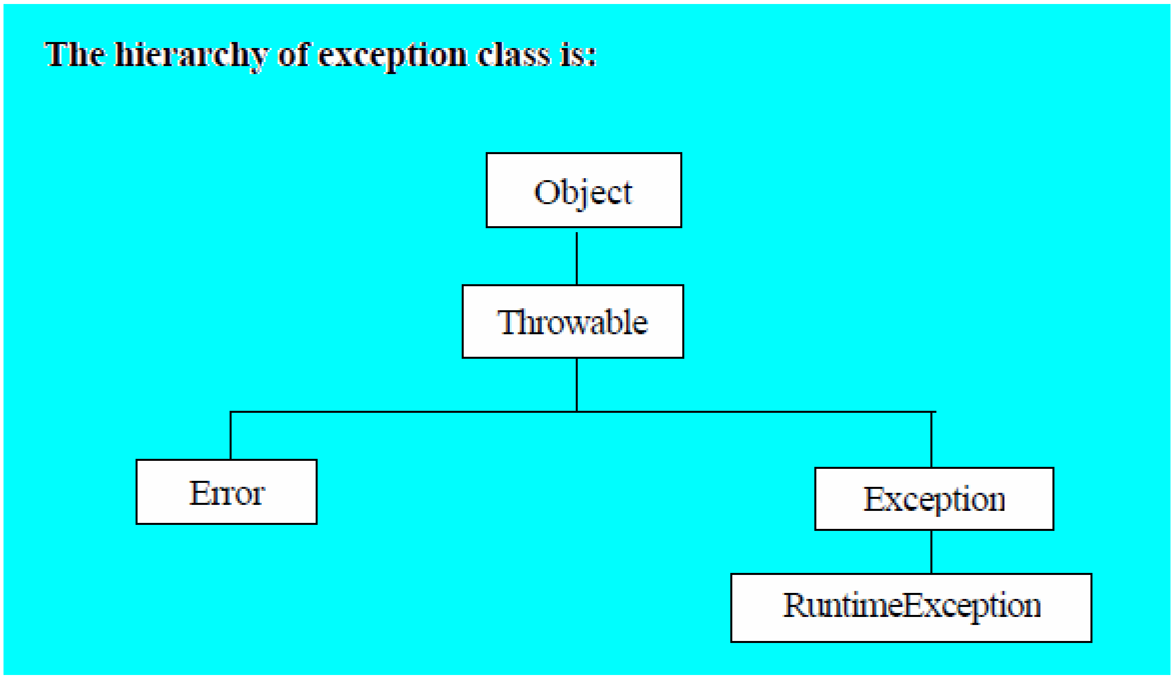
Exception :- Exception is an abnormal condition (such as errors) that occurs in a Java program.Whenever, an error is encountered in a program, it's execution abnormally terminates.

All exceptions are subclasses of **Throwable** class. It is the subclass of **Object** class and the top of the exception hierarchy. Just below **Throwable**, are two classes named **Exception** and **Error**. These classes are present in **java.lang package**.



The hierarchy of exception class is:

Java handles exceptions using five keywords **try, catch, finally** and **throw,throws.** Let us study each of them in details.

try : - If any part of code is expected to give rise to an exception, it is put inside a **try** block The statements that comprise the **try** block should be enclosed within curly braces. A **try** block cannot comprise of a single statement. It is possible to develop nested **try** blocks that is one **try** block can be placed inside another **try** block.

catch :- The **catch** block comes after the try block and performs the actual error handling functions. One **try** block may be associated with one or more **catch** blocks. All the corresponding **catch** blocks for a **try** block should come after the try block.

finally:- This comes after the try and catch block. **finally** block is optional but if defined,it will execute regardless of whether or not an exception is throw or not. If a **catch** is defined after the **try** block then the **finally** block will be executed after the execution of the **catch** block. If no **catch** block is defined after the **try** block then the **finally** block will be executed after the **try** block.

