Array Index Out Of Bounds Exception in Java

Last Updated : 26 Jul, 2025

In Java, **ArrayIndexOutOfBoundsException** is a Runtime Exception thrown only at runtime. The Java Compiler does not check for this error during the compilation of a program. It occurs when we try to access the element out of the index we are allowed to, i.e**. index >= size of the array.**

Java supports the creation and manipulation of arrays as a data structure. The index of an array is an integer value that has a value in the interval [0, n-1], where n is the size of the array. If a request for a negative or an index greater than or equal to the size of the array is made, then Java throws an **ArrayIndexOutOfBounds Exception**. This is unlike C/C++, where no index of the bound check is done.

Example 1: Here, we are ***trying to access the index which is greater than or equal to the array length***.

```java
// Java program to show ArrayIndexOutOfBoundsException
// when the index is greater than or equal to array length
public class GFG {

    public static void main(String[] args)
    {

        // taking array of integers
        int a[] = { 1, 2, 3, 4, 5 };

        for (int i = 0; i <= a.length; i++)
            System.out.println(a[i]);
    }
}
```

Runtime Error Throws an Exception:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5
        at GFG.main(GFG.java:11)
```

Here if you carefully see, the array is of size 5. Therefore while accessing its element using for loop, the maximum index value can be 4, but in our program, it is going till 5 and thus the exception.

Example 2: Here, we are ***trying to access the index of array which is negative***.

```java
// Java program to show ArrayIndexOutOfBoundsException
// when we access the negative index of array
public class GFG {

    public static void main(String[] args)
    {

        // taking array of integers
        int a[] = { 1, 2, 3 };

        // accessing the negative index of array
        System.out.println(a[-2]);
    }
}
```

Run-Time Exception:

> *Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index -2 out of bounds for length 3*
> *at GFG.main(GFG.java:12)*

Handling ArrayIndexOutOfBoundsException in Java

To handle ArrayIndexOutOfBoundsException, make sure that index of array is within the valid range. You can also use the enhanced for-loop to automatically handle this exception.

Example 1: Here, we are checking whether the index is valid or not by taking array length with in the index range i.e. [0, n-1]

```
// Java program to handle ArrayIndexOutOfBoundsException
// by taking array index within valid range
public class GFG {

    public static void main(String[] args)
    {

        // taking array of integers
        int a[] = { 1, 2, 3, 4, 5 };

        // here, we have remove equal to sign
        for (int i = 0; i < a.length; i++)
            System.out.println(a[i]);
    }
}
```

Output

```
1
2
3
4
5
```

Example 2: Here, we are ***using enhanced for loop*** that automatically handles the accessing of array's index

```
// Java program to handle ArrayIndexOutOfBoundsException
// by using enhanced for loop
public class GFG {

    public static void main(String[] args)
    {

        // taking array of integers
        int a[] = { 1, 2, 3, 4, 5 };

        // using enhanced for loop
        for (int e : a) {
            System.out.println(e);
        }
    }
}
```

Output

```
1
2
3
4
5
```

Example 3: Consider enclosing your code inside a try-catch statement and manipulate the exception accordingly. As mentioned, Java won't let you access an invalid index and will definitely throw an ArrayIndexOutOfBoundsException. However, we should be careful inside the block of the catch statement because if we don't handle the exception appropriately, we may conceal it and thus, create a bug in your application.

```java
// Java program to handle ArrayIndexOutOfBoundsException
// by using using try-catch block
public class GFG {

    public static void main(String[] args)
    {

        // taking array of integers
        int a[] = { 1, 2, 3, 4, 5 };

        // using try catch block
        try {
            for (int i = 0; i <= a.length; i++)
                System.out.print(a[i] + " ");
        }
        catch (Exception e) {
            System.out.println("\nException Caught");
        }
    }
}
```

Output

```
1 2 3 4 5
Exception Caught
```

Here in the above example, you can see that till index 4 (value 5), the loop printed all the values, but as soon as we tried to access the a[5], the program threw an exception which is caught by the catch block, and it printed the "Exception Caught" statement.

Comment

K kartik 27

Article Tags: Java Java-Exceptions Java-Arrays Java-Array-Programs

Explore

Java Basics

OOP & Interfaces

Collections

Exception Handling

Java Advanced

Practice Java

GeeksforGeeks
Sanchhaya Education Private Limited

Corporate & Communications Address:

A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)

Registered Address:

K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305

GET IT ON Google Play

Download on the App Store

Company

- About Us
- Legal
- Privacy Policy
- Contact Us
- Advertise with us
- GFG Corporate Solution
- Campus Training Program

Explore

- POTD
- Job-A-Thon
- Blogs
- Nation Skill Up

Tutorials

- Programming Languages
- DSA
- Web Technology
- AI, ML & Data Science
- DevOps
- CS Core Subjects
- Interview Preparation
- Software and Tools

Courses

- ML and Data Science
- DSA and Placements
- Web Development
- Programming Languages
- DevOps & Cloud
- GATE
- Trending Technologies

Videos

- DSA
- Python
- Java
- C++
- Web Development
- Data Science
- CS Subjects

Preparation Corner

- Interview Corner
- Aptitude
- Puzzles
- GfG 160
- System Design