row : c) {
            for (int column : row) {
                System.out.print(column + "    ");
            }
            System.out.println();
        }

    }

}
```

Prompt:
Write a java program to implement the vectors.

Output:

```
import java.util.Vector;

public class VectorExample {
   public static void main(String args[]) {
      // create default vector
      Vector v = new Vector();
      v.add(1);
      v.add(2);
      v.add("geeks");
      v.add("forGeeks");
      v.add(3);

      System.out.println("Vector is " + v);

      // create vector of initial capacity(size) of 2
      Vector v2 = new Vector(2);
      v2.add(1);
      v2.add(2);

      // will increase capacity to 4
      v2.add("geeks");
      v2.add("forGeeks");

      System.out.println("Vector 2 is " + v2);

   }
}
```

Prompt:
Write a java program to implement thread life cycle.

Output:

```
public class ThreadLifeCycle implements Runnable {

   public void run() {
      System.out.println("Thread is in running state.");
   }

   public static void main(String[] args) {
      ThreadLifeCycle tlc = new ThreadLifeCycle();
      Thread t = new Thread(tlc);

      // thread is in new state
      System.out.println("Thread is in new state.");

      // thread is in runnable state
      t.start();

      // thread is in running state

      // thread is in dead state
      try {
         t.join();
      } catch (InterruptedException ie) {
         ie.printStackTrace();
      }
      System.out.println("Thread is in dead state.");
   }
}
```

Prompt:
Write a java program to implement multithreading.

Output:

```
public class MultiThreadingExample {
  public static void main(String[] args) {
    // Create two Runnable objects
    Runnable r1 = new MyRunnable("Thread-1");
    Runnable r2 = new MyRunnable("Thread-2");

    // Create two threads
    Thread t1 = new Thread(r1);
    Thread t2 = new Thread(r2);

    // Start the threads
    t1.start();
    t2.start();
  }
}

class MyRunnable implements Runnable {
  private String name;

  public MyRunnable(String name) {
    this.name = name;
  }

  @Override
  public void run() {
    for (int i = 0; i < 10; i++) {
      System.out.println("Thread " + this.name + " is running: " + i);

      try {
        Thread.sleep(100);
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    }
  }
}
```

Prompt:
Write a java program to open a file and display the contents in the console window.

Output:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class FileDisplay {

    public static void main(String[] args) {

        BufferedReader br = null;

        try {

            String sCurrentLine;

            br = new BufferedReader(new FileReader("C:\\test\\filename.txt"));

            while ((sCurrentLine = br.readLine()) != null) {
                System.out.println(sCurrentLine);
            }

        } catch (IOException e) {
            e.printStackTrace();
        } finally {
```

```
      try {
        if (br != null)br.close();
      } catch (IOException ex) {
        ex.printStackTrace();
      }
    }

  }
}
```

Prompt:
Write a java program to copy the contents from one file to other file.

Output:

```
import java.io.*;

public class CopyFile {
  public static void main(String args[]) throws IOException {
    FileInputStream in = null;
    FileOutputStream out = null;

    try {
      in = new FileInputStream("input.txt");
      out = new FileOutputStream("output.txt");

      int c;
      while ((c = in.read()) != -1) {
        out.write(c);
      }
    }finally {
      if (in != null) {
        in.close();
      }
      if (out != null) {
        out.close();
      }
    }
  }
}
```

Prompt:
Write a java program to read the student data from user and store it in the file.

Output:

```
import java.io.*;
import java.util.Scanner;

public class StudentData {
    public static void main(String[] args) throws IOException {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number of students: ");
        int numOfStudents = sc.nextInt();

        FileWriter fw = new FileWriter("studentData.txt");
        BufferedWriter bw = new BufferedWriter(fw);

        for (int i = 0; i < numOfStudents; i++) {
            System.out.println("Enter the student " + (i + 1) + " name: ");
            String name = sc.next();

            System.out.println("Enter the student " + (i + 1) + " age: ");
            int age = sc.nextInt();

            System.out.println("Enter the student " + (i + 1) + " grade: ");
            int grade = sc.nextInt();

            bw.write(name + "," + age + "," + grade);
            bw.newLine();
        }

        bw.close();
    }
}
```

Prompt:
Design a JAVA AWT program to print the factorial for an input value.

