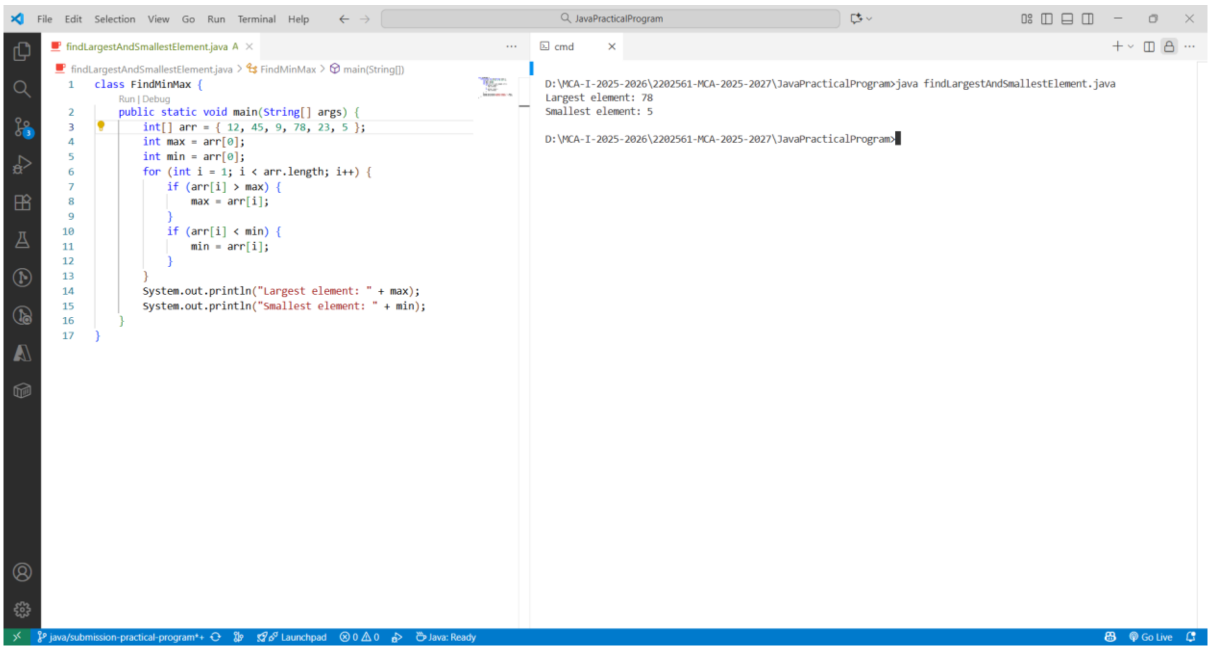
Name	Udhayarajan J
Class	MCA - I
Sub	Java Programming
Type	Experiment Practical Program
Date	2025-11-12
Roll No	2202561

Experiment 2: Introduction of arrays and different operations on arrays

Design an application by using arrays

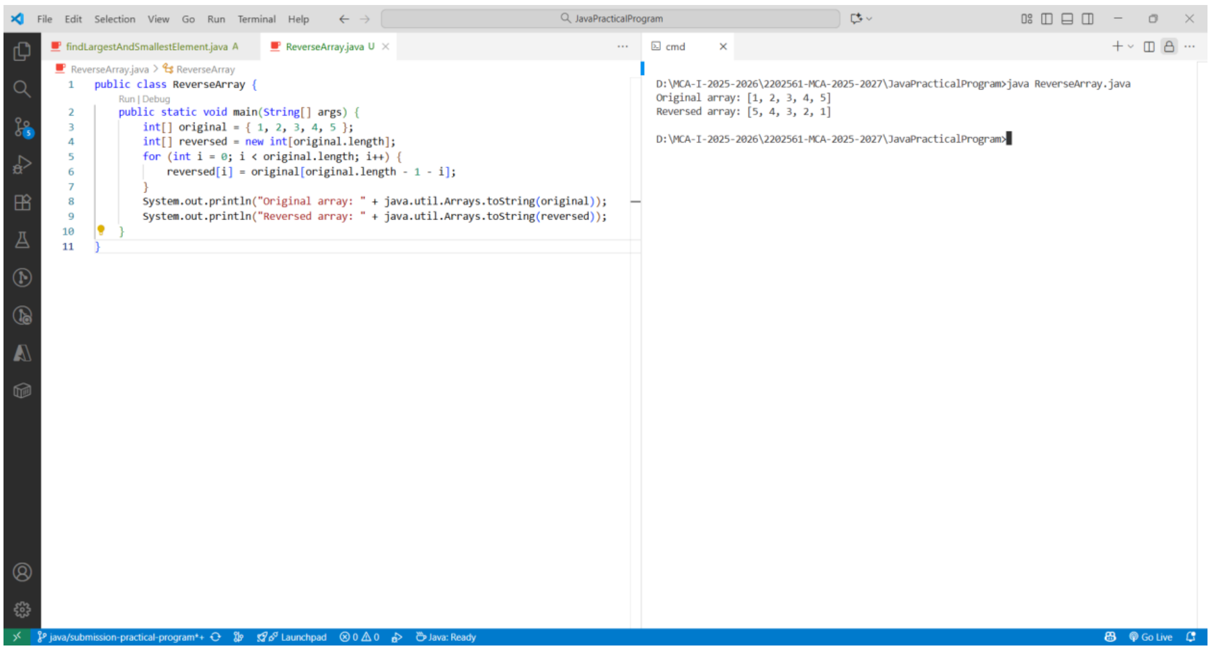
a. Finding the Largest/Smallest Element


File Edit Selection View Go Run Terminal Help
JavaPracticalProgram
findLargestAndSmallestElement.java A
cmd
findLargestAndSmallestElement.java > FindMinMax > main(String[])
Run | Debug
java/submission-practical-program*+
Launchpad
Java: Ready
Go Live

```java
class FindMinMax {
    public static void main(String[] args) {
        int[] arr = { 12, 45, 9, 78, 23, 5 };
        int max = arr[0];
        int min = arr[0];
        for (int i = 1; i < arr.length; i++) {
            if (arr[i] > max) {
                max = arr[i];
            }
            if (arr[i] < min) {
                min = arr[i];
            }
        }
        System.out.println("Largest element: " + max);
        System.out.println("Smallest element: " + min);
    }
}
```

```
D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>java findLargestAndSmallestElement.java
Largest element: 78
Smallest element: 5

D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

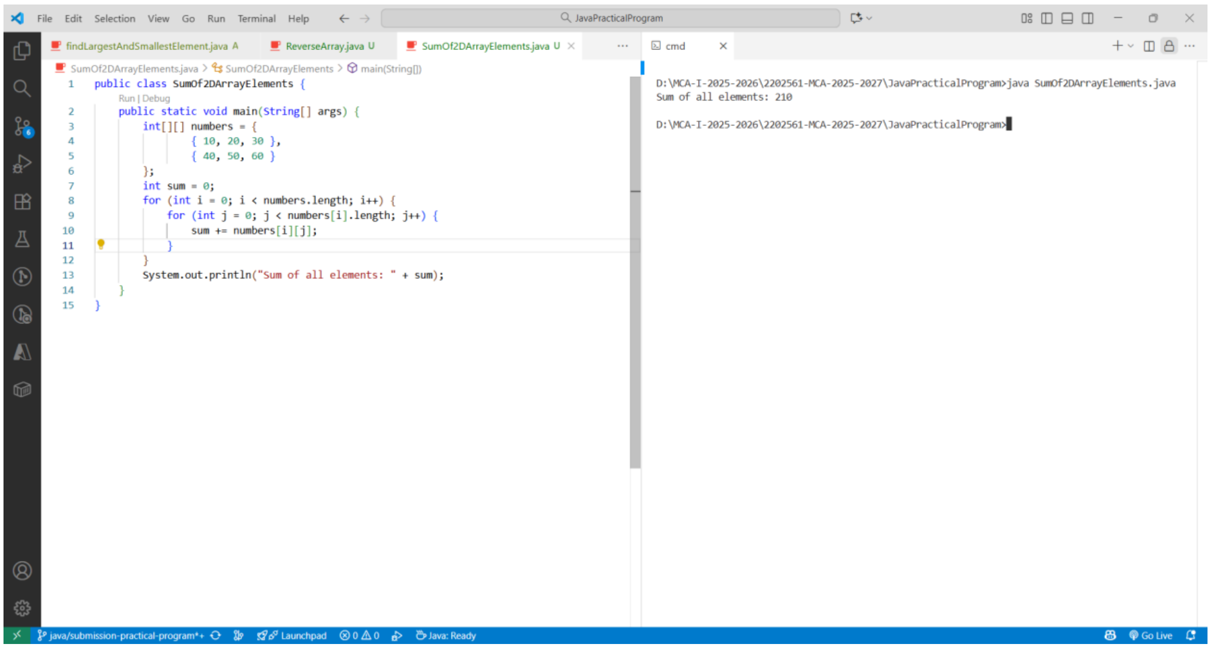
b. Reversing an Array


File Edit Selection View Go Run Terminal Help
JavaPracticalProgram
findLargestAndSmallestElement.java A
ReverseArray.java U
cmd
ReverseArray.java > ReverseArray
Run | Debug
java/submission-practical-program*+
Launchpad
Java: Ready
Go Live

