



# Exception Handling



## Topics to cover:-

- 1) Java Exceptions
- 2) Java exception handling
- 3) try catch block
- 4) finally block
- 5) throw & throws keyword.

# Java Exceptions

→

## Main.java

Class Main {

    public static void main(String args[]) {

        int arr[] = new int[20];

        System.out.println(arr[21]);

    } throws exception

    } OUT OF Bound Index Exception

    ↳ Apka code fail jayega.

}

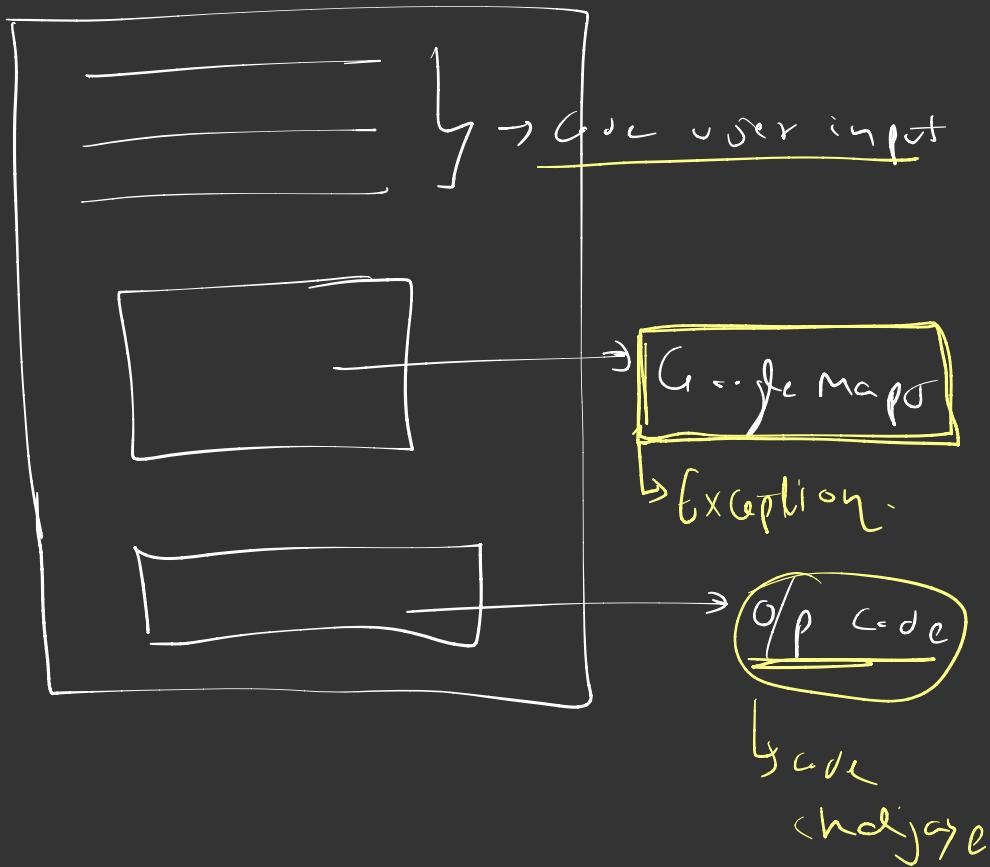
}

|   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |
|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|
| 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |
|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|

0 1 2 3

19

## Application



## Exception Handling

## Java Exceptions def

- Exception is an unexpected event.
- during program execution.
- It affects the flow of programme instructions. This lead to your programme termination.

Exceptions → occur → various reasons

- 1) invalid user input
- 2) loss of network connection
- 3) Device failure
- 4) code errors
- 5) opening an unavailable file
- 6) code run which is wrong arithmetically,  
eg) num/0

## 2 Types of Exceptions

- 1) Java Runtime Exceptions  
↳ runtime
- 2) Java IO Exception Exceptions  
↳ compiled time

### Java Runtime Exceptions

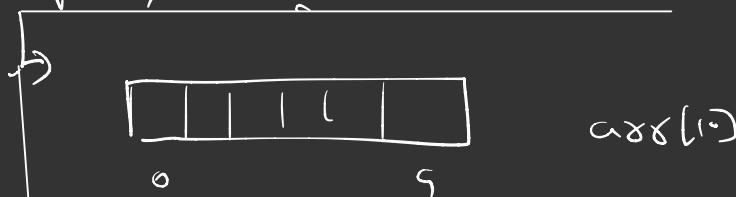
- also known as unchecked exceptions.
- happens due to a programming error.
- these exceptions are checked at runtime not compiled time.

