

# Object Oriented Programming Language Using Java Lab.

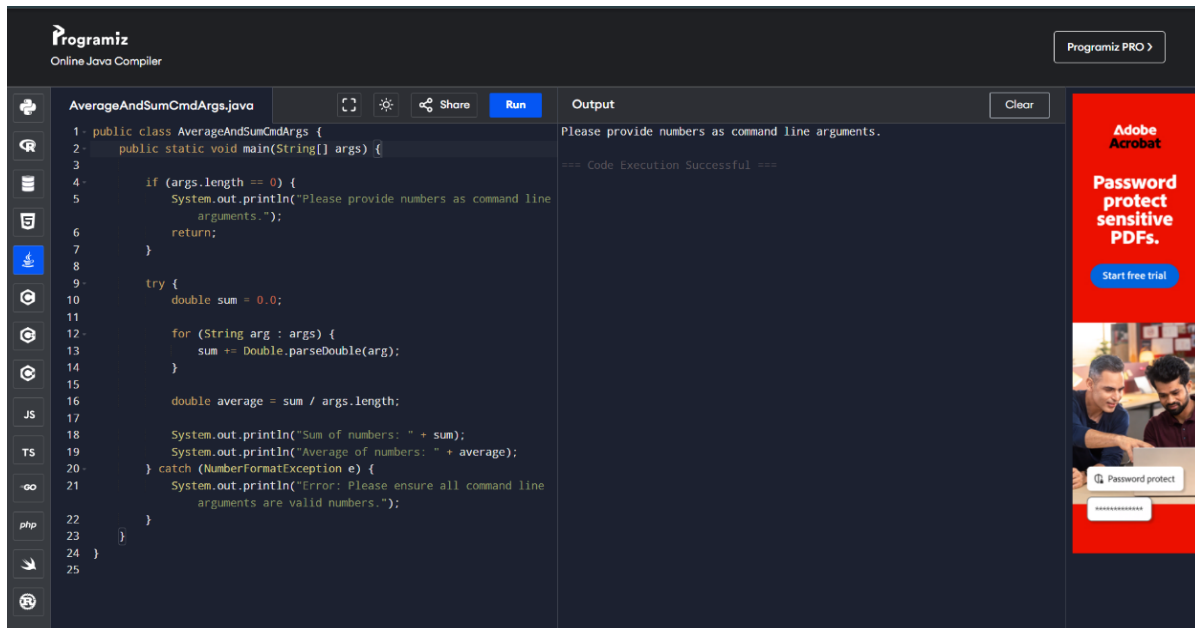
## Java Practical

### #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.

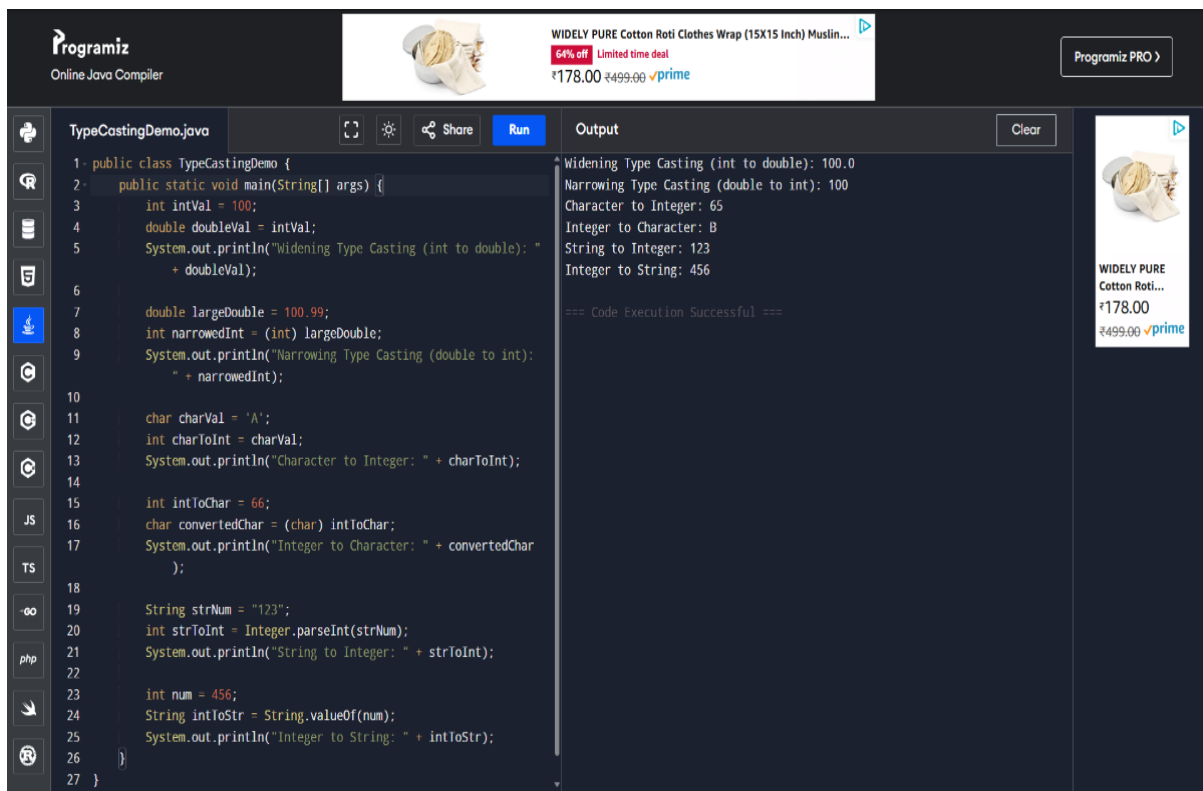


The screenshot shows the Programiz Online Java Compiler interface. The code editor contains a Java program named `AverageAndSumCmdArgs.java`. The program prompts the user to provide numbers as command line arguments. If no arguments are provided, it prints an error message. If arguments are provided, it calculates the sum and average of the numbers and prints them. The output window shows the message "Please provide numbers as command line arguments." and "=== Code Execution Successful ===".

```
1- public class AverageAndSumCmdArgs {
2-     public static void main(String[] args) {
3-
4-         if (args.length == 0) {
5-             System.out.println("Please provide numbers as command line arguments.");
6-             return;
7-         }
8-
9-         try {
10-            double sum = 0.0;
11-
12-            for (String arg : args) {
13-                sum += Double.parseDouble(arg);
14-            }
15-
16-            double average = sum / args.length;
17-
18-            System.out.println("Sum of numbers: " + sum);
19-            System.out.println("Average of numbers: " + average);
20-        } catch (NumberFormatException e) {
21-            System.out.println("Error: Please ensure all command line arguments are valid numbers.");
22-        }
23-    }
24- }
25
```

Output: Please provide numbers as command line arguments.  
=== Code Execution Successful ===