```java
public class ReverseArray {
    public static void main(String[] args) {
        int[] original = { 1, 2, 3, 4, 5 };
        int[] reversed = new int[original.length];
        for (int i = 0; i < original.length; i++) {
            reversed[i] = original[original.length - 1 - i];
        }
        System.out.println("Original array: " + java.util.Arrays.toString(original));
        System.out.println("Reversed array: " + java.util.Arrays.toString(reversed));
    }
}
```

```
D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>java ReverseArray.java
Original array: [1, 2, 3, 4, 5]
Reversed array: [5, 4, 3, 2, 1]

D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

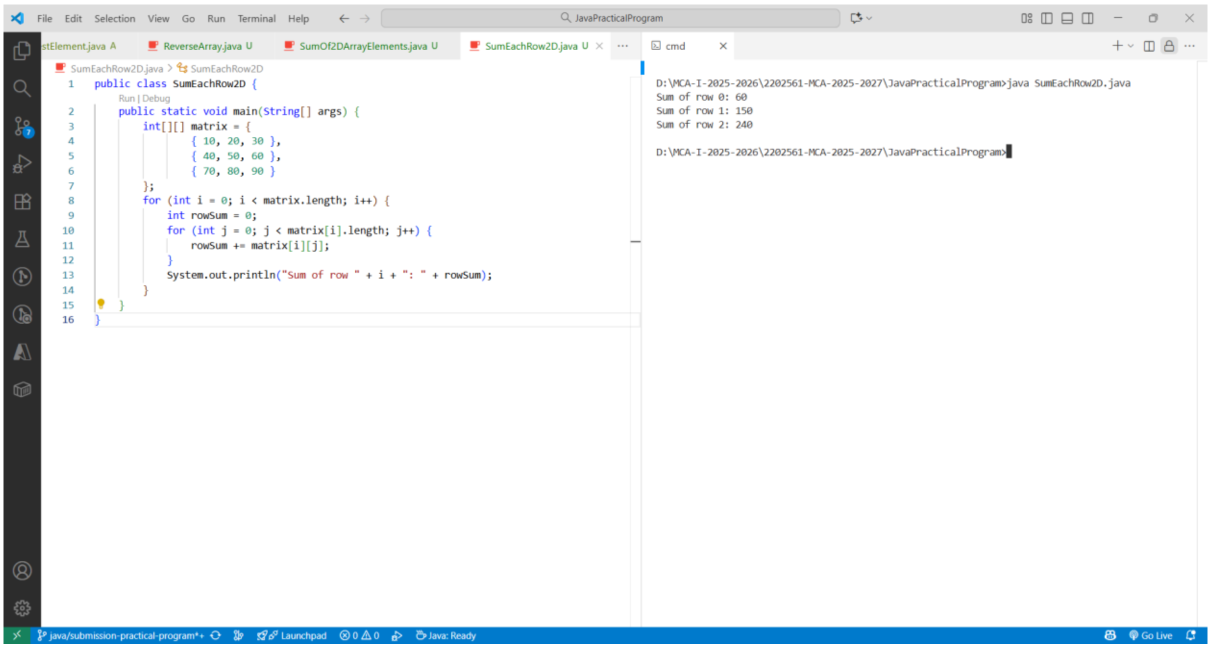
c. Sum of all elements in a 2D Array


File Edit Selection View Go Run Terminal Help
JavaPracticalProgram
findLargestAndSmallestElement.java A
ReverseArray.java U
SumOf2DArrayElements.java U
cmd
SumOf2DArrayElements.java > SumOf2DArrayElements > main(String[])
Run | Debug
java/submission-practical-program*+
Launchpad
Java: Ready
Go Live

```java
public class SumOf2DArrayElements {
    public static void main(String[] args) {
        int[][] numbers = {
                { 10, 20, 30 },
                { 40, 50, 60 }
        };
        int sum = 0;
        for (int i = 0; i < numbers.length; i++) {
            for (int j = 0; j < numbers[i].length; j++) {
                sum += numbers[i][j];
            }
        }
        System.out.println("Sum of all elements: " + sum);
    }
}
```

```
D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>java SumOf2DArrayElements.java
Sum of all elements: 210

D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

d. Sum of elements in each row of a 2D array

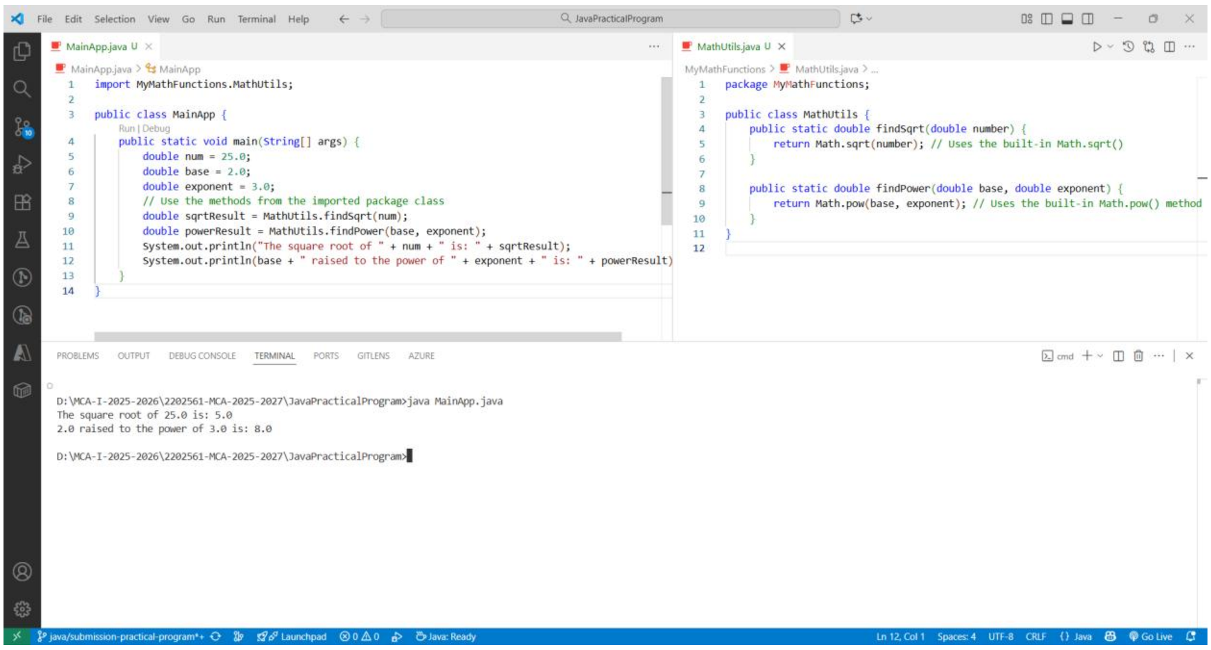

File Edit Selection View Go Run Terminal Help
JavaPracticalProgram
stElement.java A
ReverseArray.java U
SumOf2DArrayElements.java U
SumEachRow2D.java U
cmd
SumEachRow2D.java > SumEachRow2D
Run | Debug
java/submission-practical-program*+
Launchpad
Java: Ready
Go Live

```java
public class SumEachRow2D {
    public static void main(String[] args) {
        int[][] matrix = {
                { 10, 20, 30 },
                { 40, 50, 60 },
                { 70, 80, 90 }
        };
        for (int i = 0; i < matrix.length; i++) {
            int rowSum = 0;
            for (int j = 0; j < matrix[i].length; j++) {
                rowSum += matrix[i][j];
            }
            System.out.println("Sum of row " + i + ": " + rowSum);
        }
    }
}
```

