

# Catching Multiple Exceptions in single catch and Rethrowing Exceptions with Improved Type Checking - Java 7 Feature

In Java 7, catch block has been improved to handle multiple exceptions in a single catch block. If you are catching multiple exceptions and they have similar code, then using this feature will reduce code duplication. Let's understand this with an example.

Before Java 7:

```
1  catch (IOException ex) {
2      logger.error(ex);
3      throw new MyException(ex.getMessage());
4  catch (SQLException ex) {
5      logger.error(ex);
6      throw new MyException(ex.getMessage());
7  }catch (Exception ex) {
8      logger.error(ex);
9      throw new MyException(ex.getMessage());
10 }
```

In Java 7, we can catch all these exceptions in a single catch block as:

```
1  catch(IOException | SQLException | Exception ex){
2      logger.error(ex);
3      throw new MyException(ex.getMessage());
4  }
```

If a catch block handles multiple exception, you can separate them using a pipe (|) and in this case exception parameter (ex) is final, so you can't change it. The byte code generated by this feature is smaller and reduce code redundancy.

Another improvement is done in Compiler analysis of rethrown exceptions. This feature allows you to specify more specific exception types in the throws clause of a method declaration.

Let's see this with a small example:

```
1  package com.journaldev.util;
2
3  public class Java7MultipleExceptions {
4
5      public static void main(String[] args) {
6          try{
7              rethrow("abc");
8          }catch(FirstException | SecondException | ThirdException e){
9              //below assignment will throw compile time exception since e is f:
10             //e = new Exception();
11         }
```

```

11         System.out.println(e.getMessage());
12     }
13 }
14
15 static void rethrow(String s) throws FirstException, SecondException,
16     ThirdException {
17     try {
18         if (s.equals("First"))
19             throw new FirstException("First");
20         else if (s.equals("Second"))
21             throw new SecondException("Second");
22         else
23             throw new ThirdException("Third");
24     } catch (Exception e) {
25         //below assignment disables the improved rethrow exception type cl
26         // e=new ThirdException();
27         throw e;
28     }
29 }
30
31 static class FirstException extends Exception {
32
33     public FirstException(String msg) {
34         super(msg);
35     }
36 }
37
38 static class SecondException extends Exception {
39
40     public SecondException(String msg) {
41         super(msg);
42     }
43 }
44
45 static class ThirdException extends Exception {
46
47     public ThirdException(String msg) {
48         super(msg);
49     }
50 }
51
52 }

```

As you can see that in `rethrow` method, catch block is catching Exception but it's not part of throws clause. Java 7 compiler analyze the complete try block to check what types of exceptions are thrown and then rethrown from the catch block.

Note that this analysis is disabled if you change the catch block argument.

Further Readin: [Java Exception Handling Tutorial](#)