This is why it is important to handle exceptions. Here's a list of different approaches to handle exceptions in Java.

- try...catch block

- finally block

- throw and throws keyword

---

1. Java try...catch block

The try-catch (/java-programming/try-catch) block is used to handle exceptions in Java. Here's the syntax of `try...catch` block:

```
try {
  // code
}
catch(Exception e) {
  // code
}
```

Here, we have placed the code that might generate an exception inside the `try` block. Every `try` block is followed by a `catch` block.

```
        }
      }
    }
```

Run Code » (/java-programming/online-compiler)

Output

```
ArithmeticException => / by zero
```

In the example, we are trying to divide a number by `0`. Here, this code generates an exception.

To handle the exception, we have put the code, `5 / 0` inside the `try` block. Now when an exception occurs, the rest of the code inside the `try` block is skipped.

The `catch` block catches the exception and statements inside the catch block is executed.

If none of the statements in the `try` block generates an exception, the `catch` block is skipped.

Get started with Python
one step at a time.

The basic syntax of `finally` block is:

```
try {
  //code
}
catch (ExceptionType1 e1) {
  // catch block
}
finally {
  // finally block always executes
}
```

If an exception occurs, the `finally` block is executed after the `try...catch` block. Otherwise, it is executed after the try block. For each `try` block, there can be only one `finally` block.

earn/master-java?
com&utm_medium=referral&utm_audience=ORGANIC-
aign=course_promotion&utm_content=interests_learn_java&utm_term=nav_floating_button)

Run Code » (/java-programming/online-compiler)

Output

```
ArithmeticException => / by zero
This is the finally block
```

In the above example, we are dividing a number by **0** inside the `try` block. Here, this code generates an `ArithmeticException`.

The exception is caught by the `catch` block. And, then the `finally` block is executed.

> **Note:** It is a good practice to use the `finally` block. It is because it can include important cleanup codes like,
>
> - code that might be accidentally skipped by return, continue (/java-programming/continue-statement) or break (/java-programming/break-statement)
>
> - closing a file or connection

```
    }

    public static void main(String[] args) {
        divideByZero();
    }
}
```

Run Code » (/java-programming/online-compiler)

Output

```
Exception in thread "main" java.lang.ArithmeticException: Trying to divide by 0
        at Main.divideByZero(Main.java:5)
        at Main.main(Main.java:9)
```

In the above example, we are explicitly throwing the `ArithmeticException` using the `throw` keyword.

Similarly, the `throws` keyword is used to declare the type of exceptions that might occur within the method. It is used in the method declaration.

```
        System.out.println(e);
    }
  }
}
```

Run Code » (/java-programming/online-compiler)

Output

```
java.io.FileNotFoundException: test.txt (The system cannot find the file specifi
```

When we run this program, if the file **test.txt** does not exist, `FileInputStream` throws a `FileNotFoundException` which extends the `IOException` class.

The `findFile()` method specifies that an `IOException` can be thrown. The `main()` method calls this method and handles the exception if it is thrown.

If a method does not handle exceptions, the type of exceptions that may occur within it must be specified in the `throws` clause.

To learn more, visit Java throw and throws (/java-programming/throw-throws).

earn/master-java?
com&utm_medium=referral&utm_audience=ORGANIC-
aign=course_promotion&utm_content=interests_learn_java&utm_term=nav_floating_button)

tutorial: **Java Exceptions** **(/java-programming/exceptions)** **Java try...catch** → **programming/try-catch)**

Share on:

(https://www.facebook.com/sharer/sharer.php?
u=https://www.programiz.com/java-
programming/exception-handling)

(https://twitter.com/intent/tweet?
text=Check%20this%20amazing%2
programming/exception-handling)

Did you find this article helpful?

😀   🙁

Programiz PRO

Your builder path starts here.

Escape tutorial hell and ship real projects.

Try Programiz PRO
(https://programiz.pro/offer/black-
november-2025?
utm_source=programiz.com&utm_medium=referral&utm_audience=ORGANIC-

earn/master-java?
com&utm_medium=referral&utm_audience=ORGANIC-
aign=course_promotion&utm_content=interests_learn_java&utm_term=nav_floating_button)

Java catch Multiple Exceptions

(/java-programming/multiple-exceptions)

Java try-with-resources

(/java-programming/try-with-resources)