Object Oriented Programming Language Using Java Lab.

Java Practical

#Questions for Programs:

1 Write a program to find the average and sum of the N numbers using Command line argument.

2 Write a program to demonstrate type casting.

3. Write a program to generate prime numbers between 1 & given number

4. Write a program to design a class account using the inheritance and static members which show all functions of a bank (Withdrawl, deposit)

5. Write a program to create a simple class to find out the area and perimeter of rectangle using super and this keyword.

6. Write a program to find the factorial of a given number using recursion.

7. Write a program to design a class using abstract methods and abstract classes.

8. Write a program to count the number of objects created for a class

using static member function

9. Write a program to demonstrate the use of function overloading.

10. Write a program to demonstrate the use of multiple inheritance.

11. Write a program that show the partial implementation of Interface

12. Write a program to design a string class that perform string method(Equal, Reverse the string, change case).

13. Write a program to handle the exception using try and multiple catch blocks.

14. Write a program to create a package that access the member of External class as well as same package.

15. Write a program that import the user define package and access the Member variable of classes that contained by package.

16. Write a program to handle the user defined exception using throw keyword.

17. Write a program to create a class component that shows controls and event handling on that controls.(mathcalc).

18. Write a program to draw the line, Rectangle, oval, text using the graphics method.

19. Write a program to create a menu using the frame.

20. Write a program to create a menu using the frame.

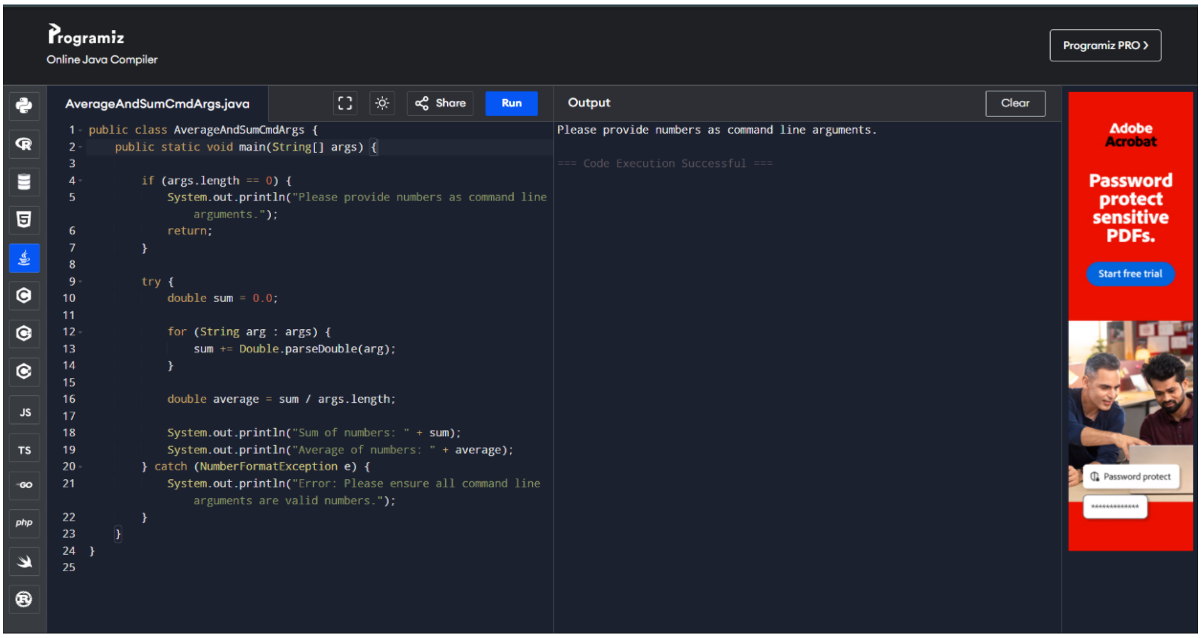
21. Write a program to implement the flow layout and border layout.

22. Write a program to imp Write a program to create a dialogbox. element the gridLayout, cardLayout.

23. Write a program to implement the gridLayout, cardLayout.

24. Write a program to create Frame that display the student information

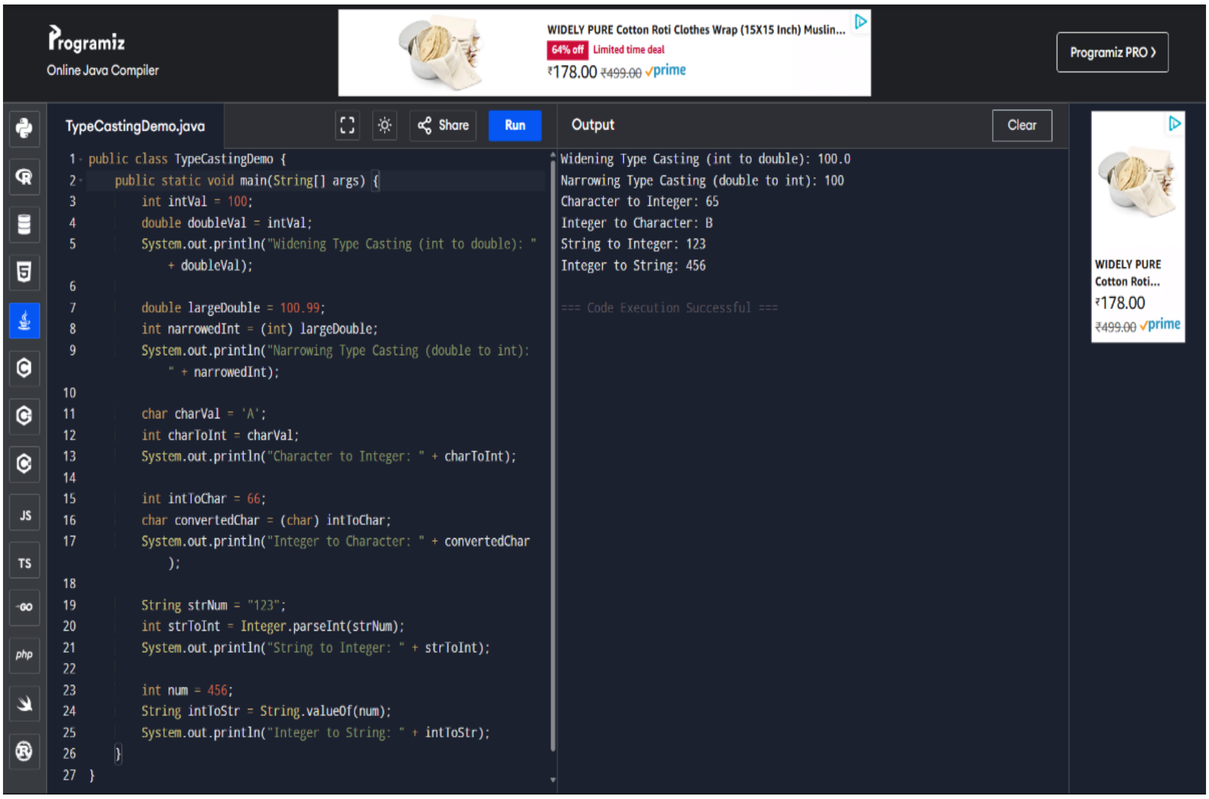
1. Write a program to find the average and sum of the N numbers using Command line argument.



```java
public class AverageAndSumCmdArgs {
    public static void main(String[] args) {

        if (args.length == 0) {
            System.out.println("Please provide numbers as command line arguments.");
            return;
        }

        try {
            double sum = 0.0;

            for (String arg : args) {
                sum += Double.parseDouble(arg);
            }

            double average = sum / args.length;

            System.out.println("Sum of numbers: " + sum);
            System.out.println("Average of numbers: " + average);
        } catch (NumberFormatException e) {
            System.out.println("Error: Please ensure all command line arguments are valid numbers.");
        }
    }
}
```

```
Please provide numbers as command line arguments.

=== Code Execution Successful ===
```

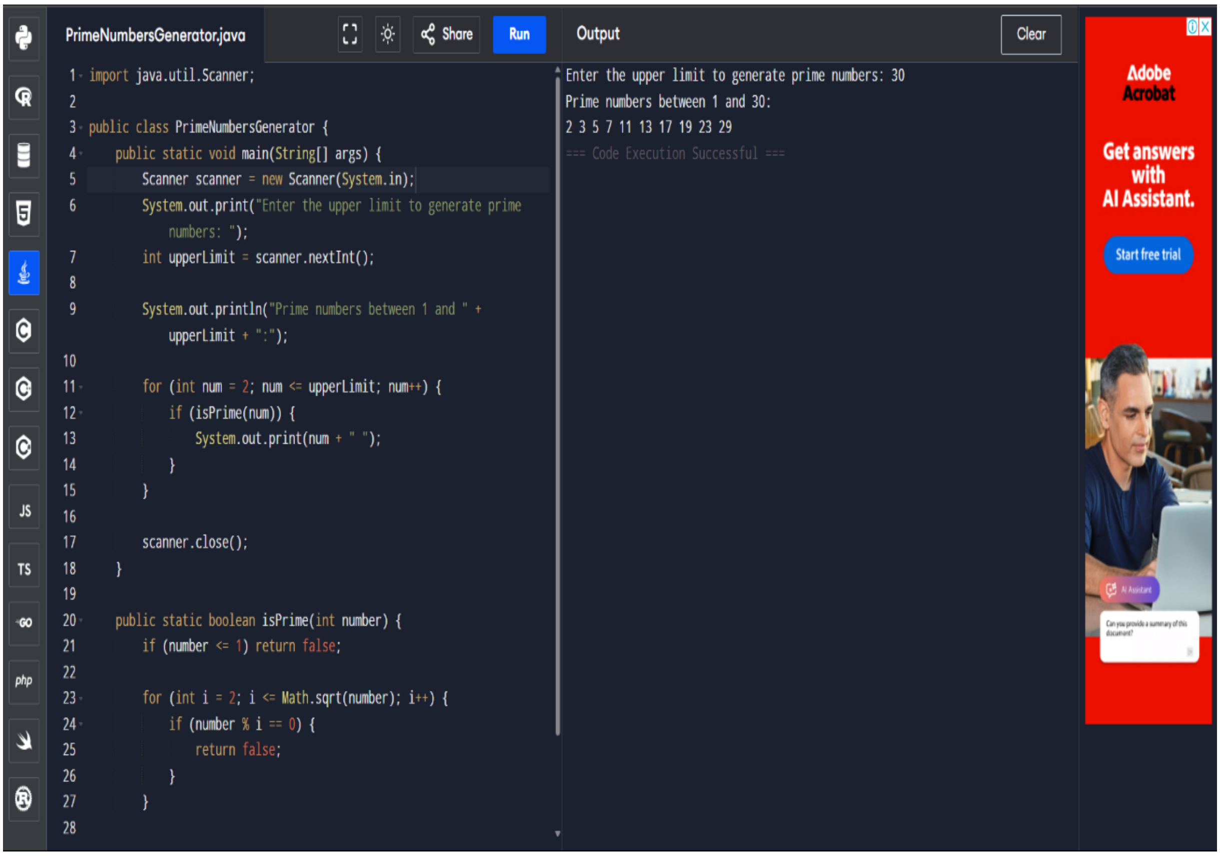
2. Write a program to demonstrate type casting.



```java
public class TypeCastingDemo {
    public static void main(String[] args) {
        int intVal = 100;
        double doubleVal = intVal;
        System.out.println("Widening Type Casting (int to double): " + doubleVal);

        double largeDouble = 100.99;
        int narrowedInt = (int) largeDouble;
        System.out.println("Narrowing Type Casting (double to int): " + narrowedInt);

        char charVal = 'A';
        int charToInt = charVal;
        System.out.println("Character to Integer: " + charToInt);

        int intToChar = 66;
        char convertedChar = (char) intToChar;
        System.out.println("Integer to Character: " + convertedChar);

        String strNum = "123";
        int strToInt = Integer.parseInt(strNum);
        System.out.println("String to Integer: " + strToInt);

        int num = 456;
        String intToStr = String.valueOf(num);
        System.out.println("Integer to String: " + intToStr);
    }
}
```

```
Widening Type Casting (int to double): 100.0
Narrowing Type Casting (double to int): 100
Character to Integer: 65
Integer to Character: B
String to Integer: 123
Integer to String: 456

=== Code Execution Successful ===
```

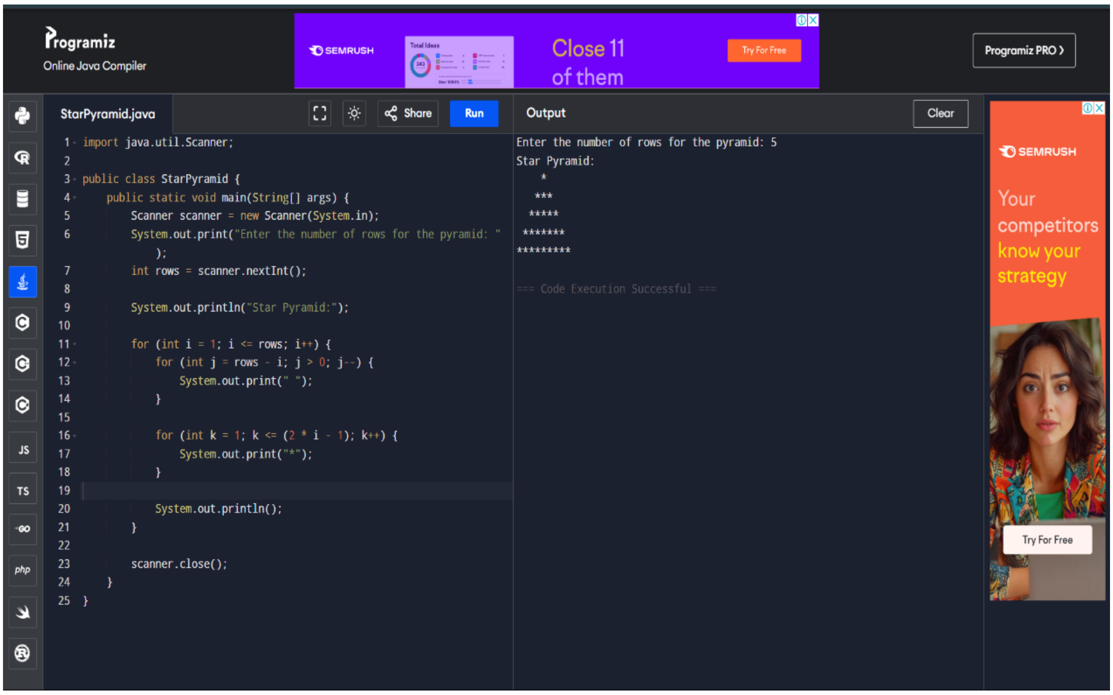
3. Write a program to generate prime numbers between 1 & given number.



```java
import java.util.Scanner;

public class PrimeNumbersGenerator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the upper limit to generate prime
            numbers: ");
        int upperLimit = scanner.nextInt();

        System.out.println("Prime numbers between 1 and " +
            upperLimit + ":");

        for (int num = 2; num <= upperLimit; num++) {
            if (isPrime(num)) {
                System.out.print(num + " ");
            }
        }

        scanner.close();
    }

    public static boolean isPrime(int number) {
        if (number <= 1) return false;

        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                return false;
            }
        }