```
D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>java SumEachRow2D.java
Sum of row 0: 60
Sum of row 1: 150
Sum of row 2: 240

D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

3: Package

Write a program for Implementation of package to create class and methods in mynmathfunction package



MainApp.java

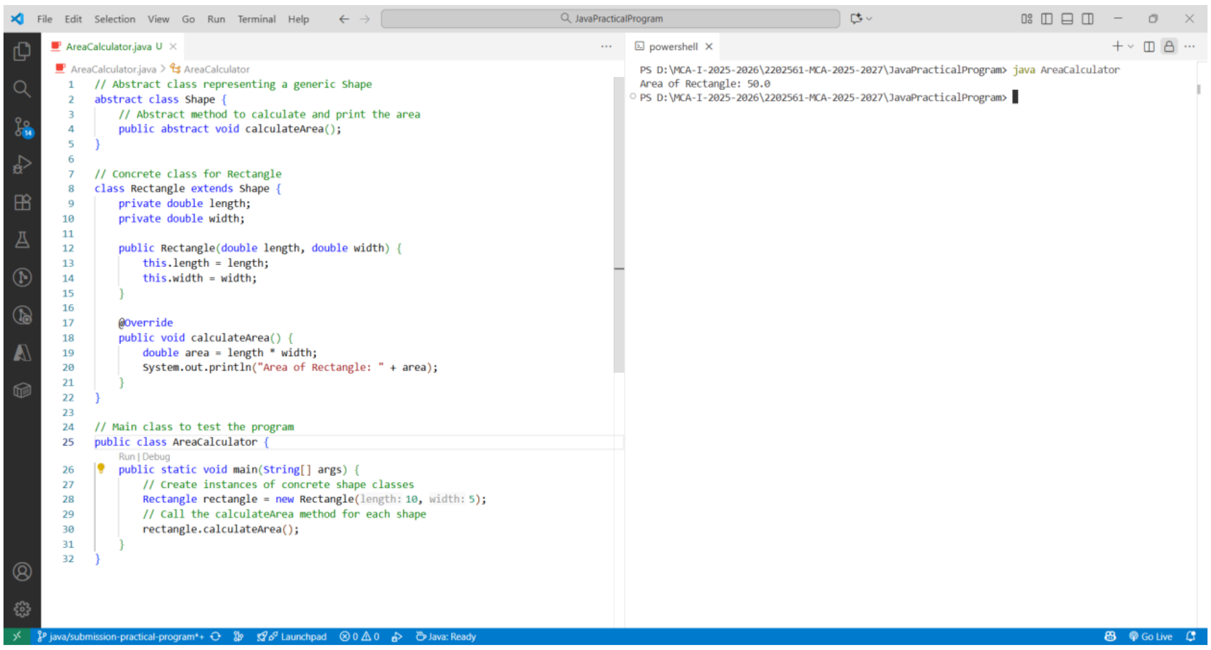
```java
import MyMathFunctions.MathUtils;

public class MainApp {
    public static void main(String[] args) {
        double num = 25.0;
        double base = 2.0;
        double exponent = 3.0;
        // Use the methods from the imported package class
        double sqrtResult = MathUtils.findSqrt(num);
        double powerResult = MathUtils.findPower(base, exponent);
        System.out.println("The square root of " + num + " is: " + sqrtResult);
        System.out.println(base + " raised to the power of " + exponent + " is: " + powerResult)
    }
}
```

MathUtils.java

```java
package MyMathFunctions;

public class MathUtils {
    public static double findSqrt(double number) {
        return Math.sqrt(number); // Uses the built-in Math.sqrt()
    }

    public static double findPower(double base, double exponent) {
        return Math.pow(base, exponent); // Uses the built-in Math.pow() method
    }
}
```

```
D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>java MainApp.java
The square root of 25.0 is: 5.0
2.0 raised to the power of 3.0 is: 8.0

D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

b. abstract class Experiment for calculating area

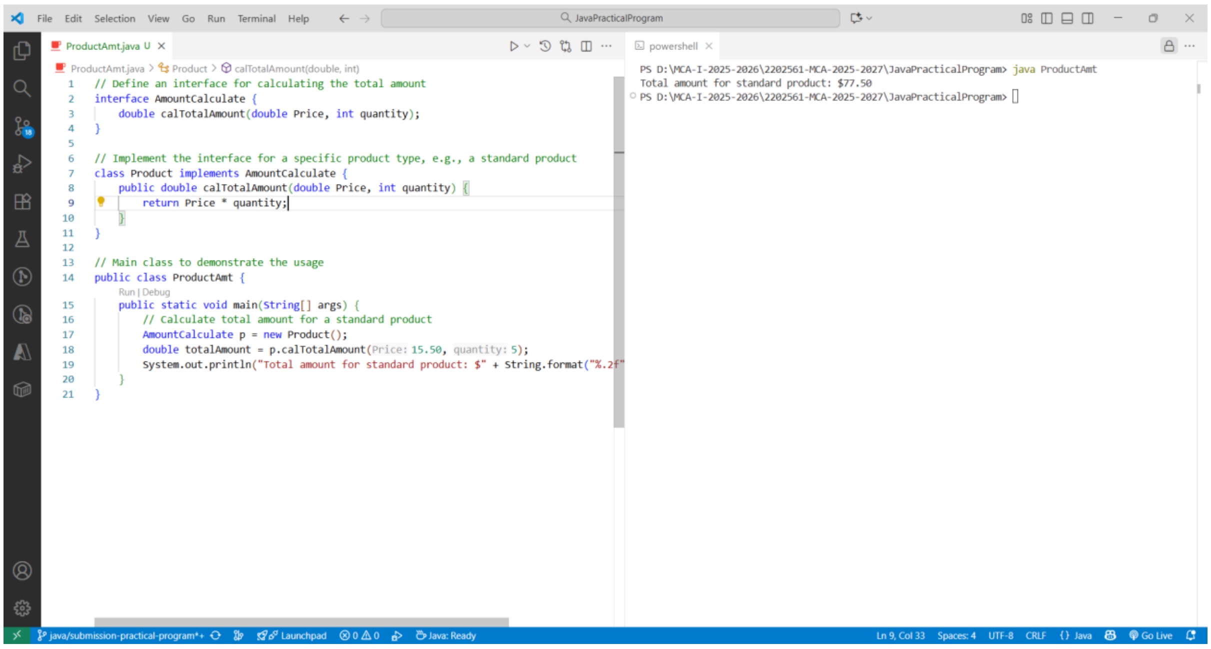


AreaCalculator.java

```java
// Abstract class representing a generic Shape
abstract class Shape {
    // Abstract method to calculate and print the area
    public abstract void calculateArea();
}

// Concrete class for Rectangle
class Rectangle extends Shape {
    private double length;
    private double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    @Override
    public void calculateArea() {
        double area = length * width;
        System.out.println("Area of Rectangle: " + area);
    }
}

