
















Main.java		Output
<pre>1 public class MultiCatchExample { 2     public static void main(String[] args) { 3         try { 4             int[] numbers = {1, 2, 3}; 5             int result = numbers[5] / 0; 6         } catch (ArithmeticException   ArrayIndexOutOfBoundsException   7                 NullPointerException e) { 8             System.out.println("Exception caught: " + e); 9         } 10    }</pre>		<pre>java -cp /tmp/1LZguwGMFn/MultiCatchExample Exception caught: java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 3  === Code Execution Successful ===</pre>

Main.java		Output
<pre>1 import java.util.Scanner; 2 public class SumOfNumbers { 3     public static void main(String[] args) { 4         Scanner scanner = new Scanner(System.in); 5         System.out.print("Enter the number of elements: "); 6         int N = scanner.nextInt(); 7         int[] numbers = new int[N]; 8         int sum = 0; 9         System.out.println("Enter the elements:"); 10        for (int i = 0; i &lt; N; i++) { 11            numbers[i] = scanner.nextInt(); 12            sum += numbers[i]; 13        } 14        System.out.println("Sum of the numbers: " + sum); 15        try { 16            numbers[N] = 10; 17        } catch (ArrayIndexOutOfBoundsException e) { 18            System.out.println("ArrayIndexOutOfBoundsException caught!"); 19        } 20    } 21 }</pre>		<pre>java -cp /tmp/8hJGTsiIvU/SumOfNumbers Enter the number of elements:  </pre>

	Main.java	Output
<div>            </div>	<div>    Share            Run         </div> <pre> 1 class Box { 2     double length; 3     double width; 4     double height; 5 6     Box() { 7         length = 1.0; 8         width = 1.0; 9         height = 1.0; 10    } 11 12    double calculateVolume() { 13        return length * width * height; 14    } 15 } 16 17 public class Main { 18     public static void main(String[] args) { 19         Box myBox = new Box(); 20         double volume = myBox.calculateVolume(); 21         System.out.println("Volume of the Box: " + volume); 22     } 23 } </pre>	<pre> java -cp /tmp/zeN2HzGKjg/Main Volume of the Box: 1.0  === Code Execution Successful === </pre>

Main.java



Run

### Output

Clear

```
1 public class FibonacciThread implements Runnable {
2     private int n;
3
4     public FibonacciThread(int n) {
5         this.n = n;
6     }
7
8     public void run() {
9         int a = 0, b = 1;
10        System.out.print("Fibonacci Series up to " + n + " terms: ");
11        for (int i = 1; i <= n; i++) {
12            System.out.print(a + " ");
13            int sum = a + b;
14            a = b;
15            b = sum;
16        }
17    }
18
19    public static void main(String[] args) {
20        int n = 10; // Number of terms in the Fibonacci series
21        FibonacciThread fibonacciThread = new FibonacciThread(n);
22        Thread thread = new Thread(fibonacciThread);
23        thread.start();
24    }
25 }
```

```
java -cp /tmp/q27QXXHLPx/FibonacciThread
Fibonacci Series up to 10 terms: 0 1 1 2 3 5 8 13 21 34
=== Code Execution Successful ===
```