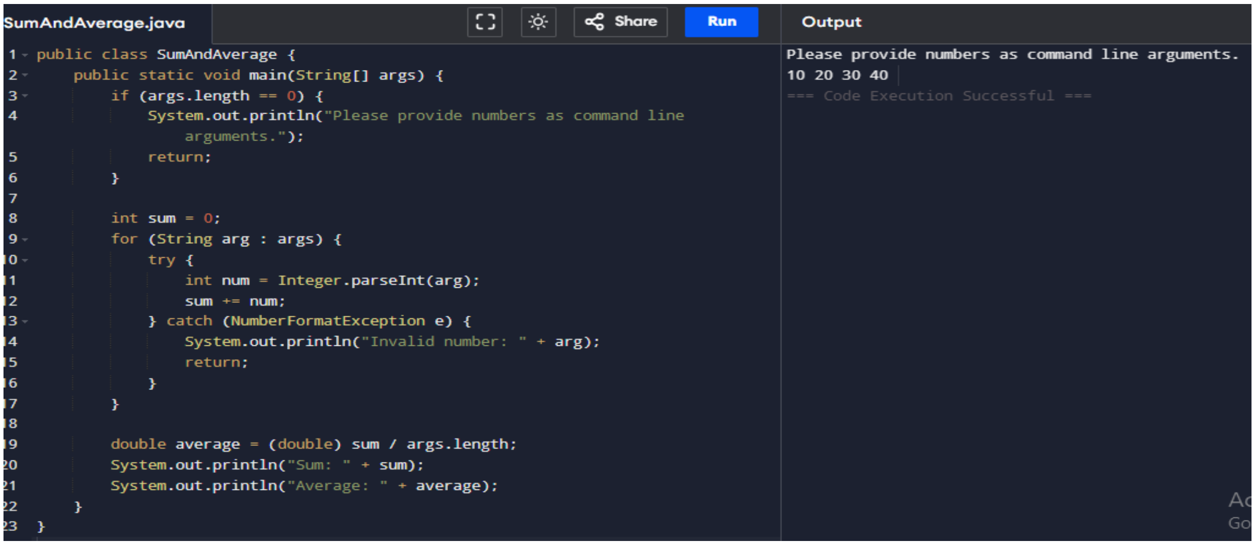
JAVA PROGRAM

C Programming Lab

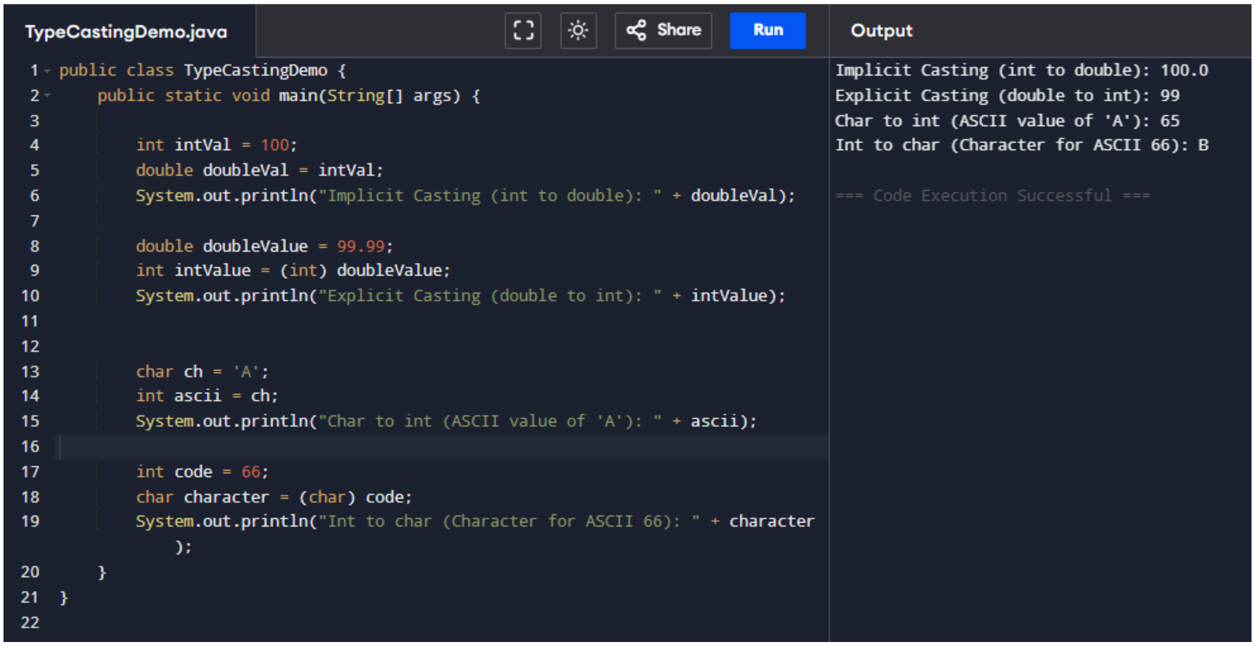
1. Write a program to find the average and sum of the N numbers using Command line argument.



```java
public class SumAndAverage {
    public static void main(String[] args) {
        if (args.length == 0) {
            System.out.println("Please provide numbers as command line arguments.");
            return;
        }

        int sum = 0;
        for (String arg : args) {
            try {
                int num = Integer.parseInt(arg);
                sum += num;
            } catch (NumberFormatException e) {
                System.out.println("Invalid number: " + arg);
                return;
            }
        }

        double average = (double) sum / args.length;
        System.out.println("Sum: " + sum);
        System.out.println("Average: " + average);
    }
}
```

```
Please provide numbers as command line arguments.
10 20 30 40
=== Code Execution Successful ===
```

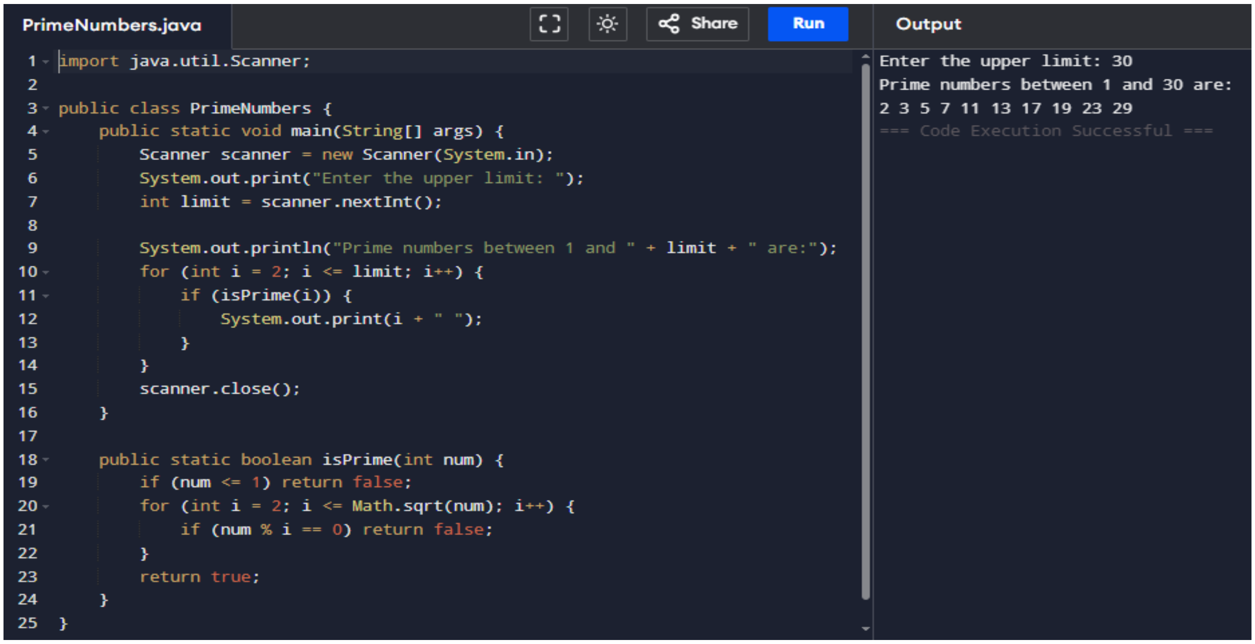
2. Write a program to demonstrate type casting.



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

        int intVal = 100;
        double doubleVal = intVal;
        System.out.println("Implicit Casting (int to double): " + doubleVal);

        double doubleValue = 99.99;
        int intValue = (int) doubleValue;
        System.out.println("Explicit Casting (double to int): " + intValue);


        char ch = 'A';
        int ascii = ch;
        System.out.println("Char to int (ASCII value of 'A'): " + ascii);

        int code = 66;
        char character = (char) code;
        System.out.println("Int to char (Character for ASCII 66): " + character);
    }
}
```

```
Implicit Casting (int to double): 100.0
Explicit Casting (double to int): 99
Char to int (ASCII value of 'A'): 65
Int to char (Character for ASCII 66): B

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

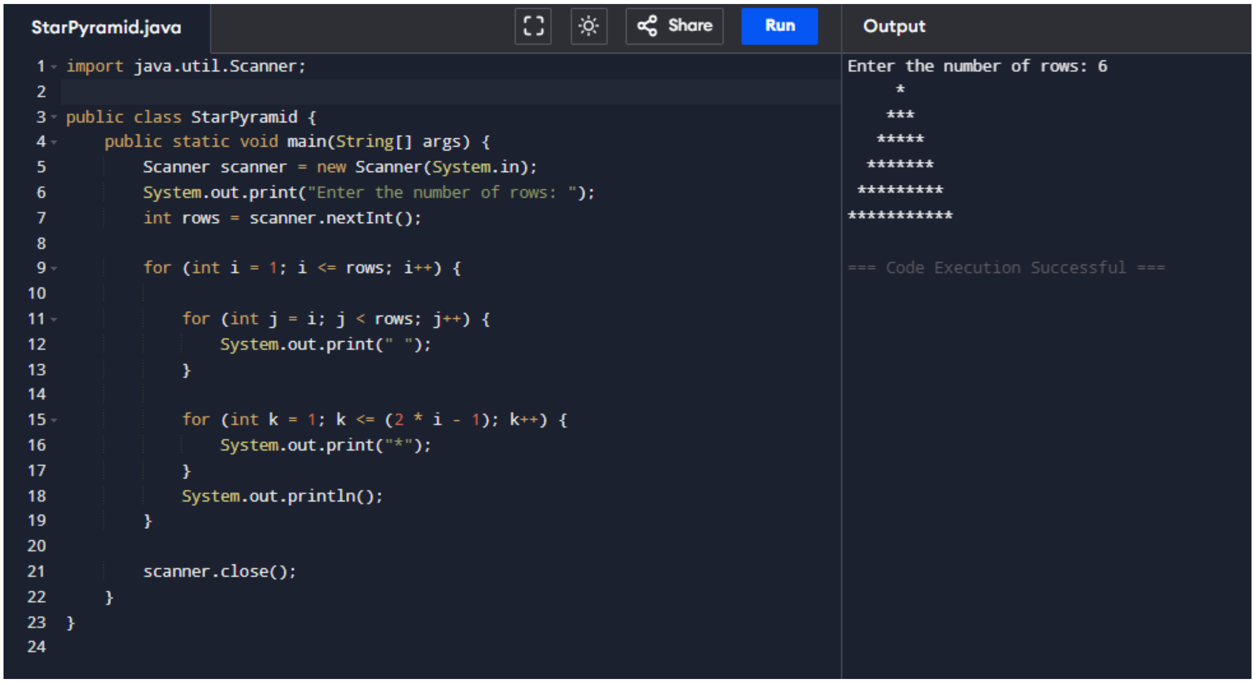
3. Write a program to generate prime numbers between 1 & given number



```java
import java.util.Scanner;

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

        System.out.println("Prime numbers between 1 and " + limit + " are:");
        for (int i = 2; i <= limit; i++) {
            if (isPrime(i)) {
                System.out.print(i + " ");
            }
        }
        scanner.close();
    }

    public static boolean isPrime(int num) {
        if (num <= 1) return false;
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) return false;
        }
        return true;
    }
}
```