// Main class to test the program
public class AreaCalculator {
    public static void main(String[] args) {
        // Create instances of concrete shape classes
        Rectangle rectangle = new Rectangle(length: 10, width: 5);
        // Call the calculateArea method for each shape
        rectangle.calculateArea();
    }
}
```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java AreaCalculator
Area of Rectangle: 50.0
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

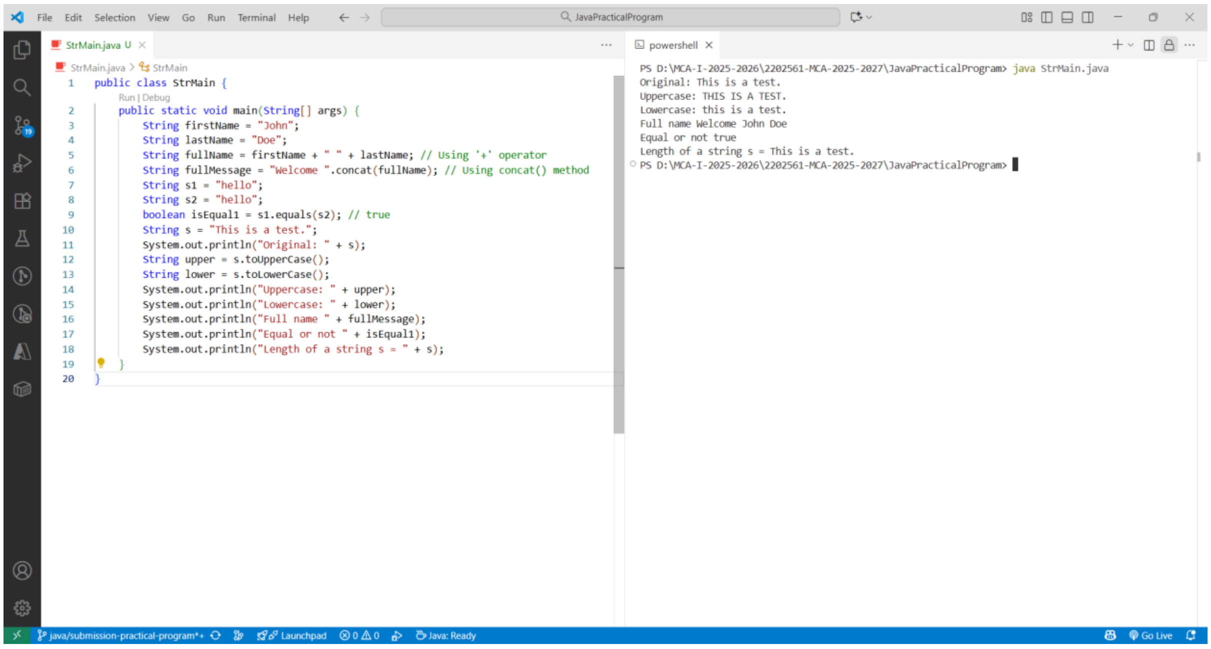
c. Interface program for calculating product amount



```java
// Define an interface for calculating the total amount
interface AmountCalculate {
    double calTotalAmount(double Price, int quantity);
}

// Implement the interface for a specific product type, e.g., a standard product
class Product implements AmountCalculate {
    public double calTotalAmount(double Price, int quantity) {
        return Price * quantity;
    }
}

// Main class to demonstrate the usage
public class ProductAmt {
    public static void main(String[] args) {
        // Calculate total amount for a standard product
        AmountCalculate p = new Product();
        double totalAmount = p.calTotalAmount(Price: 15.50, quantity: 5);
        System.out.println("Total amount for standard product: $" + String.format("%.2f"
    }
}
```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java ProductAmt
Total amount for standard product: $77.50
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

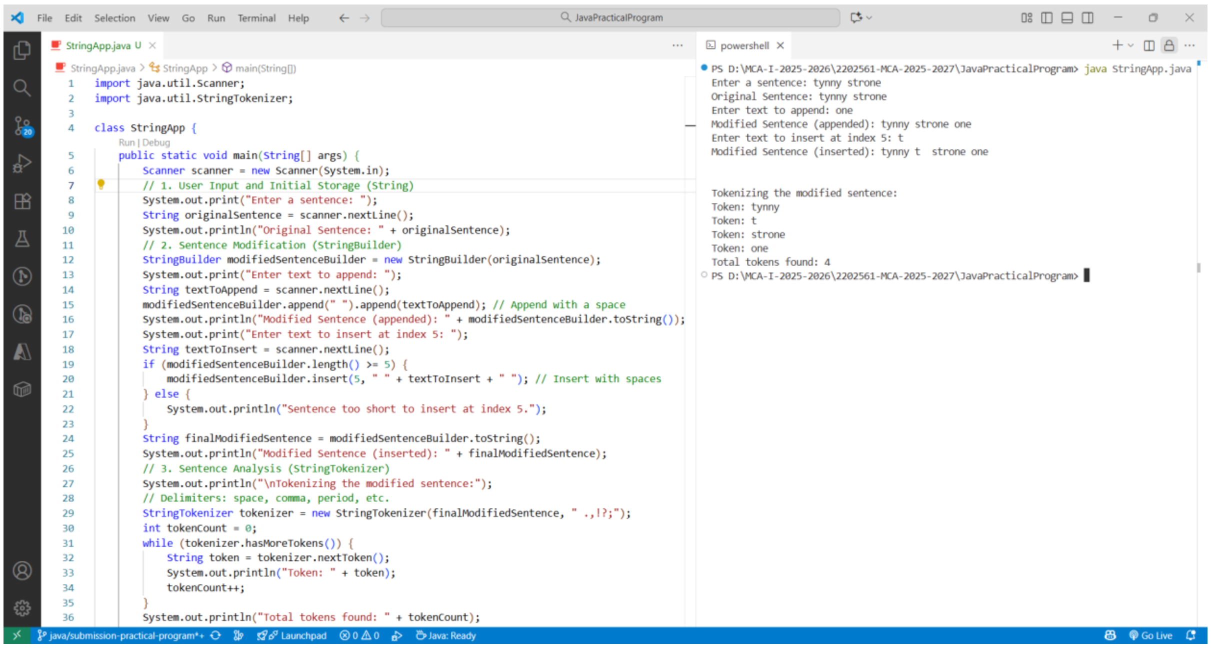
4: Introduction to string and its different operations
Use of Different string methods



```java
public class StrMain {
    public static void main(String[] args) {
        String firstName = "John";
        String lastName = "Doe";
        String fullName = firstName + " " + lastName; // Using '+' operator
        String fullMessage = "Welcome ".concat(fullName); // Using concat() method
        String s1 = "hello";
        String s2 = "hello";
        boolean isEqual1 = s1.equals(s2); // true
        String s = "This is a test.";
        System.out.println("Original: " + s);
        String upper = s.toUpperCase();
        String lower = s.toLowerCase();
        System.out.println("Uppercase: " + upper);
        System.out.println("Lowercase: " + lower);
        System.out.println("Full name " + fullMessage);
        System.out.println("Equal or not " + isEqual1);
        System.out.println("Length of a string s = " + s);
    }
}
```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java StrMain.java
Original: This is a test.
Uppercase: THIS IS A TEST.
Lowercase: this is a test.
Full name Welcome John Doe
Equal or not true
Length of a string s = This is a test.
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

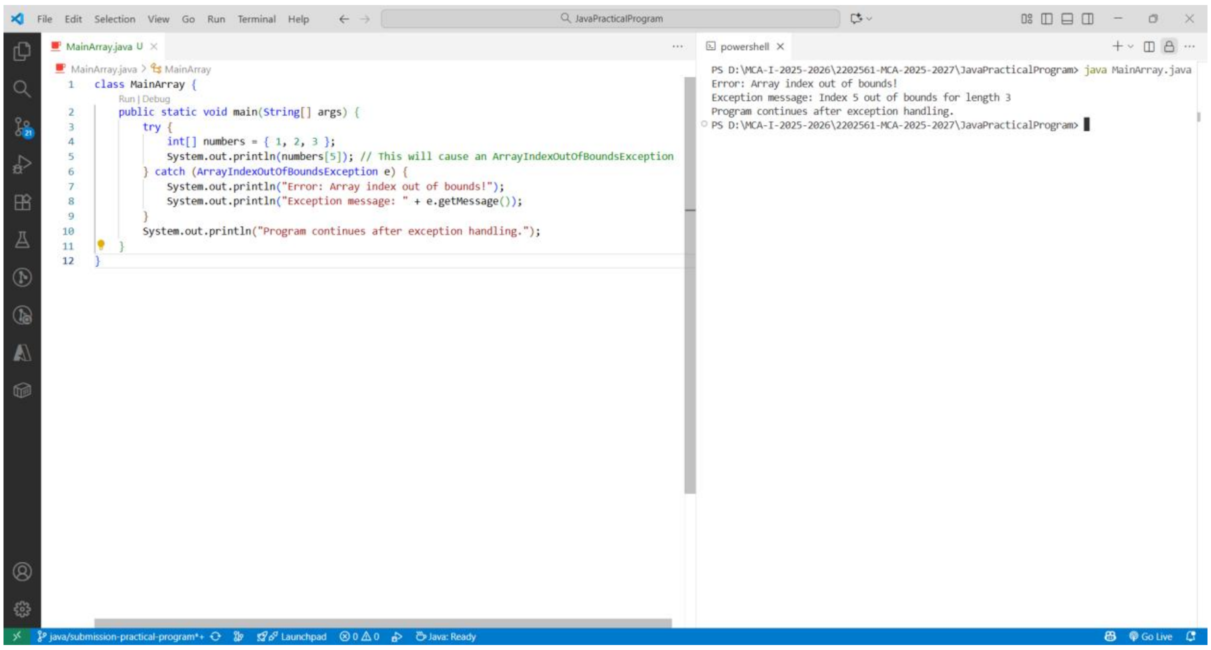
b. Design application using String, StringBuilder, StringTokenizer Experiment


File Edit Selection View Go Run Terminal Help
JavaPracticalProgram
StringApp.java U
StringApp.java > StringApp > main(String[])
Run | Debug
powershell
java/submission-practical-program*+
Launchpad
Java: Ready
Go Live