## Common runtime exceptions

### → 1) Null pointer Exception

↳ Null pointer access (missing the initialization of a variable)

### → 2) Array Index Out of Bounds Exception



access → arr[11] not possible

→ out-of-bounds array access

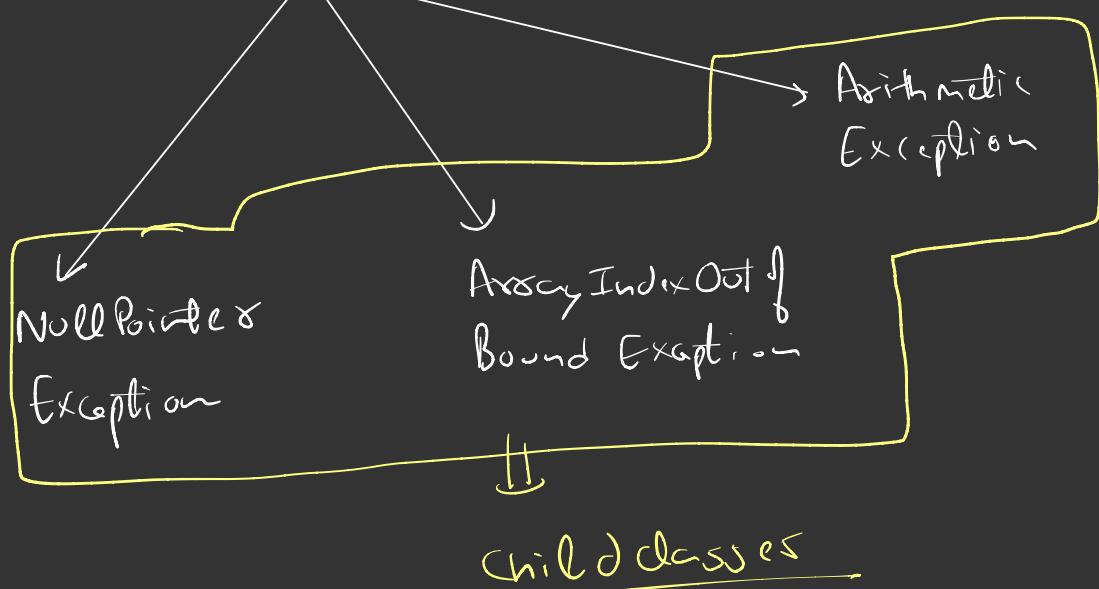
### → 3) Arithmetic Exception

↳ num/0 (Dividing a number by 0.)

Conclusion :- Runtime exception

↓  
It's your fault code

Exception (class → parent class)



## Java IO Exception Exceptions

- also known as checked exception.
- occurs at compiled time.

VJS Code :-

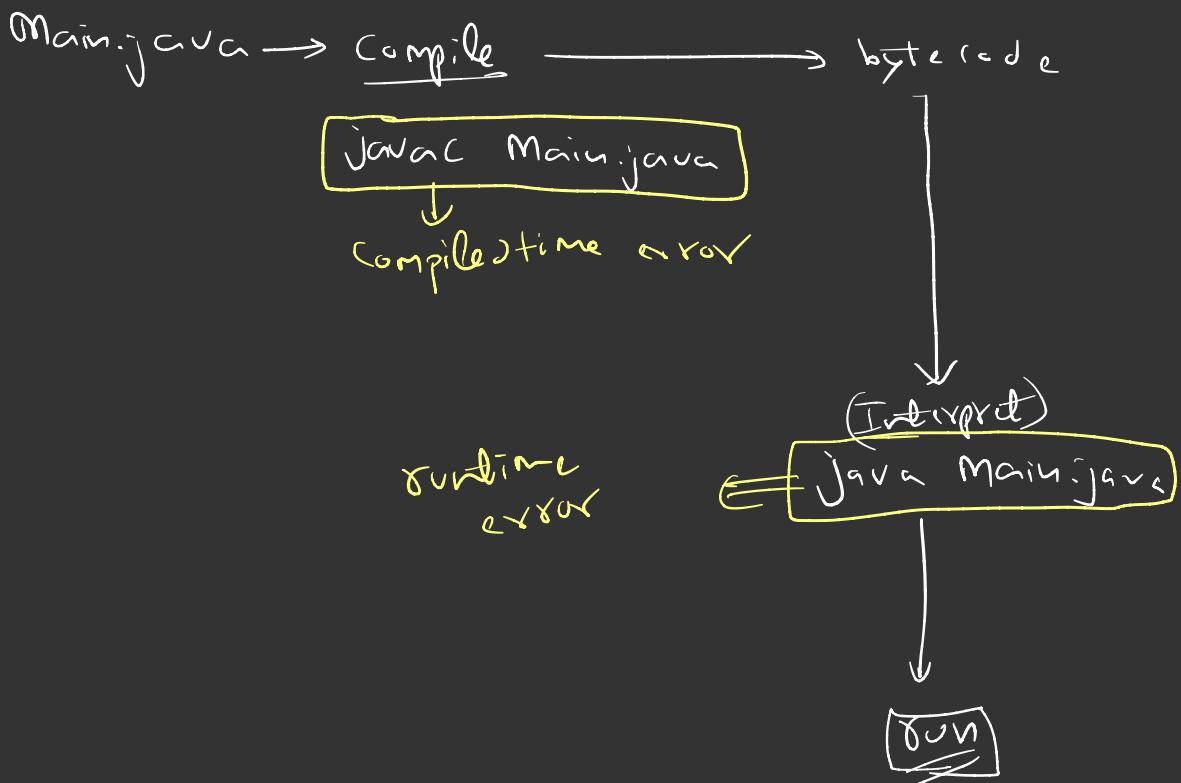
```
class ABC{  
    public void m1(){  
        int a // Semicolon Agana Bhool Gaye.  
    }  
}
```

- Computer check these exception at compiled time & programmer is prompted / show to handle these exceptions.

