Java Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc

**Major reasons why an exception Occurs**

* Invalid user input
* Device failure
* Loss of network connection
* Physical limitations (out of disk memory)
* Code errors
* Opening an unavailable file
* **Error:**An Error indicates a serious problem that a reasonable application should not try to catch.
* **Exception:**Exception indicates conditions that a reasonable application might try to catch.

The ***advantages of Exception Handling in Java***are as follows:

1. Provision to Complete Program Execution
2. Easy Identification of Program Code and Error-Handling Code
3. Propagation of Errors
4. Meaningful Error Reporting
5. Identifying Error Types

**Exception** is an unwanted or unexpected event, which occurs during the execution of a program, i.e. at run time, that disrupts the normal flow of the program’s instructions. In Java, there are two types of exceptions:

### 3.1. ****Checked Exceptions****

Checked exceptions are exceptions that we developers should anticipate and handle properly

For example ; if a file doesn’t exist maybe it got deleted , instead of letting runtime terminate our program we’d better display a friendly message to the user says ”hey file doesn’t exist ”.

Checked exceptions are exceptions that the Java compiler requires us to handle. We have to either declaratively throw the exception up the call stack, or we have to handle it ourselves. More on both of these in a moment.

### ****3.2. Unchecked Exceptions (runtime exceptions)****

Unchecked exceptions are exceptions that the Java compiler does not require us to handle.

Simply put, if we create an exception that extends RuntimeException, it will be unchecked; otherwise, it will be checked.

Some examples of unchecked exceptions are NullPointerException, IllegalArgumentException, and SecurityException.

### ****3.3. Errors****

Errors represent serious and usually irrecoverable conditions like a library incompatibility, infinite recursion, or memory leaks.

And even though they don't extend RuntimeException, they are also unchecked.

A couple of examples of errors are a StackOverflowError and OutOfMemoryError