## 2. Write a program to demonstrate type casting.

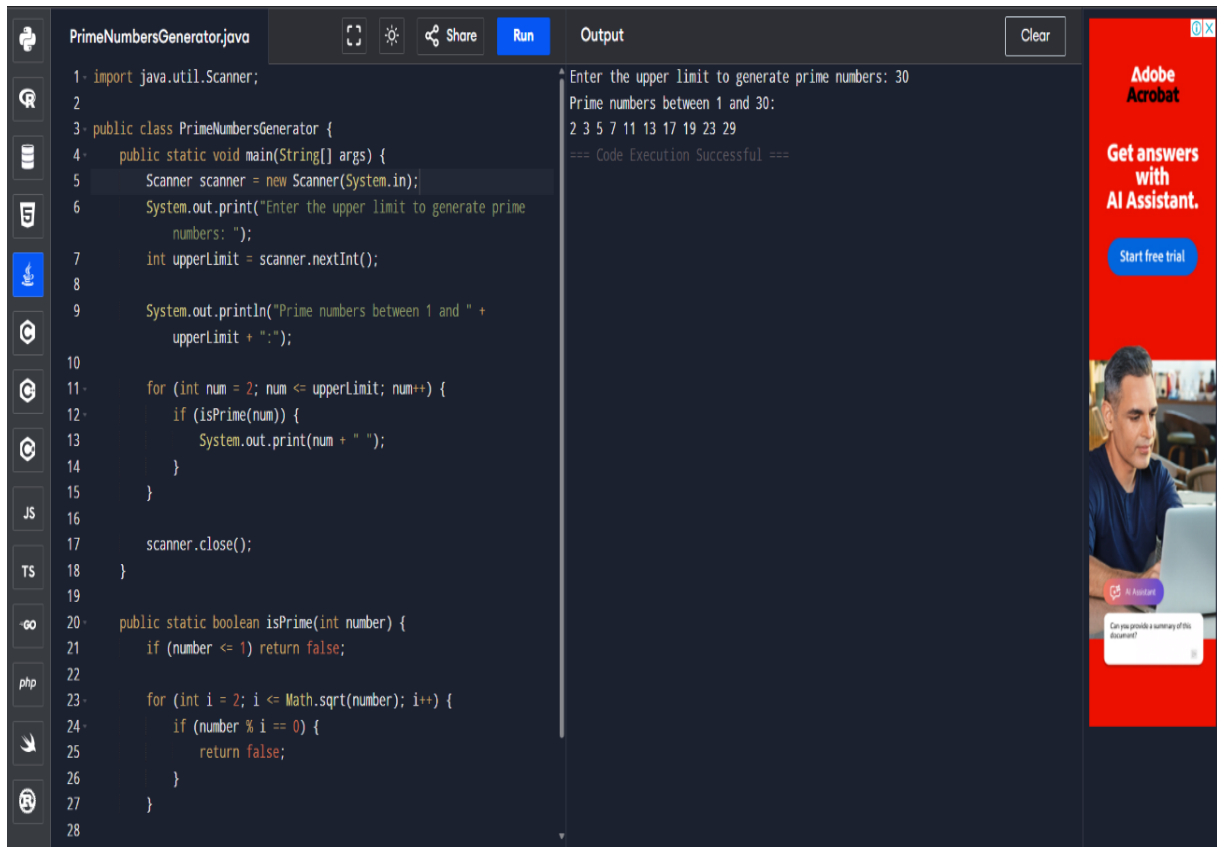


The screenshot shows the Programiz Online Java Compiler interface. The code editor contains a Java program named `TypeCastingDemo.java`. The program demonstrates various types of type casting: Widening Type Casting (int to double), Narrowing Type Casting (double to int), Character to Integer, Integer to Character, String to Integer, Integer to String, and Integer to Character. The output window shows the results of these casting operations and "=== Code Execution Successful ===".

```
1- public class TypeCastingDemo {
2-     public static void main(String[] args) {
3-         int intVal = 100;
4-         double doubleVal = intVal;
5-         System.out.println("Widening Type Casting (int to double): "
6-             + doubleVal);
7-
8-         double largeDouble = 100.99;
9-         int narrowedInt = (int) largeDouble;
10-        System.out.println("Narrowing Type Casting (double to int): "
11-            + narrowedInt);
12-
13-        char charVal = 'A';
14-        int charToInt = charVal;
15-        System.out.println("Character to Integer: " + charToInt);
16-
17-        int intToChar = 66;
18-        char convertedChar = (char) intToChar;
19-        System.out.println("Integer to Character: " + convertedChar);
20-
21-        String strNum = "123";
22-        int strToInt = Integer.parseInt(strNum);
23-        System.out.println("String to Integer: " + strToInt);
24-
25-        int num = 456;
26-        String intToStr = String.valueOf(num);
27-        System.out.println("Integer to String: " + intToStr);
28-    }
29- }
```

Output: 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.



The screenshot shows a Java IDE with a file named `PrimeNumbersGenerator.java`. The code is as follows:

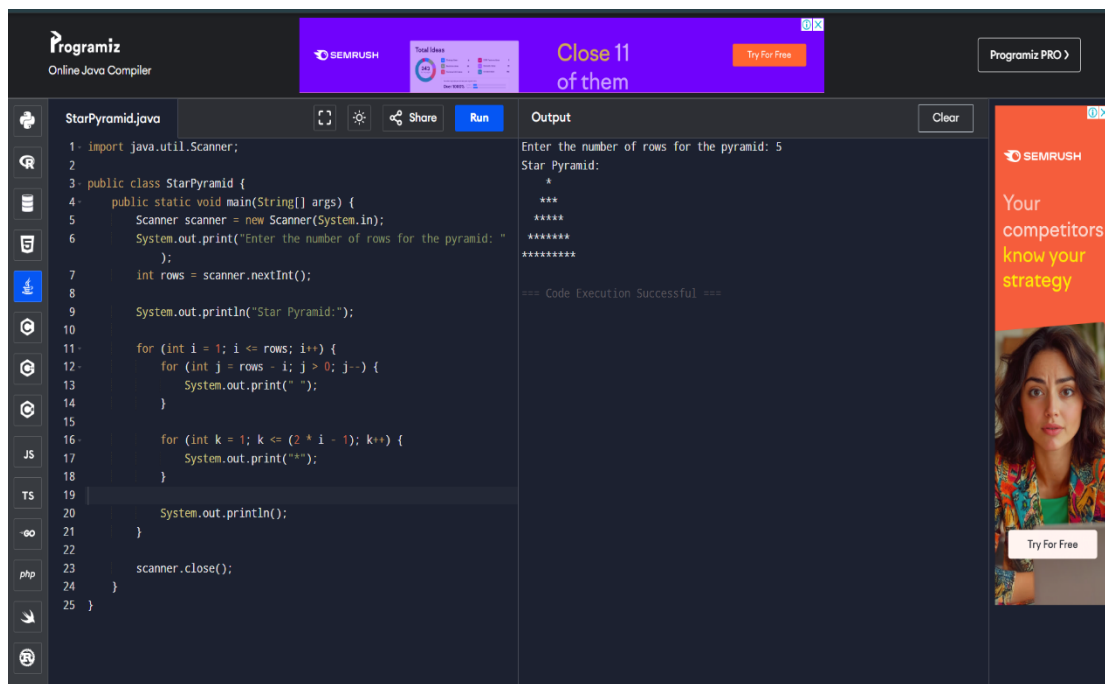
```
1 import java.util.Scanner;
2
3 public class PrimeNumbersGenerator {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Enter the upper limit to generate prime
          numbers: ");
7         int upperLimit = scanner.nextInt();
8
9         System.out.println("Prime numbers between 1 and " +
          upperLimit + ":");
10
11         for (int num = 2; num <= upperLimit; num++) {
12             if (isPrime(num)) {
13                 System.out.print(num + " ");
14             }
15         }
16
17         scanner.close();
18     }
19
20     public static boolean isPrime(int number) {
21         if (number <= 1) return false;
22
23         for (int i = 2; i <= Math.sqrt(number); i++) {
24             if (number % i == 0) {
25                 return false;
26             }
27         }
28     }
29 }
```

The output window shows the following text:

```
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 ===
```

On the right side of the IDE, there is a red sidebar with an Adobe Acrobat logo and a button that says "Get answers with AI Assistant. Start free trial". Below this, there is a small video player showing a man at a laptop.