```

```
Enter the upper limit to generate prime numbers: 30
Prime numbers between 1 and 30:
2 3 5 7 11 13 17 19 23 29
=== Code Execution Successful ===
```

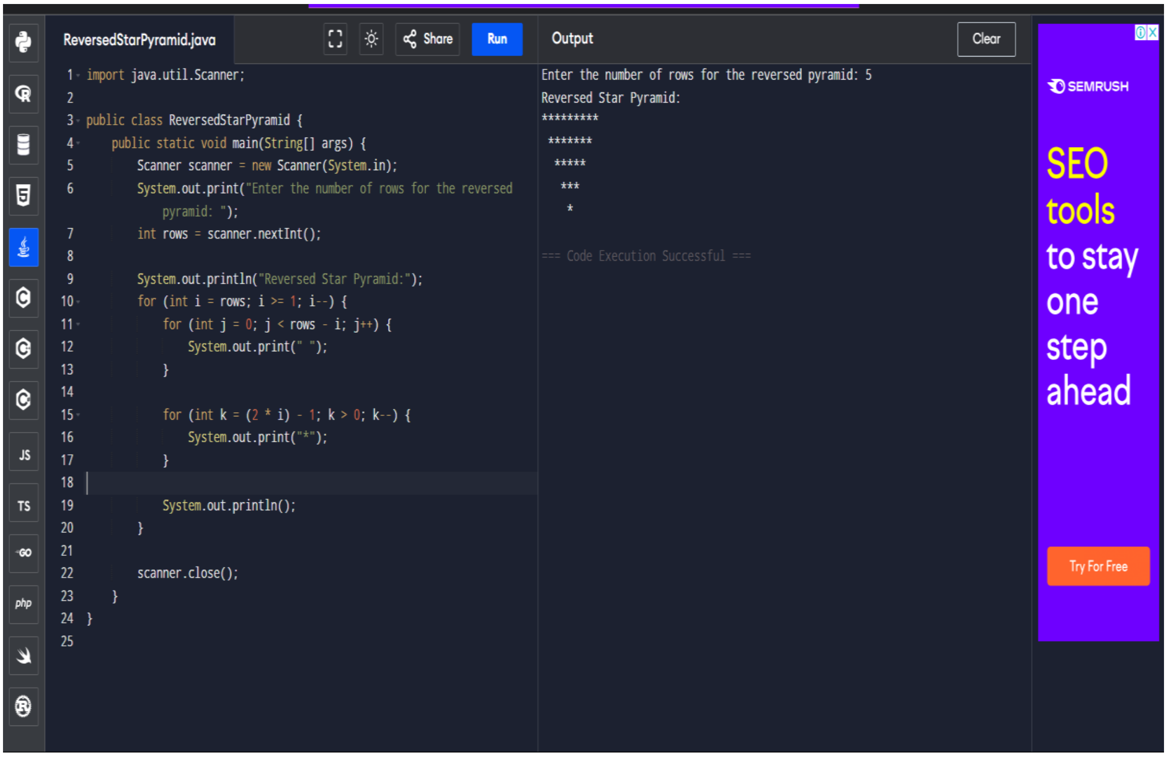
4. Write a program to generate pyramid of stars using nested for loops



```java
import java.util.Scanner;

public class StarPyramid {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of rows for the pyramid: "
            );
        int rows = scanner.nextInt();

        System.out.println("Star Pyramid:");

        for (int i = 1; i <= rows; i++) {
            for (int j = rows - i; j > 0; j--) {
                System.out.print(" ");
            }

            for (int k = 1; k <= (2 * i - 1); k++) {
                System.out.print("*");
            }

            System.out.println();
        }

        scanner.close();
    }
}
```

```
Enter the number of rows for the pyramid: 5
Star Pyramid:
    *
   ***
  *****
 *******
*********


=== Code Execution Successful ===
```

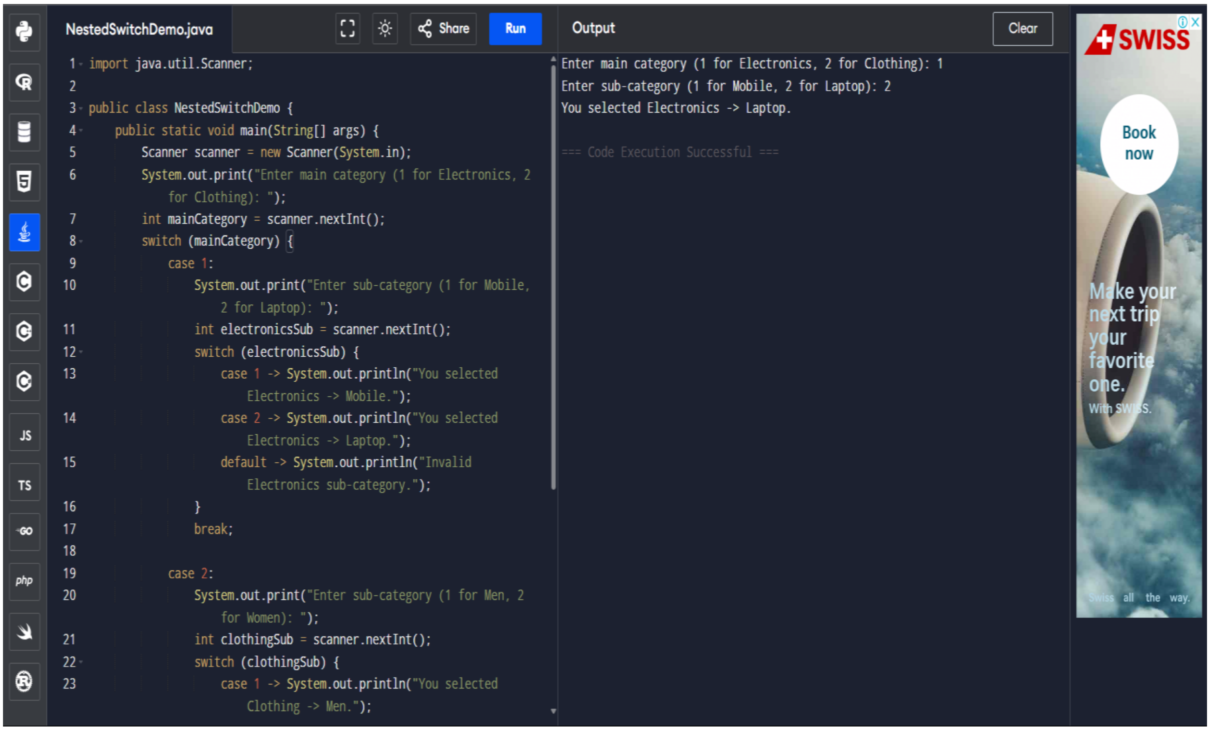
5. Write a program to reversed pyramid using for loops & decrement operator.



```java
import java.util.Scanner;

public class ReversedStarPyramid {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of rows for the reversed pyramid: ");
        int rows = scanner.nextInt();

        System.out.println("Reversed Star Pyramid:");
        for (int i = rows; i >= 1; i--) {
            for (int j = 0; j < rows - i; j++) {
                System.out.print(" ");
            }

            for (int k = (2 * i) - 1; k > 0; k--) {
                System.out.print("*");
            }

            System.out.println();
        }

        scanner.close();
    }
}
```

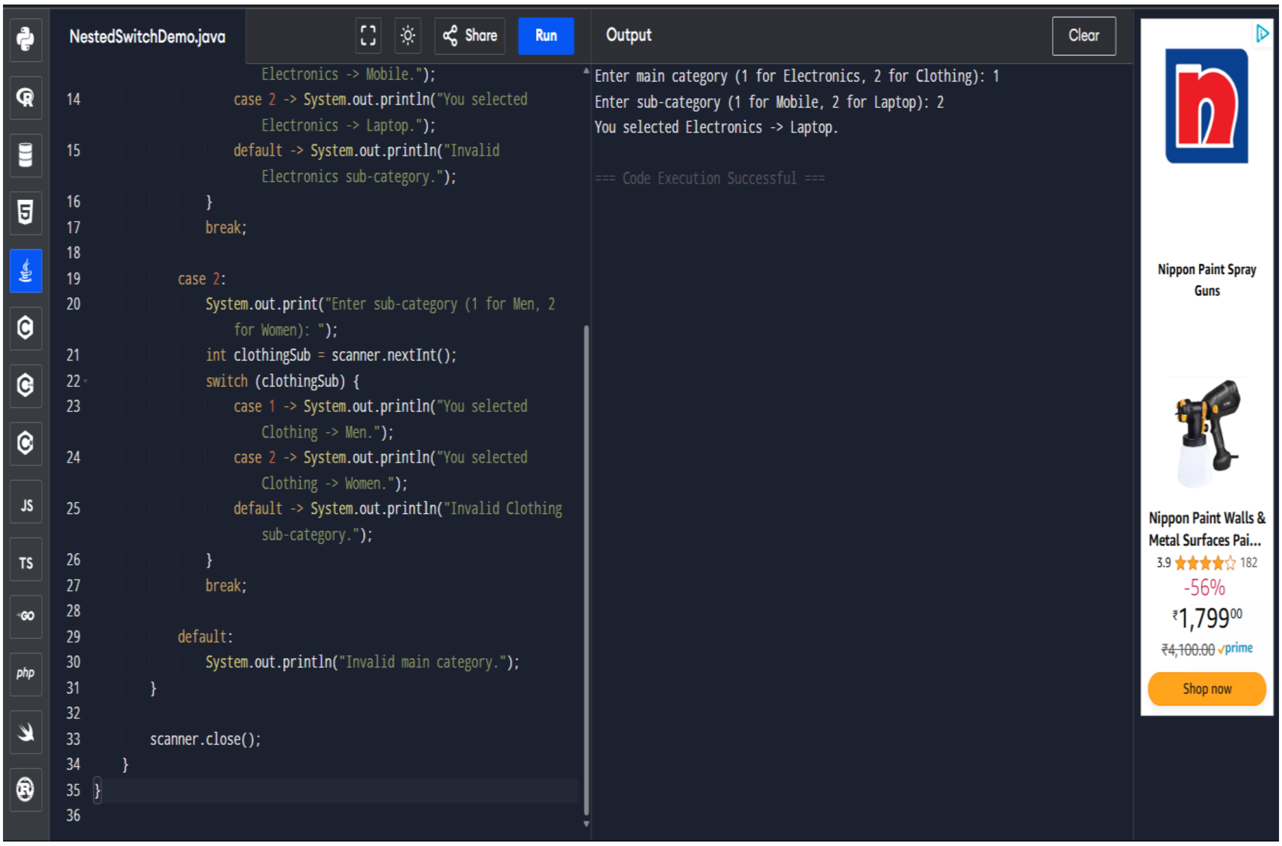
```
Enter the number of rows for the reversed pyramid: 5
Reversed Star Pyramid:
*********
 *******
  *****
   ***
    *


=== Code Execution Successful ===
```

6. Write a program for demonstrate Nested Switch.



```java
import java.util.Scanner;

public class NestedSwitchDemo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter main category (1 for Electronics, 2 for Clothing): ");
        int mainCategory = scanner.nextInt();
        switch (mainCategory) {
            case 1:
                System.out.print("Enter sub-category (1 for Mobile, 2 for Laptop): ");
                int electronicsSub = scanner.nextInt();
                switch (electronicsSub) {
                    case 1 -> System.out.println("You selected Electronics -> Mobile.");
                    case 2 -> System.out.println("You selected Electronics -> Laptop.");
                    default -> System.out.println("Invalid Electronics sub-category.");
                }
                break;

            case 2:
                System.out.print("Enter sub-category (1 for Men, 2 for Women): ");
                int clothingSub = scanner.nextInt();
                switch (clothingSub) {
                    case 1 -> System.out.println("You selected Clothing -> Men.");
```

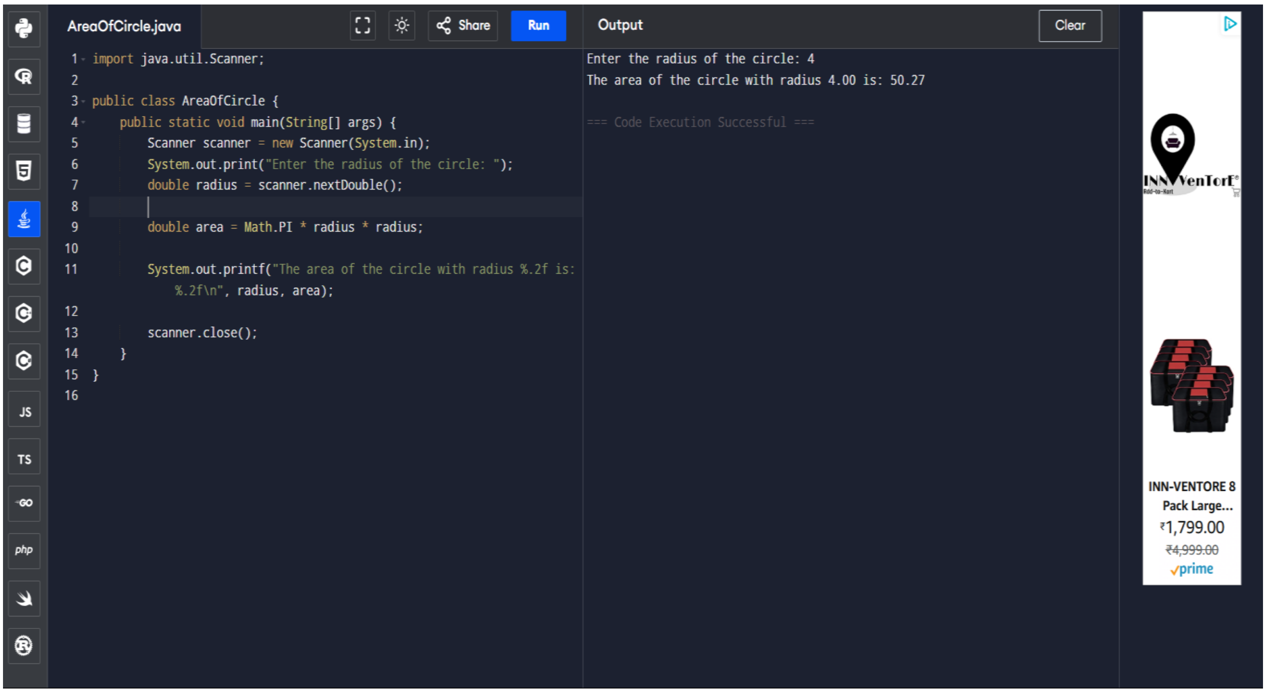
```
Enter main category (1 for Electronics, 2 for Clothing): 1
Enter sub-category (1 for Mobile, 2 for Laptop): 2
You selected Electronics -> Laptop.

=== Code Execution Successful ===
```



```
                        Electronics -> Mobile.");
                    case 2 -> System.out.println("You selected
                        Electronics -> Laptop.");
                    default -> System.out.println("Invalid
                        Electronics sub-category.");
                }
                break;

            case 2:
                System.out.print("Enter sub-category (1 for Men, 2
                    for Women): ");
                int clothingSub = scanner.nextInt();
                switch (clothingSub) {
                    case 1 -> System.out.println("You selected
                        Clothing -> Men.");
                    case 2 -> System.out.println("You selected
                        Clothing -> Women.");
                    default -> System.out.println("Invalid Clothing
                        sub-category.");
                }
                break;

            default:
                System.out.println("Invalid main category.");
        }

        scanner.close();
    }
}
```

```
Enter main category (1 for Electronics, 2 for Clothing): 1
Enter sub-category (1 for Mobile, 2 for Laptop): 2
You selected Electronics -> Laptop.

=== Code Execution Successful ===
```

