

Kotlin - Exception Handling

Exception handling is a very important part of a programming language. This technique restricts our application from generating the wrong output at runtime. In this chapter, we will learn how to handle runtime exception in Kotlin. The exceptions in Kotlin is pretty similar to the exceptions in Java. All the exceptions are descendants of the “Throwable” class. Following example shows how to use exception handling technique in Kotlin.

```
fun main(args: Array<String>) {  
    try {  
        val myVar:Int = 12;  
        val v:String = "Tutorialspoint.com";  
        v.toInt();  
    } catch(e:Exception) {  
        e.printStackTrace();  
    } finally {  
        println("Exception Handeling in Kotlin");  
    }  
}
```

In the above piece of code, we have declared a String and later tied that string into the integer, which is actually a runtime exception. Hence, we will get the following output in the browser.

```
val myVar:Int = 12;  
Exception Handeling in Kotlin
```

Note – Like Java, Kotlin also executes the finally block after executing the catch block.