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



The screenshot shows a Java IDE with a file named `StarPyramid.java`. The code is as follows:

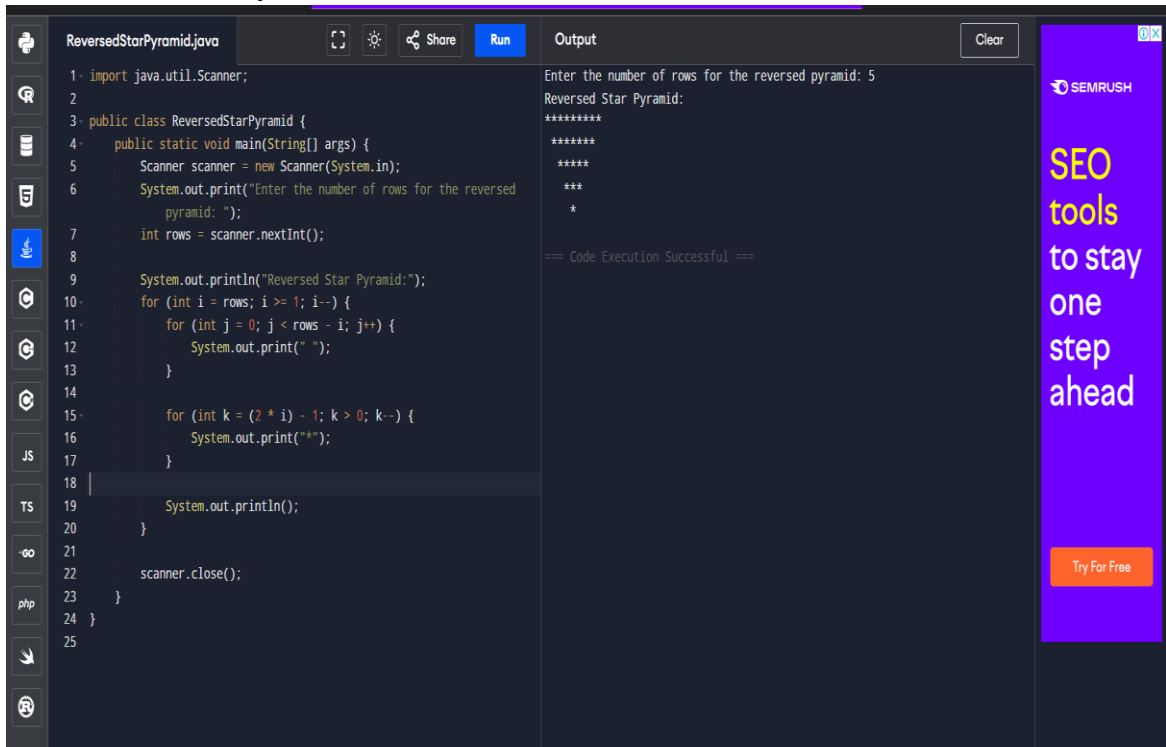
```
1 import java.util.Scanner;
2
3 public class StarPyramid {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Enter the number of rows for the pyramid: ");
7         int rows = scanner.nextInt();
8
9         System.out.println("Star Pyramid:");
10
11         for (int i = 1; i <= rows; i++) {
12             for (int j = rows - i; j > 0; j--) {
13                 System.out.print(" ");
14             }
15
16             for (int k = 1; k <= (2 * i - 1); k++) {
17                 System.out.print("*");
18             }
19
20             System.out.println();
21         }
22
23         scanner.close();
24     }
25 }
```

The output window shows the following text:

```
Enter the number of rows for the pyramid: 5
Star Pyramid:
*
***
*****
*****
*****
=== Code Execution Successful ===
```

On the right side of the IDE, there is a red sidebar with a SEMRUSH logo and a button that says "Your competitors know your strategy". Below this, there is a small video player showing a woman with long dark hair.

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



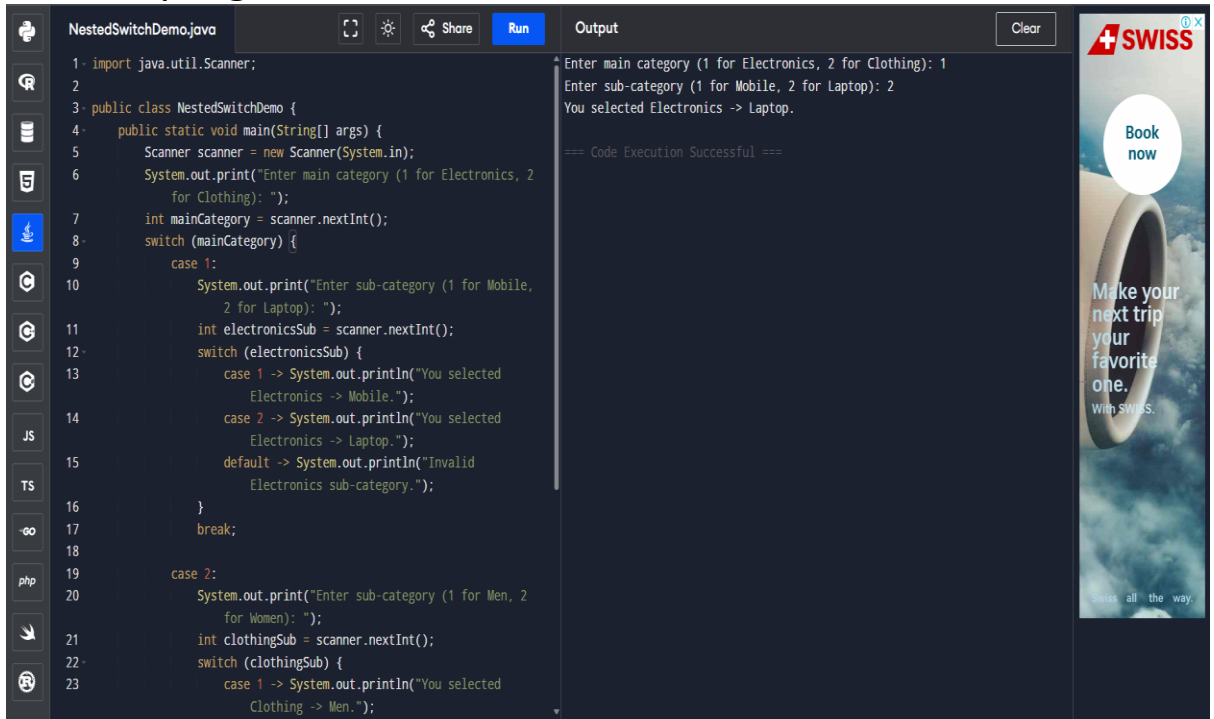
```
ReversedStarPyramid.java
1- import java.util.Scanner;
2-
3- public class ReversedStarPyramid {
4-     public static void main(String[] args) {
5-         Scanner scanner = new Scanner(System.in);
6-         System.out.print("Enter the number of rows for the reversed
pyramid: ");
7-         int rows = scanner.nextInt();
8-
9-         System.out.println("Reversed Star Pyramid:");
10-        for (int i = rows; i >= 1; i--) {
11-            for (int j = 0; j < rows - i; j++) {
12-                System.out.print(" ");
13-            }
14-
15-            for (int k = (2 * i) - 1; k > 0; k--) {
16-                System.out.print("*");
17-            }
18-
19-            System.out.println();
20-        }
21-
22-        scanner.close();
23-    }
24- }
25
```

