

BUILT-IN EXCEPTIONS

Built-in exceptions are the **exceptions which are available in Java libraries.**

These exceptions are suitable to explain certain error situations. Below is the list of important built-in exceptions in Java

Exceptions	Description
Arithmetic Exception	It is thrown when an exceptional condition has occurred in an arithmetic operation.
Array Index Out Of Bound Exception	It is thrown to indicate that an array has been accessed with an illegal index. The index is either negative or greater than or equal to the size of the array.
ClassNotFoundException	This Exception is raised when we try to access a class whose definition is not found.
FileNotFoundException	This Exception is raised when a file is not accessible or does not open.
IOException	It is thrown when an input-output operation failed or interrupted.
InterruptedException	It is thrown when a thread is waiting, sleeping, or doing some processing, and it is interrupted.
NoSuchFieldException	It is thrown when a class does not contain the field (or variable) specified.
NoSuchMethodException	It is thrown when accessing a method which is not found.
NullPointerException	This exception is raised when referring to the members of a null object. Null represents nothing.
NumberFormatException	This exception is raised when a method could not convert a string into a numeric format.
RuntimeException ★	This represents any exception which occurs during runtime.
StringIndexOutOfBoundsException	It is thrown by String class methods to indicate that an index is either negative than the size of the string

The following Java program explains **NumberFormatException**

```

class NumberFormat_Example
{
    public static void main(String args[])
    {
        try {
            int num = Integer.parseInt ("hello");
            System.out.println(num);
        }
        catch(NumberFormatException e) {
            System.out.println("Number format exception");
        }
    }
}

```

```
}  
}
```

The following Java program explains StackOverflowError exception.

```
class Example {  
    public static void main(String[] args)  
    {  
        fun1();  
    }  
    public static void fun1()  
    {  
        fun2();  
    }  
    public static void fun2()  
    {  
        fun1();  
    }  
}
```

Output:

```
Exception in thread "main" java.lang.StackOverflowError at  
Example.fun2(File.java:14)  
at Example.fun1(File.java:10)
```