Output:

```
import java.awt.*;
import java.awt.event.*;

public class Factorial extends Frame implements ActionListener {

    Label l1, l2;
    TextField t1, t2;
    Button b1;

    Factorial() {

        setLayout(new FlowLayout());

        l1 = new Label("Enter a number: ");
        add(l1);

        t1 = new TextField(20);
        add(t1);

        b1 = new Button("Calculate");
        add(b1);

        l2 = new Label("Factorial is: ");
        add(l2);

        t2 = new TextField(20);
        t2.setEditable(false);
        add(t2);

        b1.addActionListener(this);

        setTitle("Factorial Calculator");
        setSize(400, 400);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {

        int n = Integer.parseInt(t1.getText());
        int fact = 1;

        for (int i = 1; i <= n; i++) {
            fact *= i;
        }

        t2.setText(String.valueOf(fact));
    }

    public static void main(String[] args) {
        new Factorial();
    }
}
```

Prompt:
Write a java program to implement exception handling.

Output:

```
public class ExceptionHandling {
    public static void main(String[] args) {
        try {
            int a = 10;
            int b = 0;
            int c = a/b;
            System.out.println(c);
        } catch (ArithmeticException e) {
            System.out.println("Cannot divide by zero");
        } finally {
            System.out.println("Exception handling complete");
        }
    }
}
```

Prompt:
Design an JAVA AWT application that contains the interface to add student information and display the same.

Output:

```
import java.awt.*;
import java.awt.event.*;

public class StudentInfo extends Frame implements ActionListener {

    // Declare components
    Label lblName, lblRollNo, lblAddress;
    TextField txtName, txtRollNo, txtAddress;
    Button btnAdd, btnDisplay;

    // Constructor
    public StudentInfo() {

        // Set the title of the frame
        setTitle("Student Information");

        // Set the layout of the frame
        setLayout(new FlowLayout());

        // Initialize components
        lblName = new Label("Name: ");
        lblRollNo = new Label("Roll No: ");
        lblAddress = new Label("Address: ");

        txtName = new TextField(20);
        txtRollNo = new TextField(20);
        txtAddress = new TextField(20);

        btnAdd = new Button("Add");
        btnDisplay = new Button("Display");

        // Add components to the frame
        add(lblName);
        add(txtName);

        add(lblRollNo);
        add(txtRollNo);

        add(lblAddress);
        add(txtAddress);

        add(btnAdd);
        add(btnDisplay);

        // Register action listener for buttons
        btnAdd.addActionListener(this);
        btnDisplay.addActionListener(this);
        // Set the size of the frame and make it visible
        setSize(400, 200);
        setVisible(true);

    }

    // Action listener for buttons
    public void actionPerformed(ActionEvent e) {

        if (e.getSource() == btnAdd) {

            String name = txtName.getText();
            String rollNo = txtRollNo.getText();
            String address = txtAddress.getText();

            System.out.println("Name: " + name + "\nRoll No: " + rollNo + "\nAddress: " + address);

            txtName.setText("");
            txtRollNo.setText("");
            txtAddress.setText("");

            System.out.println("Student information added successfully!");

        } else if (e.getSource() == btnDisplay) {

            System.out.println("Displaying student information...");

            System.out.println("Name: John Doe\nRoll No: 12345\nAddress: 123 Main Street");

            System.out.println("Student information displayed successfully!");

        }

    }

    public static void main(String[] args) {
        new StudentInfo();
    }
}
```

Prompt:
Design a calculator based on JAVA AWT application.