Output

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

=== Code Execution Successful ===

6. Write a program for demonstrate Nested Switch.



```
NestedSwitchDemo.java
1- import java.util.Scanner;
2-
3- public class NestedSwitchDemo {
4-     public static void main(String[] args) {
5-         Scanner scanner = new Scanner(System.in);
6-         System.out.print("Enter main category (1 for Electronics, 2
for Clothing): ");
7-         int mainCategory = scanner.nextInt();
8-         switch (mainCategory) {
9-             case 1:
10-                System.out.print("Enter sub-category (1 for Mobile,
2 for Laptop): ");
11-                int electronicsSub = scanner.nextInt();
12-                switch (electronicsSub) {
13-                    case 1 -> System.out.println("You selected
Electronics -> Mobile.");
14-                    case 2 -> System.out.println("You selected
Electronics -> Laptop.");
15-                    default -> System.out.println("Invalid
Electronics sub-category.");
16-                }
17-                break;
18-
19-             case 2:
20-                System.out.print("Enter sub-category (1 for Men, 2
for Women): ");
21-                int clothingSub = scanner.nextInt();
22-                switch (clothingSub) {
23-                    case 1 -> System.out.println("You selected
Clothing -> Men.");

```

Output

```
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 ===
```

🔗

🔍

📄

📁

🔥

🔄

🔒

🔒

JS

TS

∞

php

🐦

🔒

NestedSwitchDemo.java

🔗

⚙️

🔄 Share

Run


```
14         Electronics -> Mobile.");
15         case 2 -> System.out.println("You selected
16         Electronics -> Laptop.");
17         default -> System.out.println("Invalid
18         Electronics sub-category.");
19     }
20     break;
21
22 case 2:
23     System.out.print("Enter sub-category (1 for Men, 2
24     for Women): ");
25     int clothingSub = scanner.nextInt();
26     switch (clothingSub) {
27         case 1 -> System.out.println("You selected
28         Clothing -> Men.");
29         case 2 -> System.out.println("You selected
30         Clothing -> Women.");
31         default -> System.out.println("Invalid Clothing
32         sub-category.");
33     }
34     break;
35
36 default:
37     System.out.println("Invalid main category.");
38 }
39
40 scanner.close();
41 }
```

Output


Clear

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 ===



Nippon Paint Spray Guns



Nippon Paint Walls & Metal Surfaces Pai...

3.9 ★★★★★ 182

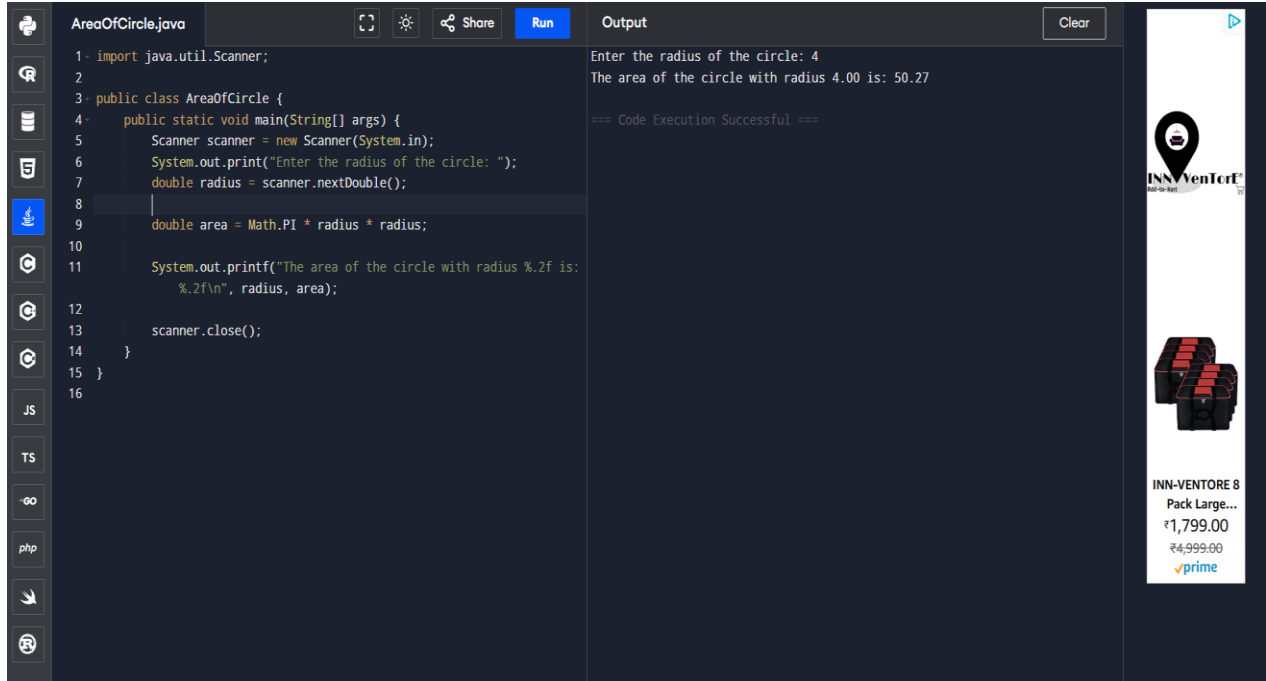
-56%

₹1,799.00

₹4,100.00 ✓prime

Shop now

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



The screenshot shows a Java IDE with a file named 'AreaOfCircle.java'. The code is as follows:

```
1- import java.util.Scanner;
2
3- public class AreaOfCircle {
4-     public static void main(String[] args) {
5-         Scanner scanner = new Scanner(System.in);
6-         System.out.print("Enter the radius of the circle: ");
7-         double radius = scanner.nextDouble();
8-
9-         double area = Math.PI * radius * radius;
10
11         System.out.printf("The area of the circle with radius %.2f is:
12                             %.2f\n", radius, area);
13
14         scanner.close();
15     }
16 }
```

