**Exception Handling**

### Exception Handling

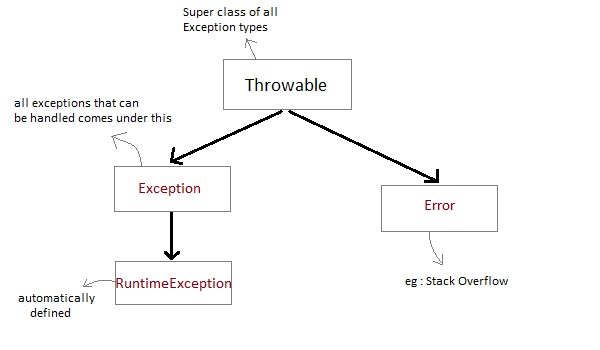
Exception Handling is the mechanism to handle runtime malfunctions. We need to handle such exceptions to prevent abrupt termination of program. The term exception means exceptional condition, it is a problem that may arise during the execution of program. A bunch of things can lead to exceptions, including programmer error, hardware failures, files that need to be opened cannot be found, resource exhaustion etc.

#### Exception

A Java Exception is an object that describes the exception that occurs in a program. When an exceptional events occurs in java, an exception is said to be thrown. The code that's responsible for doing something about the exception is called an **exception handler**.

#### Exception class Hierarchy

All exception types are subclasses of class **Throwable**, which is at the top of exception class hierarchy.



* **Exception** class is for exceptional conditions that program should catch. This class is extended to create user specific exception classes.
* **RuntimeException** is a subclass of Exception. Exceptions under this class are automatically defined for programs.
* **Exceptions** of type **Error** are used by the Java run-time system to indicate errors having to do with the run-time environment, itself. Stack overflow is an example of such an error.

#### Exception are categorized into 3 category.

* **Checked Exception**

The exception that can be predicted by the programmer at the compile time.*Example :* File that need to be opened is not found. These type of exceptions must be checked at compile time.

* **Unchecked Exception**

Unchecked exceptions are the class that extends RuntimeException. Unchecked exception are ignored at compile time. *Example :* ArithmeticException, NullPointerException, Array Index out of Bound exception. Unchecked exceptions are checked at runtime.

* **Error**

Errors are typically ignored in code because you can rarely do anything about an error. *Example :* if stack overflow occurs, an error will arise. This type of error cannot be handled in the code.

#### Uncaught Exceptions

When we don't handle the exceptions, they lead to unexpected program termination. Lets take an example for better understanding.

class UncaughtException

{

public static void main(String args[])

{

int a = 0;

int b = 7/a; // Divide by zero, will lead to exception

}

}

This will lead to an exception at runtime, hence the Java run-time system will construct an exception and then throw it. As we don't have any mechanism for handling exception in the above program, hence the default handler will handle the exception and will print the details of the exception on the terminal.