```java
import java.util.Scanner;
import java.util.StringTokenizer;

class StringApp {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // 1. User Input and Initial Storage (String)
        System.out.print("Enter a sentence: ");
        String originalSentence = scanner.nextLine();
        System.out.println("Original Sentence: " + originalSentence);
        // 2. Sentence Modification (StringBuilder)
        StringBuilder modifiedSentenceBuilder = new StringBuilder(originalSentence);
        System.out.print("Enter text to append: ");
        String textToAppend = scanner.nextLine();
        modifiedSentenceBuilder.append(" ").append(textToAppend); // Append with a space
        System.out.println("Modified Sentence (appended): " + modifiedSentenceBuilder.toString());
        System.out.print("Enter text to insert at index 5: ");
        String textToInsert = scanner.nextLine();
        if (modifiedSentenceBuilder.length() >= 5) {
            modifiedSentenceBuilder.insert(5, " " + textToInsert + " "); // Insert with spaces
        } else {
            System.out.println("Sentence too short to insert at index 5.");
        }
        String finalModifiedSentence = modifiedSentenceBuilder.toString();
        System.out.println("Modified Sentence (inserted): " + finalModifiedSentence);
        // 3. Sentence Analysis (StringTokenizer)
        System.out.println("\nTokenizing the modified sentence:");
        // Delimiters: space, comma, period, etc.
        StringTokenizer tokenizer = new StringTokenizer(finalModifiedSentence, " .,!?;");
        int tokenCount = 0;
        while (tokenizer.hasMoreTokens()) {
            String token = tokenizer.nextToken();
            System.out.println("Token: " + token);
            tokenCount++;
        }
        System.out.println("Total tokens found: " + tokenCount);
```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java StringApp.java
Enter a sentence: tynny strone
Original Sentence: tynny strone
Enter text to append: one
Modified Sentence (appended): tynny strone one
Enter text to insert at index 5: t
Modified Sentence (inserted): tynny t  strone one


Tokenizing the modified sentence:
Token: tynny
Token: t
Token: strone
Token: one
Total tokens found: 4
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

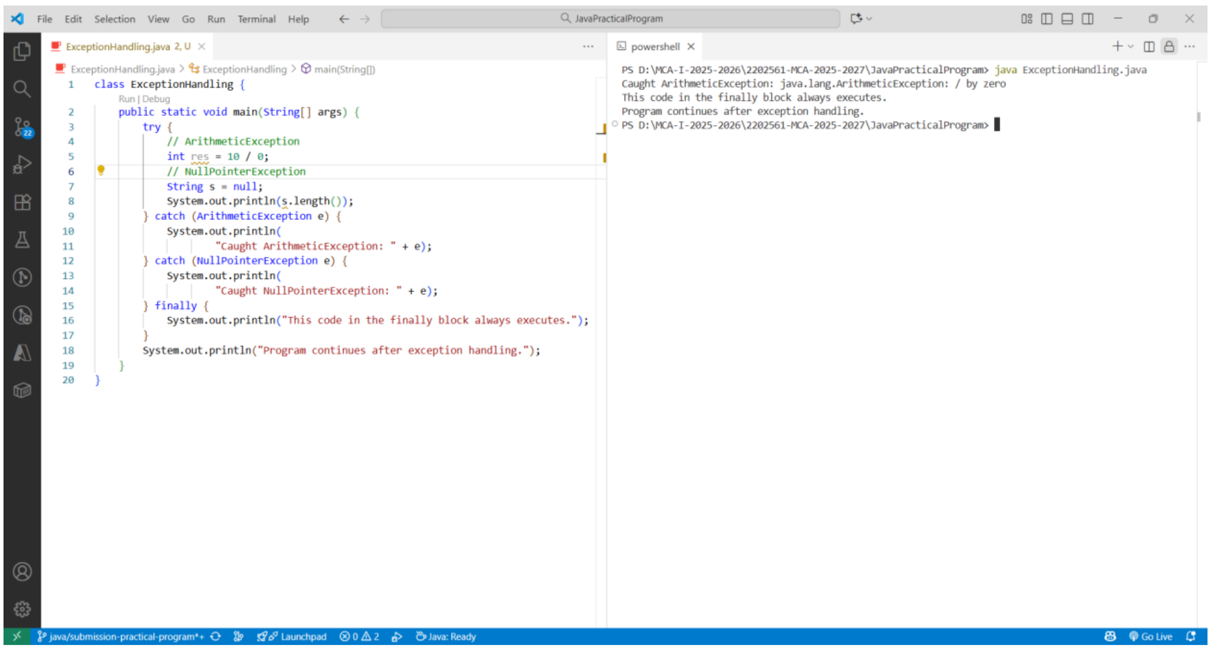
5: Exception Handling -Test any five of standard exception and user Defined Custom Exceptions in java


File Edit Selection View Go Run Terminal Help
JavaPracticalProgram
MainArray.java U
MainArray.java > MainArray
Run | Debug
powershell
java/submission-practical-program*+
Launchpad
Java: Ready
Go Live

```java
class MainArray {
    public static void main(String[] args) {
        try {
            int[] numbers = { 1, 2, 3 };
            System.out.println(numbers[5]); // This will cause an ArrayIndexOutOfBoundsException
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: Array index out of bounds!");
            System.out.println("Exception message: " + e.getMessage());
        }
        System.out.println("Program continues after exception handling.");
    }
}
```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java MainArray.java
Error: Array index out of bounds!
Exception message: Index 5 out of bounds for length 3
Program continues after exception handling.
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

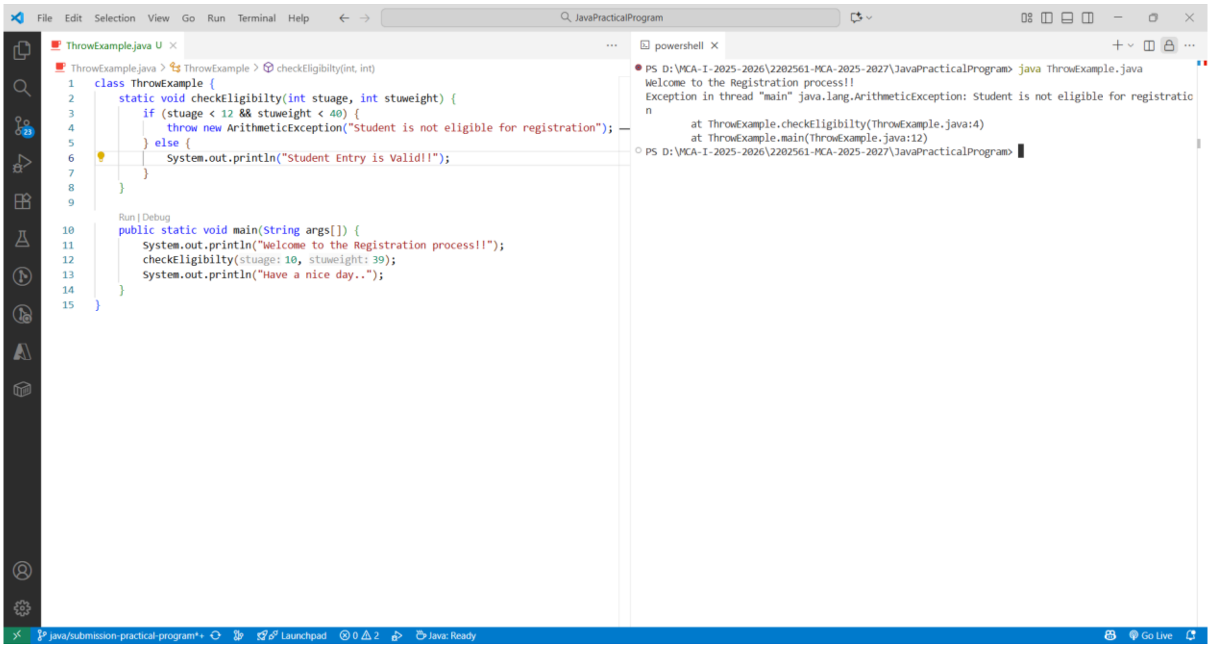
c. Try-catch-finally block with multiple catch statements.


File Edit Selection View Go Run Terminal Help
JavaPracticalProgram
ExceptionHandling.java 2, U
ExceptionHandling.java > ExceptionHandling > main(String[])
Run | Debug
powershell
java/submission-practical-program*+
Launchpad
Java: Ready
Go Live