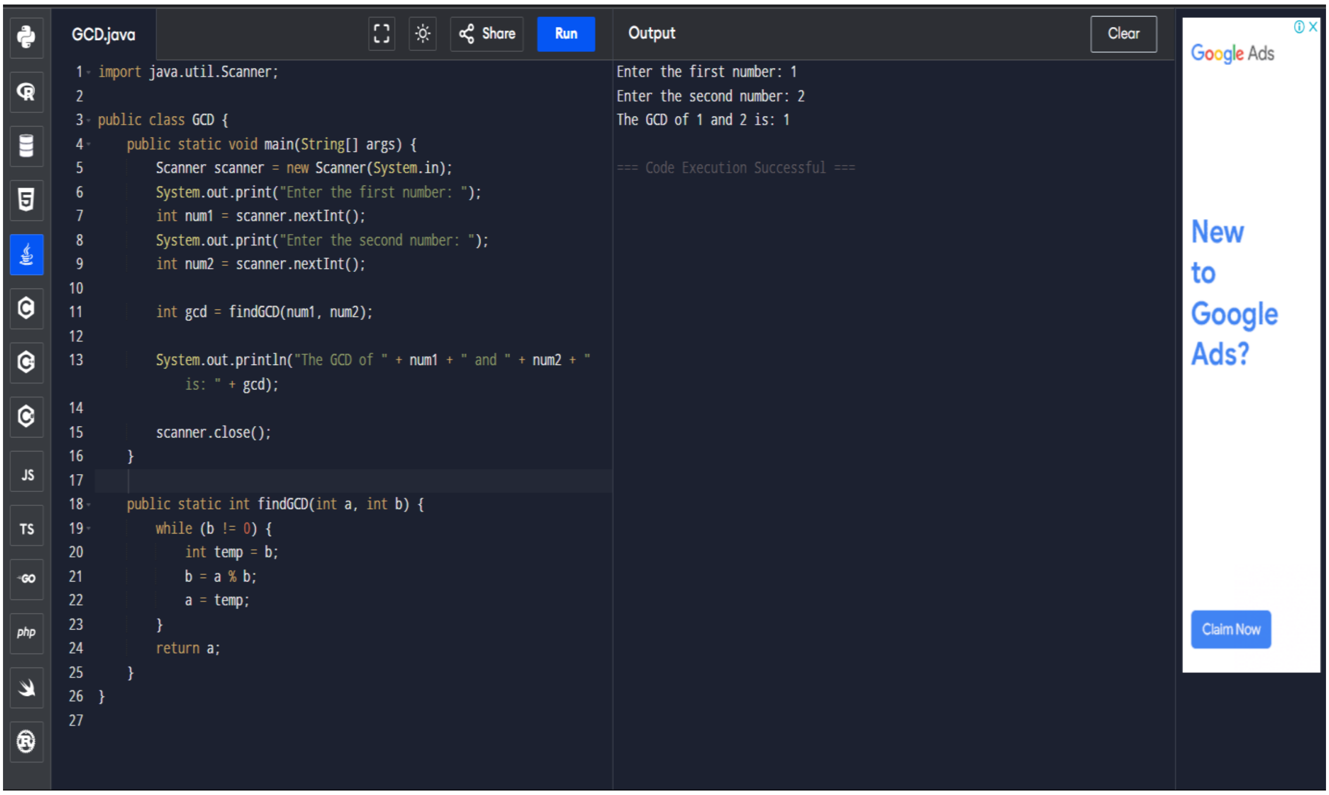
7. Write a program to calculate area of a circle using radius.



```java
import java.util.Scanner;

public class AreaOfCircle {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the radius of the circle: ");
        double radius = scanner.nextDouble();
        
        double area = Math.PI * radius * radius;

        System.out.printf("The area of the circle with radius %.2f is: %.2f\n", radius, area);

        scanner.close();
    }
}
```

```
Enter the radius of the circle: 4
The area of the circle with radius 4.00 is: 50.27

=== Code Execution Successful ===
```

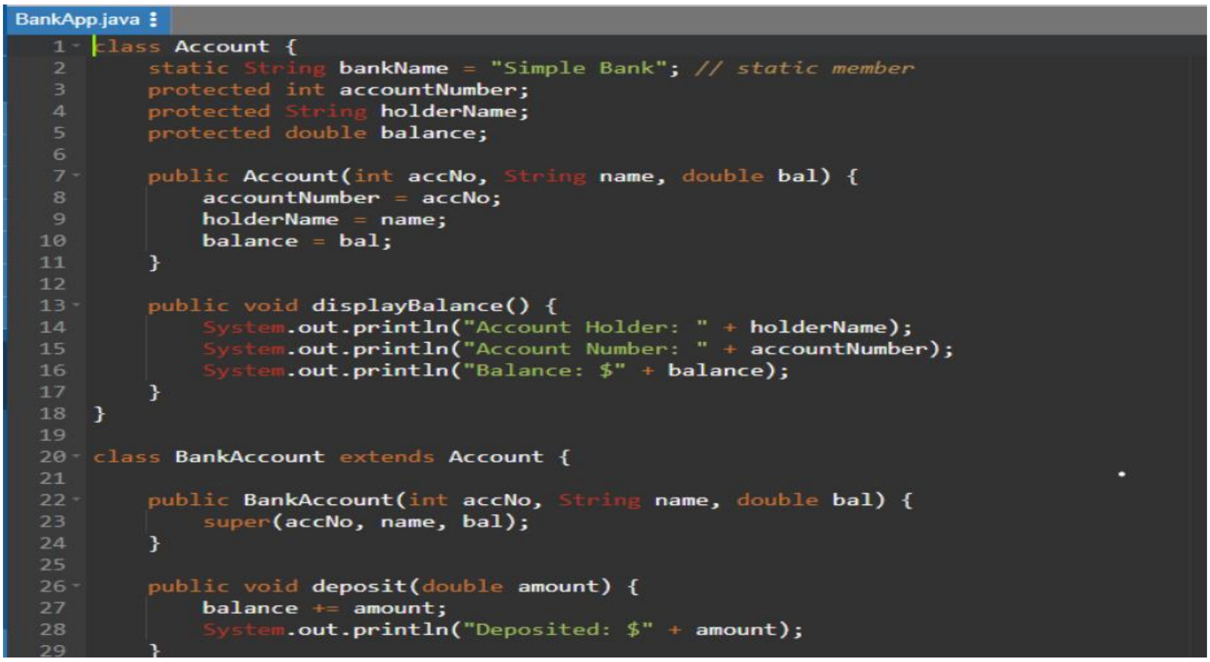
8. Write a program to find G.C.D of the number.



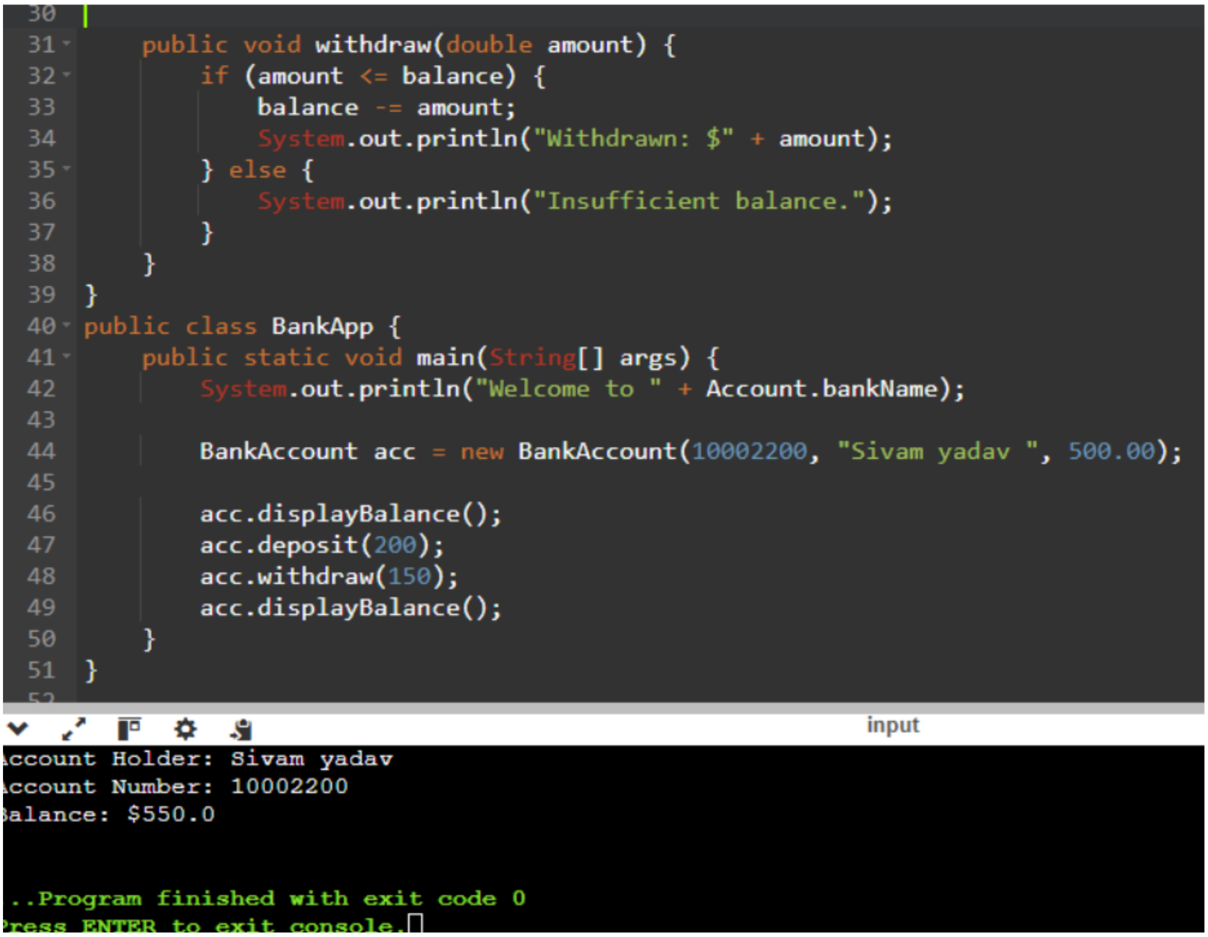
```java
import java.util.Scanner;

public class GCD {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the first number: ");
        int num1 = scanner.nextInt();
        System.out.print("Enter the second number: ");
        int num2 = scanner.nextInt();

        int gcd = findGCD(num1, num2);

        System.out.println("The GCD of " + num1 + " and " + num2 + " is: " + gcd);

        scanner.close();
    }

    public static int findGCD(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }
}
```

```
Enter the first number: 1
Enter the second number: 2
The GCD of 1 and 2 is: 1

=== Code Execution Successful ===
```

9. Write a program to design a class account using the inheritance and static members which show all functions of a bank (Withdrawal, deposit).