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

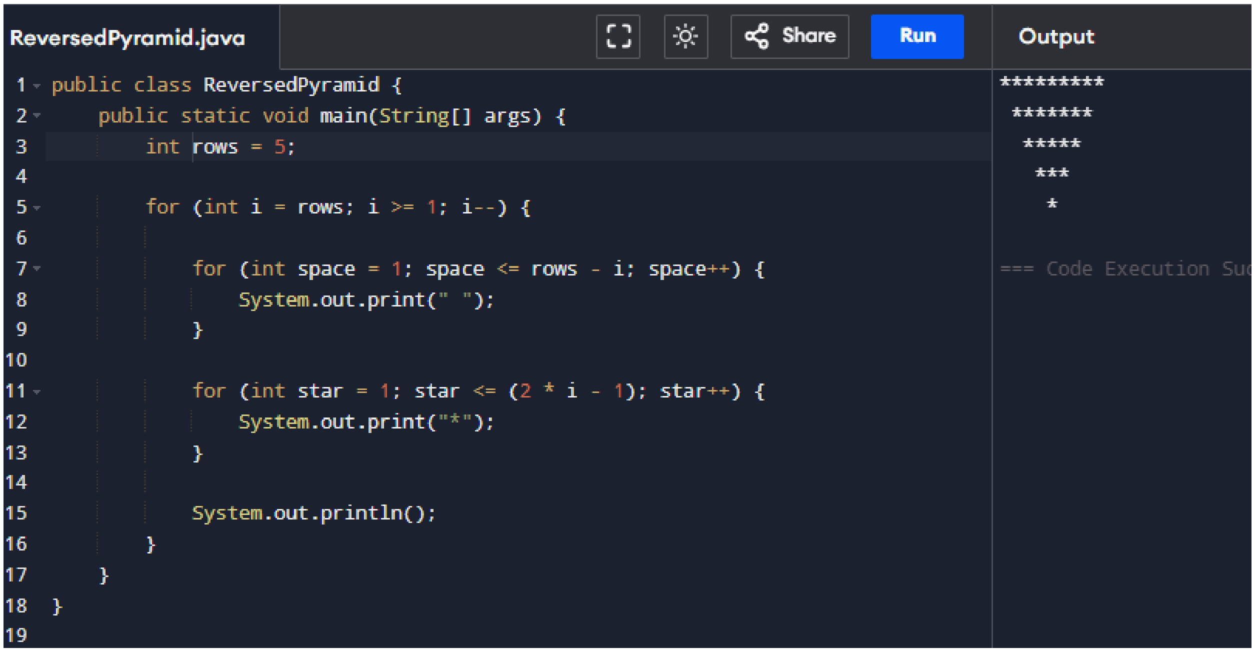
4. Write a program to generate pyramid of stars using nested for loops



```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: ");
        int rows = scanner.nextInt();

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

            for (int j = i; j < rows; 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: 6
     *
    ***
   *****
  *******
 *********
***********

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

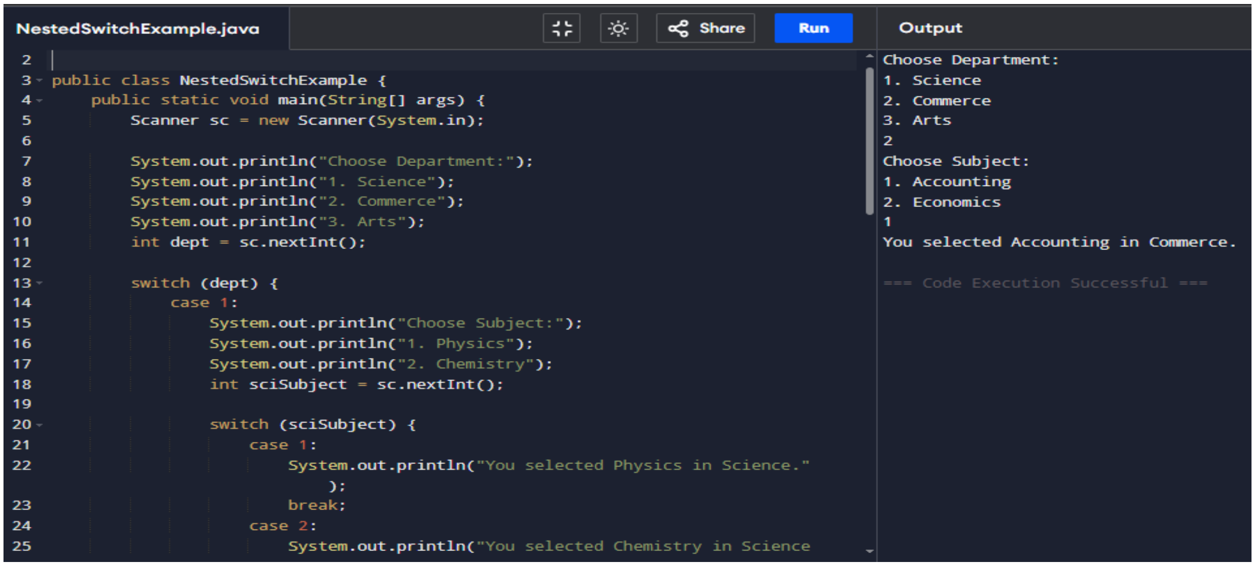
5. Write a program to reversed pyramid using for loops & decrement operator.



```java
public class ReversedPyramid {
    public static void main(String[] args) {
        int rows = 5;

        for (int i = rows; i >= 1; i--) {

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

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

            System.out.println();
        }
    }
}
```

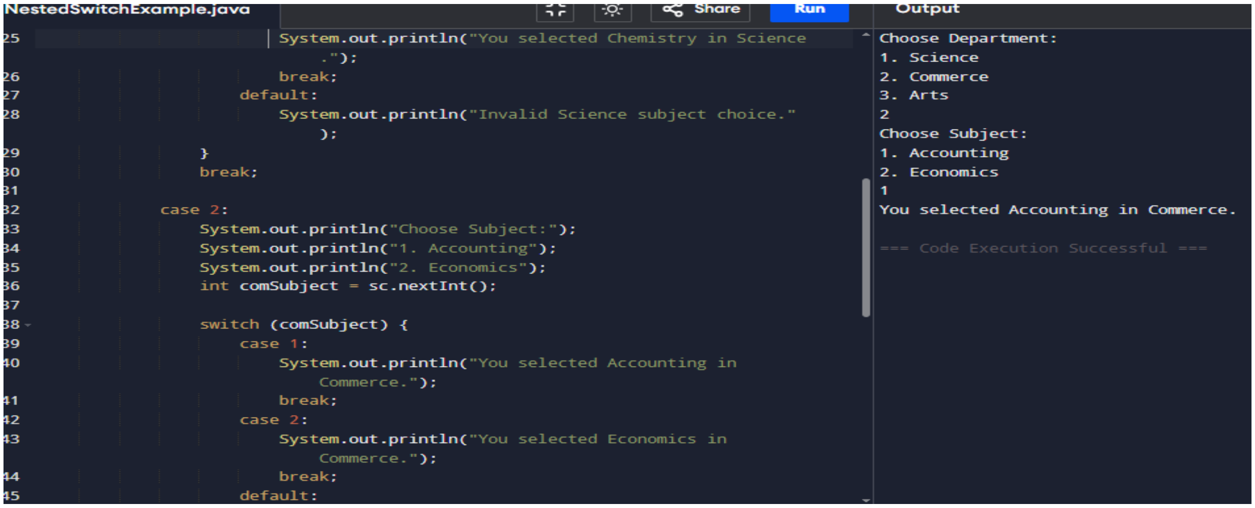
```
*********
 *******
  *****
   ***
    *

=== Code Execution Suc
```

6. Write a program for demonstrate Nested Switch



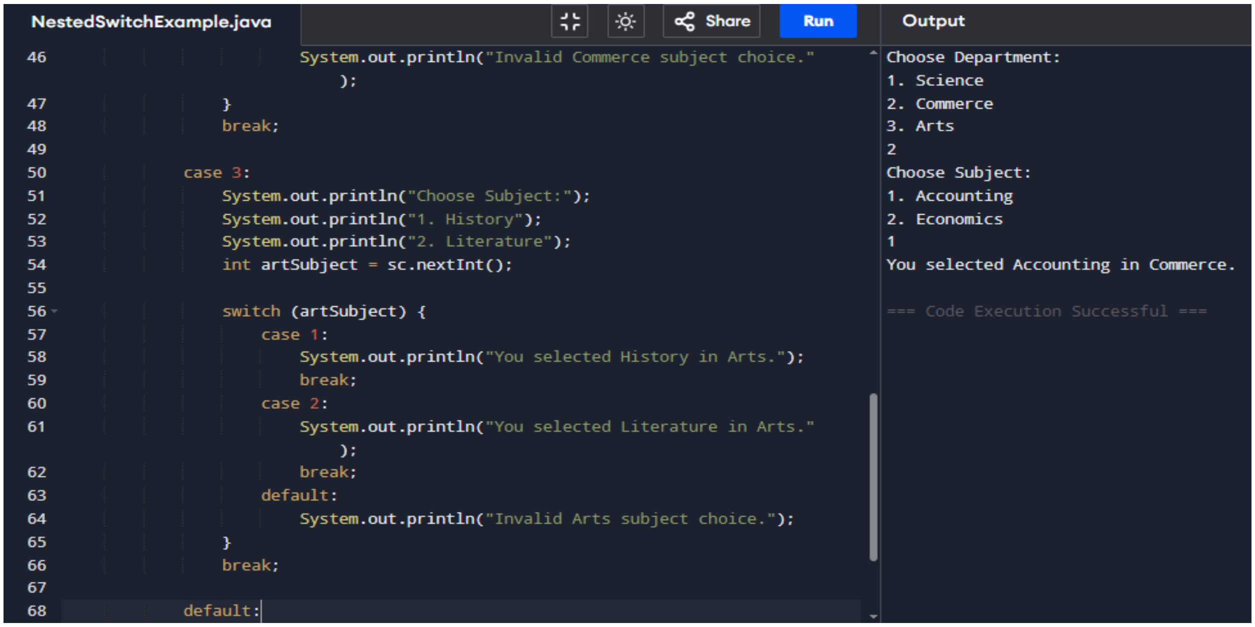
```java

public class NestedSwitchExample {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Choose Department:");
        System.out.println("1. Science");
        System.out.println("2. Commerce");
        System.out.println("3. Arts");
        int dept = sc.nextInt();

        switch (dept) {
            case 1:
                System.out.println("Choose Subject:");
                System.out.println("1. Physics");
                System.out.println("2. Chemistry");
                int sciSubject = sc.nextInt();

                switch (sciSubject) {
                    case 1:
                        System.out.println("You selected Physics in Science."
                            );
                        break;
                    case 2:
                        System.out.println("You selected Chemistry in Science
```

```
Choose Department:
1. Science
2. Commerce
3. Arts
2
Choose Subject:
1. Accounting
2. Economics
1
You selected Accounting in Commerce.

