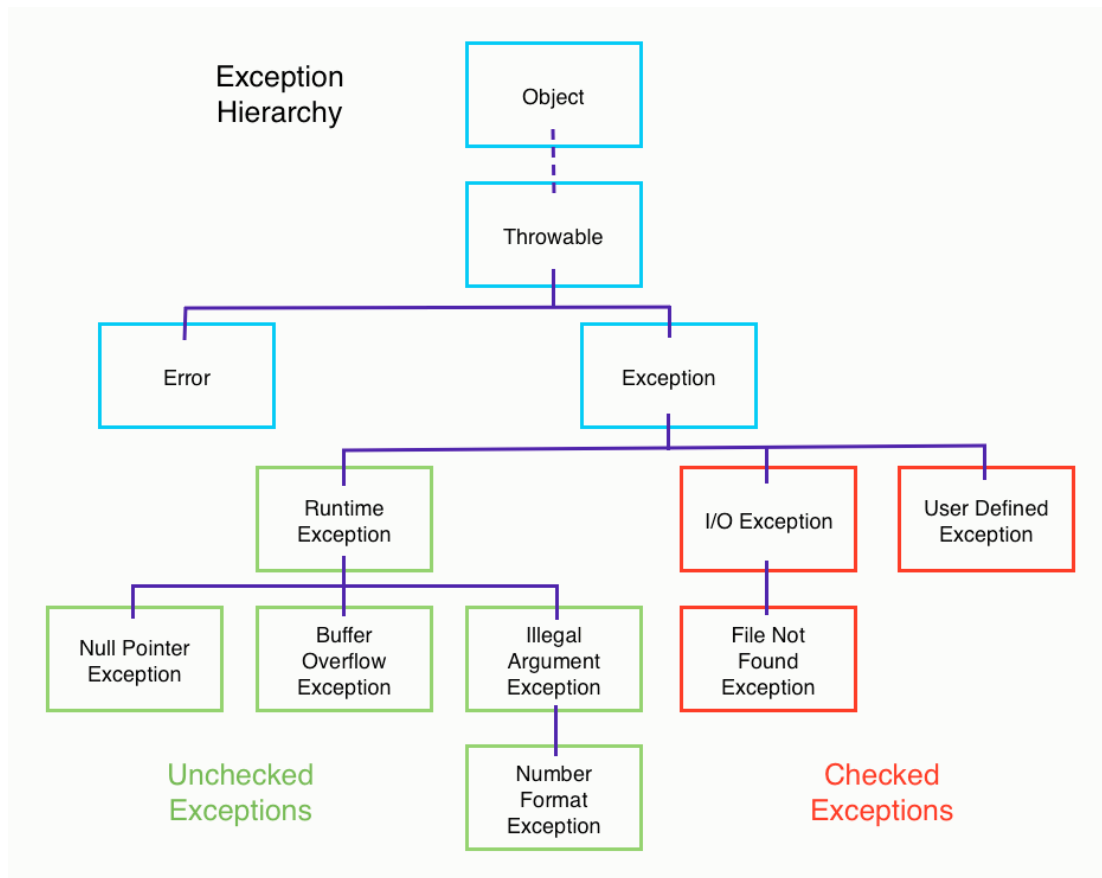


CS61B
Pre-Lab 7: Exceptions

Consider what happens when an error occurs, say, an out-of-bounds array access is made. (Commonly known as the infamous `IndexOutOfBoundsException` that I'm sure many of you have encountered). The program may be allowed to crash. Java also, however, allows a programmer to intercept the crash and take alternate action. The mechanism that enables this ability is called an `_exception_`.

Java includes an `Exception` class that may be useful to you. Feel free to view the `Exception` class here <http://docs.oracle.com/javase/7/docs/api/java/lang/Exception.html>. Similar to many other classes, the `Exception` class is part of its own hierarchy.



Throughout your work in 61B thus far, you have not had to worry about what your program would do if provided with incorrect or invalid input. An example would be inputting a negative number into a function that only cares or uses nonnegative values. Exceptions are often used to tell the program what to do when it is provided with these incorrect inputs.

Try briefly reading about try-catch blocks, the different types of exceptions, how to use `throw/throws`, and checked/unchecked exceptions for lab 7.

Here are a few questions you may want to ask yourself while learning more about Exceptions:

1. What is the difference between a Checked and Unchecked Exception in Java?
2. Is it necessary that each try block be followed by a catch block?

```

3. public class SuperClass {
    public void start() throws IOException{
        throw new IOException("Not able to open file");
    }
}

public class SubClass extends SuperClass{
    public void start() throws Exception{
        throw new Exception("Not able to start");
    }
}

```

In this code compiler will complain on sub class where start() method gets overridden. As per rules of method overriding in Java, an overridden method can not throw Checked Exception which is higher in hierarchy than original method. Since here start() is throwing IOException in super class, start() in sub class can only throw either IOException or any sub class of IOException but not super class of IOException e.g. Exception.

To prepare for the lab do the following:

- Read Chapter 11 Page 320-323 to learn more about Exceptions
- Understand questions on page 324
- Read Chapter 11 pages 326-327 for more information on try/catch blocks
- Practice the the True/False questions on page 348

(All page numbers are from Head First Java, 3rd Edition)