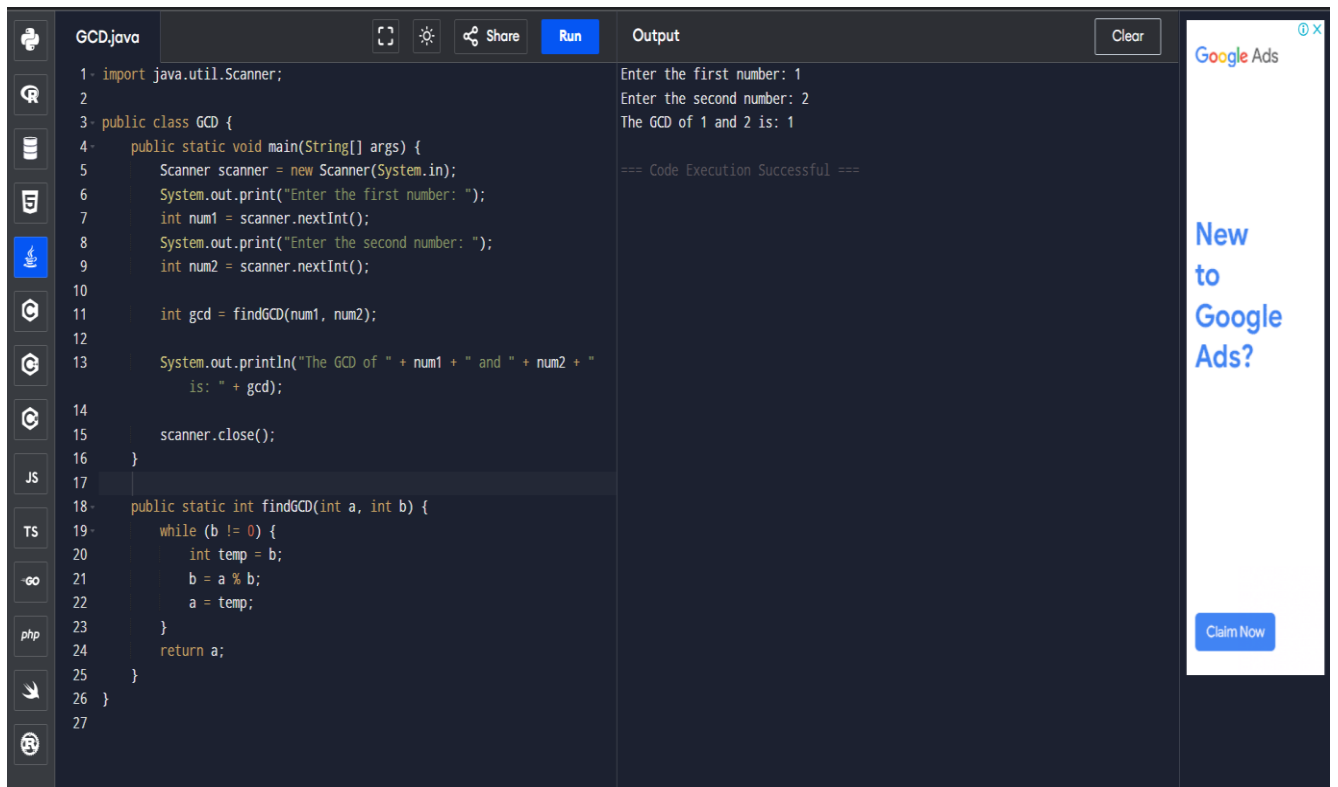
The output window shows the following text:

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

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

On the right side of the IDE, there is an advertisement for 'INN-Venture 8' with the text 'Pack Large... ₹1,799.00 ₹4,999.00' and a Prime logo.

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



The screenshot shows a Java IDE with a file named 'GCD.java'. The code implements a program to find the Greatest Common Divisor (GCD) of two numbers using the Euclidean algorithm. The code is as follows:

```
1- import java.util.Scanner;
2
3- public class GCD {
4-     public static void main(String[] args) {
5-         Scanner scanner = new Scanner(System.in);
6-         System.out.print("Enter the first number: ");
7-         int num1 = scanner.nextInt();
8-         System.out.print("Enter the second number: ");
9-         int num2 = scanner.nextInt();
10
11         int gcd = findGCD(num1, num2);
12
13         System.out.println("The GCD of " + num1 + " and " + num2 + "
14             is: " + gcd);
15
16         scanner.close();
17     }
18
19     public static int findGCD(int a, int b) {
20         while (b != 0) {
21             int temp = b;
22             b = a % b;
23             a = temp;
24         }
25         return a;
26     }
27 }
```

The 'Output' pane on the right shows the program's execution with the following text:

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

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

On the far right, there is a Google Ads banner with the text 'New to Google Ads?' and a 'Claim Now' button.

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



```

BankApp.java :
1 class Account {
2     static String bankName = "Simple Bank"; // static member
3     protected int accountNumber;
4     protected String holderName;
5     protected double balance;
6
7     public Account(int accNo, String name, double bal) {
8         accountNumber = accNo;
9         holderName = name;
10        balance = bal;
11    }
12
13    public void displayBalance() {
14        System.out.println("Account Holder: " + holderName);
15        System.out.println("Account Number: " + accountNumber);
16        System.out.println("Balance: $" + balance);
17    }
18 }
19
20 class BankAccount extends Account {
21
22     public BankAccount(int accNo, String name, double bal) {
23         super(accNo, name, bal);
24     }
25
26     public void deposit(double amount) {
27         balance += amount;
28         System.out.println("Deposited: $" + amount);
29     }

```

```

30
31     public void withdraw(double amount) {
32         if (amount <= balance) {
33             balance -= amount;
34             System.out.println("Withdrawn: $" + amount);
35         } else {
36             System.out.println("Insufficient balance.");
37         }
38     }
39 }
40 public class BankApp {
41     public static void main(String[] args) {
42         System.out.println("Welcome to " + Account.bankName);
43
44         BankAccount acc = new BankAccount(10002200, "Sivam yadav ", 500.00);
45
46         acc.displayBalance();
47         acc.deposit(200);
48         acc.withdraw(150);
49         acc.displayBalance();
50     }
51 }
52

```

```

input
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.

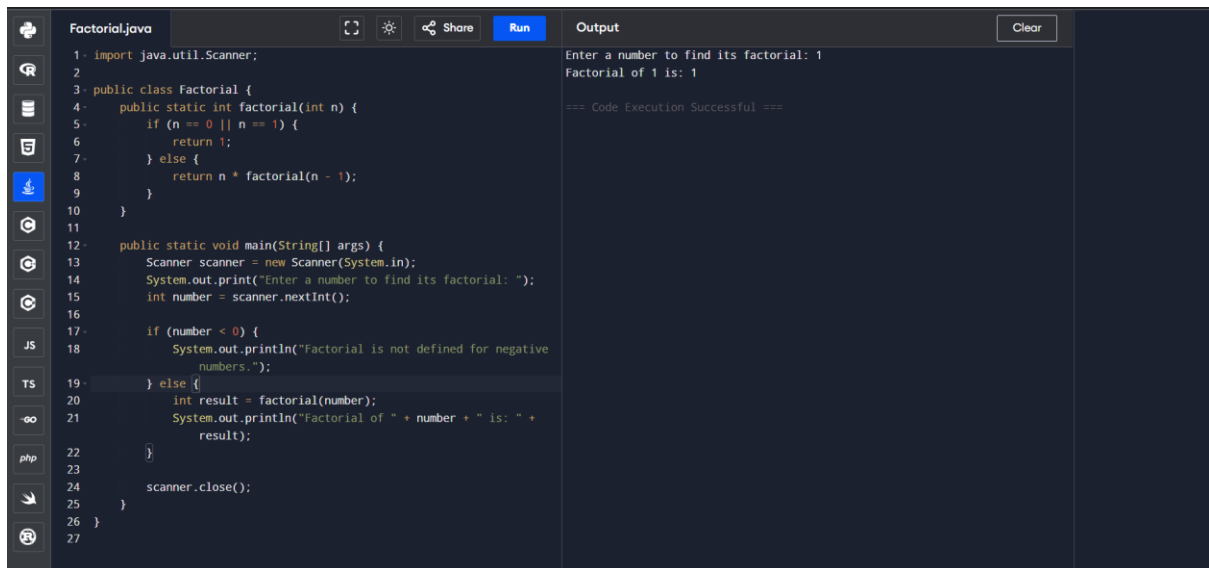
```
RectangleDemo.java :
1 class Shape {
2     double length;
3     double width;
4
5     public Shape(double length, double width) {
6         this.length = length;
7         this.width = width;
8     }
9 }
10
11 class Rectangle extends Shape {
12
13
14     public Rectangle(double length, double width) {
15         super(length, width);
16     }
17
18     public double calculateArea() {
19         return this.length * this.width;
20     }
21
22     public double calculatePerimeter() {
23         return 2 * (this.length + this.width);
24     }
25
26     public void display() {
27         System.out.println("Length: " + this.length);
28         System.out.println("Width: " + this.width);
29         System.out.println("Area: " + calculateArea());
30         System.out.println("Perimeter: " + calculatePerimeter());
31     }
32 }
33
34 public class RectangleDemo {
35     public static void main(String[] args) {
36         Rectangle rect = new Rectangle(10, 5);
37         rect.display();
38     }
39 }
40
```

input

```
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.