```java
class ExceptionHandling {
    public static void main(String[] args) {
        try {
            // ArithmeticException
            int res = 10 / 0;
            // NullPointerException
            String s = null;
            System.out.println(s.length());
        } catch (ArithmeticException e) {
            System.out.println(
                    "Caught ArithmeticException: " + e);
        } catch (NullPointerException e) {
            System.out.println(
                    "Caught NullPointerException: " + e);
        } finally {
            System.out.println("This code in the finally block always executes.");
        }
        System.out.println("Program continues after exception handling.");
    }
}
```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java ExceptionHandling.java
Caught ArithmeticException: java.lang.ArithmeticException: / by zero
This code in the finally block always executes.
Program continues after exception handling.
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

d. User-defined exceptions

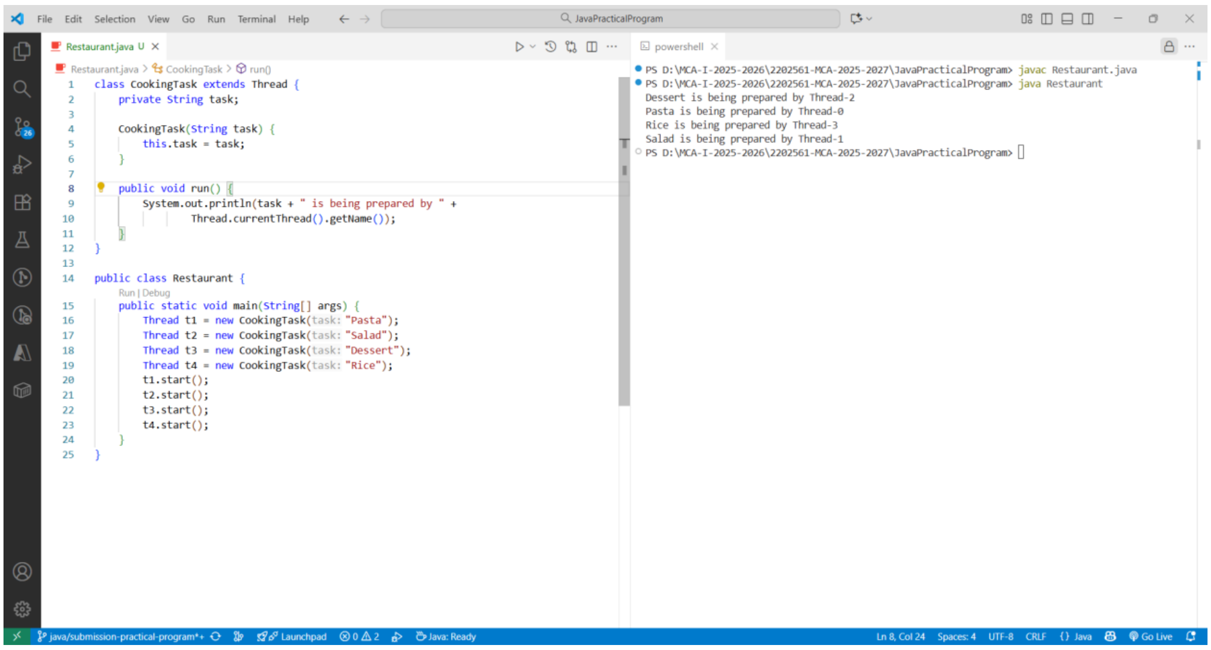

File Edit Selection View Go Run Terminal Help
JavaPracticalProgram
ThrowExample.java U
ThrowExample.java > ThrowExample > checkEligibilty(int, int)
Run | Debug
powershell
java/submission-practical-program*+
Launchpad
Java: Ready
Go Live

```java
class ThrowExample {
    static void checkEligibilty(int stuage, int stuweight) {
        if (stuage < 12 && stuweight < 40) {
            throw new ArithmeticException("Student is not eligible for registration");
        } else {
            System.out.println("Student Entry is Valid!!");
        }
    }

    public static void main(String args[]) {
        System.out.println("Welcome to the Registration process!!");
        checkEligibilty(stuage: 10, stuweight: 39);
        System.out.println("Have a nice day..");
    }
}
```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java ThrowExample.java
Welcome to the Registration process!!
Exception in thread "main" java.lang.ArithmeticException: Student is not eligible for registratio
n
        at ThrowExample.checkEligibilty(ThrowExample.java:4)
        at ThrowExample.main(ThrowExample.java:12)
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

Experiment 6: Multithreading

Implement a Thread - 1. Extending the Thread class

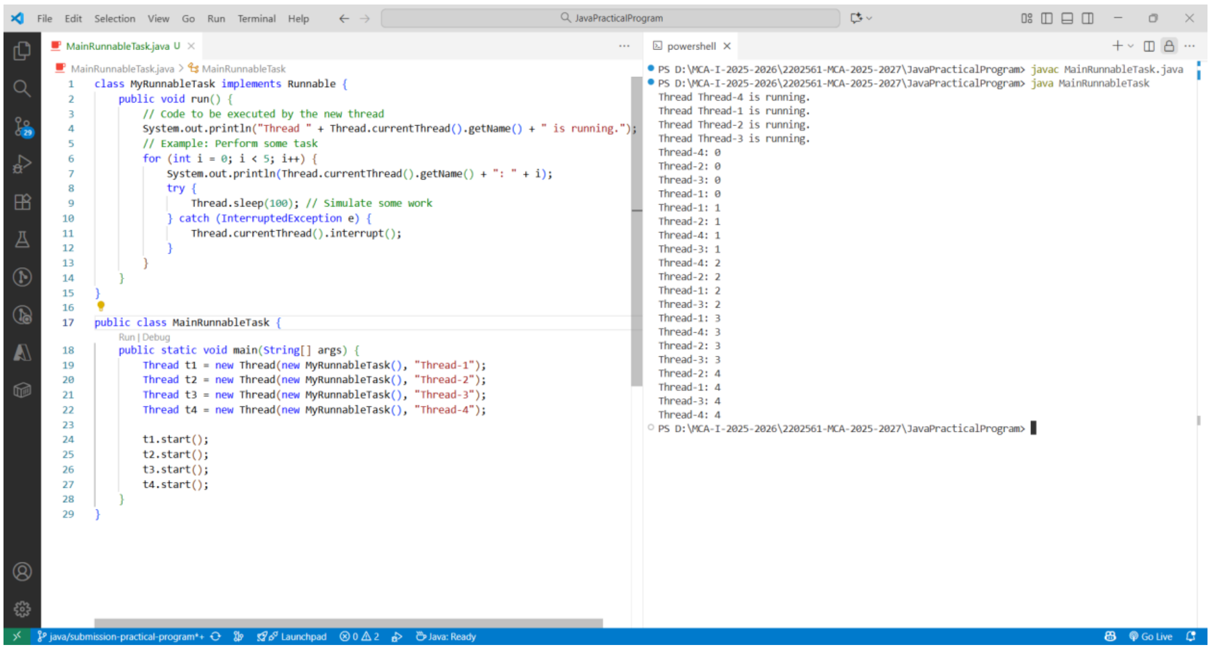


Restaurant.java > CookingTask > run()

```java
class CookingTask extends Thread {
    private String task;