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



```java
                    System.out.println("You selected Chemistry in Science.");
                    break;
                default:
                    System.out.println("Invalid Science subject choice.");
            }
            break;

        case 2:
            System.out.println("Choose Subject:");
            System.out.println("1. Accounting");
            System.out.println("2. Economics");
            int comSubject = sc.nextInt();

            switch (comSubject) {
                case 1:
                    System.out.println("You selected Accounting in Commerce.");
                    break;
                case 2:
                    System.out.println("You selected Economics in Commerce.");
                    break;
                default:
```

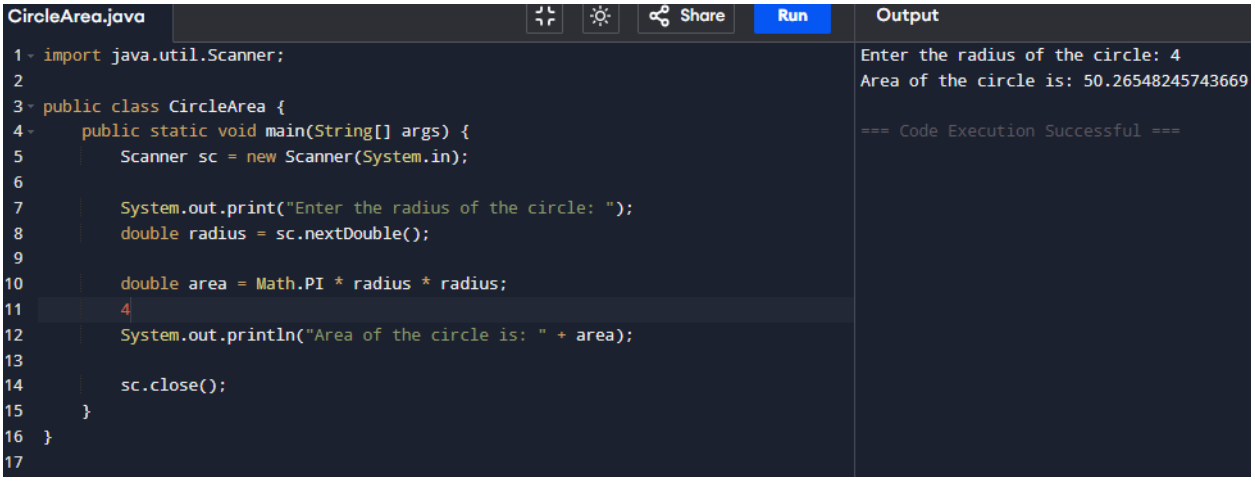
```
Choose Department:
1. Science
2. Commerce
3. Arts
2
Choose Subject:
1. Accounting
2. Economics
1
You selected Accounting in Commerce.

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



```java
                    System.out.println("Invalid Commerce subject choice.");
            }
            break;

        case 3:
            System.out.println("Choose Subject:");
            System.out.println("1. History");
            System.out.println("2. Literature");
            int artSubject = sc.nextInt();

            switch (artSubject) {
                case 1:
                    System.out.println("You selected History in Arts.");
                    break;
                case 2:
                    System.out.println("You selected Literature in Arts.");
                    break;
                default:
                    System.out.println("Invalid Arts subject choice.");
            }
            break;

        default:
```

```
Choose Department:
1. Science
2. Commerce
3. Arts
2
Choose Subject:
1. Accounting
2. Economics
1
You selected Accounting in Commerce.

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

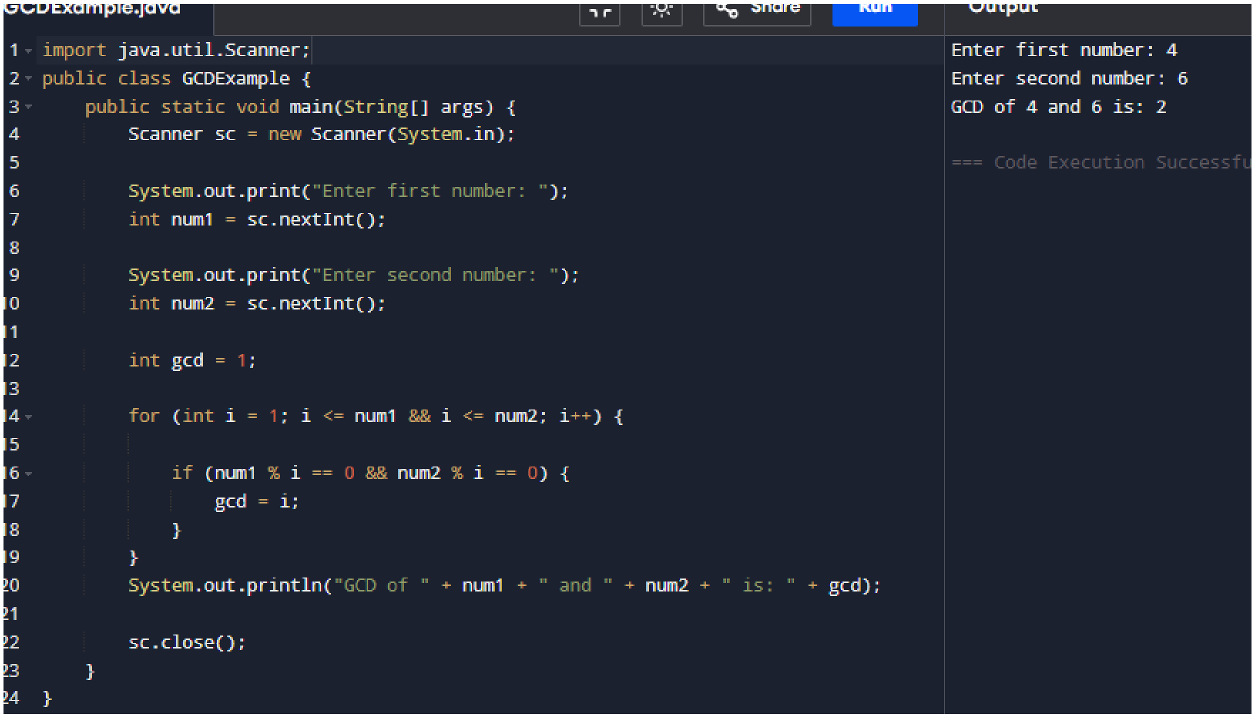
7. Write a program to calculate area of a circle using radius



```java
import java.util.Scanner;

public class CircleArea {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the radius of the circle: ");
        double radius = sc.nextDouble();

        double area = Math.PI * radius * radius;
        4
        System.out.println("Area of the circle is: " + area);

        sc.close();
    }
}
```

```
Enter the radius of the circle: 4
Area of the circle is: 50.26548245743669

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

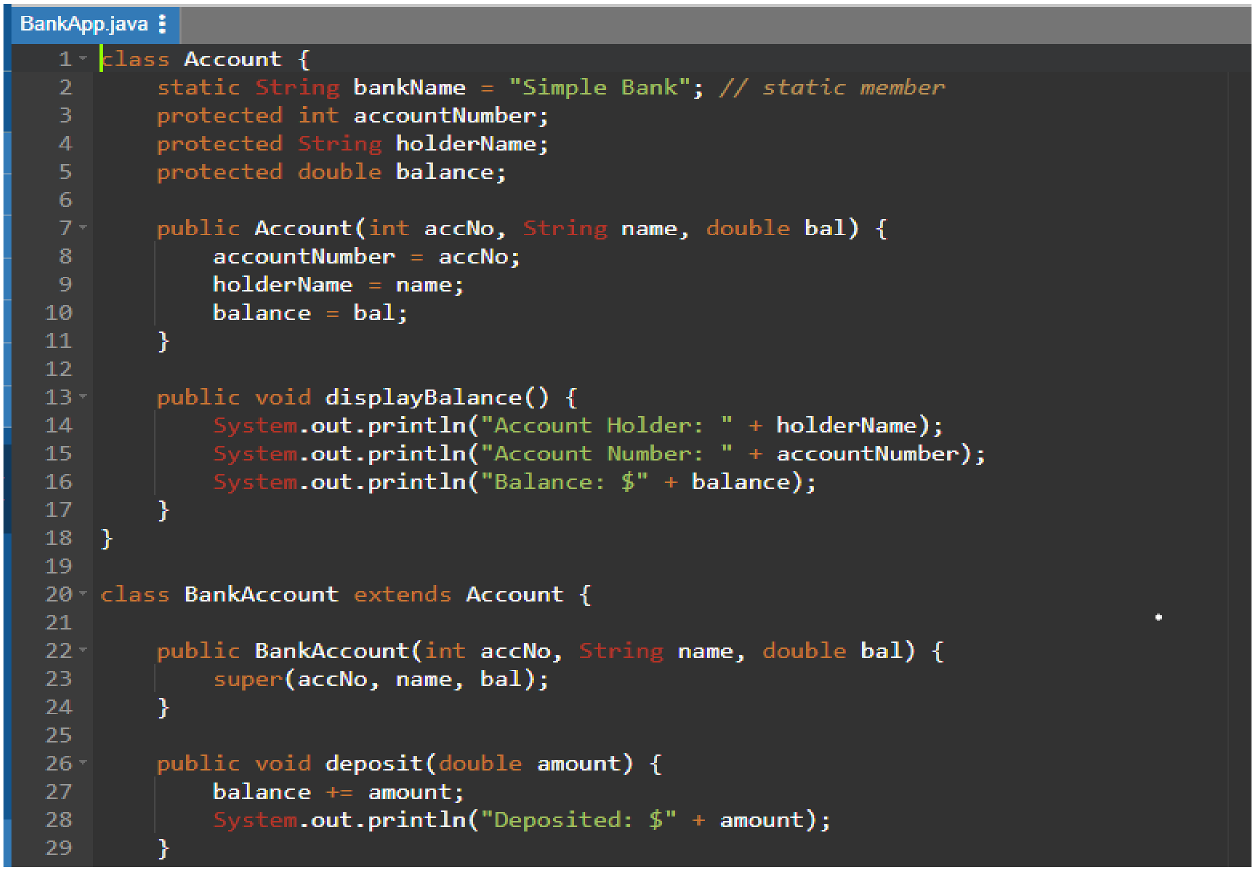
8. Write a program to find G.C.D of the number.



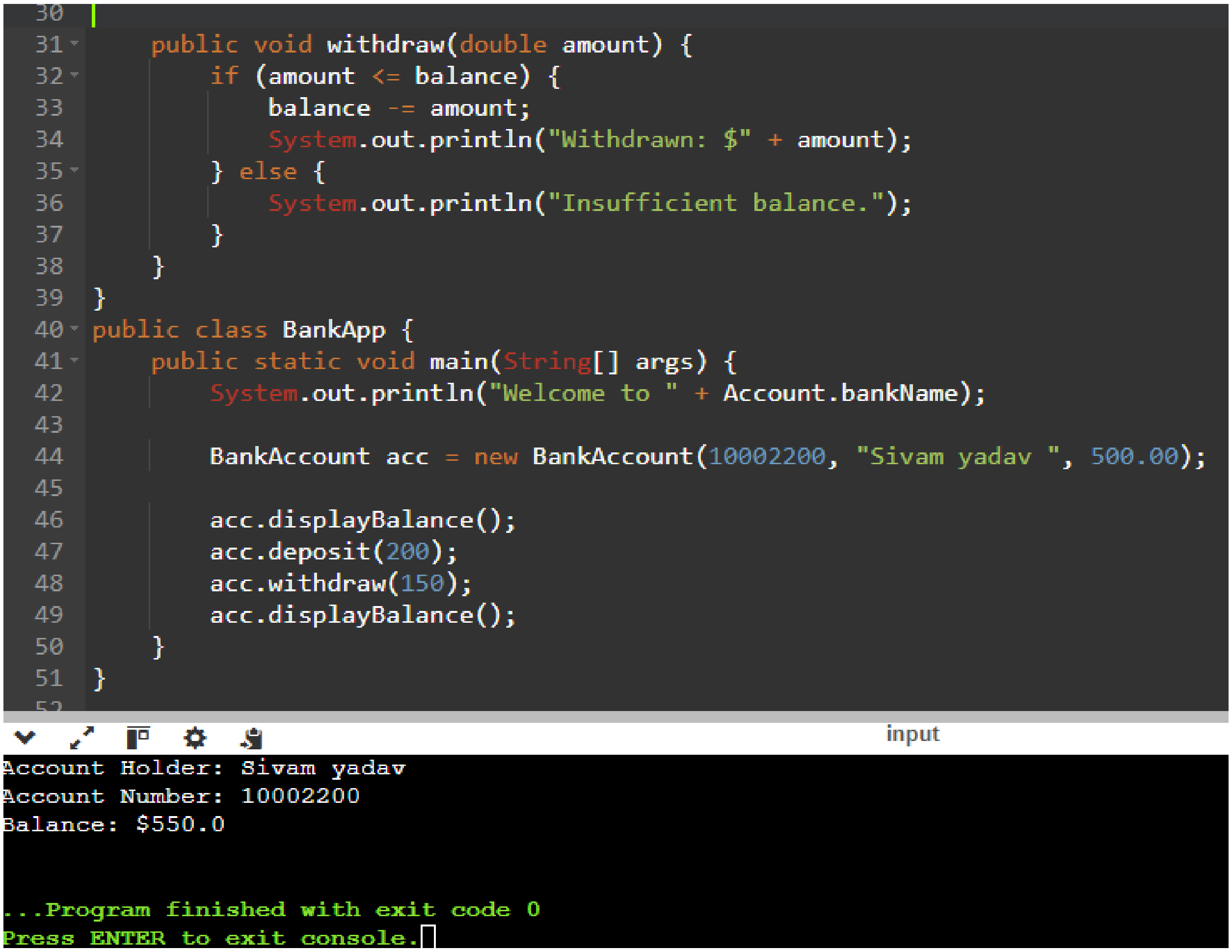
```java
import java.util.Scanner;
public class GCDExample {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter first number: ");
        int num1 = sc.nextInt();