The screenshot shows a Java IDE with a file named 'Factorial.java'. The code defines a 'Factorial' class with a recursive 'factorial' method and a 'main' method that uses a 'Scanner' to take user input. The output window shows the program running successfully, displaying the prompt 'Enter a number to find its factorial: 1' and the result 'Factorial of 1 is: 1'.

```
1 import java.util.Scanner;
2
3 public class Factorial {
4     public static int factorial(int n) {
5         if (n == 0 || n == 1) {
6             return 1;
7         } else {
8             return n * factorial(n - 1);
9         }
10    }
11
12    public static void main(String[] args) {
13        Scanner scanner = new Scanner(System.in);
14        System.out.print("Enter a number to find its factorial: ");
15        int number = scanner.nextInt();
16
17        if (number < 0) {
18            System.out.println("Factorial is not defined for negative numbers.");
19        } else {
20            int result = factorial(number);
21            System.out.println("Factorial of " + number + " is: " + result);
22        }
23        scanner.close();
24    }
25 }
26
27
```

Output: 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.

```
1 abstract class Shape {
2     abstract void calculateArea();
3
4     void display() {
5         System.out.println("Calculating area of the shape...");
6     }
7 }
8
9 class Rectangle extends Shape {
10     double length, width;
11
12     Rectangle(double l, double w) {
13         length = l;
14         width = w;
15     }
16
17     void calculateArea() {
18         double area = length * width;
19         System.out.println("Area of Rectangle: " + area);
20     }
21 }
22
23 public class AbstractDemo {
24     public static void main(String[] args) {
25         Rectangle rect = new Rectangle(10, 5);
26         rect.display();
27         rect.calculateArea();
28     }
29 }
```

```
25     Rectangle rect = new Rectangle(10, 5);
26     rect.display();
27     rect.calculateArea();
28 }
29 }
30
```

input

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.

ObjectCounter.java	Run	Output
<pre>1 public class ObjectCounter { 2     static int count = 0; 3 4     ObjectCounter() { 5         count++; 6     } 7 8     static void displayCount() { 9         System.out.println("Total objects created: " + count); 10    } 11 12    public static void main(String[] args) { 13 14        ObjectCounter obj1 = new ObjectCounter(); 15        ObjectCounter obj2 = new ObjectCounter(); 16        ObjectCounter obj3 = new ObjectCounter(); 17 18        ObjectCounter.displayCount(); 19    } 20 } 21</pre>		<p>Total objects created: 3</p> <p>=== Code Execution Successful ===</p>

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

FunctionOverloading.java	Output
<pre>1 public class FunctionOverloading { 2 3     void display(int num) { 4         System.out.println("Number: " + num); 5     } 6 7     void display(String text) { 8         System.out.println("Text: " + text); 9     } 10 11    void display(int num, String text) { 12        System.out.println("Number: " + num + ", Text: " + text); 13    } 14 15    public static void main(String[] args) { 16        FunctionOverloading obj = new FunctionOverloading(); 17 18        obj.display(10); 19        obj.display("Hello"); 20        obj.display(25, "Overloading"); 21    } 22 } 23</pre>	<pre>Number: 10 Text: Hello Number: 25, Text: Overloading  === Code Execution Successful ===</pre>

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

```
InheritanceDemo.j...
1 class Animal {
2     void eat() {
3         System.out.println("This animal eats food.");
4     }
5
6     void sleep() {
7         System.out.println("This animal sleeps.");
8     }
9 }
10 class Dog extends Animal {
11     void bark() {
12         System.out.println("The dog barks.");
13     }
14 }
15 public class InheritanceDemo {
16     public static void main(String[] args) {
17         Dog myDog = new Dog();
18
19         myDog.eat();
20         myDog.sleep();
21         myDog.bark();
22     }
23 }
```

```
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.

```
InterfaceDemo.java :
1 interface Vehicle {
2     void start();
3     void stop();
4 }
5
6 abstract class Car implements Vehicle {
7     public void start() {
8         System.out.println("Car started.");
9     }
10 }
11
12 class MyCar extends Car {
13     public void stop() {
14         System.out.println("Car stopped.");
15     }
16 }
17
18 public class InterfaceDemo {
19     public static void main(String[] args) {
20         MyCar car = new MyCar();
21         car.start();
22         car.stop();
23     }
24 }
```

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).

