

CSC207.01 2013F, Class 09: Input and Output

Overview

- Preliminaries
 - Admin.
 - A few quick notes on HW 2
 - Questions on HW 3
- Leftover types topics.
- A few notes on textual output and input.
- Output.
- Input.
- Lab.

Admin

- The strange things that CS students discuss: "If two people are dating, are they boyfriend and girlfriend (or boyfriend and boyfriend) (or girlfriend and girlfriend) or (gender-neutral-term-friend and ...)?"
- Reading for Monday (it's ready!)
 - Exceptions
- I've prepared a lab for today, but I don't know if there will be time.
- I hear that some of you are losing files. You should make it a practice to save early and often. You should also make it a practice to commit regularly. (Say, after each working procedure you write.)
- I'm in the midst of grading HW2. I should have it done by Monday. We'll discuss the averaging problem.
- Only about half the class has filled out the prologue (as of 8:30 p.m. on Thursday)
- EC opportunities:
 - CS Table, Friday: Trusting Trust.
 - CS Table, next Friday: Pair Programming.
 - More?

HW 2

- For the averaging problem, there are a variety of strategies
 - Promote the type of the parameters.
 - Many of you used `double`. We'll talk about why that's a bad idea.
 - Floating points approximate. Are you sure your code works correctly for every situation?
 - doubles are 64 IEEE floating-point numbers, they have about 52 bits, so we are okay
 - We'll also talk about why promotion may not be the best strategy. what if we have

```
public static long average(long left, long right) { return (left + right) / 2; } // average
```

will this work?

```
public static long average(long left, long right) {  
    return (long) (((double) left + (double) right) / 2);  
} // average
```

how about this?

```
public static long average(long left, long right) {  
    return (left/2) + (right/2);  
} // average
```

nope, won't work if both are odd ... so

```
public static long average(long left, long right) {  
    // Compute the average w/o overflow  
    long tmp = (left/2) + (right/2);  
    // Handle introduced inaccuracies  
    if (isOdd(left) && isOdd(right))  
        return tmp+1;  
    else  
        return tmp;  
} // average
```

coming up: generalized average: taking an array of longs as input

- Java carefully specifies all of its numeric types longs are 64 bits, twos complement
 - A few of you changed the preconditions. I was pretty clear that I wanted you to *rewrite* the procedure, but I'll accept that as an approach (particularly since the testing lab suggests that approach for a similar problem).
 - Anything else?

HW 3

- Yes, exhaustive search is an acceptable initial technique for the last problem. (And no, it won't be efficient.)
 - You might also try a strategy in which you say "Can I do this with one coin, with two coins, with three coins, ...?"
- No, you don't need to set up a lot of test cases for the last problem.
- What should we do on `splitCSV` if the input is crappy (e.g., only one double quotation mark) - You can crash and burn.
- For the de-leet-ful problem, are all 3's e's? (Not quite, some are parts of B's, but there should be no 3's in your output.)

Leftover types topics

- Two dimensions:
 - Real vs. Int
 - primitive type, object corresponding to primitive type, or Big
- Primitive types: Mostly like C, except specified
 - ints are twos-complement
- But the primitive types aren't objects
 - And so we have `java.lang.Integer` and `java.lang.Double`
- And we want arbitrary precisions
 - `java.math.BigInteger`

A few notes on textual output and input

- Textual I/O is useful for files
- We use an OO approach to I/O

Output

- `PrintWriter`
 - `print`
 - `println`
 - `format` (like `printf`)
- `PrintWriters` buffer their output
 - need `flush()` to force output
- Can create with `System.out`
- Can create with files - `new PrintWriter(new File('ooh.csv'));`

Input

- `BufferedReader`
 - `read` - read a character
 - `readLine` - read a line
- Can create with `System.in`
- Can create with files `new BufferedReader(new FileReader(new File('ooh.csv')));`

Lab

```
public static void main(String[] args) throws Exception {  
    // ...  
}
```

Copyright (c) 2013 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.