Exception Handling

- Exception is an abnormal condition.
- Exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.

Types of Exceptions:

1. Checked exception
2. Unchecked exception

Checked Exception:

Checked exceptions are called compile-time exceptions because these exceptions are checked at compile-time by the compiler.

1. **ClassNotFoundException:**

```
public class Student {
   public static void main(String args[])
   {
   try {
         Class.forName("Teacher");
      }
      catch (ClassNotFoundException exception) {
         exception.printStackTrace();
      }
   }
}
```

2. **IOException:**

```
import java.io.*;

class Main {

   public static void main(String[] args) {

      try {

        FileReader fileReader = new FileReader("input.txt");

        System.out.println(fileReader.read());

        fileReader.close();

   } catch (IOException exceptin) {

      exception.printStackTrace();

   }

 }

}
```

Output: java.io.FileNotFoundException: input.txt (No such file or directory)

3. **FileNotFoundException:**

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
 class File_notFound_Demo {

    public static void main(String args[])  {
        try {

            // Following file does not exist
            File file = new File("E://file.txt");

            FileReader fr = new FileReader(file);
        } catch (FileNotFoundException e) {
           System.out.println("File does not exist");
        }
    }
}
Output:File does not exist
```

Unchecked Exception:

The unchecked exceptions are just opposite to the checked exceptions. The compiler will not check these exceptions at compile time.

1.Arithmetic Exception:

```
class ArithmeticExceptionDemo
{
    public static void main(String args[])
    {
        try {
            int a = 30, b = 0;
            int c = a/b;
            System.out.println ("Result : " + c);
        }
        catch(ArithmeticException e) {
            System.out.println ("Can't divide a number by 0");
        }
    }
}
Output: Can't divide a number by 0
```

2.NullPointerException:

```
class NullPointerDemo
{
    public static void main(String args[])
    {
        try {
            String a = null; //null value
            System.out.println(a.charAt(0));
        } catch(NullPointerException e) {
            System.out.println("NullPointerException");
        }
    }
}
```
Output: NullPointerException

3.NumberFormatException:

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

Output:Number format Exception

4.ArrayIndexOutOfBoundException:

```
class ArrayIndexOutOfBoundDemo
{
    public static void main(String args[])
    {
        try{
            int a[] = new int[5];
            a[6] = 9;
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println ("Array Index is Out Of Bounds");
        }
    }
}
```
Output: Array Index is Out Of Bounds

5.StringIndexOutOfBoundException:

```
class StringIndexOutOfBound_Demo
{
    public static void main(String args[])
    {
        try {
            String a = "World is Amazing";
            char c = a.charAt(24);
            System.out.println(c);
        }
        catch(StringIndexOutOfBoundsException e) {
            System.out.println("StringIndexOutOfBoundsException");
        }
    }
}
```

Output: StringIndexOutOfBoundsException