Priyansh Salian

Roll No-2003148

Class-C3

Experiment 9 –Exception Handling

Theory-

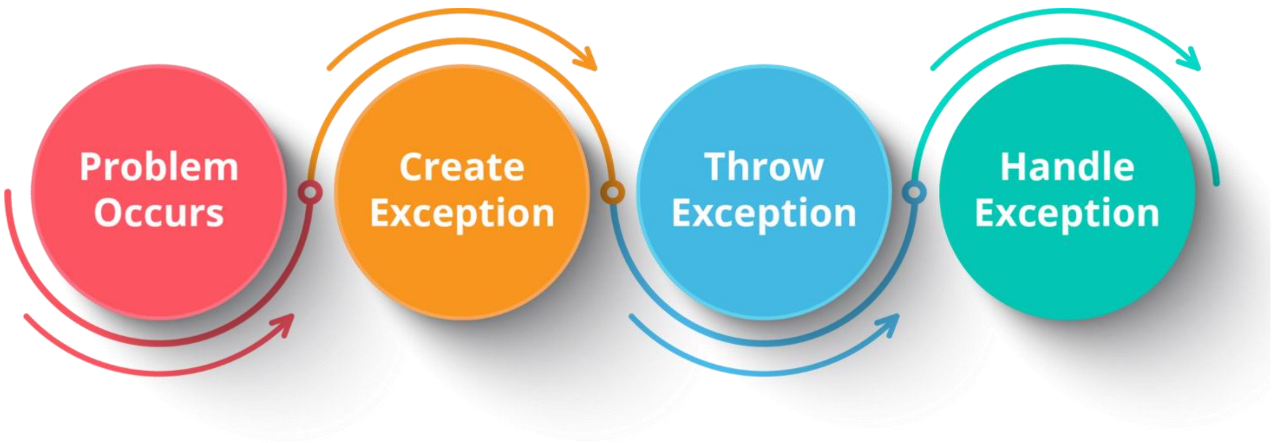
What is meant by exception handling?

Errors arise unexpectedly and can result in disrupting the normal flow of execution. This is something that every programmer faces at one point or the other while coding. **Java,** being the most prominent **object-oriented language,** provides a powerful mechanism to handle these errors/exceptions.

An exception is a problem that arises during the execution of a program. It can occur for various reasons say-

- A user has entered an invalid data
- File not found
- A network connection has been lost in the middle of communications
- The JVM has run out of a memory

Exception Handling mechanism follows a flow which is depicted in the below figure. But if an exception is not handled, it may lead to a system failure. That is why handling an exception is very important.



You may also go through this recording of Java Exception Handling where you can understand the topics in a detailed manner with examples.

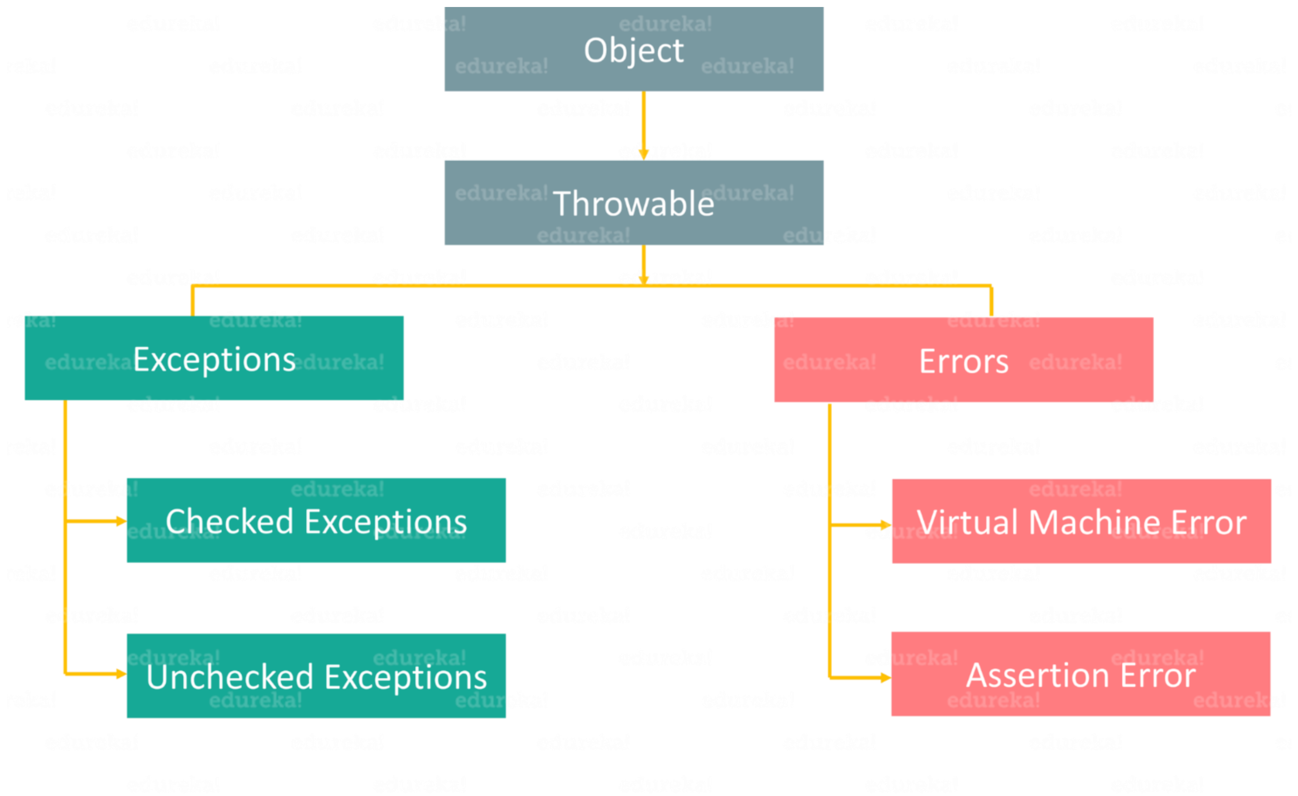
What happens if exceptions are not handled?

