

CSC207.01 2014S, Class 11: Exceptional Programming

Overview

- Preliminaries.
 - Admin.
 - Upcoming work.
 - A few notes on recent assignments.
 - Questions.
- Lab.

Preliminaries

Admin

- Thanks to Mr. Walker, we now have a copy of Weiss in the learning center.
- I've posted a new style for CSC207 to `/home/rebel sky/share/CSC207.xml`
 - It probably needs some more work, but I wanted to make some forward progress.
 - You should be able to copy the file, but I'll try to remember to post it on a Web page.
- Extra credit:
 - Wednesday Extra, February 5 at 4:30 in 3821, AppDev
 - Convo Feb. 5 at 4:15 pm in JRC 101: Pussy Riot and Putin.
 - CS Table Friday: NP Completeness
 - More?
- Review session tomorrow at 10 a.m.

Upcoming Work

- Readings for Monday: References and Autoboxing/Autounboxing.
 - Not yet written.
 - Probably available Saturday.
- Writeup due MONDAY. We will continue today's lab on Friday.
 - Subject: CSC 207 Writeup 8: Exceptions (YOUR NAME)
- HW 3 due tonight.
- HW 4 posted in draft form.

A Few Notes from Recent Labs

The `import` statement.

```
import java.io.PrintWriter
```

Introduces a shorthand. You can now refer to `PrintWriter` and the Java compiler knows that you mean `java.io.PrintWriter`. (It's not clear that `import` does much more of use.)

```
import java.io.*
```

A shorthand for importing everything from a package. Sam discourages the use, because it makes it harder to figure out what classes come from where.

Copying Two Lines

- Many of you closed your `PrintWriter` and `BufferedReader` before the end of the program. That's generally a bad idea - you might want to do some more output or input later.
- Many of you had a variable for each line that you read. That seems inefficient.

```
line1 = fileEyes.readLine(); line2 = fileEyes.readLine(); filePen.println(line1); filePen.println(line2);
```

- Note that copying lines should really be a *loop*.

```
private static final int NUMLINES = 2; ... for (int i = 0; i < NUMLINES; i++) {  
    filePen.println(fileEyes.readLine()); } // for
```

Questions on HW3

Can you explain the last test?

```
assertArrayEquals (new String[] { "a", "b", "c" },  
                   StringUtils.splitCSV("a,b,c"));  
assertArrayEquals (new String[] { "a,b", "c" },  
                   StringUtils.splitCSV("\a,b\",c"));  
assertArrayEquals (new String[] { "a", "b,b\"", "c" },  
                   StringUtils.splitCSV("a,\"b,b\"\",c"));
```

Sure. We start with the "conceptual units": We want to write a CSV line that represents the three values `[a]`, `[b,b]`, and `[c]`. (The square brackets are there to clarify what the units are, they are not part of the units.)

As we try to represent this in CSV, we realize that the middle entity has a comma, so we need to put quotation marks around it. `"b,b"`. But there's an internal quotation mark, so we need to double it. `"b,b\""`.

Our CSV line is now `a, "b,b" ", c`.

We want to represent that as a C/Java string. In C and Java, strings start and end with double-quotation marks and any internal double-quotation marks must be preceded by backslashes. Hence, we get `"a, \"b,b\\\" \"\", c"`.

Wasn't that fun? Learning to read C strings takes a bit of effort.

How much detail should we put in end braces?

Enough that you can tell what you are ending. For `if` statements, I tend to prefer that you indicate something about the test, so that you can tell which brace ends which part of a nested `if`.

```
} // if (x != 0)
} // if x is valid
```

Similarly, for a `for` loop, it's helpful if you indicate which loop variable you are using

```
} // for (j)
} // for (i)
```

Should we do unit tests for "The Name Game"?

Probably not. An experiment suffices.

Lab

Start the lab on Exceptions. We'll continue it on Friday.

Copyright (c) 2013-14 Samuel A. Rebelsky.



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.