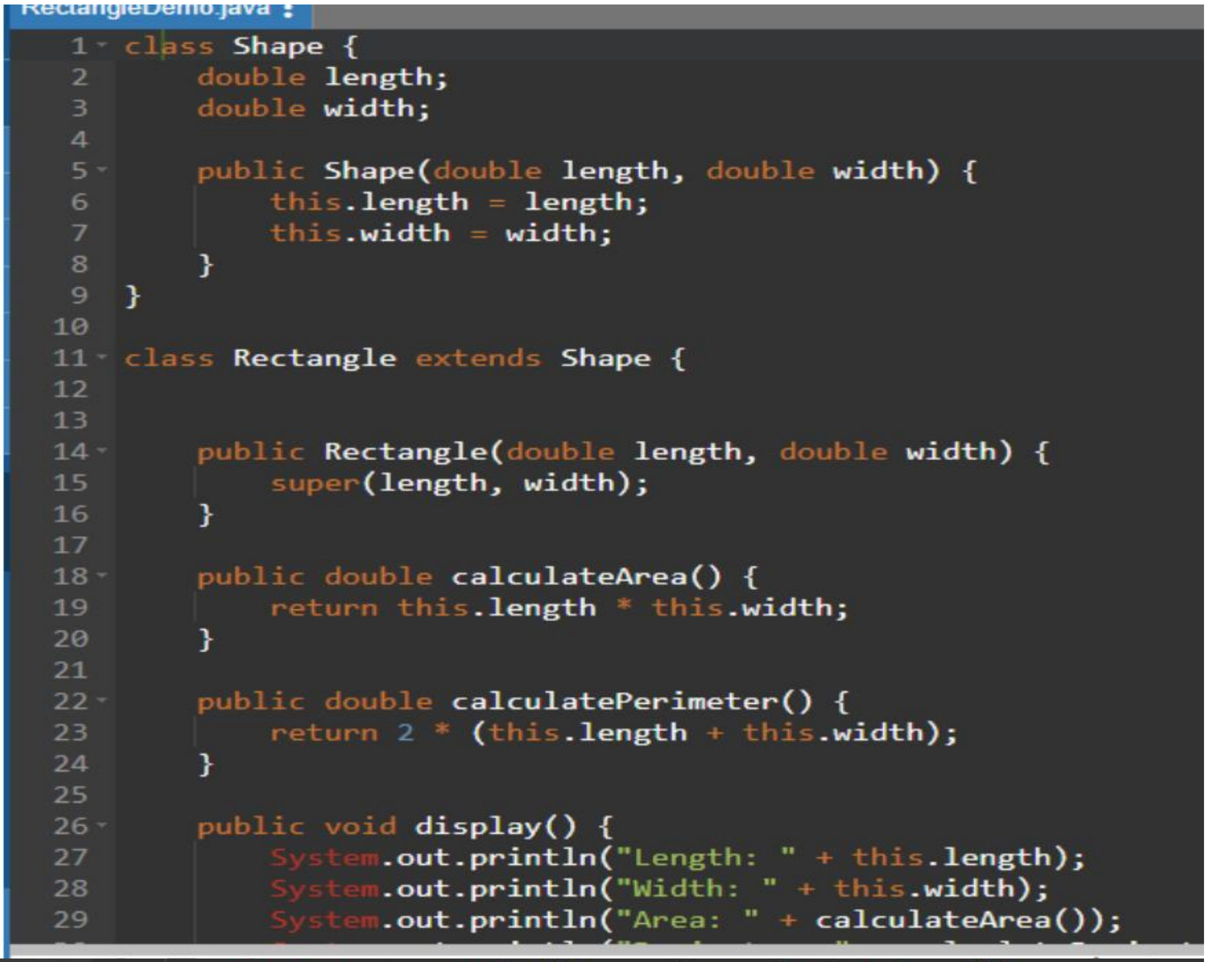
```java
class Account {
    static String bankName = "Simple Bank"; // static member
    protected int accountNumber;
    protected String holderName;
    protected double balance;

    public Account(int accNo, String name, double bal) {
        accountNumber = accNo;
        holderName = name;
        balance = bal;
    }

    public void displayBalance() {
        System.out.println("Account Holder: " + holderName);
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Balance: $" + balance);
    }
}

class BankAccount extends Account {

    public BankAccount(int accNo, String name, double bal) {
        super(accNo, name, bal);
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: $" + amount);
    }
```



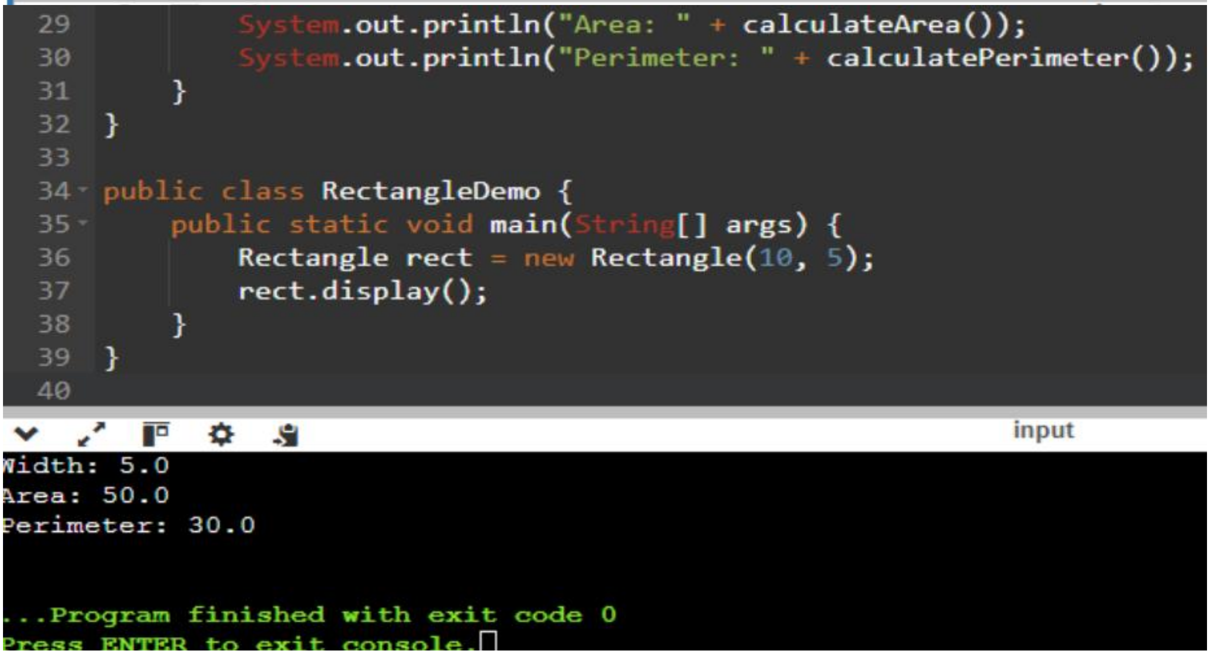
```java

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}
public class BankApp {
    public static void main(String[] args) {
        System.out.println("Welcome to " + Account.bankName);

        BankAccount acc = new BankAccount(10002200, "Sivam yadav ", 500.00);

        acc.displayBalance();
        acc.deposit(200);
        acc.withdraw(150);
        acc.displayBalance();
    }
}
```

```
ccount Holder: Sivam yadav
ccount Number: 10002200
alance: $550.0


..Program finished with exit code 0
ress ENTER to exit console.
```

10. Write a program to create a simple class to find out the area and perimeter of rectangle using super and this keyword.



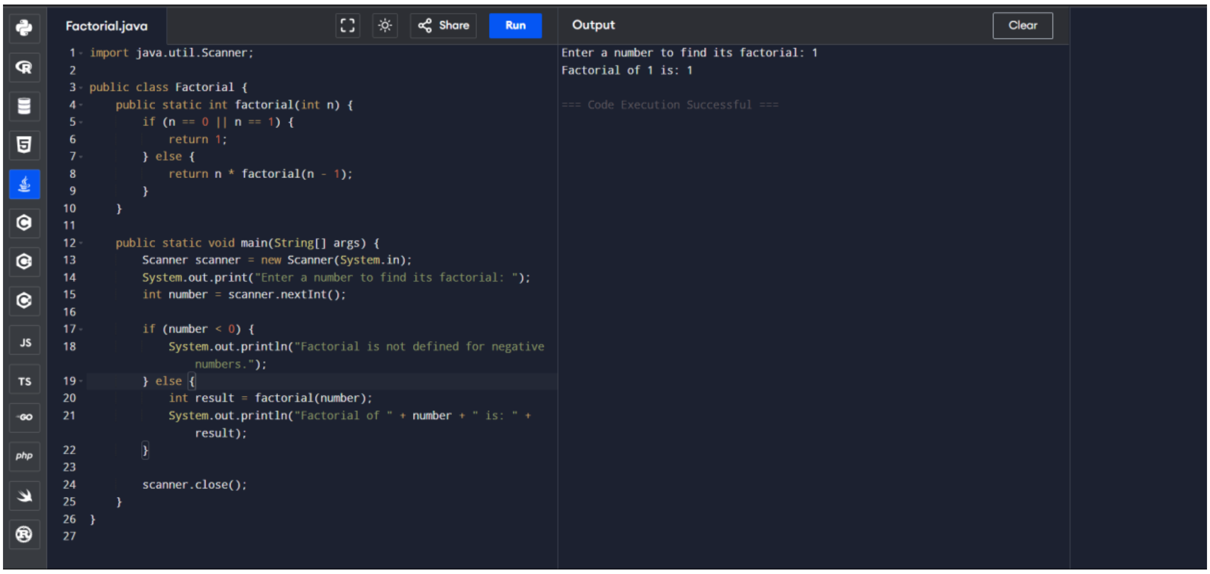
```java
class Shape {
    double length;
    double width;

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

class Rectangle extends Shape {


    public Rectangle(double length, double width) {
        super(length, width);
    }

    public double calculateArea() {
        return this.length * this.width;
    }

    public double calculatePerimeter() {
        return 2 * (this.length + this.width);
    }

    public void display() {
        System.out.println("Length: " + this.length);
        System.out.println("Width: " + this.width);
        System.out.println("Area: " + calculateArea());
```



```java
        System.out.println("Perimeter: " + calculatePerimeter());
    }
}

public class RectangleDemo {
    public static void main(String[] args) {
        Rectangle rect = new Rectangle(10, 5);
        rect.display();
    }
}
```

```
Width: 5.0
Area: 50.0
Perimeter: 30.0


...Program finished with exit code 0
Press ENTER to exit console.
```

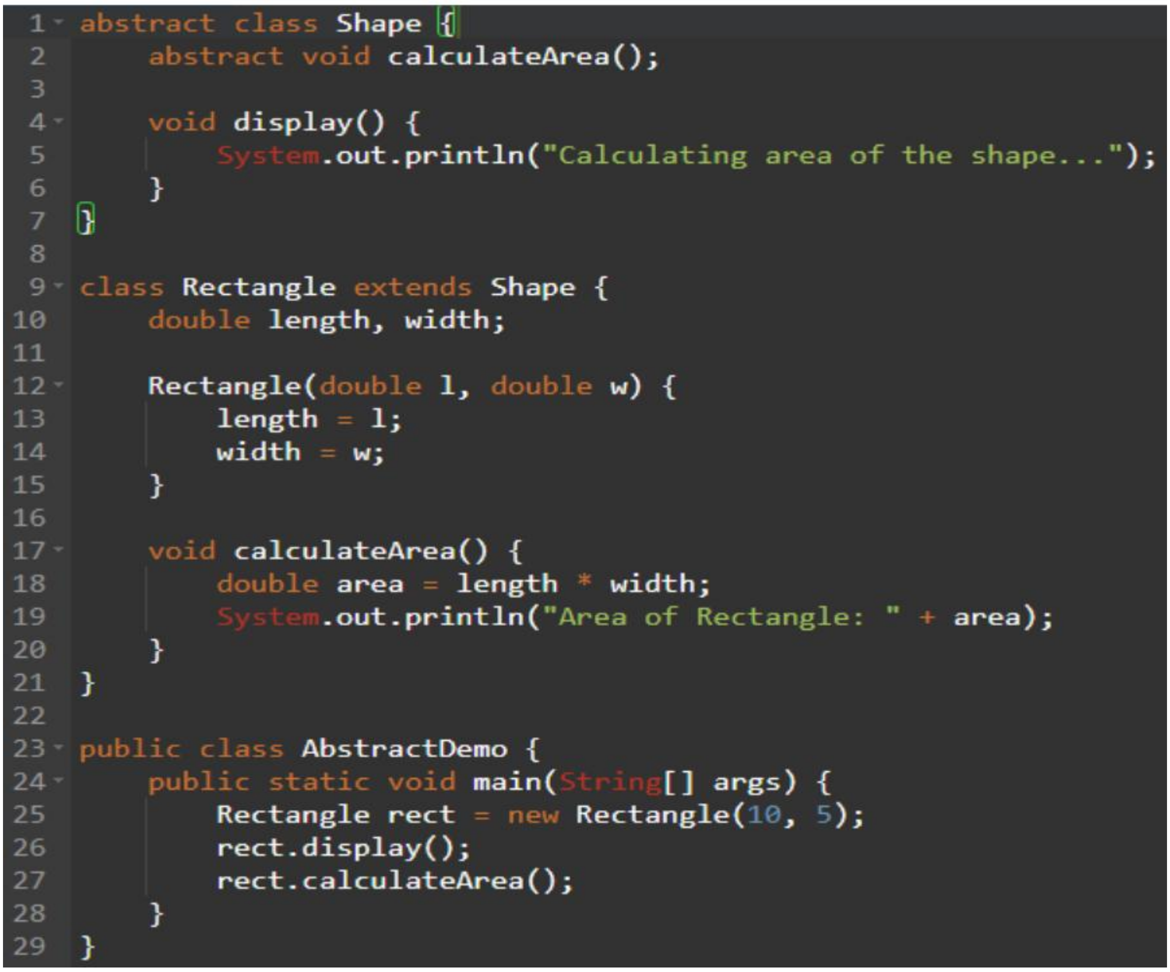
11. Write a program to find the factorial of a given number using recursion.



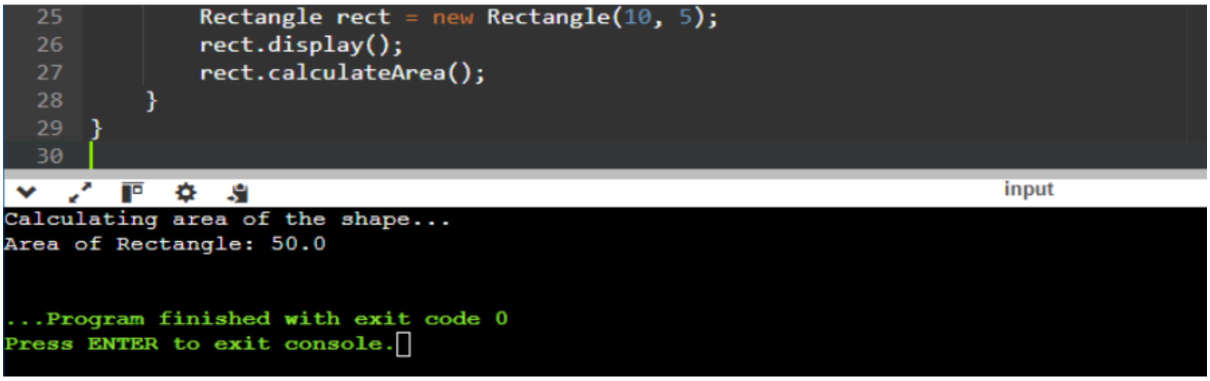
```java
import java.util.Scanner;

public class Factorial {
    public static int factorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
        } else {
            return n * factorial(n - 1);
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number to find its factorial: ");
        int number = scanner.nextInt();

        if (number < 0) {
            System.out.println("Factorial is not defined for negative
                numbers.");
        } else {
            int result = factorial(number);
            System.out.println("Factorial of " + number + " is: " +
                result);
        }

        scanner.close();
    }
}
```

```
Enter a number to find its factorial: 1
Factorial of 1 is: 1

=== Code Execution Successful ===
```

