**Exceptions**

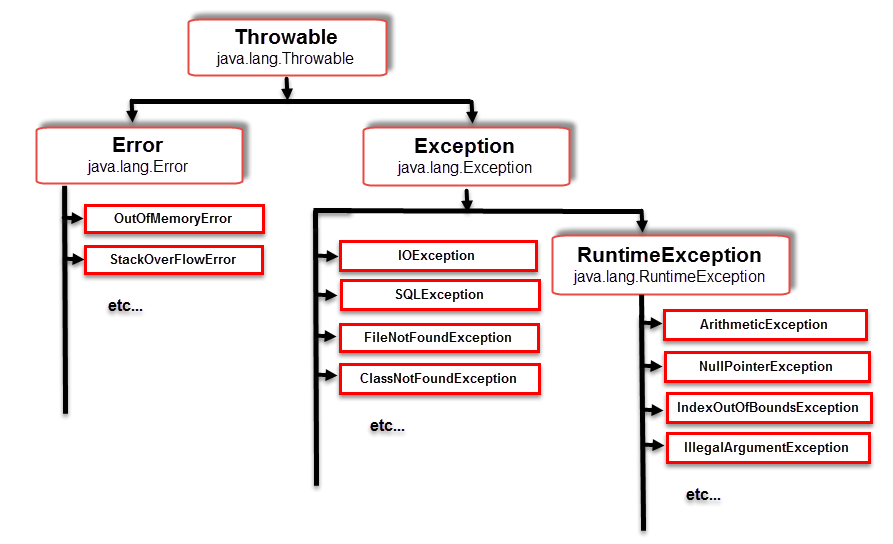
* An exception is a unwanted or unexpected condition which disturbs our normal flow of execution

Ex : 1) corona exception 2) lockdown exception

* Once exception occurred remaining part of program will not be executed.
* So it is our responsibility to handle the exception.
* Exception handling doesn’t mean we are resolving an exception it is just like providing an alternate solution so that even though exception happen our program should work properly.

**Exception Hierarchy :**

* Object class is a super class to all the predefine and userdefine classes of java
* Throwable class is a supper class to “Exception” class and “Error” class
* Exception class is a supper class to runtimeException class and other exception classes.
* All Exception classes belongs to java.lang package.



**Checked exception :**

* Exceptions which are checked during compile time by compiler, such type of exceptions are called as checked exception
* Checked exceptions are also called as compile time exceptions.

**Example :** 1)IOException 2) SQLException, 3) InterruptedException

**Unchecked Exception :**

• Exception which are checked(identified or found out) during Runtime or execution time, such type of exception are called as Unchecked Exceptions.

• Incase of Unchecked Exception our program will at least compiles successfully.

• Unchecked Exceptions are also called as Runtime Exceptions.

• RuntimeExceptionclass is a super class to all UncheckedException

classes.

• Examples(Classes) of Unchecked Exceptions are :-

-ArithmeticException

-ArrayIndexOutOfBoundsException

-NullPointerException

-StringIndexOutofBoundsException

-ClassCastException

-NumberFormatException

Error :

• An Error is an irrecoverable Condition i.e, if error occurred it is

not under programmers control to get over it.

• For Ex: if we develop any program whose size is 4gb but our system's

storage is 3gb so such condition is not in programmers control and

such situation is referred as Error.

• Examples(Classes) of Error are :-

-StackoverFlowError

-VirtualMemoryError

-404pagenotfound

class Sample

{

public static void main(String[] args)

{

try

{

int a,b,c;

a= 10;

b= 0;

c = a/b;

}

catch (ArithmeticException e)

{

System.out.println("arithmetic exception");

}

finally

{

System.out.println("finally block");

}

}

}

**Throws :**

**public** **class** Main

{

**public** **static** **void** main(String[] args) **throws** Throwable

{

Demo d1 = **new** Demo();

d1.start();

**for**(**int** i=0; i<5; i++)

{

System.***out***.println("Thread.sleep");

Thread.*sleep*(500);

}

}

}

User define Exception (or) customized Exception

-------------------------------------------------

When pre-define exceptions does not fulfil our requirement, we will go for user-define exception i.e we can create our exceptions. Such type of exception is called as User define Exceptions (or) Customized Exceptions.

* Rules for creating User define Exceptions:-
* create our Exception class and that class should be or must be extending either throwable (or) Exception (or) RuntimeException classes
* Preferable to extend RuntimeException
* Define constructor whenever requires.
* Throw Exception as per our own requirement.

class NotEligibleException extends RuntimeException

{

public NotEligibleException(String msg)

{

System.out.println(msg);

}

}

public class User

{

public static void main(String args[])

{

float percentage=56.5f;

if(percentage<60)

{

throw new NotEligibleException("Not Eligible for drive");

}

else

{

System.out.println("Register before end of the day");

}

}

}