Java - Finally Block

The finally Block in Java

The finally block follows a try block or a catch block. A finally block of code always executes, irrespective of occurrence of an Exception.

Using a finally block allows you to run any cleanup-type statements that you want to execute, no matter what happens in the protected code.

Syntax: Finally Block

A finally block appears at the end of the catch blocks and has the following syntax −

```
try {
   // Protected code
} catch (ExceptionType1 e1) {
   // Catch block
} catch (ExceptionType2 e2) {
   // Catch block
} catch (ExceptionType3 e3) {
   // Catch block
}finally {
   // The finally block always executes.
}
```

Points To Remember While Using Finally Block

- A catch clause cannot exist without a try statement.
- It is not compulsory to have finally clauses whenever a try/catch block is present.
- The try block cannot be present without either catch clause or finally clause.
- Any code cannot be present in between the try, catch, finally blocks.
- finally block is not executed in case exit() method is called before finally block or a fatal error occurs in program execution.
- finally block is executed even method returns a value before finally block.

Learn **Java** in-depth with real-world projects through our **Java certification course**. Enroll and become a certified expert to boost your career.

Why Java Finally Block Used?

- Java finally block can be used for clean-up (closing) the connections, files opened, streams, etc. those must be closed before exiting the program.
- It can also be used to print some final information.

Java Finally Block Example

Here is code segment showing how to use finally after try/catch statements after handling exception. In this example, we're creating an error accessing an element of an array using invalid index. The catch block is handling the exception and printing the same. Now in finally block, we're printing a statement signifying that finally block is getting executed.

Open Compiler

```java
package com.tutorialspoint;

public class ExcepTest {

   public static void main(String args[]) {
      int a[] = new int[2];
      try {
         System.out.println("Access element three :" + a[3]);
      } catch (ArrayIndexOutOfBoundsException e) {
         System.out.println("Exception thrown  :" + e);
      }finally {
         a[0] = 6;
         System.out.println("First element value: " + a[0]);
         System.out.println("The finally statement is executed");
      }
   }
}
```

Output

```
Exception thrown  :java.lang.ArrayIndexOutOfBoundsException: 3
First element value: 6
```

```
The finally statement is executed
```

More Examples

Example 1

Here is code segment showing how to use finally after try/catch statements even exception is not handled. In this example, we're creating an error accessing an element of an array using invalid index. As the catch block is not handling the exception, we can check in output that finally block is printing a statement signifying that finally block is getting executed.

Open Compiler

```
package com.tutorialspoint;

public class ExcepTest {

   public static void main(String args[]) {
      int a[] = new int[2];
      try {
         System.out.println("Access element three :" + a[3]);
      } catch (ArithmeticException e) {
         System.out.println("Exception thrown  :" + e);
      }finally {
         a[0] = 6;
         System.out.println("First element value: " + a[0]);
         System.out.println("The finally statement is executed");
      }
   }
}
```

Output

```
First element value: 6
The finally statement is executed
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3
	at com.tutorialspoint.ExcepTest.main(ExcepTest.java:8)
```

Example 2

Here is code segment showing how to use finally block where a method can return a value within try block. In this example, we're returning a value within try block. We can check in output that finally block is printing a statement signifying that finally block is getting executed even after method returned a value to caller function.

Open Compiler

```java
package com.tutorialspoint;

public class ExcepTest {

   public static void main(String args[]) {
      System.out.println(testFinallyBlock());
   }

   private static int testFinallyBlock() {
      int a[] = new int[2];
      try {
         return 1;
      } catch (ArrayIndexOutOfBoundsException e) {
      System.out.println("Exception thrown  :" + e);
      }finally {
         a[0] = 6;
         System.out.println("First element value: " + a[0]);
         System.out.println("The finally statement is executed");
      }
      return 0;
   }
}
```

Output

```
First element value: 6
The finally statement is executed
1
```