12. Write a program to design a class using abstract methods and abstract classes.



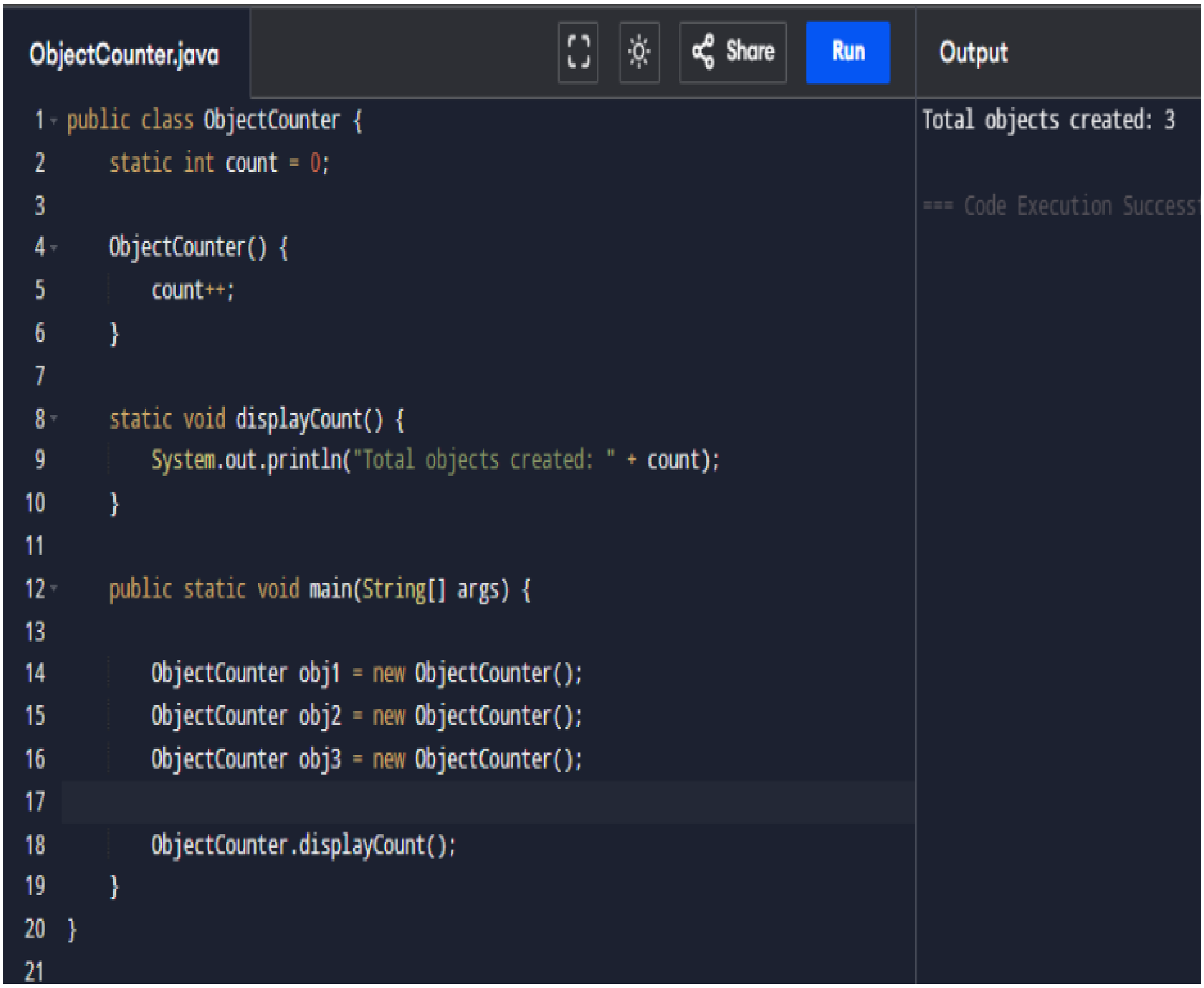
```java
abstract class Shape {
    abstract void calculateArea();

    void display() {
        System.out.println("Calculating area of the shape...");
    }
}

class Rectangle extends Shape {
    double length, width;

    Rectangle(double l, double w) {
        length = l;
        width = w;
    }

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

public class AbstractDemo {
    public static void main(String[] args) {
        Rectangle rect = new Rectangle(10, 5);
        rect.display();
        rect.calculateArea();
    }
}
```



```
        Rectangle rect = new Rectangle(10, 5);
        rect.display();
        rect.calculateArea();
    }
}

```

```
Calculating area of the shape...
Area of Rectangle: 50.0


...Program finished with exit code 0
Press ENTER to exit console.
```

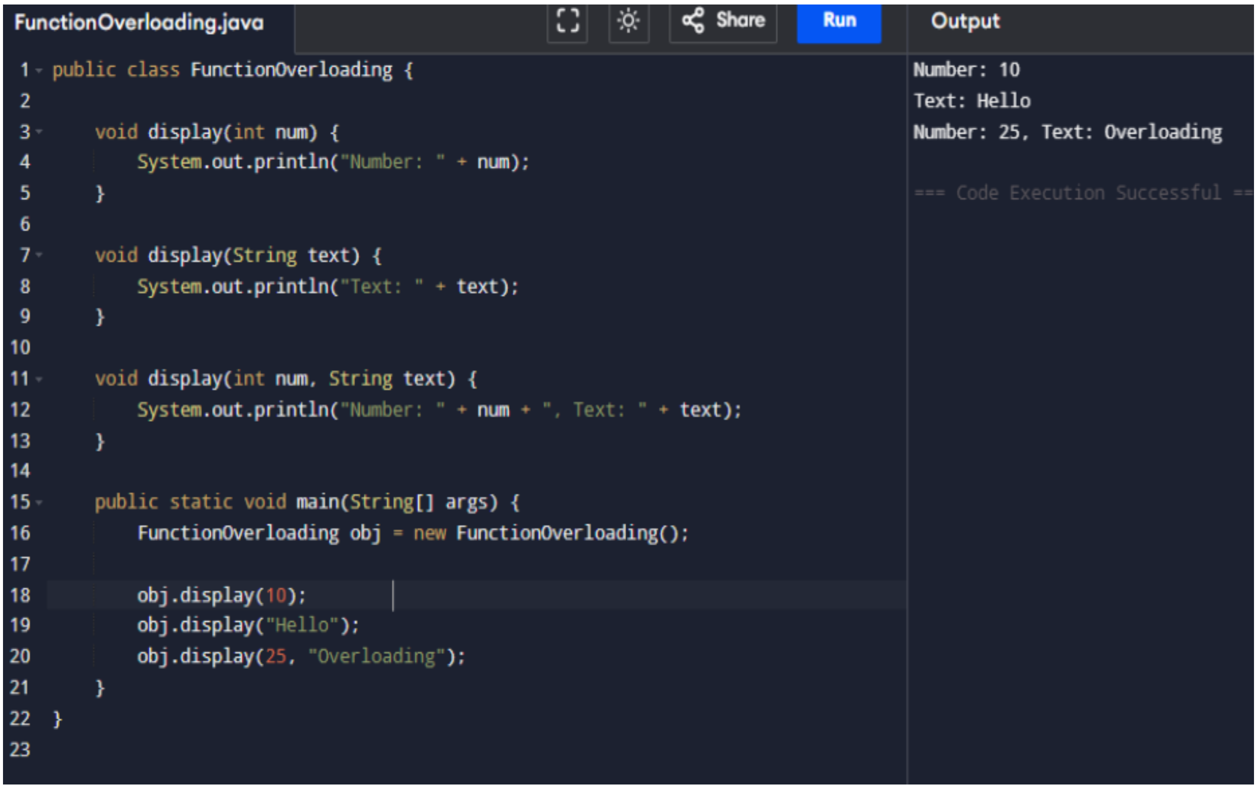
13. Write a program to count the number of objects created for a class using static member function.



```
public class ObjectCounter {
    static int count = 0;

    ObjectCounter() {
        count++;
    }

    static void displayCount() {
        System.out.println("Total objects created: " + count);
    }

    public static void main(String[] args) {

        ObjectCounter obj1 = new ObjectCounter();
        ObjectCounter obj2 = new ObjectCounter();
        ObjectCounter obj3 = new ObjectCounter();

        ObjectCounter.displayCount();
    }
}

```

```
Total objects created: 3


=== Code Execution Success
```

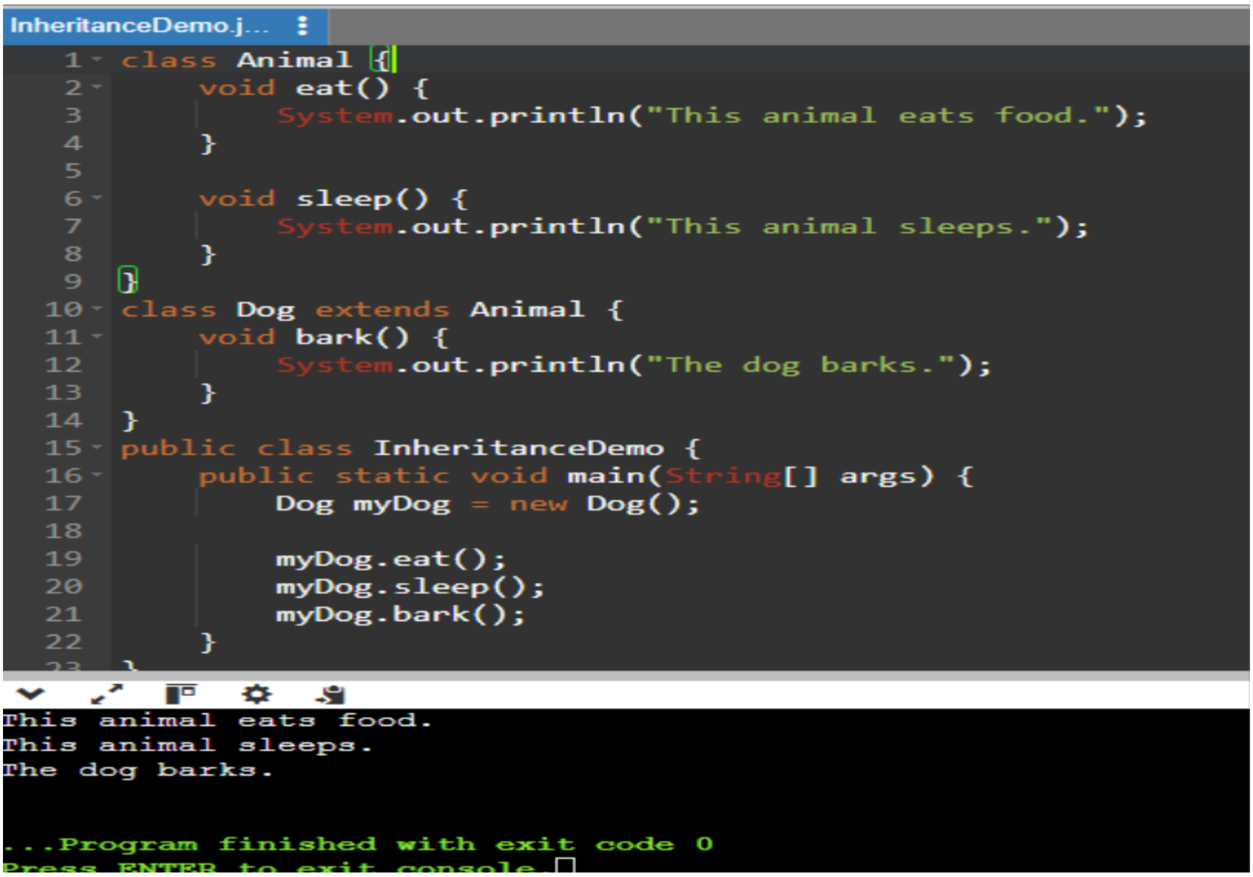
14. Write a program to demonstrate the use of function overloading.



```java
public class FunctionOverloading {

    void display(int num) {
        System.out.println("Number: " + num);
    }

    void display(String text) {
        System.out.println("Text: " + text);
    }

    void display(int num, String text) {
        System.out.println("Number: " + num + ", Text: " + text);
    }

    public static void main(String[] args) {
        FunctionOverloading obj = new FunctionOverloading();

        obj.display(10);
        obj.display("Hello");
        obj.display(25, "Overloading");
    }
}
```

```
Number: 10
Text: Hello
Number: 25, Text: Overloading

=== Code Execution Successful ==
```

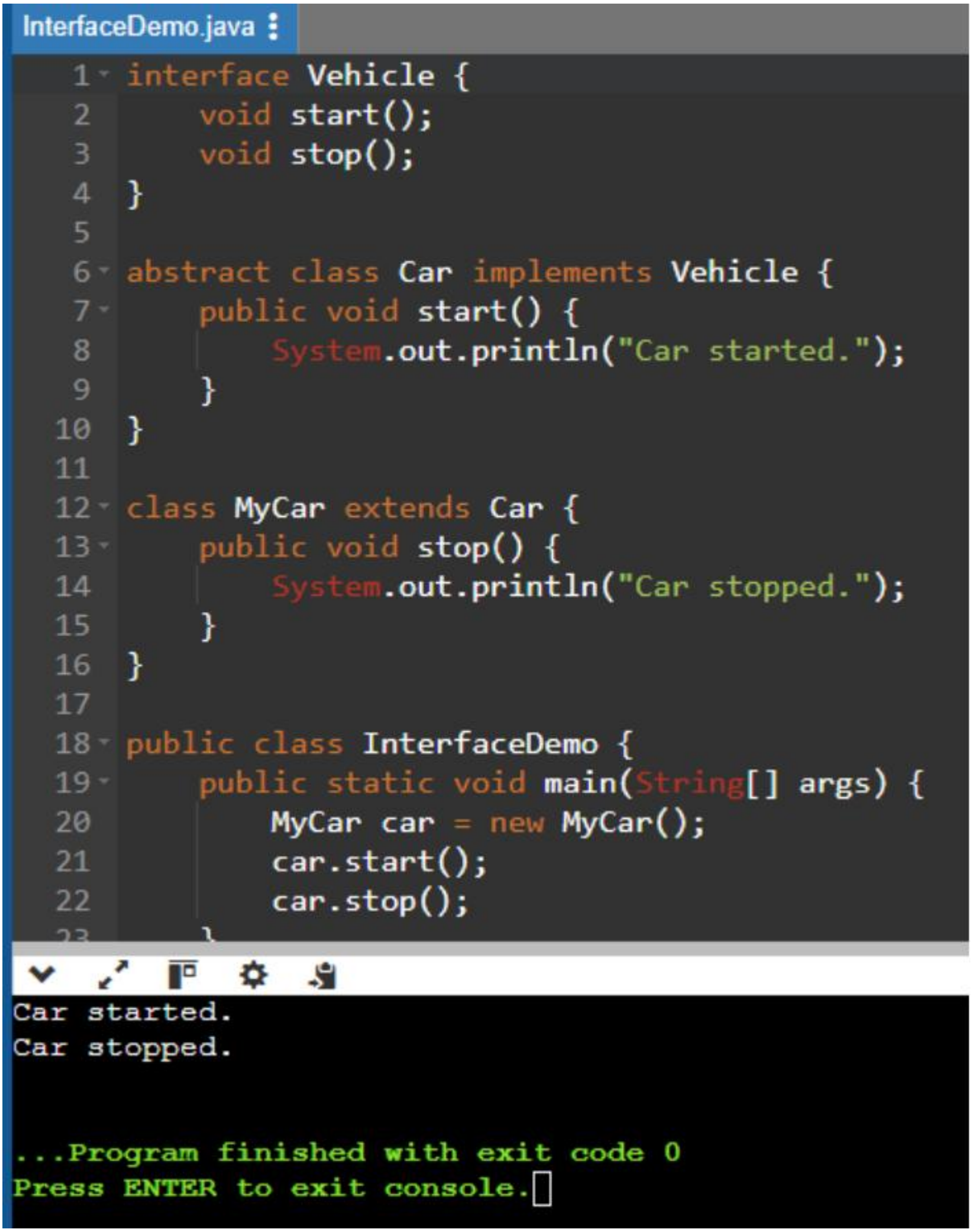
15. Write a program to demonstrate the use of inheritance.



```java
class Animal {
    void eat() {
        System.out.println("This animal eats food.");
    }

    void sleep() {
        System.out.println("This animal sleeps.");
    }
}
class Dog extends Animal {
    void bark() {
        System.out.println("The dog barks.");
    }
}
public class InheritanceDemo {
    public static void main(String[] args) {
        Dog myDog = new Dog();

        myDog.eat();
        myDog.sleep();
        myDog.bark();
    }
}
```

```
This animal eats food.
This animal sleeps.
The dog barks.


...Program finished with exit code 0
Press ENTER to exit console.
```