```
StringHandler.java  [Icons]  Share  Run  Output
1- public class StringHandler {
2     private String text;
3
4     // Constructor
5- public StringHandler(String text) {
6     this.text = text;
7 }
8
9     // Method to check equality
10- public boolean isEqual(String other) {
11     return text.equals(other);
12 }
13
14     // Method to reverse the string
15- public String reverse() {
16     StringBuilder sb = new StringBuilder(text);
17     return sb.reverse().toString();
18 }
19
20     // Method to change case
21- public String changeCase() {
22     StringBuilder result = new StringBuilder();
23
24     for (char ch : text.toCharArray()) {
25         if (Character.isUpperCase(ch))
```

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

=== Code Execution Successful ==
```

```
StringHandler.java  [Icons]  Share  Run  Output
23
24-     for (char ch : text.toCharArray()) {
25         if (Character.isUpperCase(ch))
26             result.append(Character.toLowerCase(ch));
27         else if (Character.isLowerCase(ch))
28             result.append(Character.toUpperCase(ch));
29         else
30             result.append(ch); // Leave non-letter characters unchanged
31     }
32
33     return result.toString();
34 }
35
36     // Main method to test
37- public static void main(String[] args) {
38     StringHandler sh = new StringHandler("HelloWorld");
39
40     System.out.println("Original: HelloWorld");
41     System.out.println("Is Equal to 'HelloWorld'? " + sh.isEqual
42         ("HelloWorld"));
43     System.out.println("Reversed: " + sh.reverse());
44     System.out.println("Case Changed: " + sh.changeCase());
45 }
```

```
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.

MultipleCatchExample.java	Output
<pre>1- public class MultipleCatchExample { 2-     public static void main(String[] args) { 3-         try { 4- 5-             int result = 10 / 0; 6-             int[] numbers = new int[5]; 7-             numbers[10] = 50; 8-         } 9- 10-        catch (ArithmeticException e) { 11-            System.out.println("Error: Cannot divide by zero."); 12-        } 13-        catch (ArrayIndexOutOfBoundsException e) { 14-            System.out.println("Error: Array index out of bounds."); 15-        } 16-        catch (Exception e) { 17-            System.out.println("General exception caught."); 18-        } 19- 20-        System.out.println("Program continues after exception handling."); 21-    } 22- } 23</pre>	<pre>ERROR! Error: Cannot divide by zero. Program continues after exception handling.  === Code Execution Successful ===</pre>

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

NestedTryExample.java	Output
<pre>1- public class NestedTryExample { 2-     public static void main(String[] args) { 3-         try { 4-             // Outer try block 5-             System.out.println("Outer try block started."); 6- 7-             try { 8-                 // Inner try block 1 9-                 int a = 10 / 0; // This will throw ArithmeticException 10-            } catch (ArithmeticException e) { 11-                System.out.println("Inner try 1: Cannot divide by zero."); 12-            } 13- 14-            try { 15-                // Inner try block 2 16-                int[] arr = new int[5]; 17-                arr[10] = 100; // This will throw 18-                               // ArrayIndexOutOfBoundsException 19-            } catch (ArrayIndexOutOfBoundsException e) { 20-                System.out.println("Inner try 2: Array index out of bounds."); 21-            } 22- 23-            System.out.println("Outer try block completed."); 24-        } catch (Exception e) { 25-            System.out.println("General exception caught."); 26-        } 27-    } 28- }</pre>	<pre>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 ===</pre>

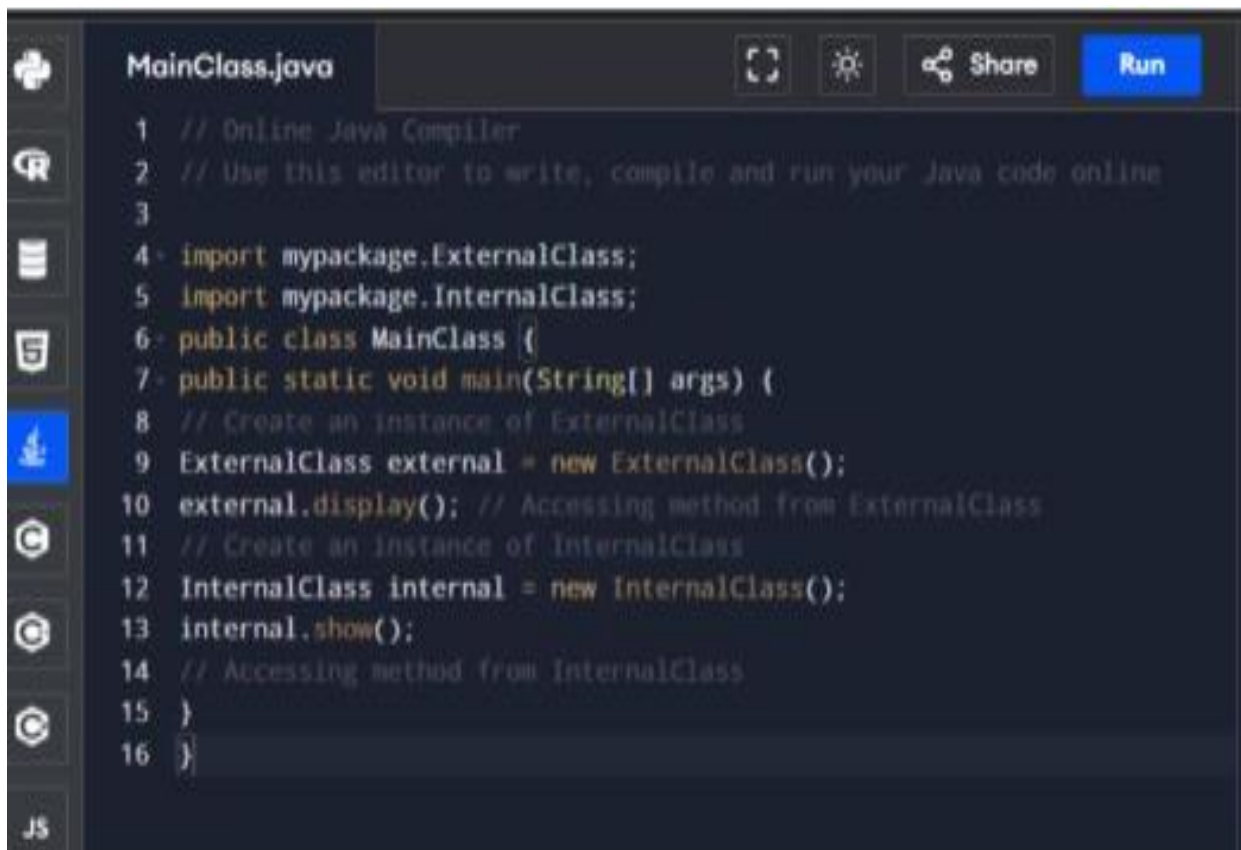


```

21
22     System.out.println("Outer try block completed.");
23 } catch (Exception e) {
24     System.out.println("Outer catch: Exception caught.");
25 }
26
27 System.out.println("Program continues after nested try blocks.");
28 }
29 }
30

```

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



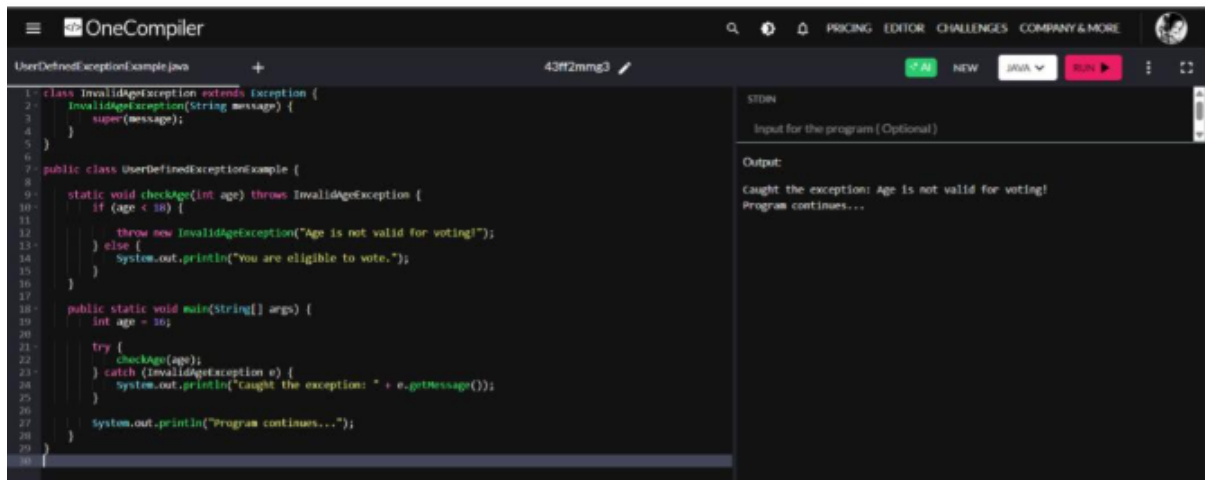
The screenshot shows an online Java compiler interface with a file named 'MainClass.java'. The code defines two classes, 'ExternalClass' and 'InternalClass', and a 'MainClass' that uses them. The 'MainClass' has a 'main' method that creates instances of 'ExternalClass' and 'InternalClass' and calls their 'display()' and 'show()' methods respectively. The interface includes a sidebar with icons for file management, a top bar with 'Share' and 'Run' buttons, and a bottom status bar showing 'JS'.