Output:

```
import java.awt.*;
import java.awt.event.*;

public class Calculator extends Frame implements ActionListener {

    // Declare all the components
    TextField t1, t2;
    Label l1, l2;
```

```
Button b1, b2, b3, b4;

// Constructor
public Calculator() {

    // Set the title of the frame
    setTitle("Calculator");

    // Set the layout of the frame
    setLayout(new FlowLayout());

    // Create the components
    t1 = new TextField(10);
    t2 = new TextField(10);
    l1 = new Label("Number 1");
    l2 = new Label("Number 2");
    b1 = new Button("Add");
    b2 = new Button("Subtract");
    b3 = new Button("Multiply");
    b4 = new Button("Divide");

    // Add the components to the frame
    add(l1);
    add(t1);
    add(l2);
    add(t2);
    add(b1);
    add(b2);
    add(b3);
    add(b4);

    // Register the action listener for each button
    b1.addActionListener(this);
    b2.addActionListener(this);
    b3.addActionListener(this);
    b4.addActionListener(this);

    // Set the size of the frame and make it visible
    setSize(250, 200);
    setVisible(true);

}

// Action listener method
public void actionPerformed(ActionEvent e) {

   int n1 = Integer.parseInt(t1.getText());
   int n2 = Integer.parseInt(t2.getText());

   if (e.getSource() == b1) {

       int sum = n1 + n2;

       t1.setText("");
       t2.setText("");
```

```
            t1.setText(Integer.toString(sum));

        } else if (e.getSource() == b2) {

            int diff = n1 - n2;

            t1.setText("");
            t2.setText("");

            t1.setText(Integer.toString(diff));

        } else if (e.getSource() == b3) {

            int prod = n1 * n2;

            t1.setText("");
            t2.setText("");

            t1.setText(Integer.toString(prod));

        } else if (e.getSource() == b4) {

            float div = (float)n1 / n2;

            t1.setText("");
            t2.setText("");

            t1.setText(Float.toString(div));

        }

    }

    public static void main (String[] args) {

        Calculator c = new Calculator();

    }
}
```

Prompt:
Design an JAVA AWT application to generate result marks sheet.

Output:

```
import java.awt.*;
import java.awt.event.*;

public class ResultMarksSheet extends Frame implements ActionListener {

    Label lblName, lblRollNo, lblMarks;
    TextField txtName, txtRollNo, txtMarks;
    Button btnGenerate;

    ResultMarksSheet() {

        setLayout(new FlowLayout());

        lblName = new Label("Name: ");
        add(lblName);

        txtName = new TextField(20);
        add(txtName);

        lblRollNo = new Label("Roll No: ");
        add(lblRollNo);

        txtRollNo = new TextField(20);
        add(txtRollNo);

        lblMarks = new Label("Marks: ");
        add(lblMarks);

        txtMarks = new TextField(20);
        add(txtMarks);

        btnGenerate = new Button("Generate");
        add(btnGenerate);

        btnGenerate.addActionListener(this);

        setTitle("Result Marks Sheet");
        setSize(400, 400);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        String name = txtName.getText();
        String rollNo = txtRollNo.getText();
        int marks = Integer.parseInt(txtMarks.getText());

        System.out.println("Name: " + name + "\nRoll No: " + rollNo + "\nMarks: " + marks);
    }

    public static void main(String[] args) {
        new ResultMarksSheet();
    }
}
```

Prompt: Write a java program to read the student data from user and store it in the file.

Output:

```
import java.io.*;
import java.util.Scanner;
```

```
public class StudentData {

    public static void main(String[] args) throws IOException {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number of students: ");
        int numOfStudents = sc.nextInt();

        FileWriter fw = new FileWriter("studentData.txt");
        BufferedWriter bw = new BufferedWriter(fw);

        for (int i = 0; i < numOfStudents; i++) {
            System.out.println("Enter the student " + (i + 1) + " name: ");
            String name = sc.next();

            System.out.println("Enter the student " + (i + 1) + " age: ");
            int age = sc.nextInt();

            System.out.println("Enter the student " + (i + 1) + " grade: ");
            int grade = sc.nextInt();

            bw.write(name + "," + age + "," + grade);
            bw.newLine();
        }

        bw.close();
    }
}
```

Prompt:
Design a JAVA AWT program to print the factorial for an input value.