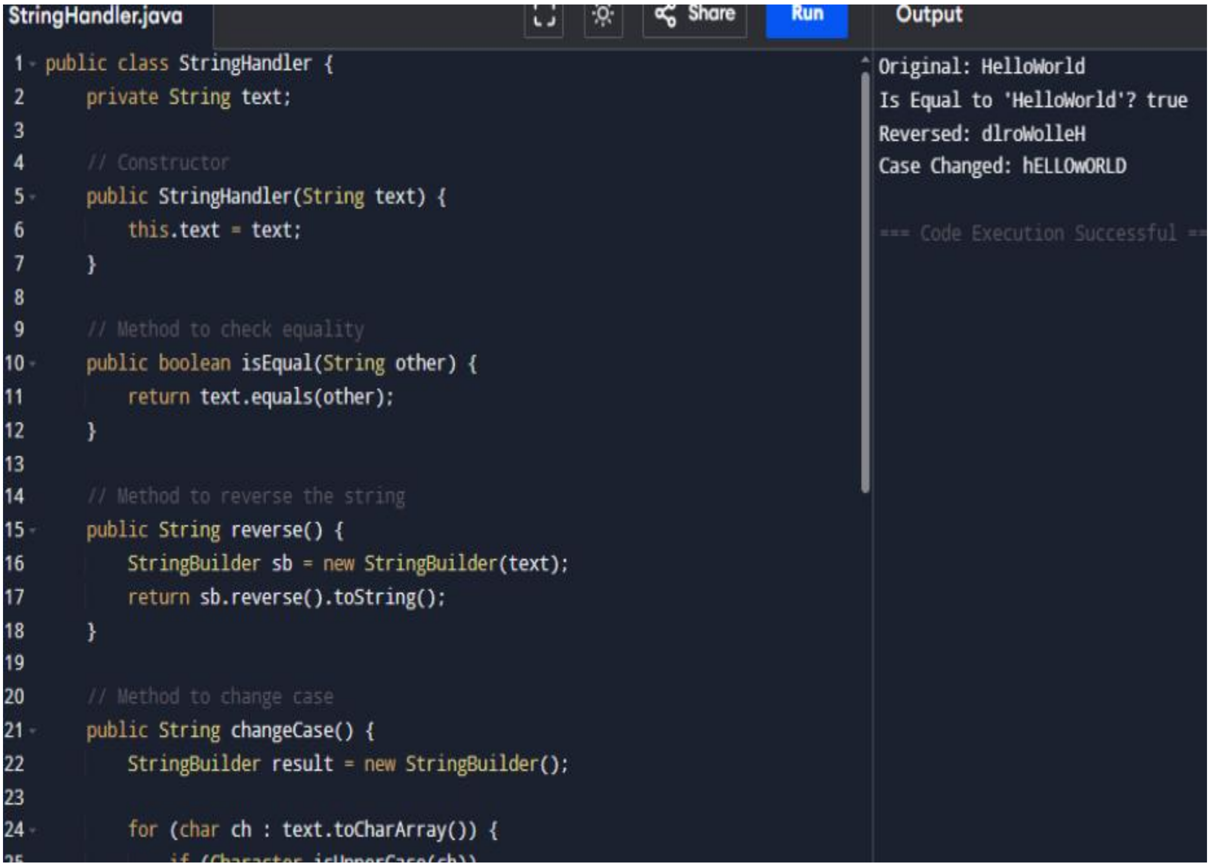
16. Write a program that show the partial implementation of Interface.



```java
interface Vehicle {
    void start();
    void stop();
}

abstract class Car implements Vehicle {
    public void start() {
        System.out.println("Car started.");
    }
}

class MyCar extends Car {
    public void stop() {
        System.out.println("Car stopped.");
    }
}

public class InterfaceDemo {
    public static void main(String[] args) {
        MyCar car = new MyCar();
        car.start();
        car.stop();
```

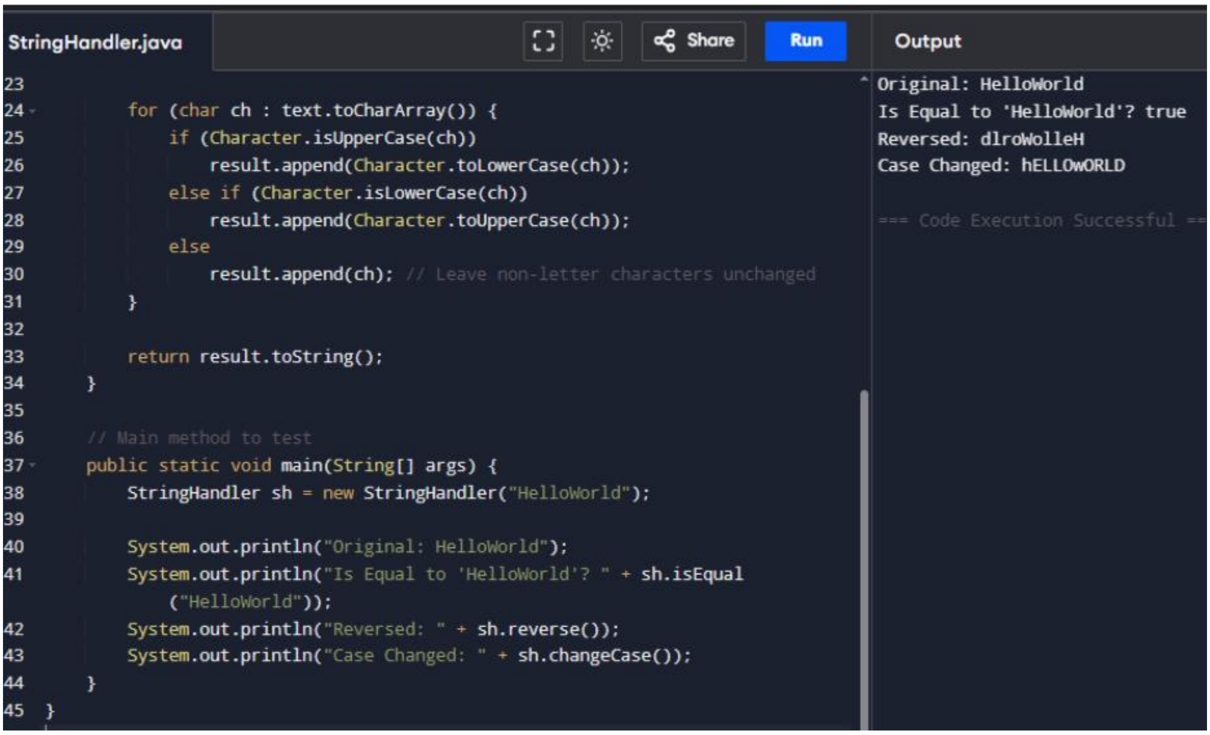
```
Car started.
Car stopped.


...Program finished with exit code 0
Press ENTER to exit console.
```

17. Write a program to design a string class that perform string method (Equal, Reverse the string, change case).



```java
public class StringHandler {
    private String text;

    // Constructor
    public StringHandler(String text) {
        this.text = text;
    }

    // Method to check equality
    public boolean isEqual(String other) {
        return text.equals(other);
    }

    // Method to reverse the string
    public String reverse() {
        StringBuilder sb = new StringBuilder(text);
        return sb.reverse().toString();
    }

    // Method to change case
    public String changeCase() {
        StringBuilder result = new StringBuilder();

        for (char ch : text.toCharArray()) {
            if (Character.isUpperCase(ch))
```

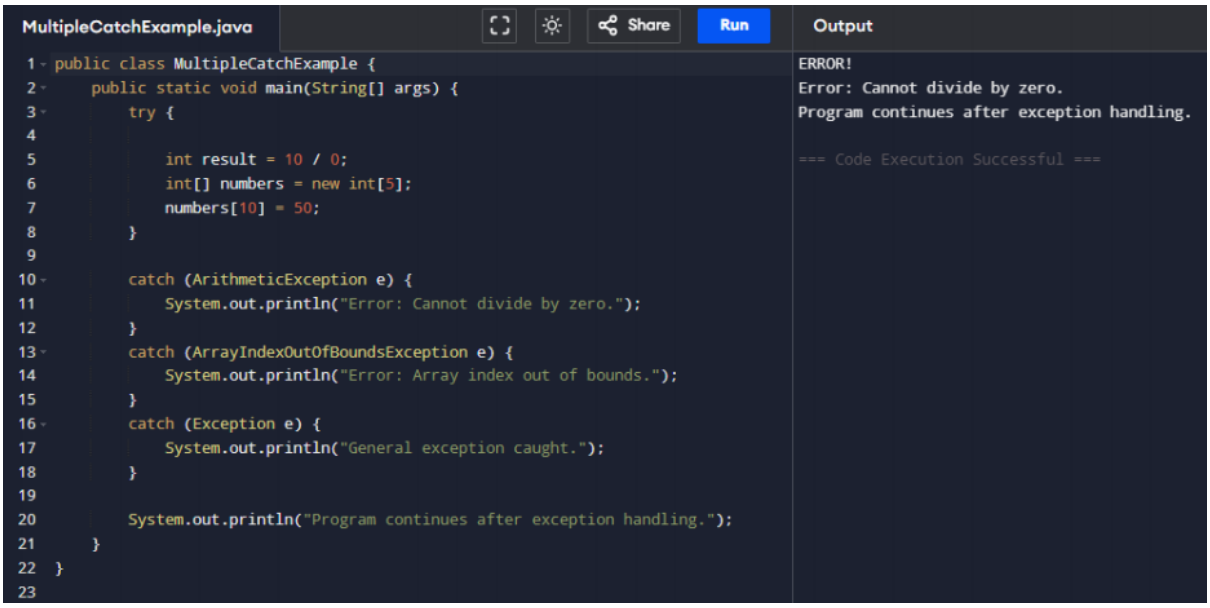
```
Original: HelloWorld
Is Equal to 'HelloWorld'? true
Reversed: dlroWolleH
Case Changed: hELLOwORLD

=== Code Execution Successful ==
```



```java

        for (char ch : text.toCharArray()) {
            if (Character.isUpperCase(ch))
                result.append(Character.toLowerCase(ch));
            else if (Character.isLowerCase(ch))
                result.append(Character.toUpperCase(ch));
            else
                result.append(ch); // Leave non-letter characters unchanged
        }

        return result.toString();
    }

    // Main method to test
    public static void main(String[] args) {
        StringHandler sh = new StringHandler("HelloWorld");

        System.out.println("Original: HelloWorld");
        System.out.println("Is Equal to 'HelloWorld'? " + sh.isEqual("HelloWorld"));
        System.out.println("Reversed: " + sh.reverse());
        System.out.println("Case Changed: " + sh.changeCase());
    }
}
```

```
Original: HelloWorld
Is Equal to 'HelloWorld'? true
Reversed: dlroWolleH
Case Changed: hELLOwORLD

=== Code Execution Successful ==
```

