Lab 7
Divisors

20 October 2015

The drill for this lab is the **Chapter 5 drill**. Come to lab on Tuesday either with it completed or with a specific written question in your notebook identifying which drill step you got to and what about it you're stuck on.

Given the format of the drill, if you do get stuck on how to fix one of them, move on to the next one! Since the 25 elements of the drill can all be typed in independently of each other, there's nothing preventing you from typing them *all* in, even if some of them aren't fixed yet.

Notes and adjustments:

- When you type in the scaffolding, omit the lines that call `keep_window_open();`.
- For each of these, you'll be saying, in your readme, what was wrong, *not just* how you fixed it. For instance (freebie!) on the first one, it's not enough to just say "replace Cout with cout", or "change C to c"; you should also say, "C in Cout is uppercase, so it's not seen as the same thing as cout". Or something like that.
- In numbers 13, 14, 15, and 19, insert the keyword `unsigned` before the keyword `int`. The absence of the `unsigned` is not the problem, if any, with those lines.
- The programs should compile, not just without error, but without any warnings either (once you fixed the `unsigned` issue I mentioned above).
- But to fix a program, it is not sufficient to just get it to compile and run without throwing an exception; it needs to print, exactly, the string "`Success!`" followed by a newline. (If it isn't followed by a newline, your prompt will get printed on the same line as the "`Success!`", which looks funny and is incorrect.)
- Below is a list of "par" values for each of the 25 problems. As in golf, you want to try to find a fix with the lowest number of strokes—in this case keystrokes. The number represents, roughly, the number of

characters you can add, delete, or modify in order to fix the line. In a few cases, you might do it with an even smaller number than I've given. If you go over by one or two, that's probably fine. If you go over by a lot, give it a closer look—you might not really be understanding the problem(s) with the line. (But, if you don't come up with anything better, still put that in as your answer! There's partial credit to be had here.)

Prob	Par	Prob	Par	Prob	Par	Prob	Par
1	1	8	5	14	2	20	1
2	1	9	2	15	1	21	2
3	2	10	1	16	4	22	6
4	2	11	2	17	4	23	2
5	5	12	2	18	2	24	2
6	5	13	1	19	5	25	3
7	4						

- Don't forget that some lines may have multiple errors, and some lines may have none. (There's at least one that has none, but I won't tell you which one(s) or how many. I've given it/them nonzero par value(s) in the table above just to keep things fun.)

- When you type them in, it's ok to spread the given code onto multiple lines rather than typing them as one-liners as in the book. (This is especially helpful for the later ones that have several statements and control structures.)