Idris Ratlamwala
C31 - 2003145

# Experiment 9: Program on Exception handling

## Theory :

An **Exception** is an unwanted event that interrupts the normal flow of the program. When an exception occurs program execution gets terminated. In such cases we get a system generated error message. The good thing about exceptions is that they can be handled in Java. By handling the exceptions we can provide a meaningful message to the user about the issue rather than a system generated message, which may not be understandable to a user.

If an exception occurs, which has not been handled by programmer then program execution gets terminated and a system generated error message is shown to the user.

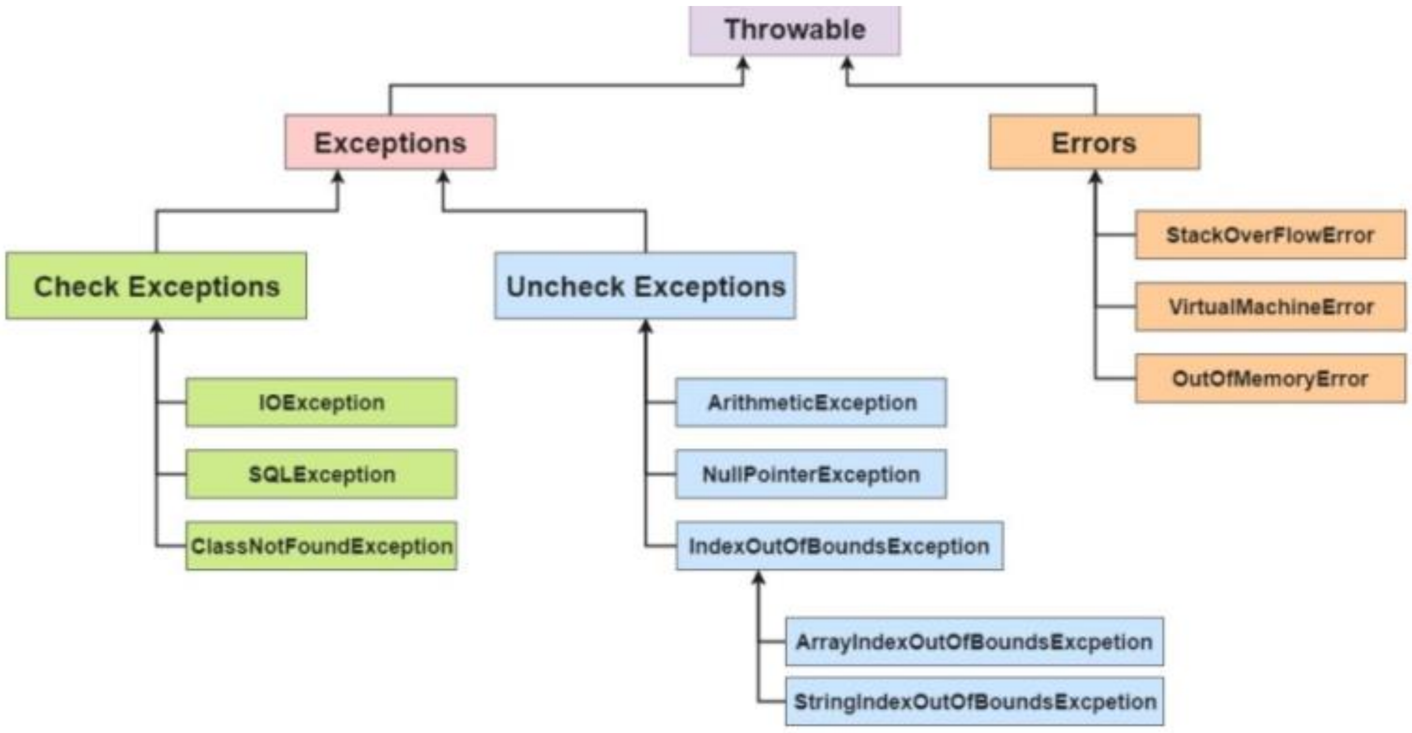
**Advantage of exception handling**
Exception handling ensures that the flow of the program doesn't break when an exception occurs. For example, if a program has bunch of statements and an exception occurs mid way after executing certain statements then the statements after the exception will not execute and the program will terminate abruptly.
By handling we make sure that all the statements execute and the flow of program doesn't break.