Output:

```
import java.awt.*;
import java.awt.event.*;

public class Factorial extends Frame implements ActionListener {

    Label l1, l2;
    TextField t1, t2;
    Button b1;

    Factorial() {

        setLayout(new FlowLayout());

        l1 = new Label("Enter a number: ");
        add(l1);

        t1 = new TextField(20);
        add(t1);

        b1 = new Button("Calculate");
        add(b1);

        l2 = new Label("Factorial is: ");
        add(l2);

        t2 = new TextField(20);
        t2.setEditable(false);
        add(t2);

        b1.addActionListener(this);

        setTitle("Factorial Calculator");
        setSize(400, 400);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {

        int n = Integer.parseInt(t1.getText());
        int fact = 1;

        for (int i = 1; i <= n; i++) {
            fact *= i;
        }

        t2.setText(String.valueOf(fact));
    }

    public static void main(String[] args) {
        new Factorial();
    }
}
```

Prompt:
Design an JAVA AWT program to perform various string operations like reverse string, string concatenation, etc.

Output:

```
import java.awt.*;
import java.awt.event.*;

public class StringOperations extends Frame implements ActionListener {

    // Declare components
    Label lblString1, lblString2, lblResult;
    TextField txtString1, txtString2, txtResult;
    Button btnReverse, btnConcatenate;

    // Constructor
    public StringOperations() {
```

```
    // Set the title of the frame
    setTitle("String Operations");

    // Set the layout of the frame
    setLayout(new FlowLayout());

    // Initialize components
    lblString1 = new Label("Enter String 1:");
    lblString2 = new Label("Enter String 2:");
    lblResult = new Label("Result:");

    txtString1 = new TextField(20);
    txtString2 = new TextField(20);
    txtResult = new TextField(20);

    btnReverse = new Button("Reverse");
    btnConcatenate = new Button("Concatenate");

    // Add components to the frame
    add(lblString1);
    add(txtString1);

    add(lblString2);
    add(txtString2);

    add(btnReverse);
    add(btnConcatenate);

    add(lblResult);
    add(txtResult);

    // Register action listeners for the buttons
    btnReverse.addActionListener(this);
    btnConcatenate.addActionListener(this);

    // Set the size of the frame and make it visible
    setSize(400, 200);
    setVisible(true);

}

// Action listener method for the buttons
public void actionPerformed(ActionEvent e) {

    if (e.getSource() == btnReverse) {

        // Get the strings from the text fields
        String str1 = txtString1.getText();

        // Reverse the string
        String reversedStr = new StringBuilder(str1).reverse().toString();

        // Set the result in the result text field
        txtResult.setText(reversedStr);
        } else if (e.getSource() == btnConcatenate) {

            // Get the strings from the text fields
            String str1 = txtString1.getText();
            String str2 = txtString2.getText();

            // Concatenate the strings
            String concatenatedStr = str1 + str2;

            // Set the result in the result text field
            txtResult.setText(concatenatedStr);

        }

    }

    public static void main(String[] args) {

      new StringOperations();

    }
}
```

Prompt:
Write a java program to implement exception handling.

Output:

```
public class ExceptionHandling {
    public static void main(String[] args) {
        try {
            int a = 10;
            int b = 0;
            int c = a/b;
            System.out.println(c);
        } catch (ArithmeticException e) {
            System.out.println("Cannot divide by zero");
        } finally {
            System.out.println("Exception handling complete");
        }
    }
}
```

Prompt:
Design an JAVA AWT application that contains the interface to add student information and display the same.

Output:

```
import java.awt.*;
import java.awt.event.*;
```

```
public class StudentInfo extends Frame implements ActionListener {

    // Declare components
    Label lblName, lblRollNo, lblAddress;
    TextField txtName, txtRollNo, txtAddress;
    Button btnAdd, btnDisplay;

    // Constructor to setup GUI components and event handling
    public StudentInfo() {
        setLayout(new FlowLayout());

        // Create labels and text fields
        lblName = new Label("Name: ");
        add(lblName);

        txtName = new TextField(20);
        add(txtName);

        lblRollNo = new Label("Roll No: ");
        add(lblRollNo);

        txtRollNo = new TextField(20);
        add(txtRollNo);

        lblAddress = new Label("Address: ");
        add(lblAddress);

        txtAddress = new TextField(20);
        add(txtAddress);

        // Create buttons and add action listeners
        btnAdd = new Button("Add");
        btnAdd.addActionListener(this);
        add(btnAdd);

        btnDisplay = new Button("Display");
        btnDisplay.addActionListener(this);
        add(btnDisplay);

        setTitle("Student Information");
        setSize(400, 200);
        setVisible(true);
    }

    // ActionEvent handler - Called back upon button-click.
    @Override
    public void actionPerformed(ActionEvent evt) {
        String name = txtName.getText();
        String rollNo = txtRollNo.getText();
        String address = txtAddress.getText();

        if (evt.getSource() == btnAdd) {
            System.out.println("Adding student information...");
            System.out.println("Name: " + name + ", Roll No: " + rollNo + ", Address: " + address);
        } else if (evt.getSource() == btnDisplay) {
            System.out.println("Displaying student information...");
            System.out.println("Name: " + name + ", Roll No: " + rollNo + ", Address: " + address);
        }
    }

    public static void main(String[] args) {
        new StudentInfo();
    } }
```