        System.out.print("Enter second number: ");
        int num2 = sc.nextInt();

        int gcd = 1;

        for (int i = 1; i <= num1 && i <= num2; i++) {

            if (num1 % i == 0 && num2 % i == 0) {
                gcd = i;
            }
        }
        System.out.println("GCD of " + num1 + " and " + num2 + " is: " + gcd);

        sc.close();
    }
}
```

```
Enter first number: 4
Enter second number: 6
GCD of 4 and 6 is: 2

=== Code Execution Successfu
```

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



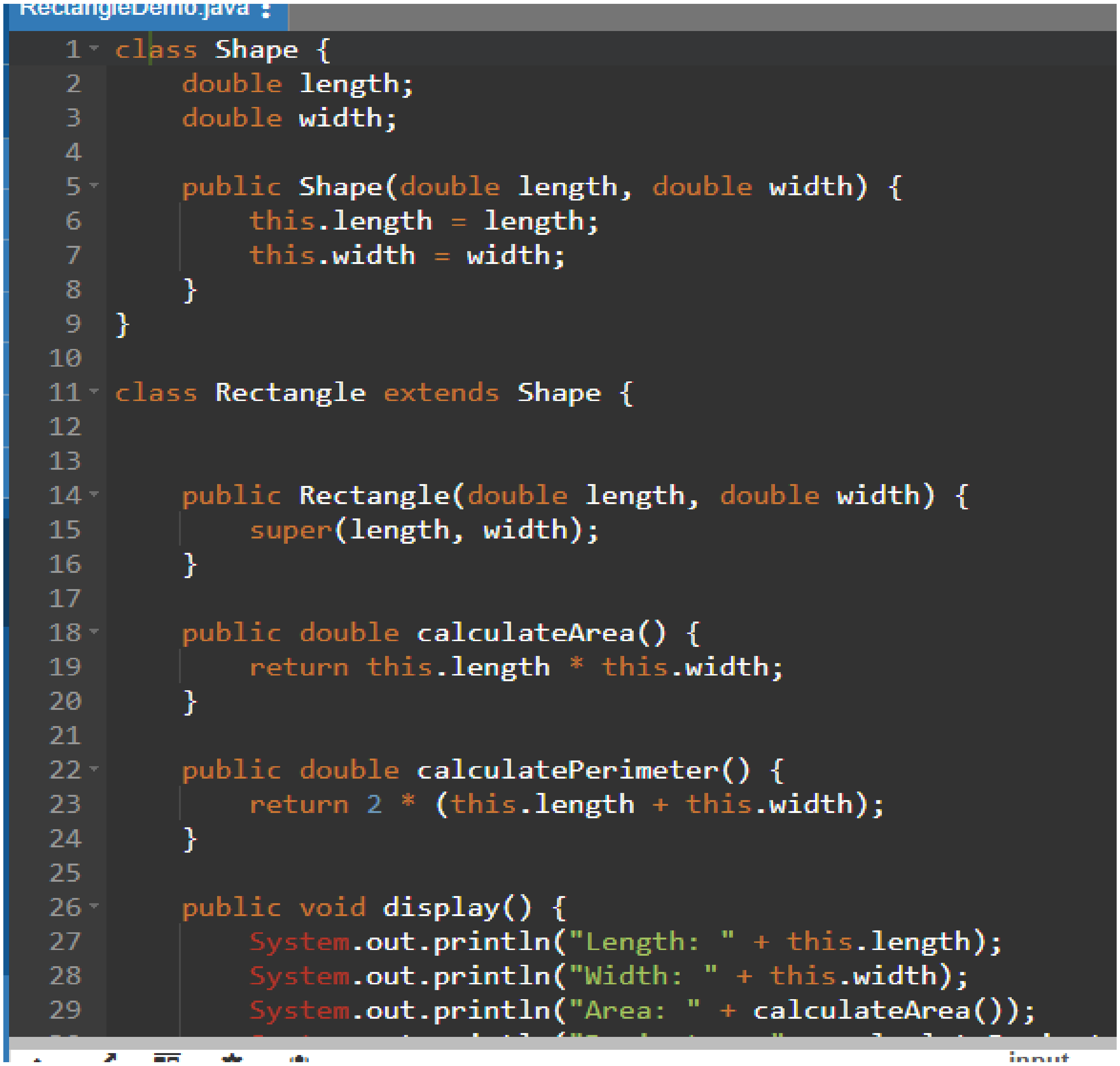
```java
class Account {
    static String bankName = "Simple Bank"; // static member
    protected int accountNumber;
    protected String holderName;
    protected double balance;

    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);
    }
```



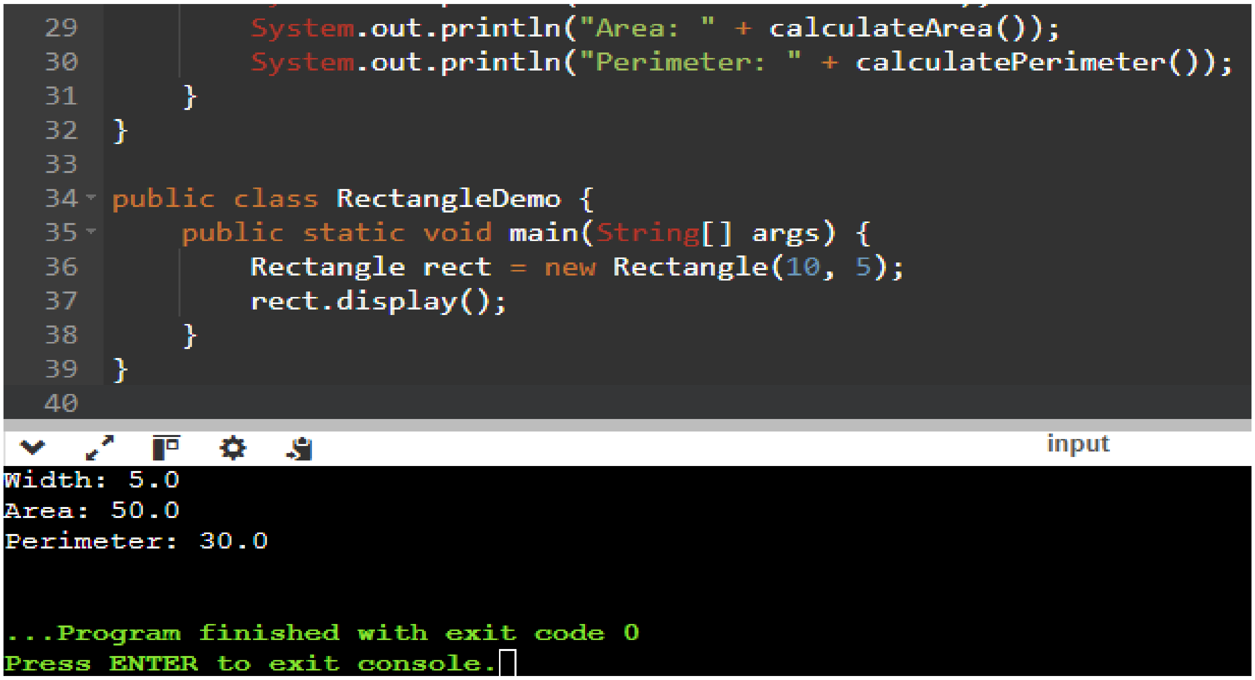
```
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}
public class BankApp {
    public static void 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();
    }
}
```

```
Account Holder: Sivam yadav
Account Number: 10002200
Balance: $550.0


...Program finished with exit code 0
Press 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.