## Some examples :-

- 1) Trying to open a file that doesn't exist result in FileNotFoundException.
- 2) Trying to read past the end of a file.

## Code



# Compiled-time exception

A screenshot of the Visual Studio Code interface. The Explorer sidebar shows files: OPEN EDITORS (Hello.java), G2JAVACODES (Hello.class, Hello.java). The main editor window displays Java code with a syntax error:

```
1 public class Hello {  
2     Run|Debug  
3     public static void main(String[] args) {  
4         int a // compile time exception  
5     }  
6 }
```

The Problems panel shows the error: "Hello.java:3: error: ';' expected". The terminal shows the command "javac Hello.java" being run.

# Run-time Exception

A screenshot of the Visual Studio Code interface. The Explorer sidebar shows files: OPEN EDITORS (Hello.java), G2JAVACODES (Hello.class, Hello.java). The main editor window displays Java code with a runtime exception:

```
1 public class Hello {  
2     Run|Debug  
3     public static void main(String[] args) {  
4         int arr[] = new int[10]; // 10 size - 0 to 9  
5         System.out.println(x:"Doramon"); // Doramon  
6         System.out.println(arr[15]); // out of bound  
7         System.out.println(x:"Shinchan"); //try and catch block  
8     }  
9 }  
10  
11  
12 }  
13 }
```

The Problems panel shows the error: "Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 15 out of bounds for length 10". The terminal shows the command "java Hello.java" being run.

## try-catch block

- try & catch block is used to handle the exceptions.
- This prevents abnormal termination of the program.

Syntax :-

```
try {  
    // Code  
  
    }  
    catch (exception) {  
        // Code  
  
    }
```



G2JavaCodes

EXPLORER OPEN EDITORS J Hello.java

G2JAVA CODES J Hello.class J Hello.java

```
1 public class Hello {  
2     public static void main(String[] args) {  
3         int arr[] = new int[10]; // 10 size - 0 to 9  
4         System.out.println(x:"Doramon"); // Doramon  
5         try{  
6             System.out.println(arr[15]); // out of bound  
7         }catch(ArrayIndexOutOfBoundsException e){  
8             System.out.println(x:"Ooper pankha chalta hai");  
9         }  
10    }  
11    System.out.println(x:"Shinchan"); //try and catch block  
12 }  
13  
14  
15  
16  
17  
18 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS + ⌂ ⌂ ⌂ ⌂ ⌂

rahulmacbook@Rahuls-Laptop G2JavaCodes % /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/rahu1macbook/Library/Application Support/Code/User/workspaceStorage/18d0aed49b4f1e227b3d617500416d5b/redhat.java/jdt\_ws/G2JavaCodes\_a397fdf1/bin Hello  
Doramon  
Ooper pankha chalta hai  
Shinchan  
rahulmacbook@Rahuls-Laptop G2JavaCodes %

OUTLINE TIMELINE JAVA PROJECTS

Java: Ready Ln 12, Col 1 Spaces: 4 UTF-8 LF () Java Go Live Tabnine: Sign-in is required

G2JavaCodes

EXPLORER OPEN EDITORS J Hello.java

G2JAVA CODES J Hello.class J Hello.java

```
1 public class Hello {  
2     public static void main(String[] args) {  
3         int arr[] = new int[10]; // 10 size - 0 to 9  
4         System.out.println(x:"Doramon"); // Doramon  
5         try{  
6             System.out.println(arr[15]); // out of bound  
7         }catch(ArrayIndexOutOfBoundsException e){  
8             System.out.println(e);  
9         }  
10    }  
11    System.out.println(x:"Shinchan"); //try and catch block  
12 }  
13  
14  
15  
16  
17  
18 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS + ⌂ ⌂ ⌂ ⌂ ⌂

G2JavaCodes : /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/rahu1macbook/Library/Application Support/Code/User/workspaceStorage/18d0aed49b4f1e227b3d617500416d5b/redhat.java/jdt\_ws/G2JavaCodes\_a397fdf1/bin Hello  
Doramon  
java.lang.ArrayIndexOutOfBoundsException: Index 15 out of bounds for length 10  
Shinchan  
rahulmacbook@Rahuls-Laptop G2JavaCodes %

OUTLINE TIMELINE JAVA PROJECTS

## Multiple exception handling

The screenshot shows a Java code editor with the following code:

```
public class Hello {
    public static void main(String[] args) {
        int arr[] = new int[10]; // 10 size - 0 to 9
        System.out.println("Doramon"); // Doramon

        try{
            System.out.println(arr[15]); // ArrayIndexOutOfBoundsException
            int result = 5/0; // java.lang.ArithmetricException
        }catch(ArrayIndexOutOfBoundsException e){
            System.out.println(e);
        }
        System.out.println("Shinchan"); //try and catch block
    }
}
```

The code contains a try-catch block. The catch block is currently active, indicated by a yellow dot on line 14. A tooltip for the variable `result` is visible, showing its type as `int`. The output pane at the bottom shows the following error message:

G2JavaCodes ; /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/rahu/Downloads/Library/Application\_Support/Code/User/workspaceStorage/18dbdaed49b4f1e227b3d617500416d5b/redhat.java/dt\_ws/G2JavaCodes\_a397df1/bin Hello  
Doramon  
java.lang.ArrayIndexOutOfBoundsException: Index 15 out of bounds for length 10  
Shinchan  
rahu@macbook:~/Downloads/Library/Application\_Support/Code/User/workspaceStorage/18dbdaed49b4f1e227b3d617500416d5b/redhat.java/dt\_ws/G2JavaCodes\_a397df1/bin Hello

The screenshot shows a Java application running on a Mac. The main window is a code editor titled "Hello.java" with the file path "G2.javaCodes". The code contains a main method that prints "Doramon" and then attempts to divide by zero, which is caught and handled. Below the code editor is a terminal window showing the execution of the program and the resulting output.

```
public static void main(String[] args) {
    int arr[] = new int[10]; // 10 size - 0 to 9
    System.out.println("Doramon"); // Doramon

    try{
        int result = 5/0; // java.lang.ArithmaticException
        System.out.println(arr[15]); // ArrayIndexOutOfBoundsException

    }catch(ArrayIndexOutOfBoundsException e){
        System.out.println(e);
    }

    System.out.println("Shinchan"); //try and catch block
}
```

```
rahulmacbook@Rahuls-Laptop:~/Documents/G2JavaCodes$ cd /Users/rahu/macbook/Documents/G2JavaCodes ; /usr/bin/env /Library/Java/VirtualMachines/jdk-21.1.0/Contents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/rahu/macbook/Library/Application_Support/Code/Workspaces/Forge/18d01d49b4f1150015001500410d55/eduat.java@/dt-ws/G2JavaCodes_3537df0d/bin Hello
Doramon
Exception in thread "main" java.lang.ArithmaticException: / by zero
        at Hello.main(Hello.java:10)
rahulmacbook@Rahuls-Laptop:~/Documents/G2JavaCodes$
```

Handling that :-

The screenshot shows a Java code editor interface with the following details:

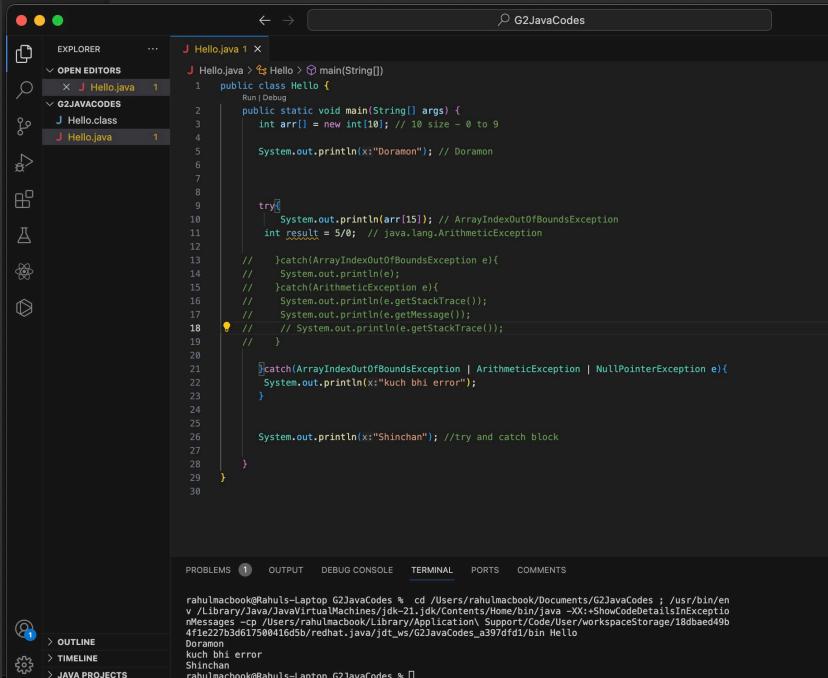
- EXPLORER:** Shows files: `Hello.java`, `Hello.class`, and `G2JAVACODES`.
- OPEN EDITORS:** Shows `Hello.java` (1 tab).
- EDITOR:** Displays the `Hello.java` file content:

```
public static void main(String[] args) {
    int arr[] = new int[10]; // 10 size - 0 to 9
    System.out.println(x:"Doramon"); // Doramon

    try{
        int result = 5/0; // java.lang.ArithmaticException
        System.out.println(arr[15]); // ArrayIndexOutOfBoundsException

    }catch(ArrayIndexOutOfBoundsException e){
        System.out.println(e);
    }catch(ArithmaticException e){
        System.out.println(e.getStackTrace());
        System.out.println(e.getMessage());
        // System.out.println(e.getStackTrace());
    }
}
```
- PROBLEMS:** Shows 1 error: `G2JavaCodes ; /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/rahu...4f1e227b3d617500416d5b/redhat.java/jdt_ws/G2JavaCodes_a397dfd1/bin Hello Doramon [Ljava.lang.StackTraceElement;@251a69d7 / by zero`.
- OUTPUT:** Shows command-line output: `Shinchan rahulmacbook@Rahuls-Laptop G2JavaCodes %`.
- STATUS BAR:** Shows: Java: Ready, Ln 17, Col 44, Spaces: 4, UTF-8, LF, Java, Go Live, Tabnine: Sign-in is required.

