**User defined exception in java**

**User defined exceptions** **in java** are also known as **Custom exceptions**. Most of the times when we are developing an application in java, we often feel a need to create and throw [**our own exceptions**](http://beginnersbook.com/2013/04/java-exception-handling/). These exceptions are known as **User defined or Custom exceptions.**In this [**tutorial**](http://beginnersbook.com/2013/05/java-tutorial/) we will see how to create and [**throw such  exceptions in a java program**](http://beginnersbook.com/2013/04/throw-in-java/).

**Example of User defined exception in Java**

class MyException extends Exception{

String str1;

MyException(String str2) {

str1=str2;

}

public String toString(){

return ("Output String = "+str1) ;

}

}

class CustomException{

public static void main(String args[]){

try{

throw new MyException("Custom");

// I'm throwing user defined custom exception above

}

catch(MyException exp){

System.out.println("Hi this is my catch block") ;

System.out.println(exp) ;

}

}

}

**Output:**

Hi this is my catch block

Output String = Custom

**Key-points from above example**:  
You can see that while throwing my custom exception I gave a string in parenthesis ( throw new MyException("Custom");). That’s the reason we have a parametric constructor (with a String parameter) in my custom exception class.

**Notes:**

* User defined exception needs to [**inherit (extends)**](http://beginnersbook.com/2013/03/inheritance-in-java/) Exception class in order to [**act as an exception**](http://beginnersbook.com/2013/04/java-checked-unchecked-exceptions-with-examples/).
* throw keyword is used to throw such exceptions.

**Another example wherein we will modify the error message of Exception Class**

Part 1: I have created my own exception class-**MyException** by inheriting the parent class **Exception** then I have [**defined a parametric constructor**](http://beginnersbook.com/2013/03/constructors-in-java/) of my class with a String parameter. In the constructor I called super(), super refers to the super class { My class has inherited Exception class so Exception class is my superclass}. In this way I have modified the system generated message by my own message.

public class MyException extends Exception

{

public MyException(String mymsg)

{

super(mymsg);

}

}

Part2:

public class ExceptionSample

{

public static void main(String args[]) throws Exception

{

ExceptionSample es = new ExceptionSample();

es.displayMymsg();

}

public void displayMymsg() throws MyException

{

for(int j=8;j>0;j--)

{

System.out.println("j= "+j);

if(j==7)

{

throw new MyException("This is my own Custom Message");

}

}

}

}

**Output:**

j = 8

j = 7

Exception in thread "main" MyException: This is my own Custom Message

at ExceptionSample.displayMymsg( ExceptionSample.java.19)

...