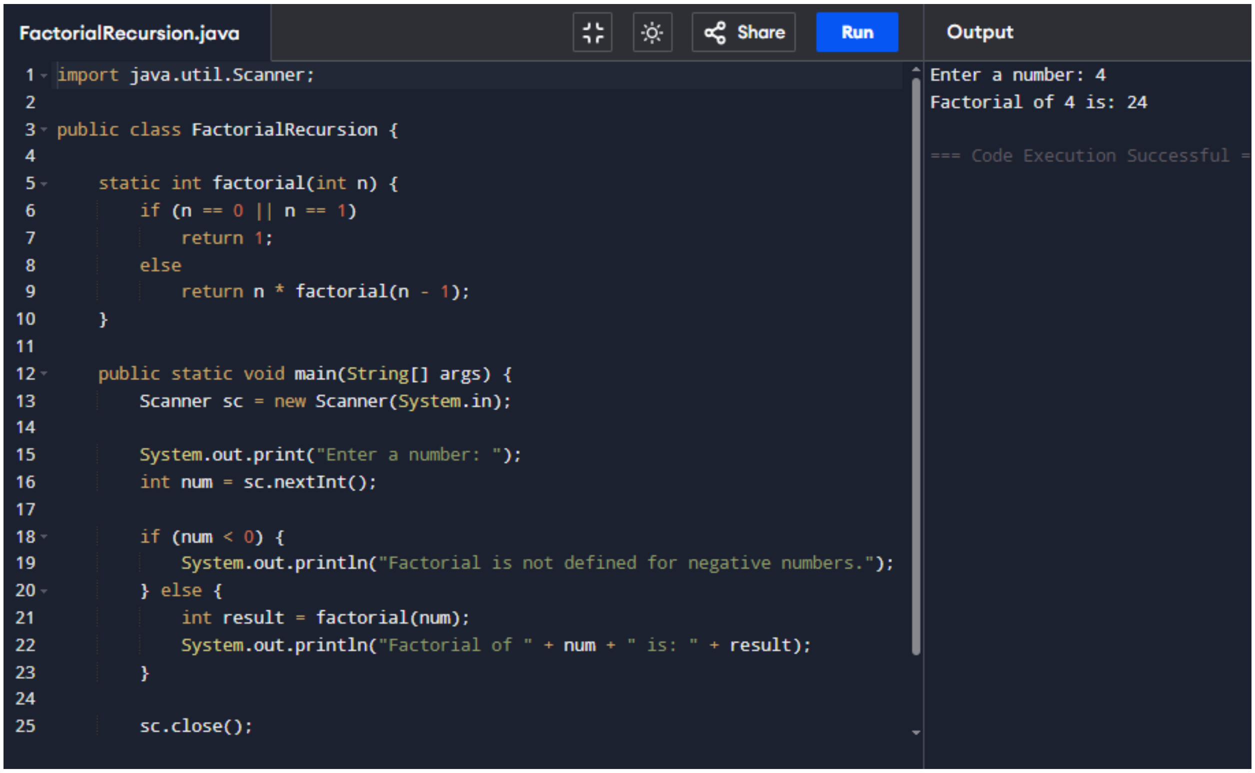
```java
class Shape {
    double length;
    double width;

    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("Area: " + calculateArea());
        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.
```

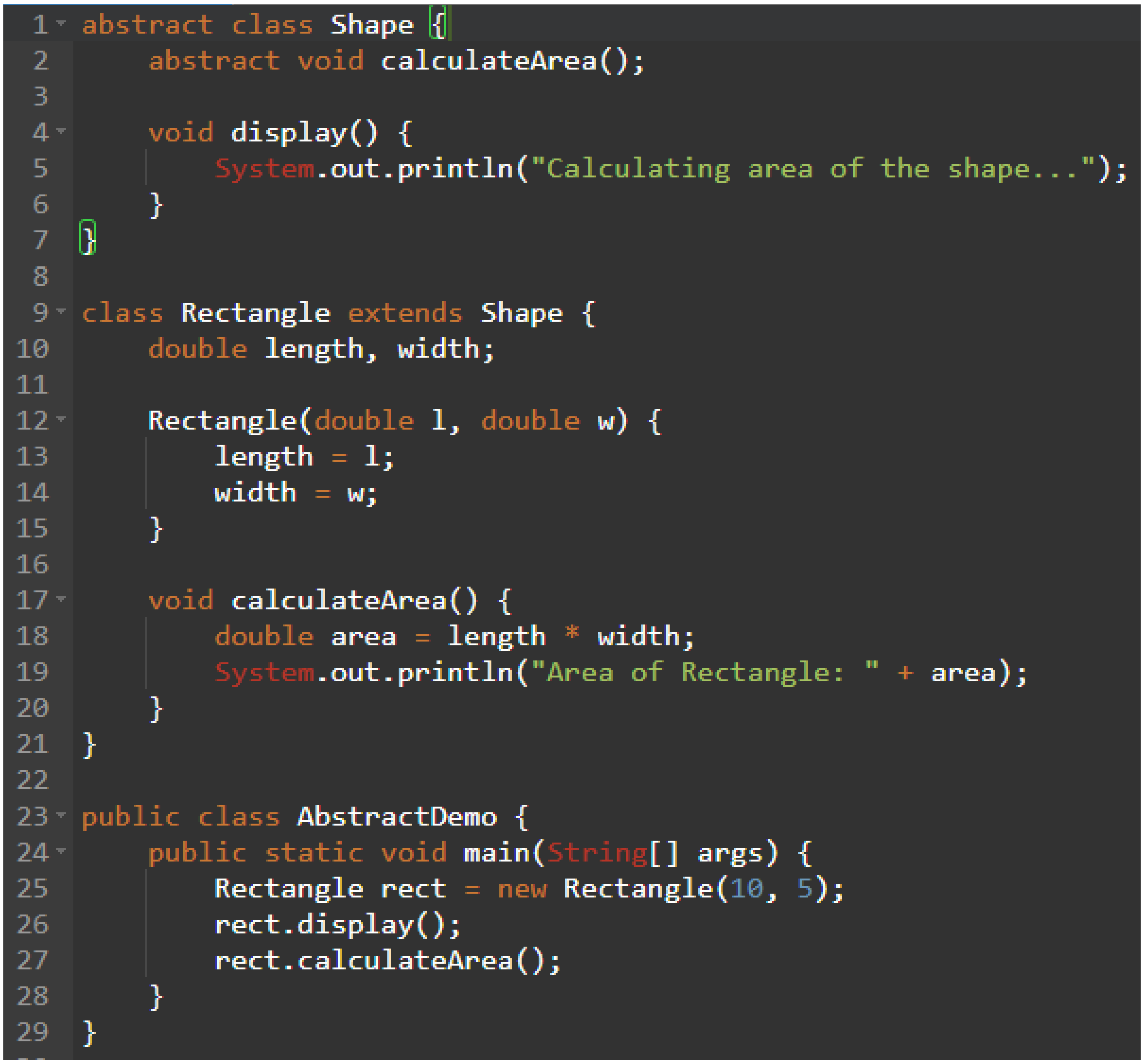
11. Write a program to find the factorial of a given number using recursion.



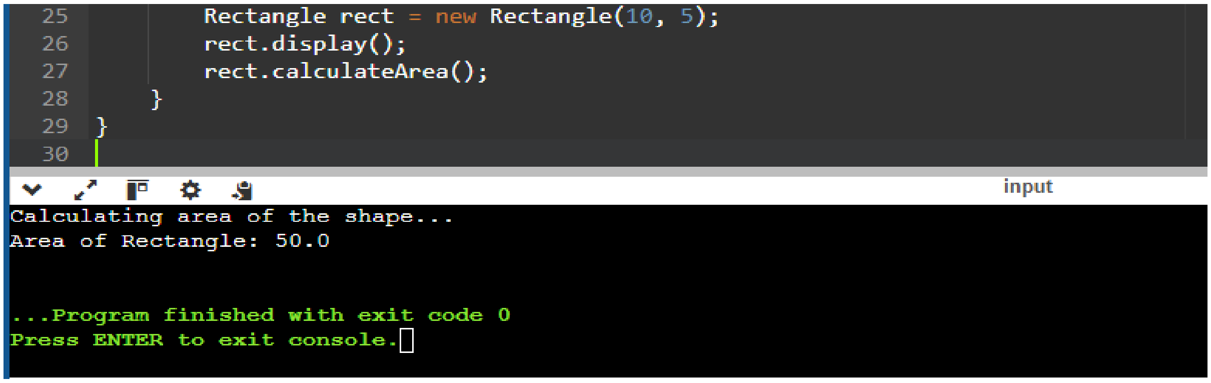
```java
import java.util.Scanner;

public class FactorialRecursion {

    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 sc = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int num = sc.nextInt();

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

        sc.close();
```

```
Enter a number: 4
Factorial of 4 is: 24

=== Code Execution Successful =
```