in a single line :-



The screenshot shows the Visual Studio Code interface with a Java file named `Hello.java` open in the editor. The code is a simple program that prints "Doramon" and then attempts to print the 15th element of an array, which results in an `ArrayIndexOutOfBoundsException`. The code is written in a single horizontal line, demonstrating a coding style where multiple lines of code are combined into one.

```
J Hello.java 1
J Hello.java > main(String[])
public class Hello {
    public static void main(String[] args) {
        int arr[] = new int[10]; // 10 size - 0 to 9
        System.out.println("Doramon"); // Doramon

        try{
            System.out.println(arr[15]); // ArrayIndexOutOfBoundsException
            int result = 5/0; // java.lang.ArithmetricException
        }
        // catch(ArrayIndexOutOfBoundsException e){
        //     System.out.println(e);
        // } catch(ArithmetricException e){
        //     System.out.println(e.getStackTrace());
        // } catch(NullPointerException e){
        //     System.out.println(e.getMessage());
        // }
        // System.out.println(e.getStackTrace());
    }

    System.out.println("Shinchan"); //try and catch block
}

```

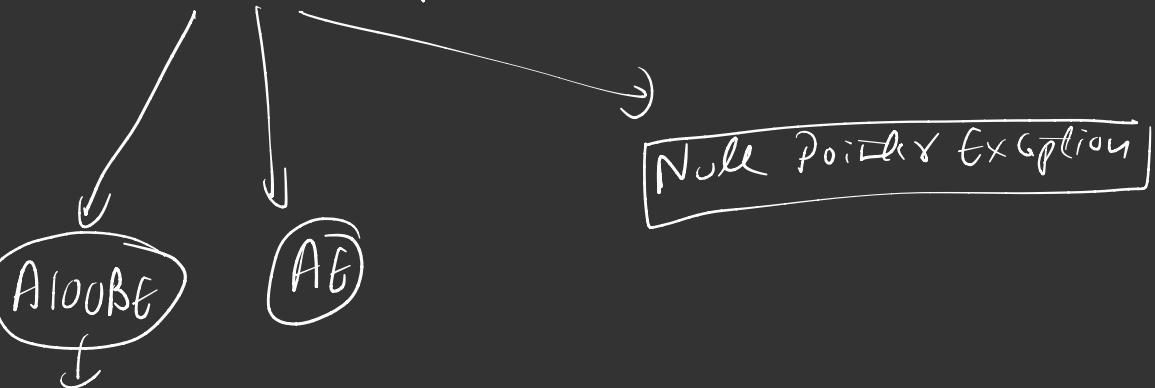
The terminal at the bottom shows the execution of the Java code and its output:

```
rahuImacbook@Rahuls-MacBook-Pro G2JavaCodes % cd /Users/rahuImacbook/Documents/G2JavaCodes ; /usr/bin/java v /Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp "/Users/rahuImacbook/Library/Application Support/Code/User/workspaceStorage/180baded49b4f1e227b3d617500416d5b/redhat.java/dt_ws/G2JavaCodes_a397dfdf1/bin Hello
Doramon
kuch bhi error
Shinchan
rahuImacbook@Rahuls-MacBook-Pro G2JavaCodes %
```

Exception (parent)



Runtime Exception



Array Index  
Out of bound  
Exception

→ how replace & save handling  
Exception save

The screenshot shows a Java development environment with the following interface elements:

- EXPLORER**: Shows the project structure with files `Hello.java`, `NullPointerExc...`, and `Hello.class`.
- OPEN EDITORS**: Displays two tabs: `Hello.java` (active) and `NullPointerExc...`.
- G2JAVACODES**: A folder containing `Hello.class`.
- CODE**: The main editor area showing the code for `Hello.java`. The code includes a try-catch block that prints "Doramon" and handles ArrayIndexOutOfBoundsException and ArithmeticException.
- PROBLEMS**: Shows 1 error.
- OUTPUT**: Terminal output showing the command to change directory to `/Users/rahulmacbook/Documents/G2JavaCodes` and set the Java path.
- DEBUG CONSOLE**: Not visible in the screenshot.
- TERMINAL**: Shows the terminal command: `rahulmacbook@Rahuls-Laptop G2JavaCodes % cd /Users/rahulmacbook/Documents/G2JavaCodes ; /usr/bin/env /Library/Java/Ja`.
- PORTS**: Not visible in the screenshot.
- COMMENTS**: Not visible in the screenshot.

```
1  public class Hello {
2      public static void main(String[] args) {
3          int arr[] = new int[10]; // 10 size - 0 to 9
4          System.out.println(x:"Doramon"); // Doramon
5
6
7
8
9
try{
10     System.out.println(arr[15]); // ArrayIndexOutOfBoundsException
11     int result = 5/0; // java.lang.ArithmetricException
12
13     // }catch(ArrayIndexOutOfBoundsException e){
14     // System.out.println(e);
15     // }catch(ArithmetricException e){
16     // System.out.println(e.getStackTrace());
17     // System.out.println(e.getMessage());
18     // // System.out.println(e.getStackTrace());
19     // }
20
21     }catch(Exception e){
22         System.out.println(x:"kuch bhi error");
23     }
24
25
26     System.out.println(x:"Shinchan"); //try and catch block
27
28 }
29
30 }
```

## Try-catch-finally block

- We use finally after try-catch block.
- finally block is always executed whether there is an exception inside the try block or not.