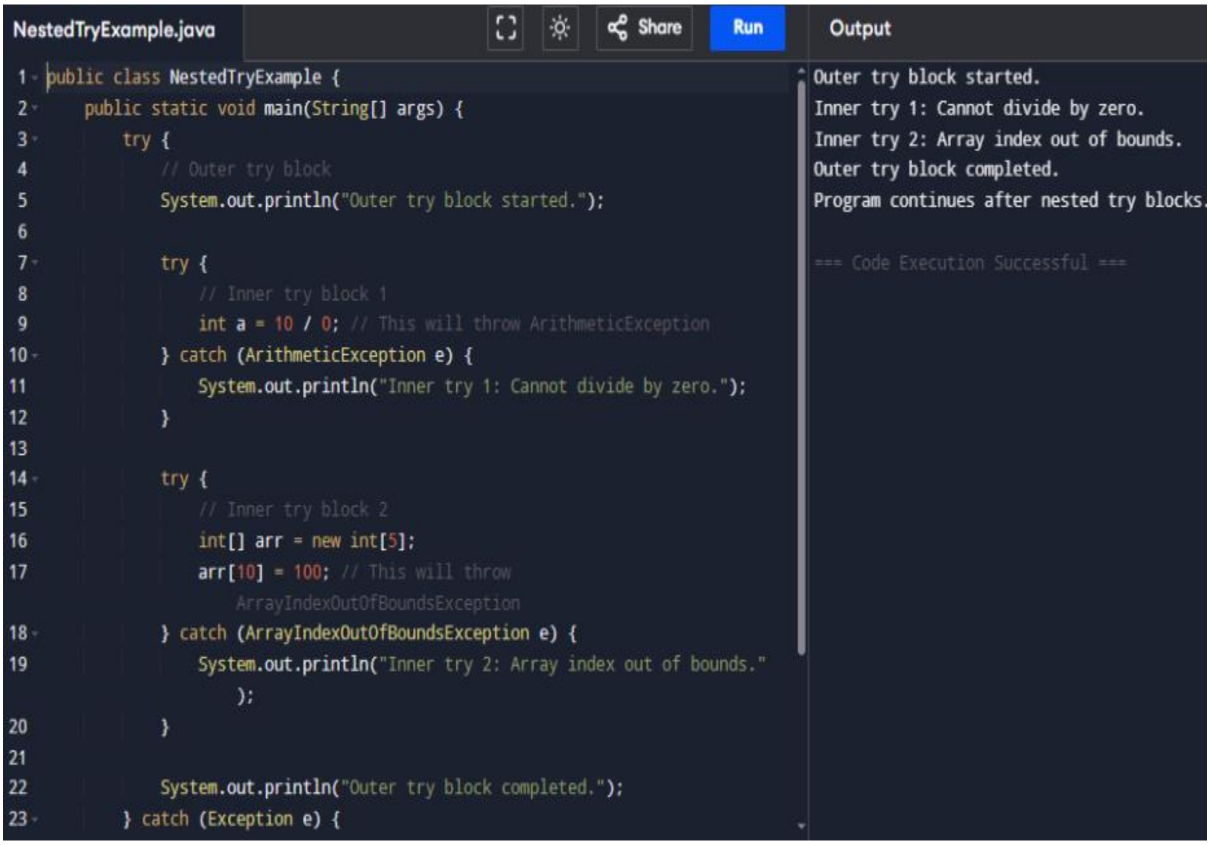
18. Write a program to handle the exception using try and multiple catch block.



```java
public class MultipleCatchExample {
    public static void main(String[] args) {
        try {

            int result = 10 / 0;
            int[] numbers = new int[5];
            numbers[10] = 50;
        }

        catch (ArithmeticException e) {
            System.out.println("Error: Cannot divide by zero.");
        }
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: Array index out of bounds.");
        }
        catch (Exception e) {
            System.out.println("General exception caught.");
        }

        System.out.println("Program continues after exception handling.");
    }
}
```

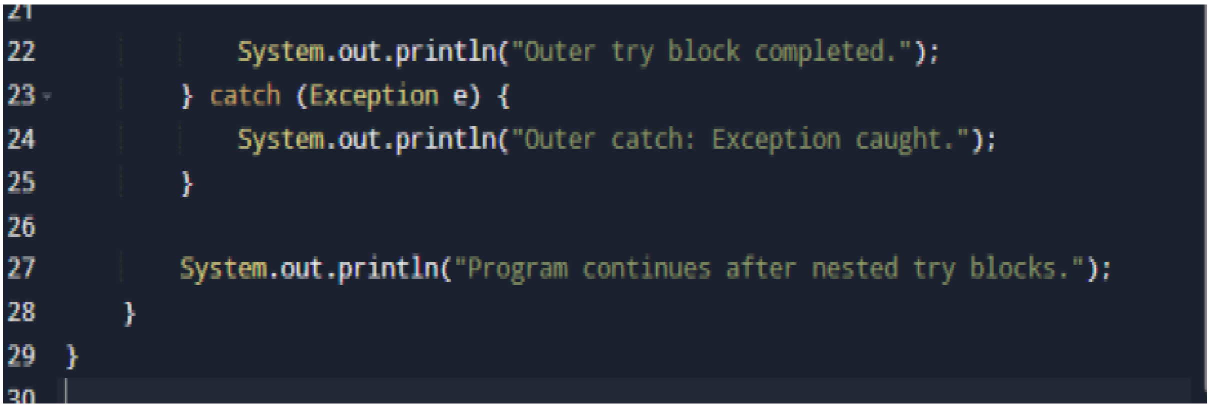
```
ERROR!
Error: Cannot divide by zero.
Program continues after exception handling.

=== Code Execution Successful ===
```

19. Write a program that implement the Nested Try Statements.



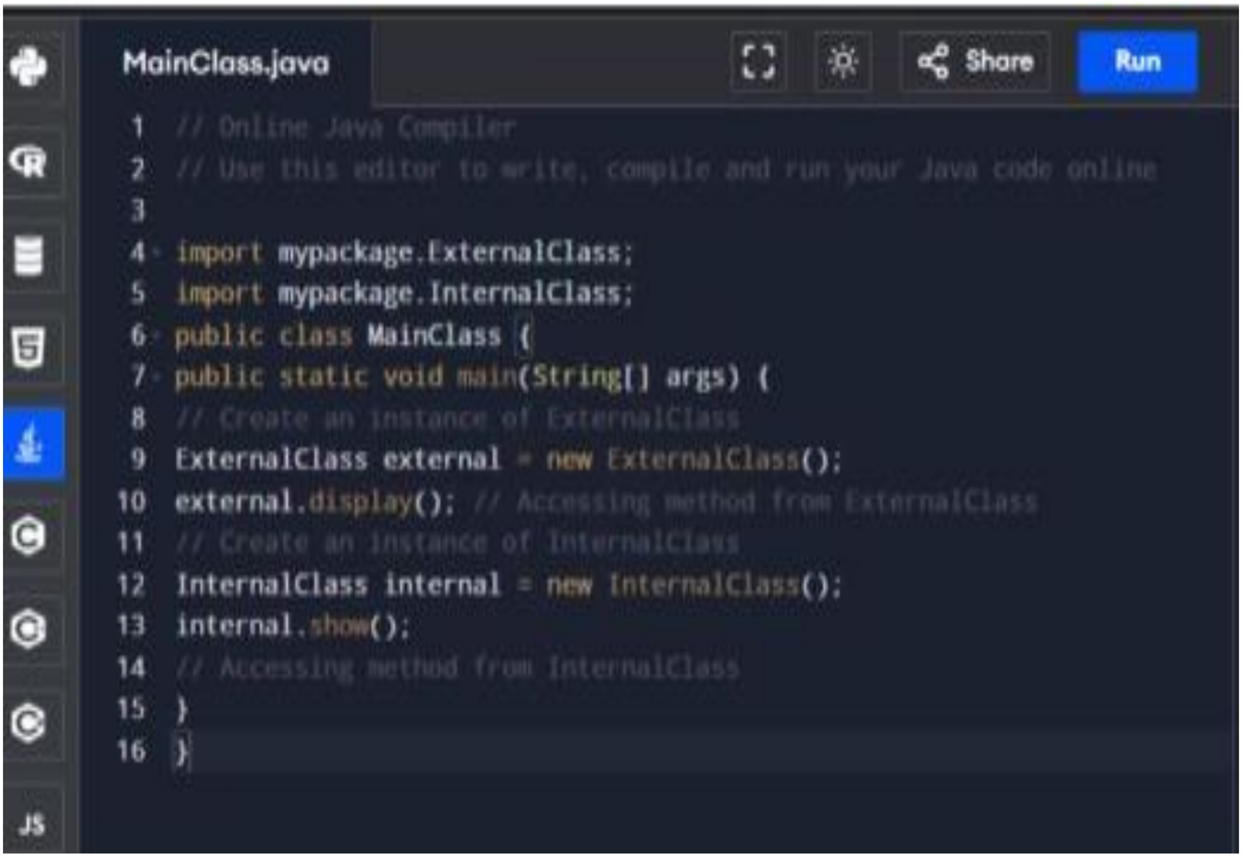
```java
public class NestedTryExample {
    public static void main(String[] args) {
        try {
            // Outer try block
            System.out.println("Outer try block started.");

            try {
                // Inner try block 1
                int a = 10 / 0; // This will throw ArithmeticException
            } catch (ArithmeticException e) {
                System.out.println("Inner try 1: Cannot divide by zero.");
            }

            try {
                // Inner try block 2
                int[] arr = new int[5];
                arr[10] = 100; // This will throw ArrayIndexOutOfBoundsException
            } catch (ArrayIndexOutOfBoundsException e) {
                System.out.println("Inner try 2: Array index out of bounds.");
            }

            System.out.println("Outer try block completed.");
        } catch (Exception e) {
```

```
Outer try block started.
Inner try 1: Cannot divide by zero.
Inner try 2: Array index out of bounds.
Outer try block completed.
Program continues after nested try blocks.

=== Code Execution Successful ===
```



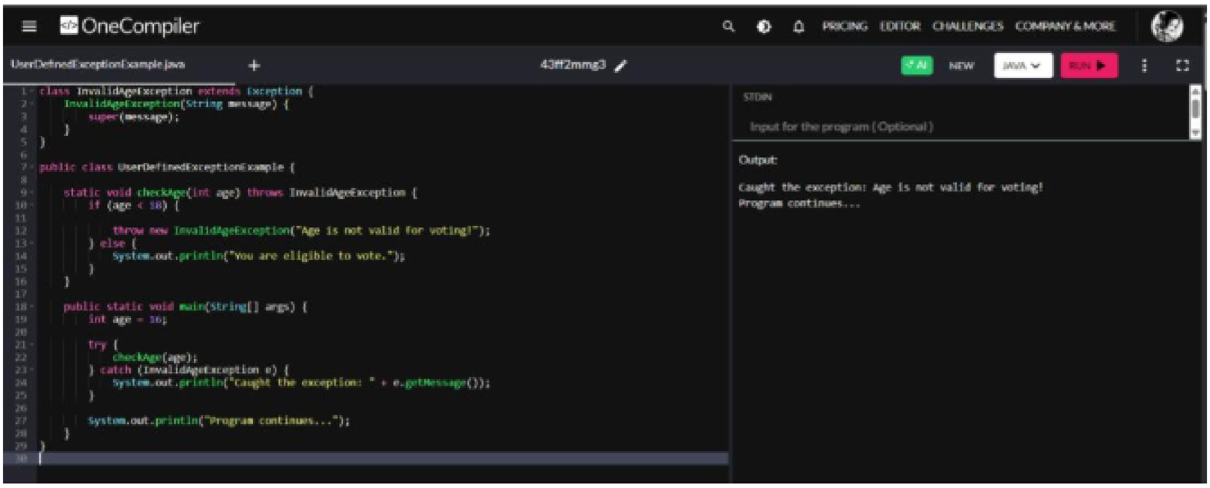
```
            System.out.println("Outer try block completed.");
        } catch (Exception e) {
            System.out.println("Outer catch: Exception caught.");
        }

        System.out.println("Program continues after nested try blocks.");
    }
}
```

20. Write a program to create a menu using the frame.



```
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

import mypackage.ExternalClass;
import mypackage.InternalClass;
public class MainClass {
public static void main(String[] args) {
// Create an instance of ExternalClass
ExternalClass external = new ExternalClass();
external.display(); // Accessing method from ExternalClass
// Create an instance of InternalClass
InternalClass internal = new InternalClass();
internal.show();
// Accessing method from InternalClass
}
}
```

22. Write a program to handle the user defined exception using throw keyword.

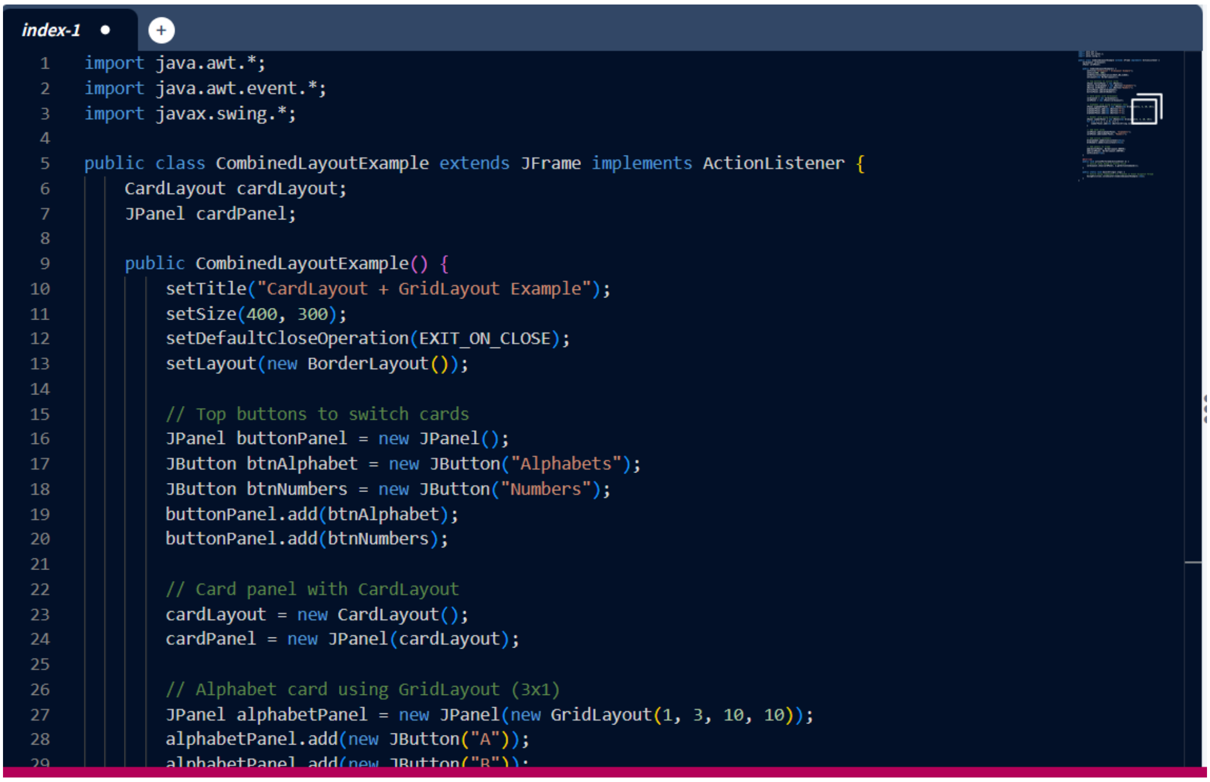


```java
class InvalidAgeException extends Exception {
    InvalidAgeException(String message) {
        super(message);
    }
}

public class UserDefinedExceptionExample {

    static void checkAge(int age) throws InvalidAgeException {
        if (age < 18) {

            throw new InvalidAgeException("Age is not valid for voting!");
        } else {
            System.out.println("You are eligible to vote.");
        }
    }

    public static void main(String[] args) {
        int age = 16;

        try {
            checkAge(age);
        } catch (InvalidAgeException e) {
            System.out.println("Caught the exception: " + e.getMessage());
        }

        System.out.println("Program continues...");
    }
}
```