There are two types of exceptions in Java:
1)Checked exceptions
2)Unchecked exceptions

**Checked exceptions**
All exceptions other than Runtime Exceptions are known as Checked exceptions as the compiler checks them during compilation to see whether the programmer has handled them or not. If these exceptions are not handled/declared in the program,

you will get compilation error. For example, SQLException, IOException, ClassNotFoundException etc.

**Unchecked Exceptions**
Runtime Exceptions are also known as Unchecked Exceptions. These exceptions are not checked at compile-time so compiler does not check whether the programmer has handled them or not but it's the responsibility of the programmer to handle these exceptions and provide a safe exit. For example, ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc.



## Try block :

The try block contains set of statements where an exception can occur. A try block is always followed by a catch block, which handles the exception that occurs in associated try block. A try block must be followed by catch blocks or finally block or both.

**Syntax of try block**

```
try{
   //statements that may cause an exception
}
```

While writing a program, if we suspect that certain statements in a program can throw a exception, enclosed them in try block and handle that exception

## Catch block :

A catch block is where you handle the exceptions, this block must follow the try block. A single try block can have several catch blocks associated with it. You can catch different exceptions in different catch blocks. When an exception occurs in try block, the corresponding catch block that handles that particular exception executes. For example if an arithmetic exception occurs in try block then the statements enclosed in catch block for arithmetic exception executes.

### Syntax of try catch in java

```
try
{
     //statements that may cause an exception
}
catch (exception(type) e(object))
{
     //error handling code
}
```

**A.**

**Aim :** WAP to catch any three built-in exceptions.

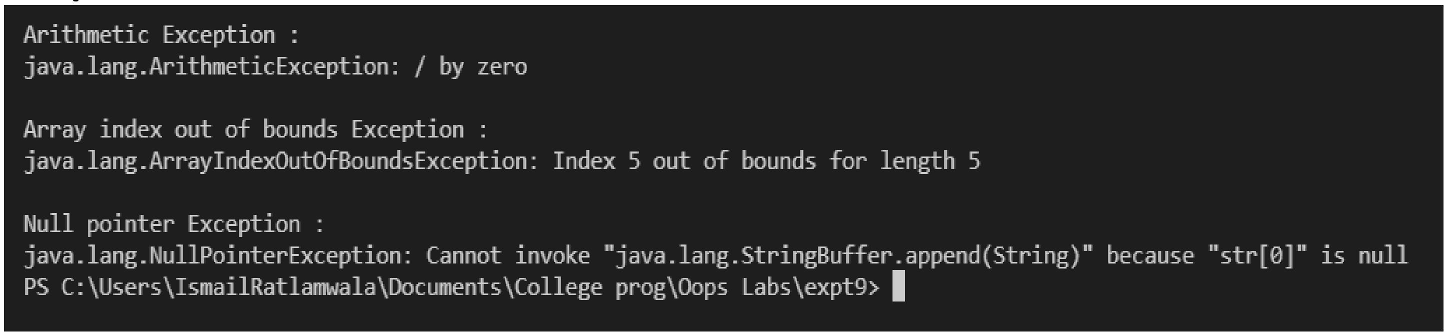
**Program :**

```java
public class ExceptionHandling{
    public static void main(String[] args) {
        System.out.println("\nArithmetic Exception :");
        try {
            int a=10,b=0;
            int c=a/b;
            System.out.println(c);
        } catch (ArithmeticException e) {
            System.out.println(e);
        }

        System.out.println("\nArray index out of bounds Exception :");
        try {
            int[] a = {10,20,30,40,50};
            System.out.println(a[5]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println(e);
        }

        System.out.println("\nNull pointer Exception :");
        try{
            StringBuffer[] str = new StringBuffer[4];
            str[0].append("Hello");
            System.out.println(str[0]);
        } catch(NullPointerException e){
            System.out.println(e);
        }
    }
}
```

**Output :**



```
Arithmetic Exception :
java.lang.ArithmeticException: / by zero

Array index out of bounds Exception :
java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5

Null pointer Exception :
java.lang.NullPointerException: Cannot invoke "java.lang.StringBuffer.append(String)" because "str[0]" is null
PS C:\Users\IsmailRatlamwala\Documents\College prog\Oops Labs\expt9>
```