

# SCALA EXCEPTION HANDLING

[http://www.tutorialspoint.com/scala/scala\\_exception\\_handling.htm](http://www.tutorialspoint.com/scala/scala_exception_handling.htm)

Copyright © tutorialspoint.com

Scala's exceptions work like exceptions in many other languages like Java. Instead of returning a value in the normal way, a method can terminate by throwing an exception. However, Scala doesn't actually have checked exceptions.

When you want to handle exceptions, you use a `try{...}catch{...}` block like you would in Java except that the catch block uses matching to identify and handle the exceptions.

## Throwing exceptions:

Throwing an exception looks the same as in Java. You create an exception object and then you throw it with the **throw** keyword:

```
throw new IllegalArgumentException
```

## Catching exceptions:

Scala allows you to **try/catch** any exception in a single block and then perform pattern matching against it using **case** blocks as shown below:

```
import java.io.FileReader
import java.io.FileNotFoundException
import java.io.IOException

object Test {
  def main(args: Array[String]) {
    try {
      val f = new FileReader("input.txt")
    } catch {
      case ex: FileNotFoundException => {
        println("Missing file exception")
      }
      case ex: IOException => {
        println("IO Exception")
      }
    }
  }
}
```

When the above code is compiled and executed, it produces the following result:

```
C:/>scalac Test.scala
C:/>scala Test
Missing file exception

C:/>
```

The behavior of this **try-catch** expression is the same as in other languages with exceptions. The body is executed, and if it throws an exception, each **catch** clause is tried in turn.

## The finally clause:

You can wrap an expression with a **finally** clause if you want to cause some code to execute no matter how the expression terminates.

```
import java.io.FileReader
import java.io.FileNotFoundException
import java.io.IOException

object Test {
  def main(args: Array[String]) {
```

```
try {
  val f = new FileReader("input.txt")
} catch {
  case ex: FileNotFoundException => {
    println("Missing file exception")
  }
  case ex: IOException => {
    println("IO Exception")
  }
} finally {
  println("Exiting finally...")
}
}
```

When the above code is compiled and executed, it produces the following result:

```
C:/>scalac Test.scala
C:/>scala Test
Missing file exception
Exiting finally...

C:/>
```