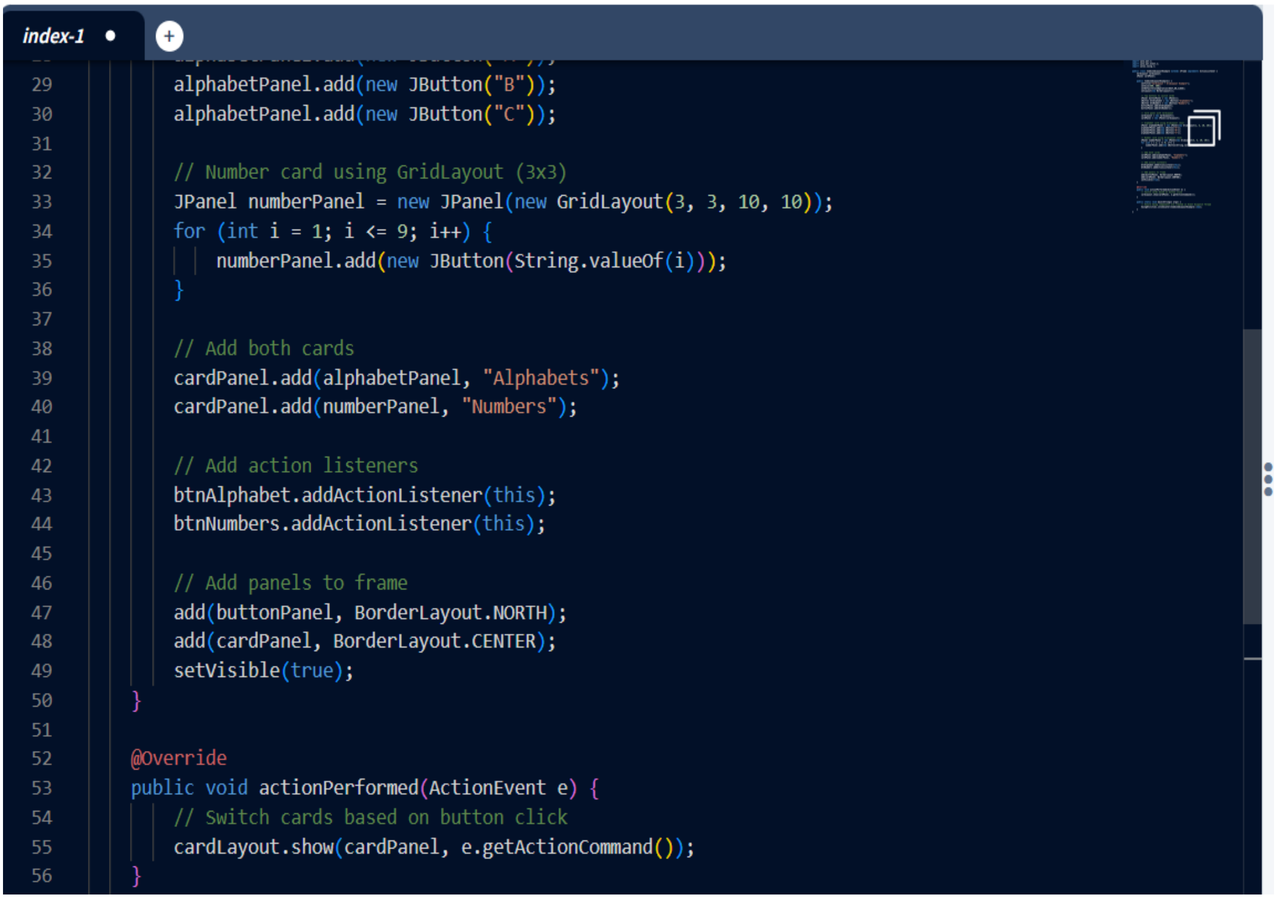
Output:

```
Caught the exception: Age is not valid for voting!
Program continues...
```

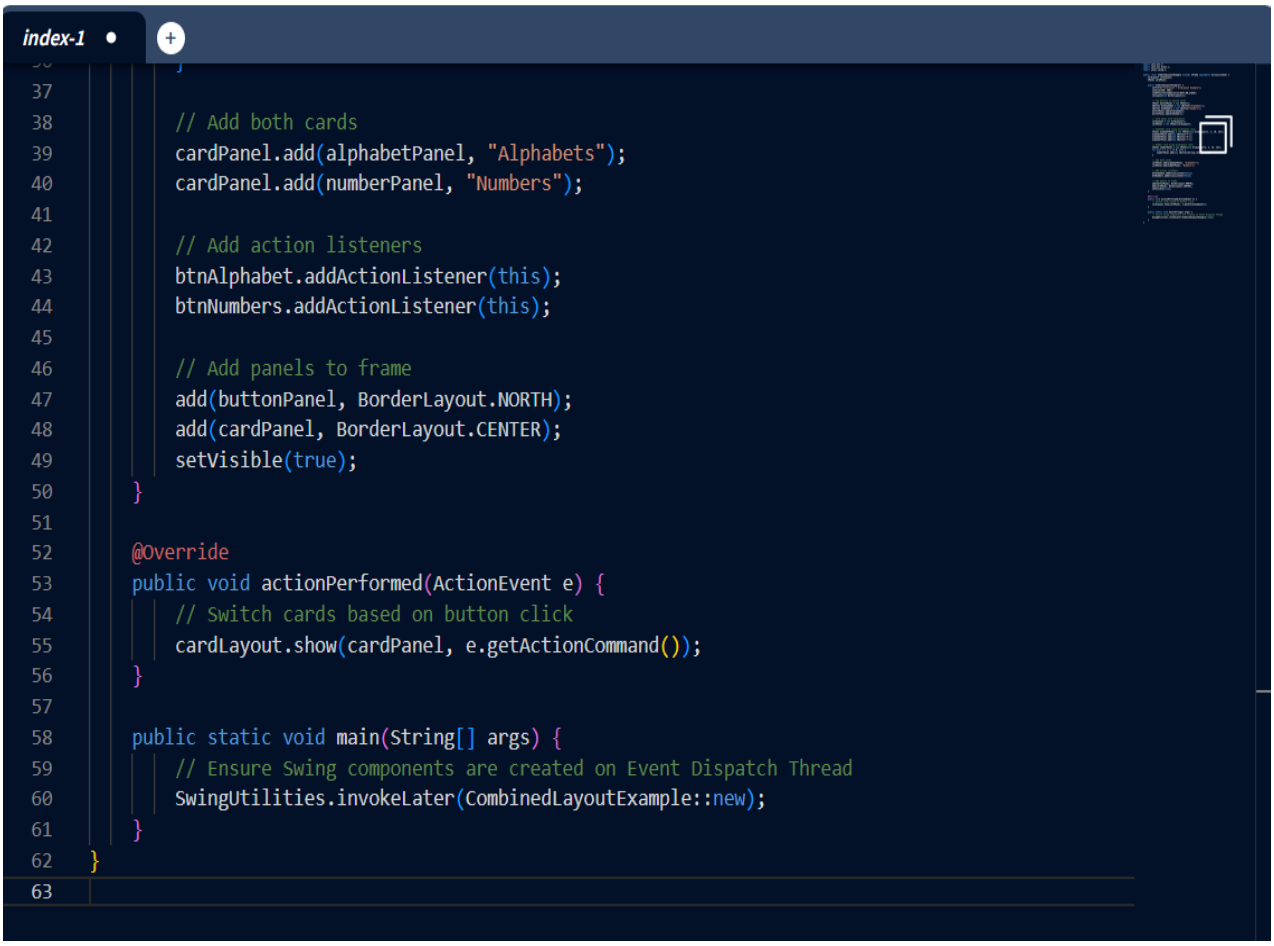
23. Write a program to implement the gridLayout, cardLayout.



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

public class CombinedLayoutExample extends JFrame implements ActionListener {
    CardLayout cardLayout;
    JPanel cardPanel;

    public CombinedLayoutExample() {
        setTitle("CardLayout + GridLayout Example");
        setSize(400, 300);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        // Top buttons to switch cards
        JPanel buttonPanel = new JPanel();
        JButton btnAlphabet = new JButton("Alphabets");
        JButton btnNumbers = new JButton("Numbers");
        buttonPanel.add(btnAlphabet);
        buttonPanel.add(btnNumbers);

        // Card panel with CardLayout
        cardLayout = new CardLayout();
        cardPanel = new JPanel(cardLayout);

        // Alphabet card using GridLayout (3x1)
        JPanel alphabetPanel = new JPanel(new GridLayout(1, 3, 10, 10));
        alphabetPanel.add(new JButton("A"));
        alphabetPanel.add(new JButton("B"));
```



```
        alphabetPanel.add(new JButton("B"));
        alphabetPanel.add(new JButton("C"));

        // Number card using GridLayout (3x3)
        JPanel numberPanel = new JPanel(new GridLayout(3, 3, 10, 10));
        for (int i = 1; i <= 9; i++) {
            numberPanel.add(new JButton(String.valueOf(i)));
        }

        // Add both cards
        cardPanel.add(alphabetPanel, "Alphabets");
        cardPanel.add(numberPanel, "Numbers");

        // Add action listeners
        btnAlphabet.addActionListener(this);
        btnNumbers.addActionListener(this);

        // Add panels to frame
        add(buttonPanel, BorderLayout.NORTH);
        add(cardPanel, BorderLayout.CENTER);
        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        // Switch cards based on button click
        cardLayout.show(cardPanel, e.getActionCommand());
    }
```



```

        // Add both cards
        cardPanel.add(alphabetPanel, "Alphabets");
        cardPanel.add(numberPanel, "Numbers");

        // Add action listeners
        btnAlphabet.addActionListener(this);
        btnNumbers.addActionListener(this);

        // Add panels to frame
        add(buttonPanel, BorderLayout.NORTH);
        add(cardPanel, BorderLayout.CENTER);
        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        // Switch cards based on button click
        cardLayout.show(cardPanel, e.getActionCommand());
    }
```

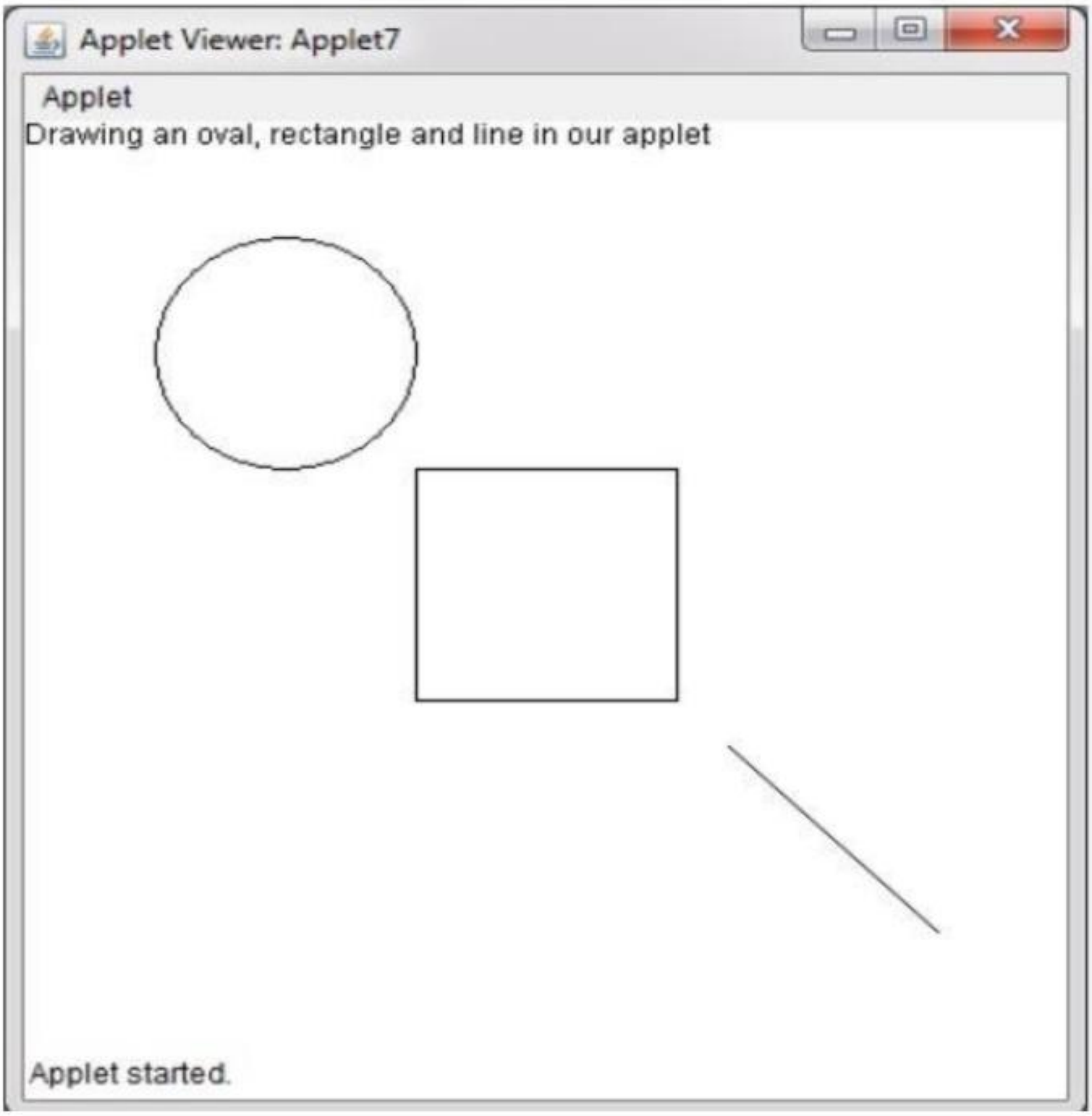
```

    public static void main(String[] args) {
        // Ensure Swing components are created on Event Dispatch Thread
        SwingUtilities.invokeLater(CombinedLayoutExample::new);
    }
}

```

24. Write a program to create Frame that display the student information.

Solution-

```
import javax.swing.; import java.awt.; public class ShapeDrawing extends JPanel {
@Override
protected void paintComponent(Graphics g) {
super.paintComponent(g);
// Draw a line
g.drawLine(50, 50, 200, 50);
// Draw a rectangle
g.drawRect(50, 70, 150, 100);
// Draw an oval
g.drawOval(50, 180, 150, 100);
// Draw text
g.drawString("Hello, Graphics!", 50, 300);
}
public static void main(String[] args) {
JFrame frame = new JFrame("Shape Drawing Example");
ShapeDrawing panel = new ShapeDrawing(); frame.add(panel);
frame.setSize(300, 400);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
}
}
```

Output will be :



Drawing an oval, rectangle and line in our applet

Applet started.