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



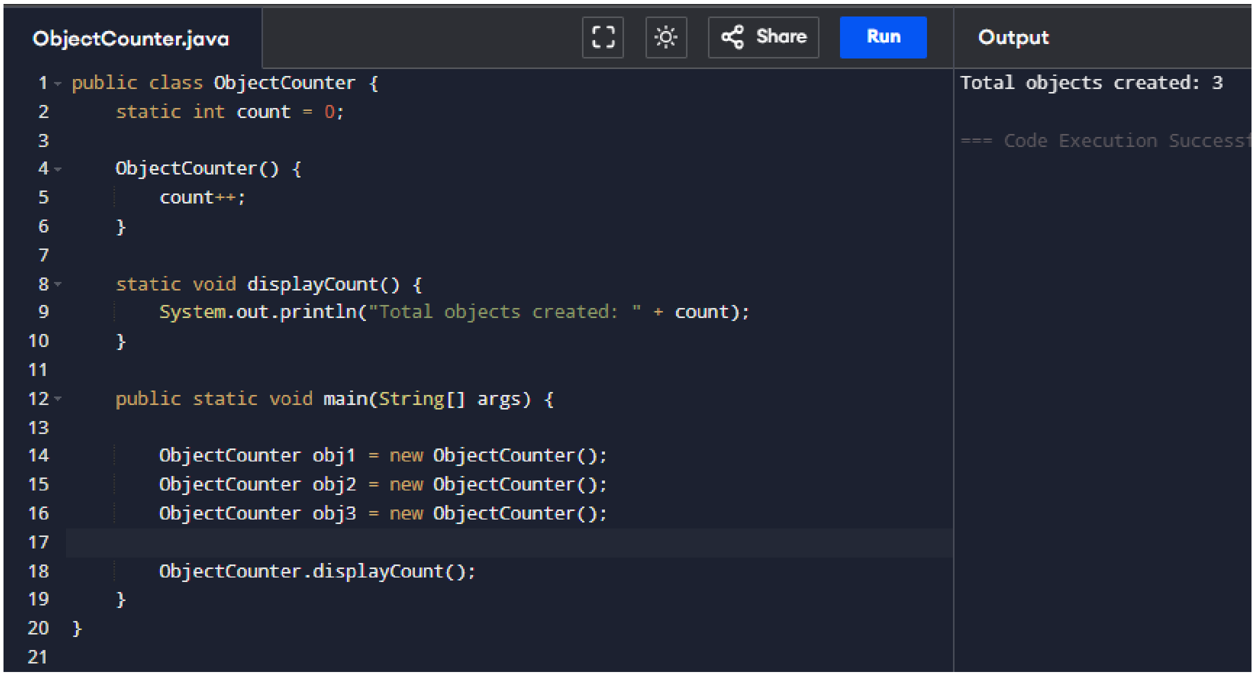
```java
abstract class Shape {
    abstract void calculateArea();

    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();
    }
}
```



```java
        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.
```

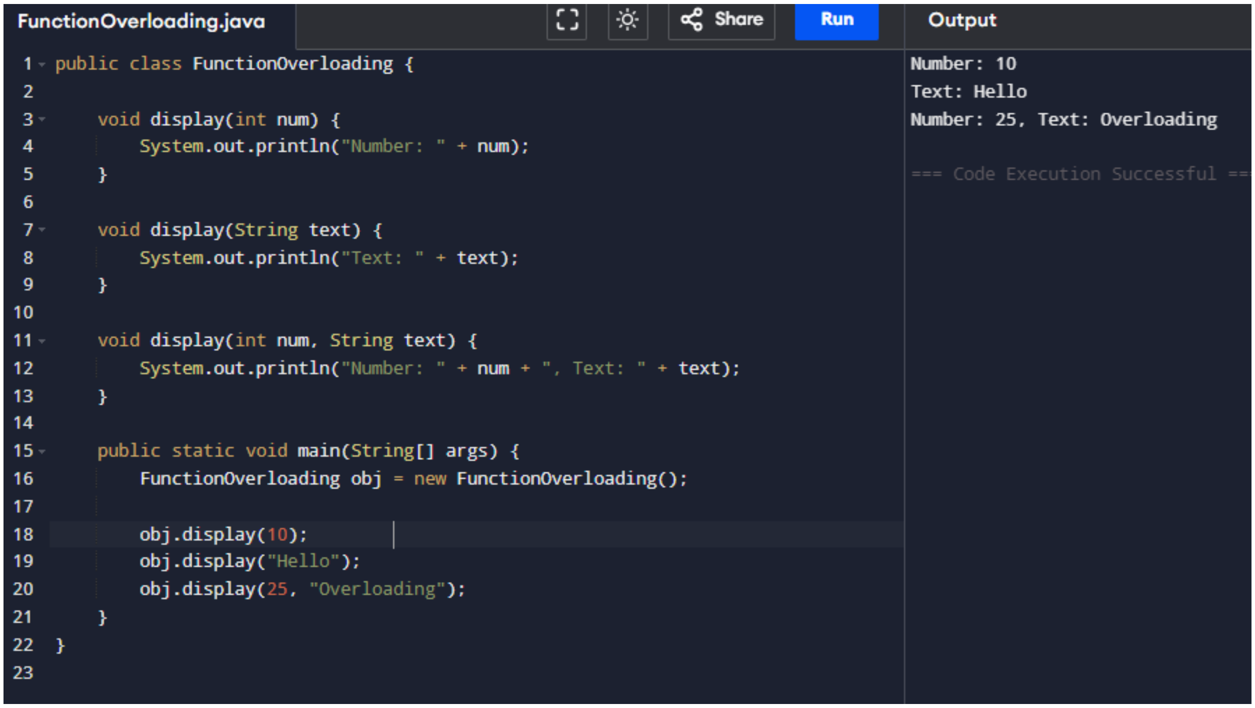
13. Write a program to count the number of objects created for a class using static member function



```java
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
```

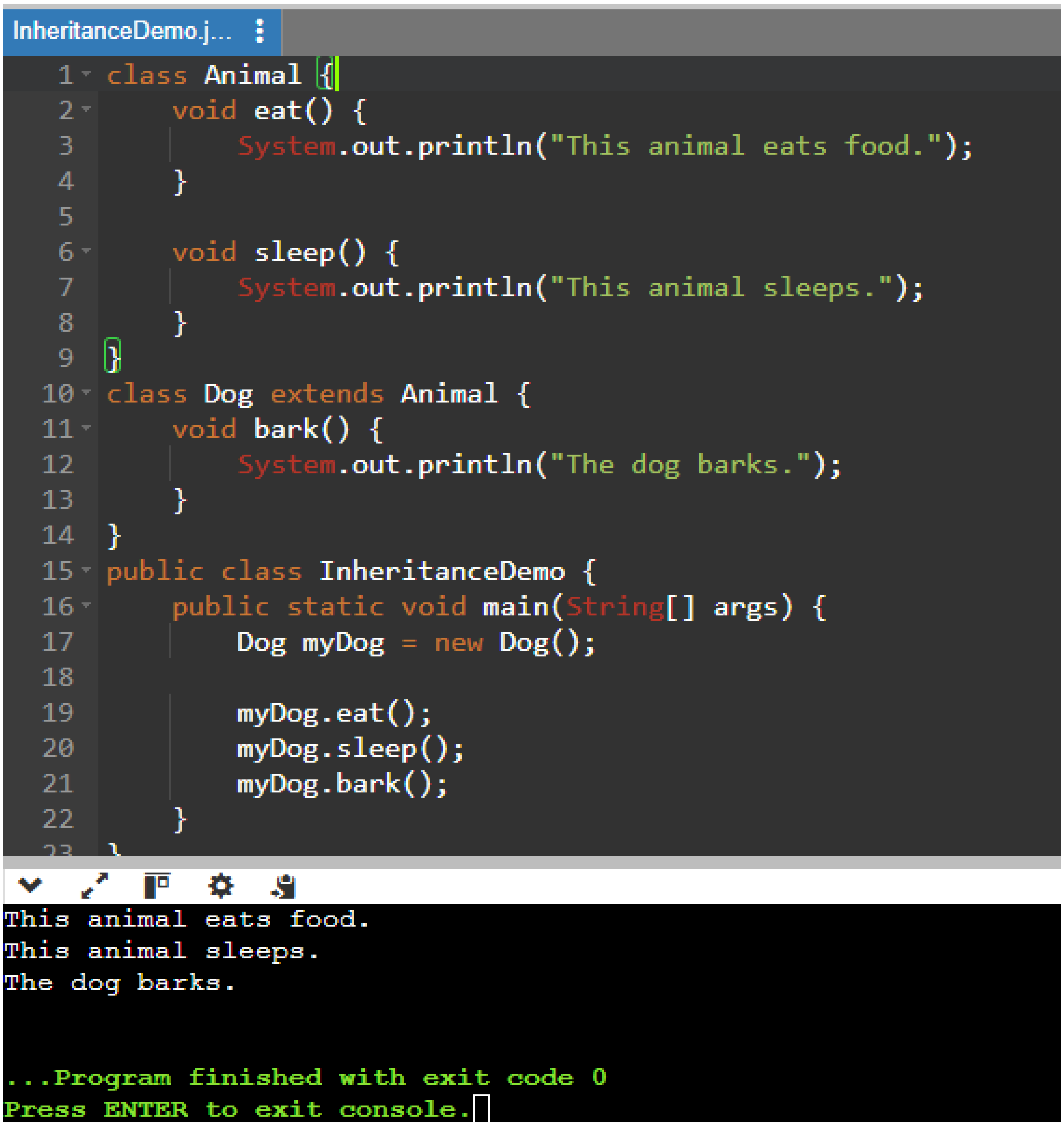
14. Write a program to demonstrate the use of function overloading.



```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 ==
```