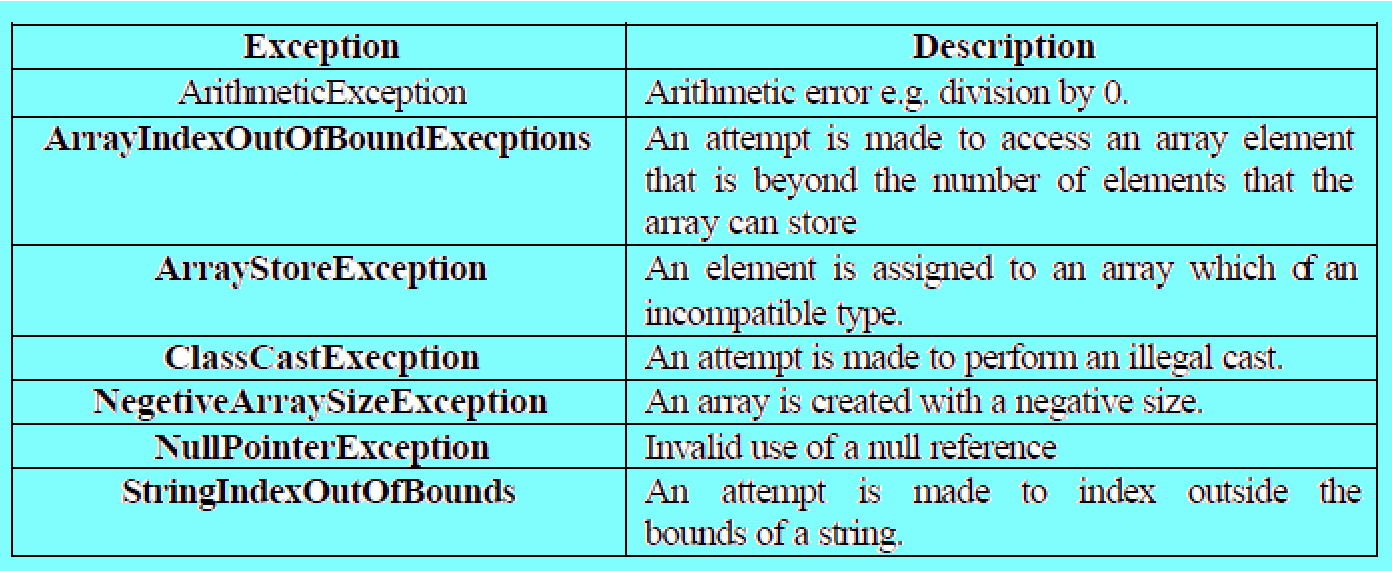
throw:- The **throw** statement can be used to explicitly throw an exception. Its syntax is

```
throw Throwable_Instance
```

The exception that can arise in a Java program can be categorized into:

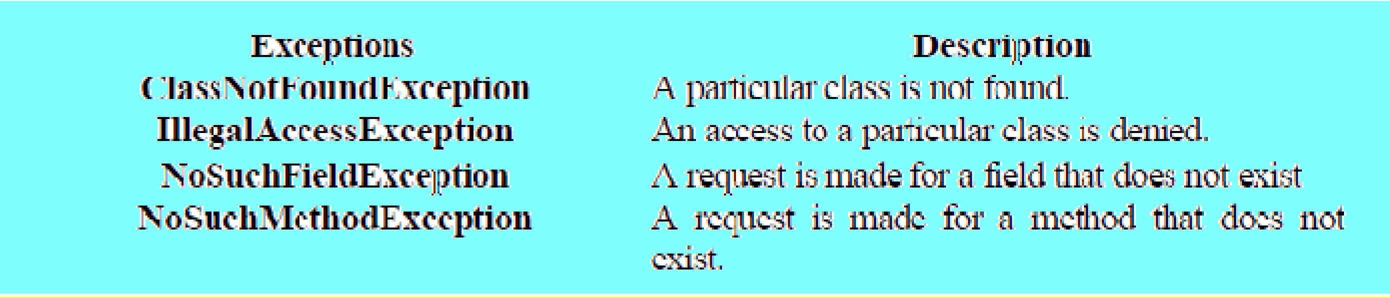
Ø Checked Exceptions
Ø Unchecked Exceptions

The following table illustrates some of the important **unchecked exceptions**:



Exception	Description
ArithmeticException	Arithmetic error e.g. division by 0.
ArrayIndexOutOfBoundExecptions	An attempt is made to access an array element that is beyond the number of elements that the array can store
ArrayStoreException	An element is assigned to an array which of an incompatible type.
ClassCastExecption	An attempt is made to perform an illegal cast.
NegetiveArraySizeException	An array is created with a negative size.
NullPointerException	Invalid use of a null reference
StringIndexOutOfBounds	An attempt is made to index outside the bounds of a string.

Checked Exceptions are checked during compile time by the compiler. They must be included in a method's **throws** list if that method is capable of throwing that exception but does not handle such exception by itself. Some of the important checked exceptions are listed in the following table.



Exceptions	Description
ClassNotFoundException	A particular class is not found.
IllegalAccessException	An access to a particular class is denied.
NoSuchFieldException	A request is made for a field that does not exist
NoSuchMethodException	A request is made for a method that does not exist.