# SYSC 2004 Object-0riented Software Development

# Lab 12

---

**Lab 12:**

**Background Reading**

- *Objects First with Java*, Chapter 14.

**Objective**

The objective of this lab is to gain further experience with Java Exceptions.

**Getting Started**

1. Download file `money-exceptions.zip` from cuLearn. Save the file to the desktop.
2. Right-click on the `money-exceptions.zip` folder and select Extract All... to extract all the files into a folder called `money-exceptions`.
3. Launch BlueJ and open the `money-exceptions` project.

**Part 0 - Exploring the Example**

1. Have a look at class `Test`'s constructor. You will see that it includes six tests of the newly added `addMonies` method in the `Money` class.
2. Look at the `addMonies` method. It's at the end of the `Money`class.
3. Create a `Test` object by invoking its constructor. What happened?
4. Comment out test #4 (don't delete it!) and create a `Test` object. What happened?
5. Leave test #4 commented out and also comment out test #5 (don't delete it). Create a `Test` object. What happened?
6. Remove the comments from test #4 and #5.

**Part 1 - Improving `addMonies`**

1. Update the `addMonies` method so that it does the following:
    a. Throws an IllegalArgumentException if number is not between 1 and 10.
    b. Throws an IllegalArgumentException if cents is not between 0 and 99.
    c. Throws a NullPointerException if obj is null.
    d. Throws a ClassCastException if obj is not a Money object.
2. Ensure that a useful message is provided with each of these exceptions, so that the user knows what happened.
3. Don't forget your javadoc comments (i.e. @throws <ExceptionName> <explanation>). Generate your documentation to check that it looks correct.
4. Test your changes by running the tests (i.e. creating `Test` objects). Comment out the tests that cause exceptions one after another (as in Part 0) so you can check that tests 2, 3, 4, and 5 now throw exceptions but tests 1 and 6 do not.
5. Get Part 1 checked by a TA.

**Part 2 - Checked Exceptions**

The `Exception` (super-)class is a checked exception. This means that if we tell the compiler that our method is going to throw an `Exception`, any calls to that method require try/catch blocks for `Exception`.

1. At the end of the signature for `addMonies` add "throws Exception".
2. Compile your project. What happened?
3. Add a try /catch block around all the tests in class `Test`'s constructor. In the catch block just output the exception message.

    Syntax reminder:

```
try {
     // all the comments and code for the six tests INDENTED!!
} catch (Exception e) {
     System.out.println(e);
}
```

4. Test your changes by running the tests (i.e. creating `Test` objects). Comment out the tests that cause exceptions one after another (as in Part 0) so you can check that tests 2, 3, 4, and 5 still throw exceptions but tests 1 and 6 do not.
5. Generate your Java documentation and look closely at the `addMonies` method. What has been added?
6. Get Part 2 checked by a TA.

**Part 3 - Improving the Code**

With one try/catch block we can deal with only one exception. Here we have multiple ones, and it would be nice to see all the exception messages without having to comment things in/out.

1. Add a try/catch block around **each** of the six tests.
2. Test your changes.
3. Get Part 3 checked by a TA.

---