    CookingTask(String task) {
        this.task = task;
    }

    public void run() {
        System.out.println(task + " is being prepared by " +
                Thread.currentThread().getName());
    }
}

public class Restaurant {
    Run | Debug
    public static void main(String[] args) {
        Thread t1 = new CookingTask(task: "Pasta");
        Thread t2 = new CookingTask(task: "Salad");
        Thread t3 = new CookingTask(task: "Dessert");
        Thread t4 = new CookingTask(task: "Rice");
        t1.start();
        t2.start();
        t3.start();
        t4.start();
    }
}
```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> javac Restaurant.java
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java Restaurant
Dessert is being prepared by Thread-2
Pasta is being prepared by Thread-0
Rice is being prepared by Thread-3
Salad is being prepared by Thread-1
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

b. Implement Thread using the Runnable Interface

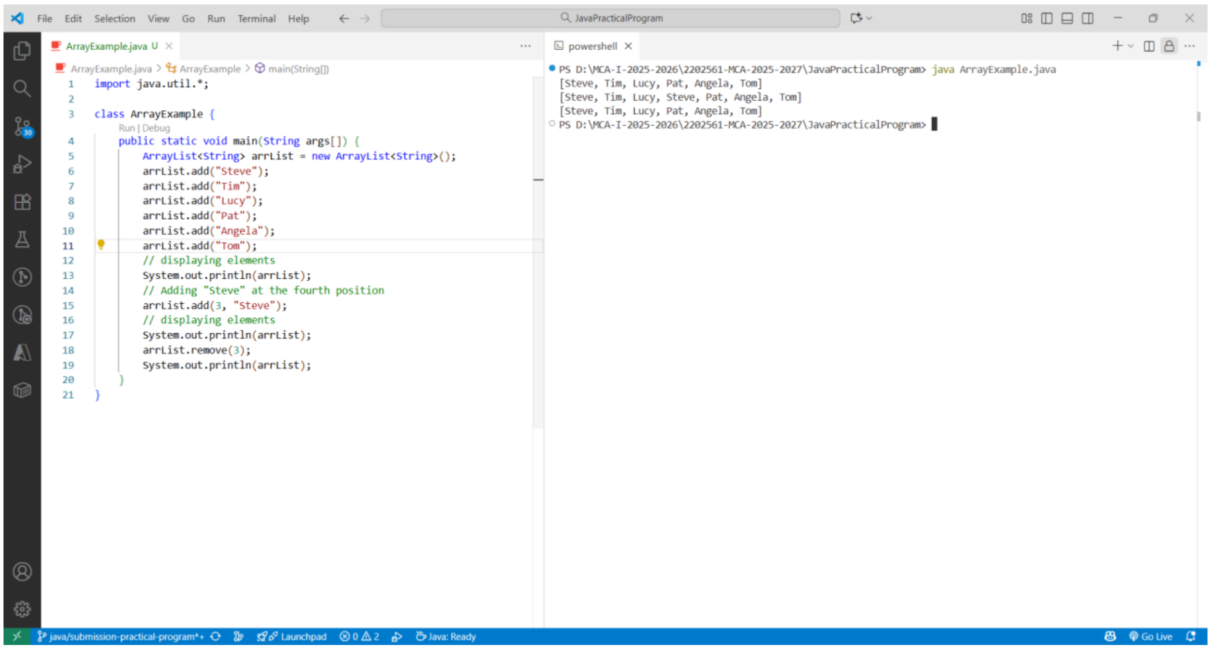


MainRunnableTask.java > MainRunnableTask

```java
class MyRunnableTask implements Runnable {
    public void run() {
        // Code to be executed by the new thread
        System.out.println("Thread " + Thread.currentThread().getName() + " is running.");
        // Example: Perform some task
        for (int i = 0; i < 5; i++) {
            System.out.println(Thread.currentThread().getName() + ": " + i);
            try {
                Thread.sleep(100); // Simulate some work
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
    }
}

public class MainRunnableTask {
    Run | Debug
    public static void main(String[] args) {
        Thread t1 = new Thread(new MyRunnableTask(), "Thread-1");
        Thread t2 = new Thread(new MyRunnableTask(), "Thread-2");
        Thread t3 = new Thread(new MyRunnableTask(), "Thread-3");
        Thread t4 = new Thread(new MyRunnableTask(), "Thread-4");

        t1.start();
        t2.start();
        t3.start();
        t4.start();
    }
}
```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> javac MainRunnableTask.java
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java MainRunnableTask
Thread Thread-4 is running.
Thread Thread-1 is running.
Thread Thread-2 is running.
Thread Thread-3 is running.
Thread-4: 0
Thread-2: 0
Thread-3: 0
Thread-1: 0
Thread-1: 1
Thread-2: 1
Thread-4: 1
Thread-3: 1
Thread-4: 2
Thread-2: 2
Thread-1: 2
Thread-3: 2
Thread-1: 3
Thread-4: 3
Thread-2: 3
Thread-3: 3
Thread-2: 4
Thread-1: 4
Thread-3: 4
Thread-4: 4
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

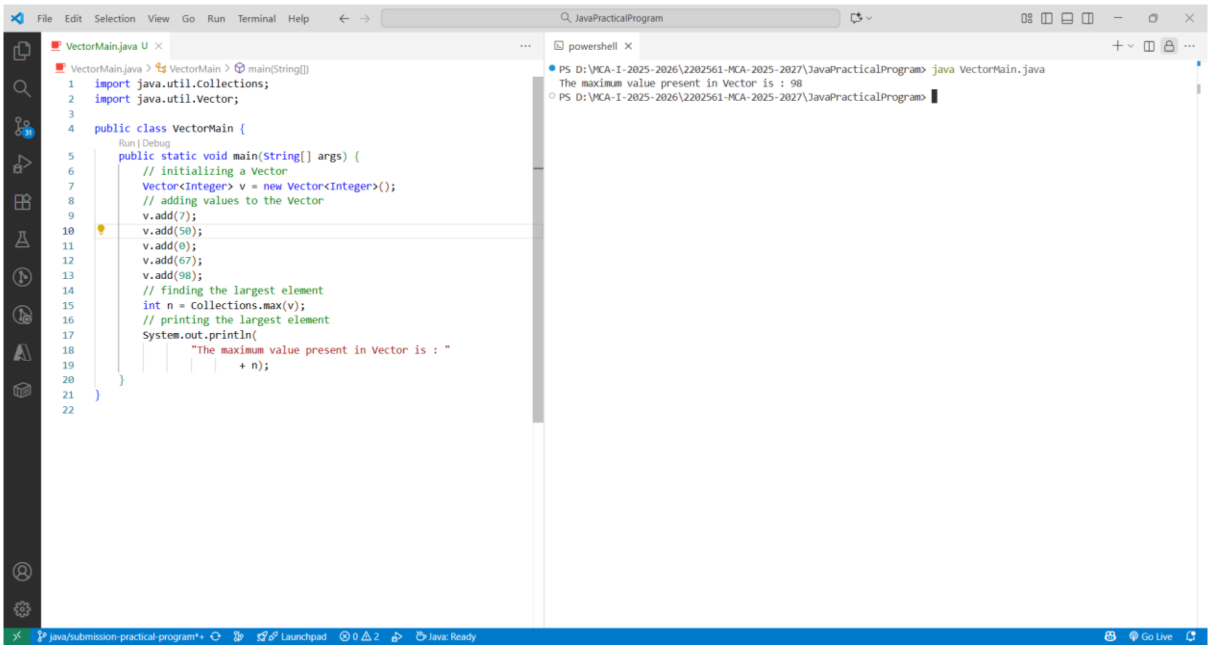
Experiment 7: Collections
Add and remove element using ArrayList


File Edit Selection View Go Run Terminal Help
JavaPracticalProgram
ArrayExample.java U
powershell
ArrayExample.java > ArrayExample > main(String[])
Run | Debug
java/submission-practical-program*+
Launchpad
Java: Ready
Go Live