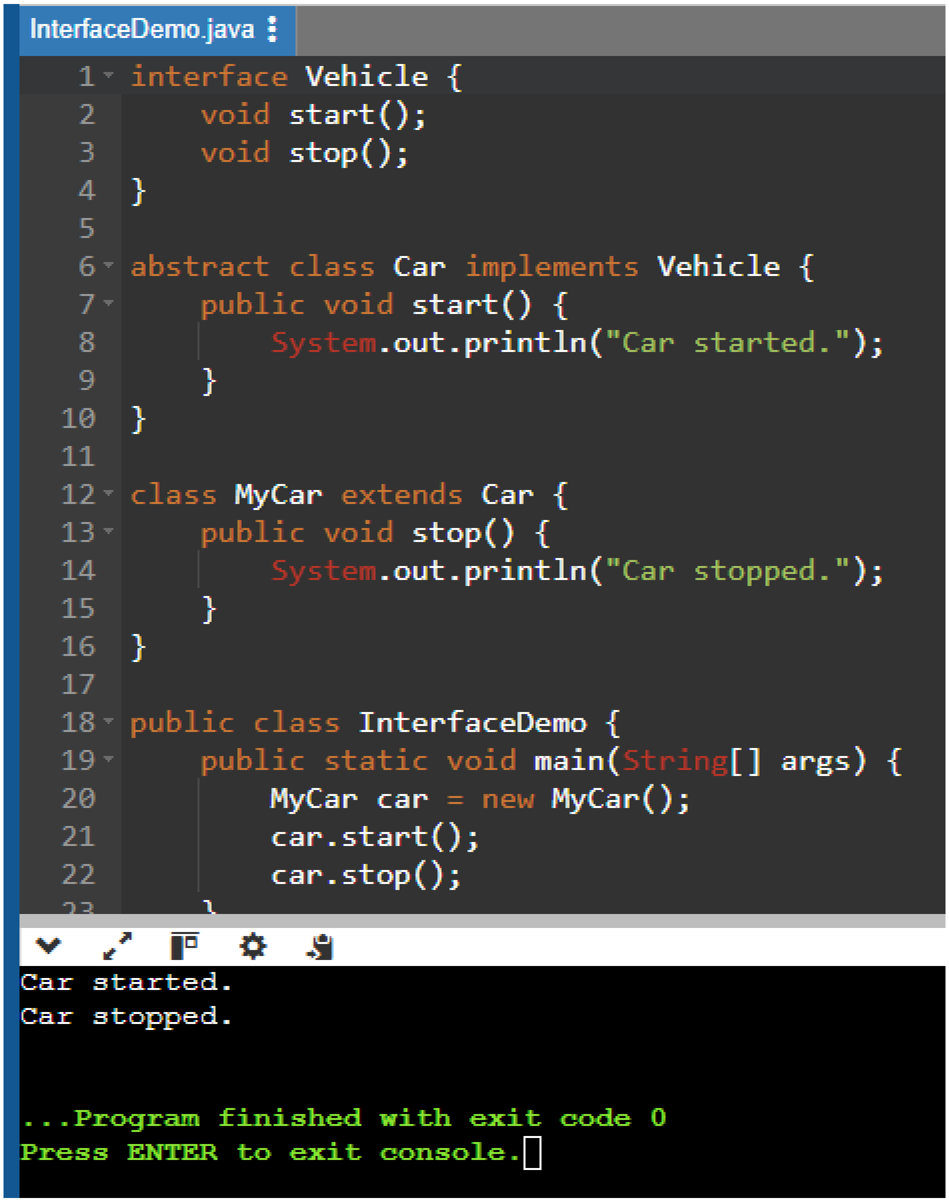
15. Write a program to demonstrate the use of inheritance



```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.
```

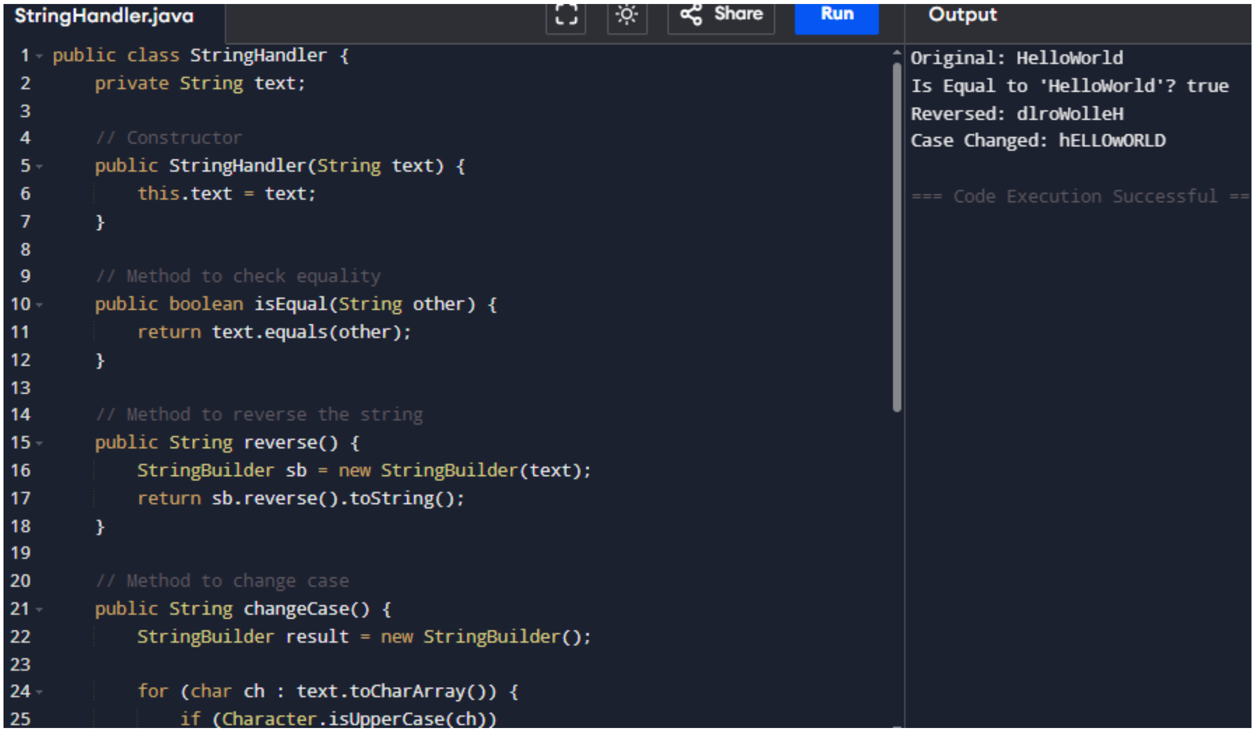
16. Write a program that show the partial implementation of Interface



```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();
    }
```

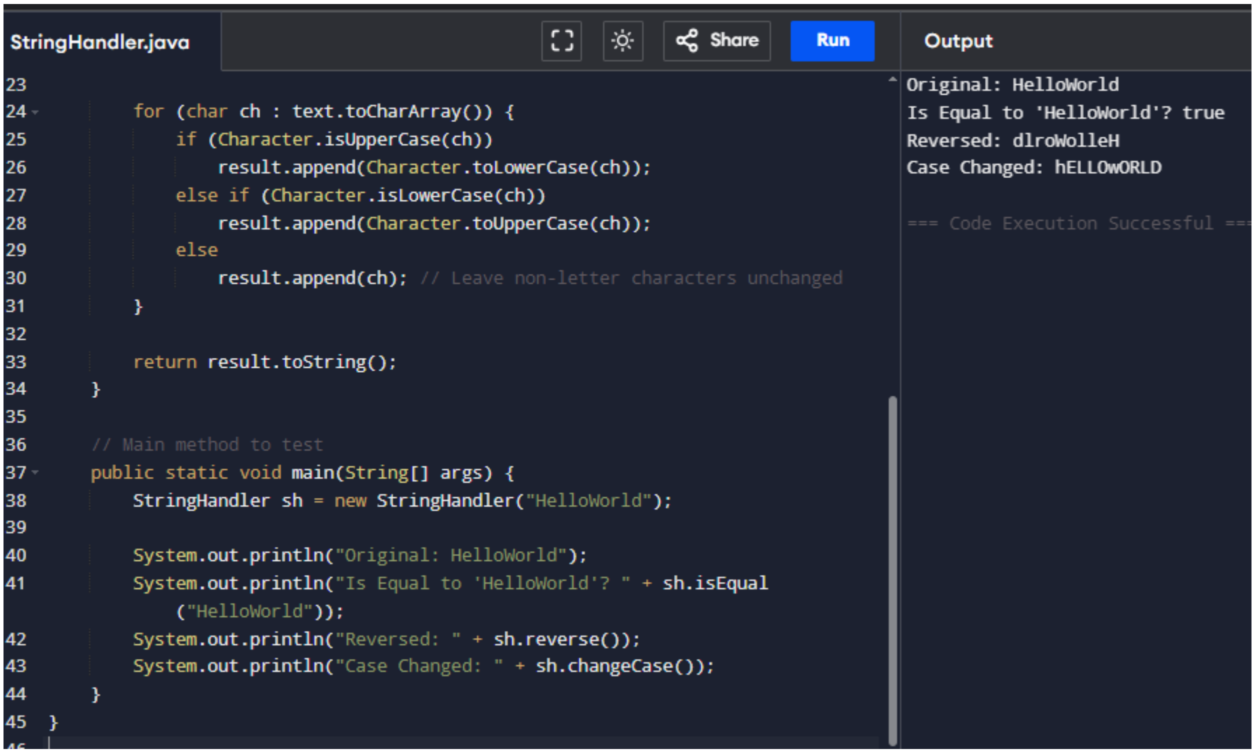
```
Car started.
Car stopped.


...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))
```

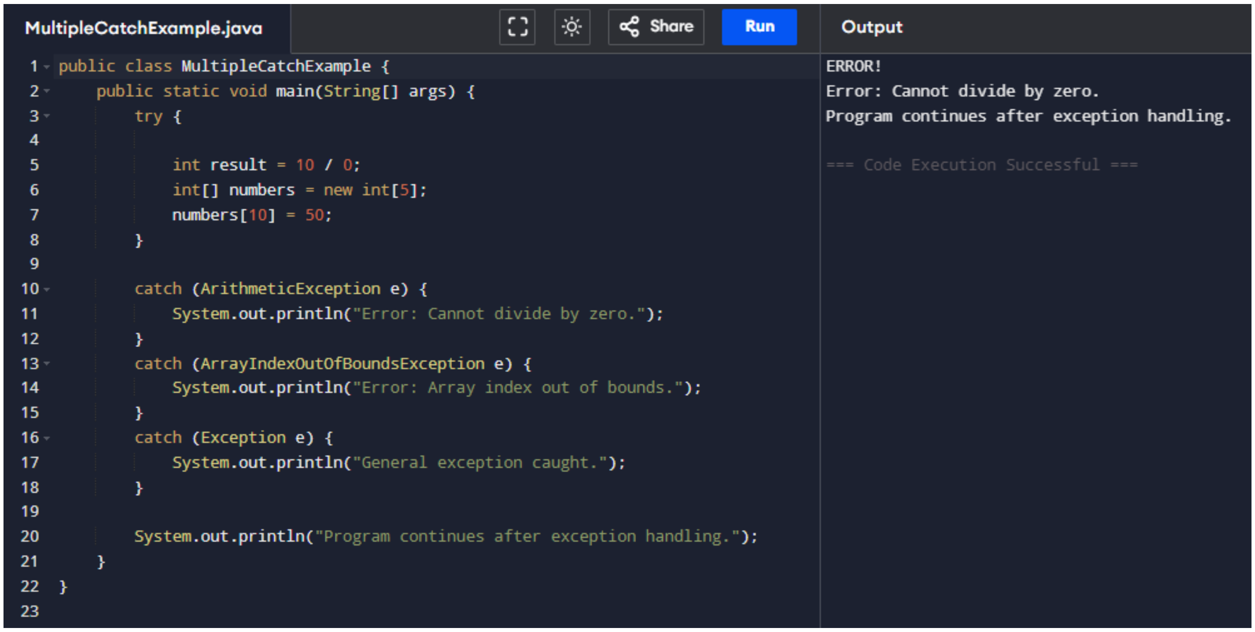
```
Original: HelloWorld
Is Equal to 'HelloWorld'? true
Reversed: dlroWolleH
Case Changed: hELLOwORLD

=== 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 ==
```