Prompt:
Design a calculator based on JAVA AWT application.

Output:

```
import java.awt.*;
import java.awt.event.*;

public class Calculator extends Frame implements ActionListener {

    // Declare all the components
    TextField t1, t2;
    Label l1, l2;
    Button b1, b2, b3, b4;

    // Constructor
    public Calculator() {

        // Set the title of the frame
        setTitle("Calculator");

        // Set the layout of the frame
        setLayout(new FlowLayout());

        // Create the components
        t1 = new TextField(10);
        t2 = new TextField(10);
        l1 = new Label("Number 1");
        l2 = new Label("Number 2");
        b1 = new Button("Add");
        b2 = new Button("Subtract");
        b3 = new Button("Multiply");
        b4 = new Button("Divide");

        // Add the components to the frame
        add(l1);
        add(t1);
        add(l2);
        add(t2);
        add(b1);
        add(b2);
        add(b3);
        add(b4);
```

```
   // Register the action listener for each button
   b1.addActionListener(this);
   b2.addActionListener(this);
   b3.addActionListener(this);
   b4.addActionListener(this);

   // Set the size of the frame and make it visible
   setSize(250, 200);
   setVisible(true);

}

// Action listener method
public void actionPerformed(ActionEvent e) {

  int n1 = Integer.parseInt(t1.getText());
  int n2 = Integer.parseInt(t2.getText());

  if (e.getSource() == b1) {

     int sum = n1 + n2;

     t1.setText("");
     t2.setText("");

     t1.setText(Integer.toString(sum));

  } else if (e.getSource() == b2) {

     int diff = n1 - n2;

     t1.setText("");
     t2.setText("");

     t1.setText(Integer.toString(diff));

  } else if (e.getSource() == b3) {

     int prod = n1 * n2;

     t1.setText("");
     t2.setText("");

     t1.setText(Integer.toString(prod));

  } else if (e.getSource() == b4) {

     float div = (float)n1 / n2;

     t1.setText("");
     t2.setText("");

     t1.setText(Float.toString(div));
```

```
        }

    }

    public static void main (String[] args) {

        Calculator c = new Calculator();

    }
}
```

Prompt:
Design an JAVA AWT application to generate result marks sheet.

Output:

```
import java.awt.*;
import java.awt.event.*;

public class ResultMarksSheet extends Frame implements ActionListener {

    Label lblName, lblRollNo, lblMarks;
    TextField txtName, txtRollNo, txtMarks;
    Button btnGenerate;

    ResultMarksSheet() {

        setLayout(new FlowLayout());

        lblName = new Label("Name");
        add(lblName);

        txtName = new TextField(20);
        add(txtName);

        lblRollNo = new Label("Roll No");
        add(lblRollNo);

        txtRollNo = new TextField(20);
        add(txtRollNo);

        lblMarks = new Label("Marks");
        add(lblMarks);

        txtMarks = new TextField(20);
        add(txtMarks);

        btnGenerate = new Button("Generate");
        add(btnGenerate);

        btnGenerate.addActionListener(this);

        setTitle("Result Marks Sheet");
        setSize(400, 400);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {

        String name = txtName.getText();
        String rollNo = txtRollNo.getText();
        int marks = Integer.parseInt(txtMarks.getText());

        System.out.println("Name: " + name);
        System.out.println("Roll No: " + rollNo);
        System.out.println("Marks: " + marks);

    }

    public static void main(String[] args) {
        new ResultMarksSheet();
    }
}
```