```java
import java.util.*;

class ArrayExample {
    public static void main(String args[]) {
        ArrayList<String> arrList = new ArrayList<String>();
        arrList.add("Steve");
        arrList.add("Tim");
        arrList.add("Lucy");
        arrList.add("Pat");
        arrList.add("Angela");
        arrList.add("Tom");
        // displaying elements
        System.out.println(arrList);
        // Adding "Steve" at the fourth position
        arrList.add(3, "Steve");
        // displaying elements
        System.out.println(arrList);
        arrList.remove(3);
        System.out.println(arrList);
    }
}
```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java ArrayExample.java
[Steve, Tim, Lucy, Pat, Angela, Tom]
[Steve, Tim, Lucy, Steve, Pat, Angela, Tom]
[Steve, Tim, Lucy, Pat, Angela, Tom]
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

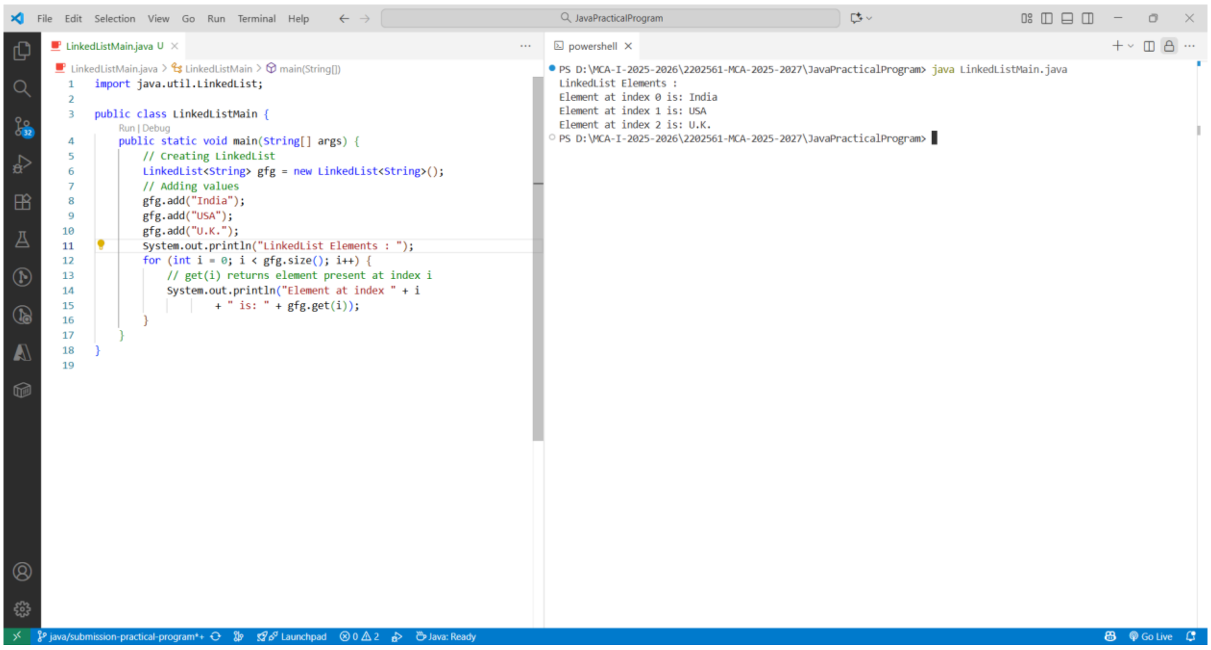
b. To find maximum element vector using predefined method


File Edit Selection View Go Run Terminal Help
JavaPracticalProgram
VectorMain.java U
powershell
VectorMain.java > VectorMain > main(String[])
Run | Debug
java/submission-practical-program*+
Launchpad
Java: Ready
Go Live

```java
import java.util.Collections;
import java.util.Vector;

public class VectorMain {
    public static void main(String[] args) {
        // initializing a Vector
        Vector<Integer> v = new Vector<Integer>();
        // adding values to the Vector
        v.add(7);
        v.add(50);
        v.add(0);
        v.add(67);
        v.add(98);
        // finding the largest element
        int n = Collections.max(v);
        // printing the largest element
        System.out.println(
                "The maximum value present in Vector is : "
                        + n);
    }
}

```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java VectorMain.java
The maximum value present in Vector is : 98
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

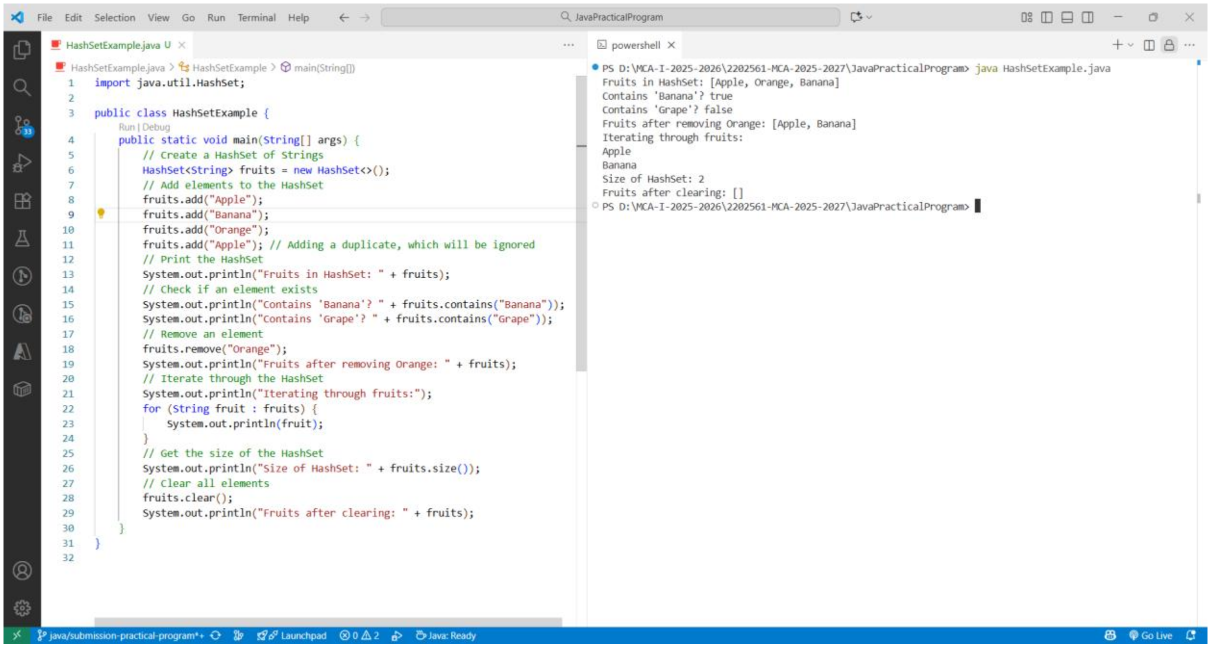
c. to get the elements of Linkedlist



```java
import java.util.LinkedList;

public class LinkedListMain {
    public static void main(String[] args) {
        // Creating LinkedList
        LinkedList<String> gfg = new LinkedList<String>();
        // Adding values
        gfg.add("India");
        gfg.add("USA");
        gfg.add("U.K.");
        System.out.println("LinkedList Elements : ");
        for (int i = 0; i < gfg.size(); i++) {
            // get(i) returns element present at index i
            System.out.println("Element at index " + i
                    + " is: " + gfg.get(i));
        }
    }
}
```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java LinkedListMain.java
LinkedList Elements :
Element at index 0 is: India
Element at index 1 is: USA
Element at index 2 is: U.K.
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```

d. to get the elements of HashSet



```java
import java.util.HashSet;

public class HashSetExample {
    public static void main(String[] args) {
        // Create a HashSet of Strings
        HashSet<String> fruits = new HashSet<>();
        // Add elements to the HashSet
        fruits.add("Apple");
        fruits.add("Banana");
        fruits.add("Orange");
        fruits.add("Apple"); // Adding a duplicate, which will be ignored
        // Print the HashSet
        System.out.println("Fruits in HashSet: " + fruits);
        // Check if an element exists
        System.out.println("Contains 'Banana'? " + fruits.contains("Banana"));
        System.out.println("Contains 'Grape'? " + fruits.contains("Grape"));
        // Remove an element
        fruits.remove("Orange");
        System.out.println("Fruits after removing Orange: " + fruits);
        // Iterate through the HashSet
        System.out.println("Iterating through fruits:");
        for (String fruit : fruits) {
            System.out.println(fruit);
        }
        // Get the size of the HashSet
        System.out.println("Size of HashSet: " + fruits.size());
        // Clear all elements
        fruits.clear();
        System.out.println("Fruits after clearing: " + fruits);
    }
}
```

```
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram> java HashSetExample.java
Fruits in HashSet: [Apple, Orange, Banana]
Contains 'Banana'? true
Contains 'Grape'? false
Fruits after removing Orange: [Apple, Banana]
Iterating through fruits:
Apple
Banana
Size of HashSet: 2
Fruits after clearing: []
PS D:\MCA-I-2025-2026\2202561-MCA-2025-2027\JavaPracticalProgram>
```