When an exception occurs, and if you don't handle it, the program will terminate abruptly (the piece of code after the line causing the exception will not get executed).

Through this article on Java Exception Handling, I will give you a complete insight into the fundamentals and various methods of Exception Handling.

Exceptions Hierarchy

All exception and error types are subclasses of class Throwable, which is the base class of hierarchy. One branch is headed by Error which occurs at run-time and other by Exception that can happen either at compile time or run-time.



Basically, an Error is used by the Java run-time system (JVM) to indicate errors that are associated with the run-time environment (JRE). StackOverflowError is an example of such an error. Whereas Exception is used for exceptional conditions that user programs should catch. NullPointerException is an example of such an exception.

Now that you know what errors and exceptions are, let’s find out the basic difference between them. Take a look at the below table which draws a clear line between both of them.

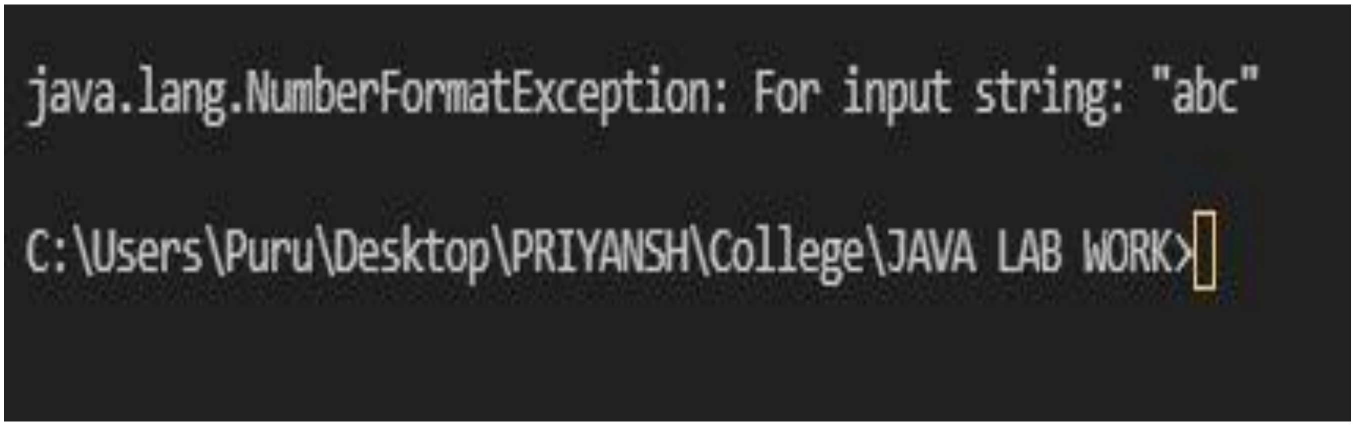
Errors	Exceptions
1. Impossible to recover from an error	1. Possible to recover from exceptions
2. Errors are of type 'unchecked'	2. Exceptions can be either 'checked' or 'unchecked'
3. Occur at runtime	3. Can occur at compile time or run time
4. Caused by the application running environment	4. Caused by the application itself

AIM-WAP to catch any three built-in exceptions

Program-

```
import java.util.*;

public class expa {

    public static void main(String[] args) {

        int a =50,b=0;

        int []arr= {1,5,6,7};

        String s="abc";

        try{

           // int data= (a/b);
        //int data2=arr[6];

        int i=Integer.parseInt(s);

    }

    catch(ArithmeticException e){

        System.out.println(e);

    }

    catch(ArrayIndexOutOfBoundsException e){

        System.out.println(e);

    }

    catch(NumberFormatException e){

        System.out.println(e);

    }

    catch(Exception e){

        System.out.println(e);

    }


}
```

Output-



```
java.lang.NumberFormatException: For input string: "abc"

C:\Users\Puru\Desktop\PRIYANSH\College\JAVA LAB WORK>
```