**Throws clause in java – Exception handling**

**Use of throws keyword in Java**

1. The [**throws keyword**](http://beginnersbook.com/2013/04/difference-between-throw-and-throws-in-java/) is used in method declaration, in order to explicitly specify the exceptions that a particular method might throw. When a method declaration has one or more exceptions defined using throws clause then the method-call must handle all the defined exceptions.  
2. When defining a method you must include a throws clause to [**declare those exceptions**](http://beginnersbook.com/2013/04/java-exception-handling/) that might be thrown but doesn’t get caught in the method.  
3. If a method is using throws clause along with few exceptions then this implicitly tells other methods that – “ If you call me, you must handle these exceptions that I throw”.

**Syntax of Throws in java:**

void MethodName() throws ExceptionName{

Statement1

...

...

}

E.g:

public void sample() throws IOException{

//Statements

//if (somethingWrong)

IOException e = new IOException();

throw e;

//More Statements

}

**Note:** In case a method throws more than one exception, all of them should be listed in throws clause. PFB the example to understand the same.

public void sample() throws IOException, SQLException

{

//Statements

}

The above method has both IOException and SQLException listed in throws clause. There can be [**any number of exceptions**](http://beginnersbook.com/2013/04/user-defined-exception-in-java/) defined using throws clause.

**Complete Example of Java throws Clause**

class Demo

{

static void throwMethod() throws NullPointerException

{

System.out.println ("Inside throwMethod");

throw new NullPointerException ("Demo");

}

public static void main(String args[])

{

try

{

throwMethod();

}

catch (NullPointerException exp)

{

System.out.println ("The exception get caught" +exp);

}

}

}

**The output of the above program is:**

Inside throwMethod

The exception get caught java.lang.IllegalAccessException: Demo