Some cases when a finally block does not execute,

- 1) Use of system.exit() method
- 2) An exception occurs in the finally block
- 3) The death of a thread.

Chrome File Edit View History Bookmarks Profiles Tab Window Help Thu 29 Aug 9:58 AM

Untitled - LeetCode Playground +

leetcode.com/playground/new/empty

Explore Problems Contest Discuss Interview Store Register or Sign in Premium

Run Code Untitled Save Java Settings

Output: Finished

Clear Console

```
1 //try catch and finally block
2 public class Main {
3     public static void main(String[] args) {
4         int arr[] = new int[5];
5
6         try{
7             System.out.println(arr[100]); // 100 toh hai hin ni toh
ayega exception- array index out of bounds ka
8         }catch(Exception e){
9             System.out.println(e);
10    }finally{
11        System.out.println("barish ka mosam bemisaal,
barish ka mosam bemisal ismai kya hai kamal");
12    }
13
14 }
15 }
```

Share Live Add Snippet

stdin

This screenshot shows a successful execution of Java code on LeetCode's playground. The code attempts to print the 100th element of an array of length 5, which triggers an `IndexOutOfBoundsException`. The `finally` block prints a message in Hindi: "barish ka mosam bemisaal, barish ka mosam bemisal ismai kya hai kamal". The output window shows the message "Finished in 92 ms" and the exception details in English.

Untitled - LeetCode Playground +

leetcode.com/playground/new/empty

Explore Problems Contest Discuss Interview Store Register or Sign in Premium

Run Code Untitled Save Java Settings

Output: Runtime Error

Clear Console

```
1 //try catch and finally block
2 public class Main {
3     public static void main(String[] args) {
4         int arr[] = new int[5];
5
6         try{
7             System.out.println(arr[100]); // 100 toh hai hin ni toh
ayega exception- array index out of bounds ka
8         }finally{
9             System.out.println("barish ka mosam bemisaal,
barish ka mosam bemisal ismai kya hai kamal");
10    }
11
12 }
13 }
```

Share Live Add Snippet

stdin

This screenshot shows a runtime error on LeetCode's playground. The code is identical to the one above, but the output indicates a "Runtime Error" and "Error." with a message asking to ensure a working internet connection. The final output is "Finished in N/A" with the same exception details as the first screenshot.

## Java throw & throws Keyword

- we use the throws keyword in the method declaration to declare the type of exceptions that might occur within it.

Cg)

public class ABCD

public static void

}

Explore Problems Contest Discuss Interview **Store** Register or Sign in Premium

Run Code Untitled  Java 

```
1 // "static void main" must be defined in a public class.
2 public class Main {
3     public static void main(String[] args) {
4         int arr[] = new int[5];
5         customeFunction(arr);
6     }
7 }
8
9 public static void customeFunction(int arr[]){
10     System.out.println(arr[100]);
11 }
12 }
```

Output: **Runtime Error** 

Finished in N/A  
java.lang.ArrayIndexOutOfBoundsException: Index 100 o  
ut of bounds for length 5  
at line 10, Main.customeFunction  
at line 5, Main.main

Share Live  stdin 

Explore Problems Contest Discuss Interview **Store** Register or Sign in Premium

Run Code Untitled  Java 

```
1 // "static void main" must be defined in a public class.
2 public class Main {
3     public static void main(String[] args) {
4         int arr[] = new int[5];
5         try{
6             customeFunction(arr);
7         }catch(Exception e){
8             System.out.println(e);
9         }
10        System.out.println("After exception code");
11    }
12 }
13
14 public static void customeFunction(int arr[]){
15     System.out.println(arr[100]);
16 }
17 }
18 }
```

Output: **Finished** 

Finished in 95 ms  
java.lang.ArrayIndexOutOfBoundsException: Index 100 o  
ut of bounds for length 5

Finished in 121 ms  
java.lang.ArrayIndexOutOfBoundsException: Index 100 o  
ut of bounds for length 5  
After exception code

Share Live  stdin 

## Throw Key word

↳ It's used to explicitly throw a single exception.

Q) Class Main {

    public static void main() {

        throw new ArithmeticException(" Trying to divide by 0");

}  
y

o/f

Trying to divide by 0 .

throw → we can create custom exceptions

    → we can throw pre-defined exceptions.

[Explore](#) [Problems](#) [Contest](#) [Discuss](#) [Interview](#) [Store](#)[Register](#) or [Sign in](#)[Premium](#)[Run Code](#)

Untitled

Save

Java

Output: [Finished](#)[Clear Console](#)

```
1 public class Main {  
2     public static void main(String[] args) {  
3         Scanner scn = new Scanner(System.in);  
4         int age = scn.nextInt();  
5         try{  
6             if(age>100){  
7                 throw new MyException();  
8             }  
9         }  
10        }catch(Exception e){  
11            System.out.println("I am good ");  
12        }  
13        System.out.println("I am good 2");  
14    }  
15    class MyException extends Exception{  
16    }  
17 }
```

Finished in N/A

Line 8: error: unreported exception MyException; must  
be caught or declared to be thrown [in Main.java]  
 throw new MyException();  
 ^

Finished in 188 ms

I am good

I am good 2

stdin

1000

 [Share](#) [Live](#) [Add Snippet](#)

If we have to an exception with a message.