```

MainClass.java
1 // Online Java Compiler
2 // Use this editor to write, compile and run your Java code online
3
4 import mypackage.ExternalClass;
5 import mypackage.InternalClass;
6 public class MainClass {
7     public static void main(String[] args) {
8         // Create an instance of ExternalClass
9         ExternalClass external = new ExternalClass();
10        external.display(); // Accessing method from ExternalClass
11        // Create an instance of InternalClass
12        InternalClass internal = new InternalClass();
13        internal.show();
14        // Accessing method from InternalClass
15    }
16 }
JS

```

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

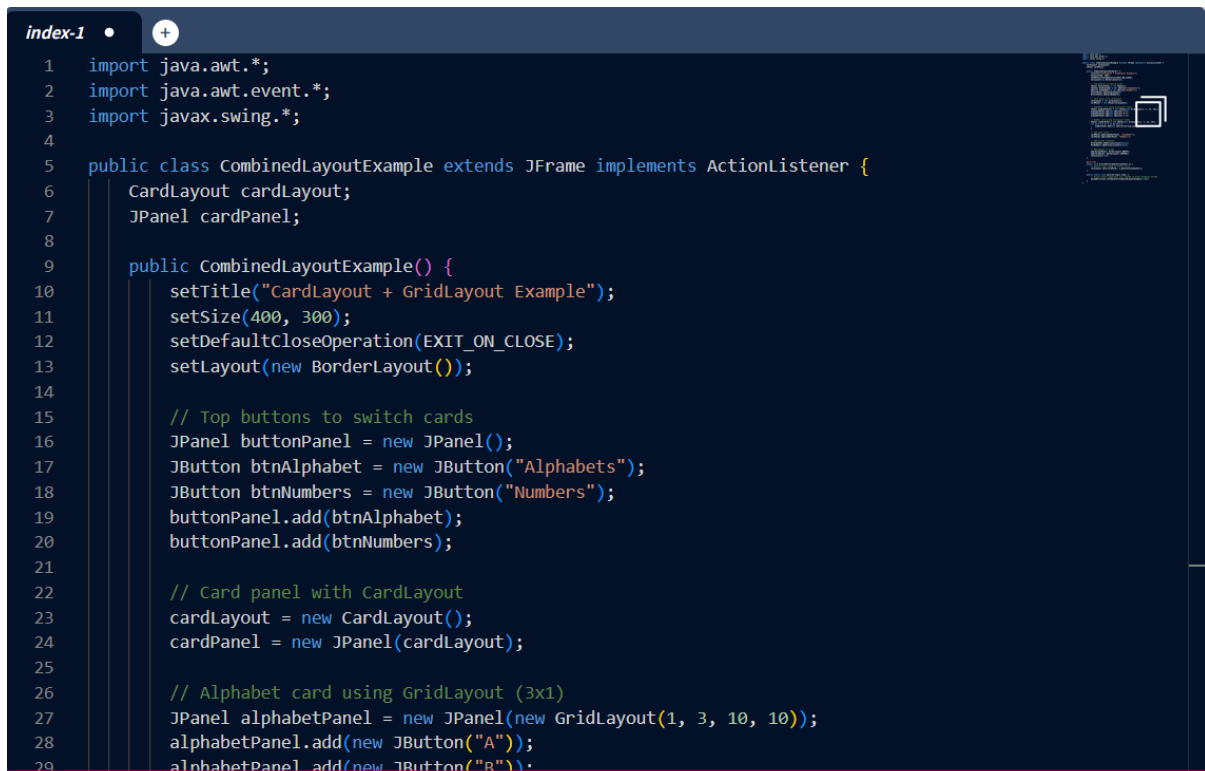


The screenshot shows the OneCompiler IDE with a Java file named 'UserDefinedExceptionExample.java'. The code defines an 'InvalidAgeException' class that extends 'Exception'. It then defines a 'UserDefinedExceptionExample' class with a 'checkAge' method that throws 'InvalidAgeException' if the age is less than 18. The 'main' method calls 'checkAge' and catches the exception, printing a message and continuing the program. The output shows the exception being caught and the program continuing.

```
1 class InvalidAgeException extends Exception {  
2     InvalidAgeException(String message) {  
3         super(message);  
4     }  
5 }  
6  
7 public class UserDefinedExceptionExample {  
8  
9     static void checkAge(int age) throws InvalidAgeException {  
10         if (age < 18) {  
11             throw new InvalidAgeException("Age is not valid for voting!");  
12         } else {  
13             System.out.println("you are eligible to vote.");  
14         }  
15     }  
16  
17     public static void main(String[] args) {  
18         int age = 10;  
19  
20         try {  
21             checkAge(age);  
22         } catch (InvalidAgeException e) {  
23             System.out.println("caught the exception: " + e.getMessage());  
24         }  
25  
26         System.out.println("Program continues...");  
27     }  
28 }  
29 }
```

Output:  
caught the exception: Age is not valid for voting!  
Program continues...

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



The screenshot shows an IDE with a Java file named 'CombinedLayoutExample.java'. The code imports 'java.awt.\*', 'java.awt.event.\*', and 'javax.swing.\*'. It defines a 'CombinedLayoutExample' class that extends 'JFrame' and implements 'ActionListener'. The constructor sets the title, size, and layout to 'BorderLayout'. It creates a 'JPanel' named 'buttonPanel' with two buttons: 'Alphabets' and 'Numbers'. It also creates a 'CardLayout' and a 'JPanel' named 'cardPanel'. The 'cardPanel' is added to the 'buttonPanel'. The 'Alphabets' button is added to the 'cardPanel' with a 'GridLayout' of 3x1. The 'Numbers' button is added to the 'cardPanel'.

```
1 import java.awt.*;  
2 import java.awt.event.*;  
3 import javax.swing.*;  
4  
5 public class CombinedLayoutExample extends JFrame implements ActionListener {  
6     CardLayout cardLayout;  
7     JPanel cardPanel;  
8  
9     public CombinedLayoutExample() {  
10         setTitle("CardLayout + GridLayout Example");  
11         setSize(400, 300);  
12         setDefaultCloseOperation(EXIT_ON_CLOSE);  
13         setLayout(new BorderLayout());  
14  
15         // Top buttons to switch cards  
16         JPanel buttonPanel = new JPanel();  
17         JButton btnAlphabet = new JButton("Alphabets");  
18         JButton btnNumbers = new JButton("Numbers");  
19         buttonPanel.add(btnAlphabet);  
20         buttonPanel.add(btnNumbers);  
21  
22         // Card panel with CardLayout  
23         cardLayout = new CardLayout();  
24         cardPanel = new JPanel(cardLayout);  
25  
26         // Alphabet card using GridLayout (3x1)  
27         JPanel alphabetPanel = new JPanel(new GridLayout(1, 3, 10, 10));  
28         alphabetPanel.add(new JButton("A"));  
29         alphabetPanel.add(new JButton("B"));
```

index-1

```
29     alphabetPanel.add(new JButton("B"));
30     alphabetPanel.add(new JButton("C"));
31
32     // Number card using GridLayout (3x3)
33     JPanel numberPanel = new JPanel(new GridLayout(3, 3, 10, 10));
34     for (int i = 1; i <= 9; i++) {
35         numberPanel.add(new JButton(String.valueOf(i)));
36     }
37
38     // Add both cards
39     cardPanel.add(alphabetPanel, "Alphabets");
40     cardPanel.add(numberPanel, "Numbers");
41
42     // Add action listeners
43     btnAlphabet.addActionListener(this);
44     btnNumbers.addActionListener(this);
45
46     // Add panels to frame
47     add(buttonPanel, BorderLayout.NORTH);
48     add(cardPanel, BorderLayout.CENTER);
49     setVisible(true);
50 }
51
52 @Override
53 public void actionPerformed(ActionEvent e) {
54     // Switch cards based on button click
55     cardLayout.show(cardPanel, e.getActionCommand());
56 }
```

index-1

```
37
38     // Add both cards
39     cardPanel.add(alphabetPanel, "Alphabets");
40     cardPanel.add(numberPanel, "Numbers");
41
42     // Add action listeners
43     btnAlphabet.addActionListener(this);
44     btnNumbers.addActionListener(this);
45
46     // Add panels to frame
47     add(buttonPanel, BorderLayout.NORTH);
48     add(cardPanel, BorderLayout.CENTER);
49     setVisible(true);
50 }
51
52 @Override
53 public void actionPerformed(ActionEvent e) {
54     // Switch cards based on button click
55     cardLayout.show(cardPanel, e.getActionCommand());
56 }
57
58 public static void main(String[] args) {
59     // Ensure Swing components are created on Event Dispatch Thread
60     SwingUtilities.invokeLater(CombinedLayoutExample::new);
61 }
62 }
63
```

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 :**