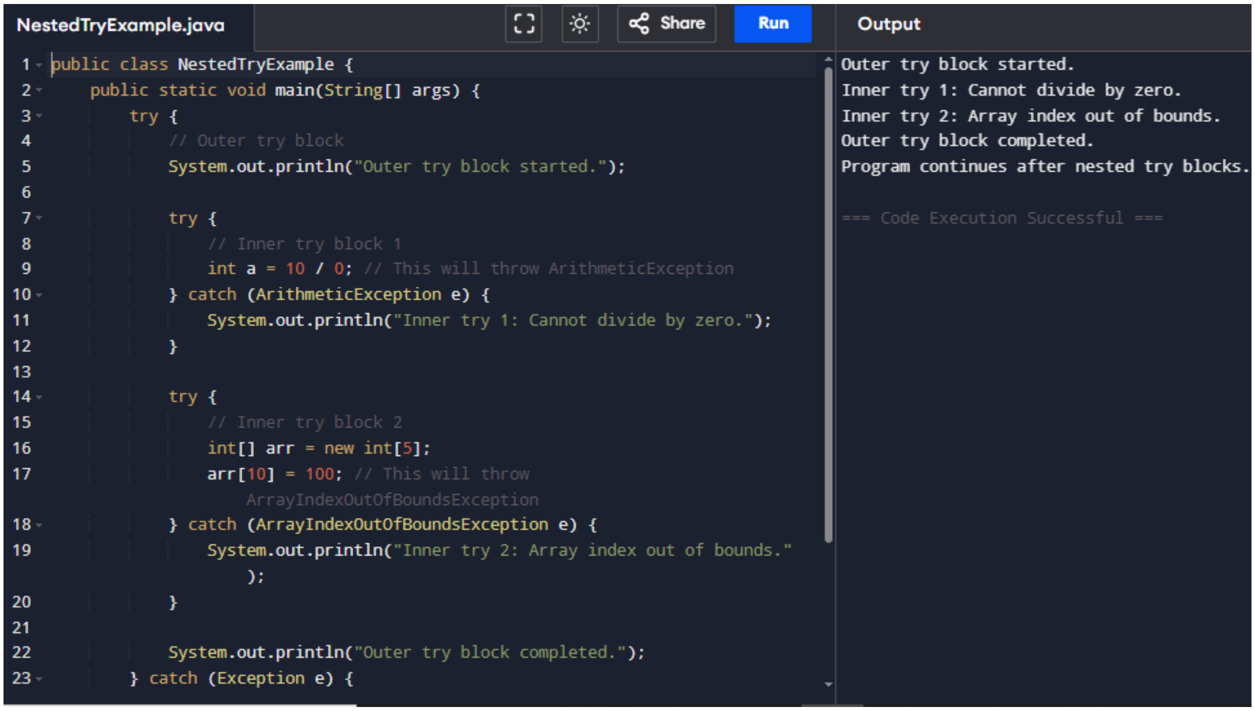
18. Write a program to handle the exception using try and multiple catch block.



```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.");
    }
}

```

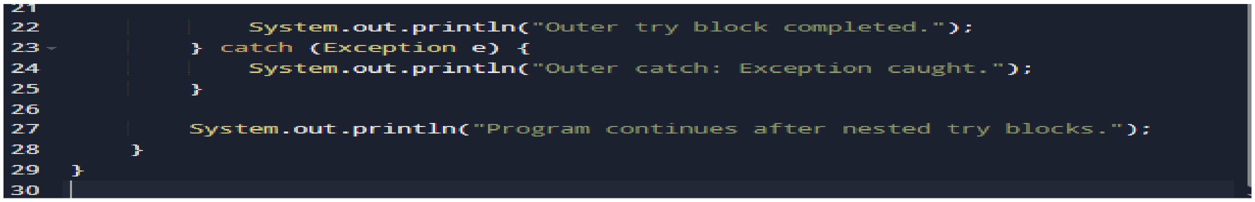
```
ERROR!
Error: Cannot divide by zero.
Program continues after exception handling.

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

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



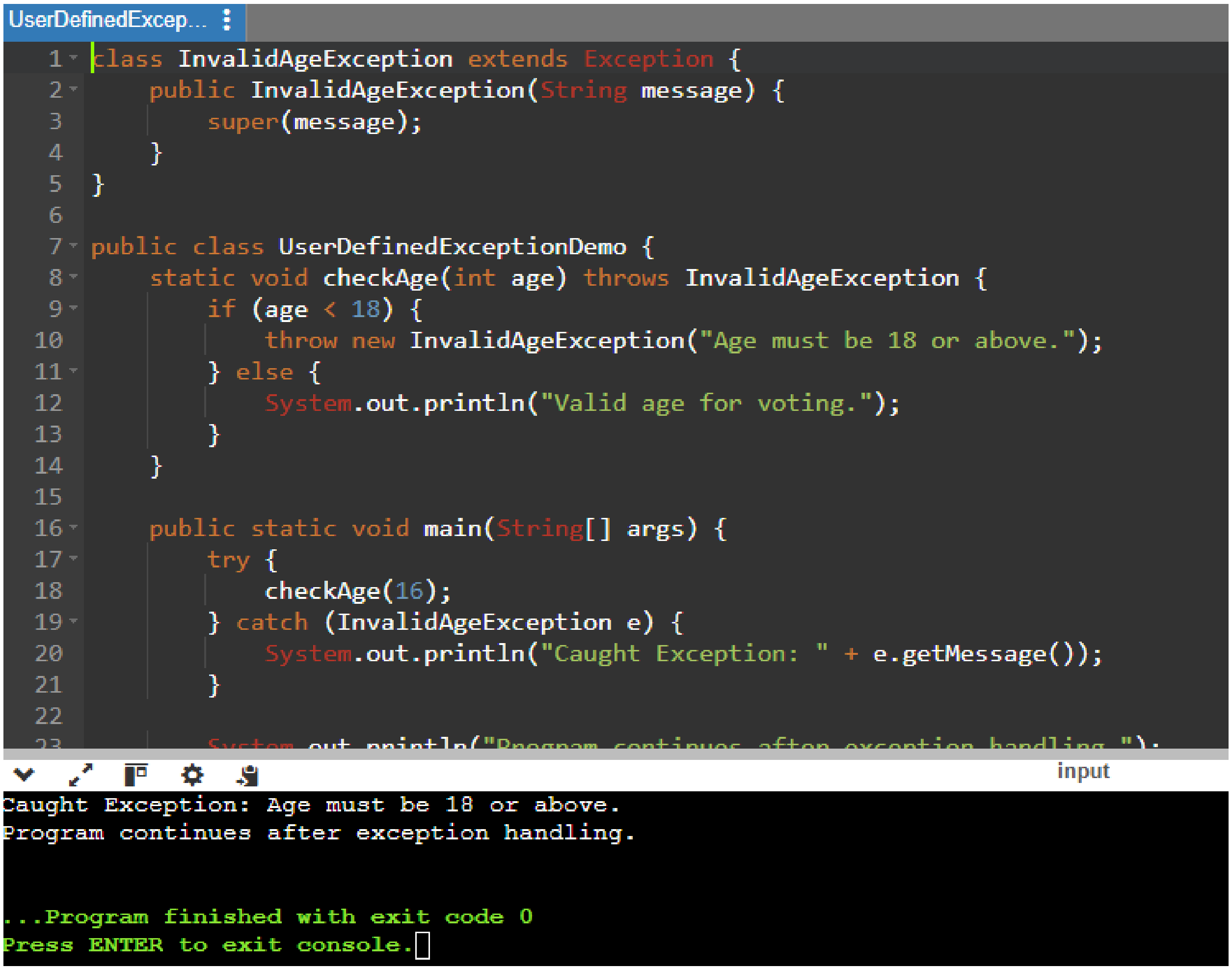
```java
public class NestedTryExample {
    public static void main(String[] args) {
        try {
            // Outer try block
            System.out.println("Outer try block started.");

            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 ===
```



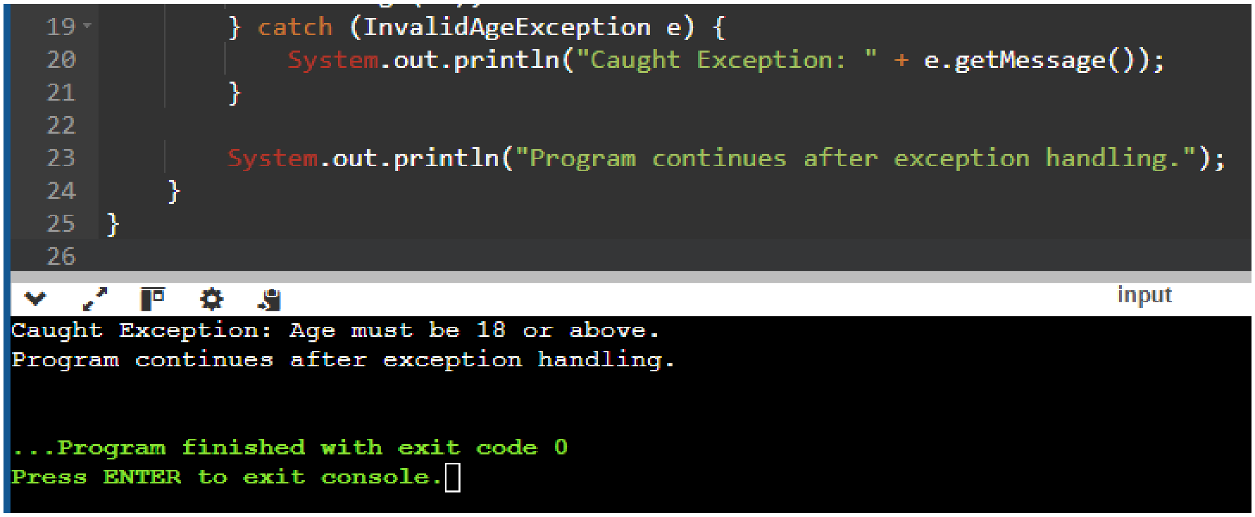
```
            System.out.println("Outer try block completed.");
        } catch (Exception e) {
            System.out.println("Outer catch: Exception caught.");
        }

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

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



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

public class UserDefinedExceptionDemo {
    static void checkAge(int age) throws InvalidAgeException {
        if (age < 18) {
            throw new InvalidAgeException("Age must be 18 or above.");
        } else {
            System.out.println("Valid age for voting.");
        }
    }

    public static void main(String[] args) {
        try {
            checkAge(16);
        } catch (InvalidAgeException e) {
            System.out.println("Caught Exception: " + e.getMessage());
        }

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

```
Caught Exception: Age must be 18 or above.
Program continues after exception handling.


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



```java
        } catch (InvalidAgeException e) {
            System.out.println("Caught Exception: " + e.getMessage());
        }

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

```

```
Caught Exception: Age must be 18 or above.
Program continues after exception handling.


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