We have to ~~super~~ keyword  
it

parent class  
constructor  
call.

super(message);

e.g)  
class MyException extends Exception {  
 public MyException(String message) {  
 super(message);  
 }  
}

Explore Problems Contest Discuss Interview Store Register or Sign In Premium

Run Code Untitled

```
1 import java.util.*;
2 public class Test2 {
3     public static void main(String[] args) {
4         Scanner scn = new Scanner(System.in);
5         System.out.println("Enter age");
6
7         try{
8             int age = scn.nextInt();
9             if(age > 100){
10                 throw new MyException("MY ERROR IS THIS");
11             }
12         }catch(Exception e){
13             System.out.println(e);
14         }
15     }
16 }
17
18 class MyException extends Exception {
19     public MyException(String message){
20         super(message); //calling parent constructor
21     }
22 }
23
24
```

To exit full screen, press [fn F] Output: Finished

Finished in 201 ms  
Enter age  
MyException: MY ERROR IS THIS

stdin 10000000

Share Live Add Snippet

## Preddefined Exception throwing

Explore Problems Contest Discuss Interview Store Register or Sign In Premium

Run Code Untitled

```
1 import java.util.*;
2 public class Test {
3     public static void main(String[] args) {
4         Scanner scn = new Scanner(System.in);
5         System.out.println("Enter age");
6
7         try{
8             int age = scn.nextInt();
9             if(age > 100){
10                 throw new ArithmeticException();
11             }
12         }catch(Exception e){
13             System.out.println(e);
14         }
15     }
16 }
17
18 class MyException extends Exception {
19     public MyException(String message){
20         super(message); //calling parent constructor
21     }
22 }
23
24
```

Output: Finished

Finished in N/A  
java.lang.ArrayIndexOutOfBoundsException: Index 100 out of bounds for length 5  
at line 10, Main.customFunction  
at line 5, Main.main

Finished in 219 ms  
Enter age  
java.util.NoSuchElementException

Finished in 208 ms  
Enter age  
java.lang.ArithmaticException

stdin 10000

Share Live Add Snippet

## Nested Try-Catch

→ we can use multiple try catch as follows:

```
try {  
    try {  
        try {  
            catch (exception e) {}  
        }  
    }  
    catch (exception e) {}  
}
```

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Untitled - LeetCode Playground

leetcode.com/playground/new/empty

Explore Problems Contest Discuss Interview Store Premium

Run Code Untitled Save Java Settings

Output: Finished

Clear Console

```
1 // "static void main" must be defined in a public class.
2 public class Main {
3     public static void main(String[] args) {
4         int arr[] = new int[20];
5         try{
6             System.out.println("i am block1 try");
7             try{
8                 System.out.println("i am block 2 try");
9                 System.out.println(arr[200]);
10            }catch(Exception e){
11                System.out.println("i am block2 catch");
12            }
13        }catch(Exception e){
14            System.out.println("i am catch1 block");
15        }
16    }
17 }
```

Share Live Add Snippet

stdin

Q-11 Write a program that allows you to keep accessing an array until a valid index is given by the user.



Q.2) Password too short exception

(Q-3)

Email validation.

M/Q 15

1. What will be the output of the following code?

```
1~ public class Test {  
2~     public static void main(String[] args) {  
3~         try {  
4~             int data = 50 / 0;  
5~             System.out.println("Try block executed");  
6~         } catch (ArithmaticException e) {  
7~             System.out.println("ArithmaticException caught");  
8~         } catch (Exception e) {  
9~             System.out.println("General exception caught");  
10~        } finally {  
11~            System.out.println("Finally block executed");  
12~        }  
13~        System.out.println("Code after try-catch-finally");  
14~    }  
15~}  
16~
```

Soln) Try block executed

finally block executed

2)

```
1 public class Test {
2     public static void main(String[] args) {
3         try {
4             String s = null;
5             System.out.println(s.length());
6         } catch (NullPointerException e) {
7             System.out.println("NullPointerException caught");
8         } finally {
9             System.out.println("Finally block executed");
10        }
11    }
12 }
```

- A) NullPointerException caught
- B) Finally block executed
- C) NullPointerException caught, Finally block executed
- D) No output due to exception

Soln) NullPointerException caught, finally  
block executed.

3) 

```
public class Test {
    public static void main(String[] args) {
        try {
            int[] arr = new int[5];
            System.out.println(arr[10]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("ArrayIndexOutOfBoundsException caught");
        } finally {
            System.out.println("Finally block executed");
        }
    }
}
```

- A) ArrayIndexOutOfBoundsException caught  
B) Finally block executed  
C) ArrayIndexOutOfBoundsException caught, Finally block executed  
D) No output due to exception

4) 

```
public class Test {
    public static void main(String[] args) {
        try {
            int num = Integer.parseInt("XYZ");
        } catch (NumberFormatException e) {
            System.out.println("NumberFormatException caught");
        } finally {
            System.out.println("Finally block executed");
        }
    }
}
```

- A) NumberFormatException caught  
B) Finally block executed  
C) NumberFormatException caught, Finally block executed  
D) No output due to exception

