

Exceptions

Purpose:

The purpose of this laboratory is to acquaint you with Java's exception mechanism.

The class *Whoopser*:

Create a directory name *Lab210.whoops* in your *Java* directory, and copy the files from *~labCourse/Labs/Lab210/whoops* to your *whoops* directory.

Take a few moments to look at the two files you have copied. Compile these two files.

- Run *Lab210.whoops.WhoopsTest*. The program will crash. What exception is thrown?
- The constructor for *Whoopser* tries to access the file */home/labCourse/Labs/Lab210/VaporFile.txt*. Examine the directory */home/labCourse/Labs/Lab210/*. What in particular caused this crash?

Edit the *Whoopser* constructor so that the variable *inFile* accesses */home/labCourse/Labs/Lab210/RealFile.txt*. Re-compile and rerun. Again you get a crash.

- What exception was thrown? What in particular caused this crash?

Edit the *Whoopser* constructor so *inFile* accesses file */home/labCourse/Labs/Lab210/NiceFile.txt*. Re-compile and rerun. You are prompted for an integer. Instead, enter your first name. Another crash results.

- What exception is thrown?

Rerun *WhoopsTest*. This time enter the integer 0 at the prompt. You are informed that an "arithmetic calamity" occurred.

- What caused this arithmetic calamity?

Next enter the integer 13.

- What kind of calamity are you told of this time? What in particular caused this calamity?

Next enter the integer 10.

- What kind of calamity are you told of this time? What in particular caused this calamity?

Next enter the integer 14. To stop the program's infinite loop of messages, press Control-C.

- What kind of calamity are you told of this time?

This time the program did not pause after its prompt for you to enter an integer, as with the earlier calamities. The infinite looping is due to the following. The pause comes when the program executes the statement

```
input.readInt();
```

which waits for a line of data to be entered. But once variable *input* gets assigned the value *null*, executing this statement throws an exception since *input* no longer references a *BasicFileReader* instance.

Edit file *Whoopser.java* and add a third statement

```
        throw npe;
```

to the *NullPointerException* handler. Re-compile and rerun. Again, when prompted, 14.

- What happens this time that is different? Why did this occur?

A robust input method:

Assume a variable *input* references a *BasicFileReader*, and a variable *output* references a *BasicFileWriter*. Write a method

```
    public int readInputValue ()
```

which prompts the user for an integer in the range 1 to 100, successfully obtains the integer, and returns it. The method should be implemented with a *while* loop that continually prompts the user until a proper value is input.

If the user enters something other than an integer, the message “Please enter a properly formatted integer” should be displayed. If the user enters an integer that is not in range, the message “Please enter an integer in the range 1 to 100” should be displayed. In either case, the user should be again prompted for input.

Note that if the user enters a non-integer, then *input.readInt()* will not consume any characters. You should discard those characters with *input.readLine()*. Otherwise, the method will loop infinitely, balking at the same unacceptable characters in the input stream.

Post-lab:

Submit the following, as directed by your lab instructor:

- responses to the questions posed in the lab;
- a listing of your modified file *Whoopser.java*;
- a listing of tester with method *readInputValue*;
- a script testing *readInputValue*.