S)

```
1 - public class Test {
2 -     public static void main(String[] args) {
3 -         try {
4 -             return;
5 -         } finally {
6 -             System.out.println("Finally block executed");
7 -         }
8 -     }
9 - }
10 }
```

- A) No output
- B) Finally block executed
- C) Program exits without output
- D) Runtime error

→

finally block execute word

b)

```
1 - public class Test {
2 -     public static void main(String[] args) {
3 -         try {
4 -             throw new RuntimeException("Runtime Exception");
5 -         } catch (RuntimeException e) {
6 -             System.out.println(e.getMessage());
7 -         } finally {
8 -             System.out.println("Finally block executed");
9 -         }
10    }
11 }
```

- A) Runtime Exception
- B) Finally block executed
- C) Runtime Exception, Finally block executed
- D) No output due to exception

**Correct Answer:** C) Runtime Exception, Finally block executed

```
1 public class Test {
2     public static void main(String[] args) {
3         try {
4             System.out.println("Inside try");
5         } finally {
6             System.out.println("Inside finally");
7         }
8     }
9 }
10
```

- A) Inside try  
B) Inside finally  
C) Inside try, Inside finally  
D) No output

**Correct Answer:** C) Inside try, Inside finally

```
1 public class Test {
2     public static void main(String[] args) {
3         try {
4             int[] arr = new int[1];
5         } catch (ArrayIndexOutOfBoundsException e) {
6             System.out.println("ArrayIndexOutOfBoundsException caught");
7         } finally {
8             System.out.println("Finally block executed");
9         }
10    }
11 }
```

- A) ArrayIndexOutOfBoundsException caught  
B) Finally block executed  
C) ArrayIndexOutOfBoundsException caught, Finally block executed  
D) No output due to exception

**Correct Answer:** C) ArrayIndexOutOfBoundsException caught, Finally block executed

```
1 public class Test {
2     public static void main(String[] args) {
3         try {
4             System.out.println("Try block");
5         } finally {
6             System.out.println("Finally block");
7         }
8     }
9 }
```

- A) Try block  
B) Finally block  
C) Try block, Finally block  
D) No output due to return statement

**Correct Answer:** C) Try block, Finally block

16)

```
1 public class Test {
2     public static void main(String[] args) {
3         try {
4             throw new NullPointerException("Custom Message");
5         } catch (NullPointerException e) {
6             System.out.println("Exception: " + e.getMessage());
7         } finally {
8             System.out.println("Finally block executed");
9         }
10 }
```

- A) Exception: Custom Message
- B) Finally block executed
- C) Exception: Custom Message, Finally block executed
- D) No output due to exception

**Correct Answer:** C) Exception: Custom Message, Finally block executed

# Problems

## ↳ Password Validation

The screenshot shows a Java code editor and a terminal window. The code editor displays a file named `PasswordValidation.java` with the following content:

```
1 - class PasswordTooShortException extends Exception {
2 -     public PasswordTooShortException(String message) {
3 -         super(message);
4 -     }
5 - }
6
7 - public class PasswordValidation {
8 -     public static void validatePassword(String password) throws PasswordTooShortException {
9 -         if (password.length() < 8) {
10 -             throw new PasswordTooShortException("Password must be at least 8 characters long.");
11 -         } else {
12 -             System.out.println("Password is valid.");
13 -         }
14 -     }
15
16 -     public static void main(String[] args) {
17 -         try {
18 -             validatePassword("pass123");
19 -         } catch (PasswordTooShortException e) {
20 -             System.out.println("Exception caught: " + e.getMessage());
21 -         }
22 -     }
23 }
```

The terminal window below shows the execution of the program. It outputs the exception message and the exit code.

```
input
Exception caught: Password must be at least 8 characters long.

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

## ↳ Validate Email

```
>EmailValidation.java :  
1- public class EmailValidation {  
2-     public static void validateEmail(String email) {  
3-         if (email.contains("@") && email.indexOf(.) > email.indexOf("@")) {  
4-             System.out.println("Email is valid.");  
5-         } else {  
6-             System.out.println("Invalid email address.");  
7-         }  
8-     }  
9-  
10-    public static void main(String[] args) {  
11-        validateEmail("example@domain.com"); // Valid email  
12-        validateEmail("invalidemail.com"); // Invalid email  
13-    }  
14- }  
15
```

```
Input  
Email is valid.  
Invalid email address.
```

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

### 3) Validate Index

The screenshot shows a Java code editor and a terminal window. The code in the editor is as follows:

```
1. public class ArrayIndexValidation {
2.     public static void validateIndex(int[] array, int index) {
3.         try {
4.             System.out.println("Element at index " + index + ": " + array[index]);
5.         } catch (ArrayIndexOutOfBoundsException e) {
6.             System.out.println("Exception caught: Index " + index + " is out of bounds.");
7.         }
8.     }
9.
10.    public static void main(String[] args) {
11.        int[] numbers = {10, 20, 30, 40, 50};
12.
13.        validateIndex(numbers, 2); // Valid index
14.        validateIndex(numbers, 5); // Invalid index
15.    }
16. }
```

The terminal window below shows the program's output:

```
Element at index 2: 30
Exception caught: Index 5 is out of bounds.

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