

## CSC 101 Lab Week 4

### Using a Debugger to Investigate Program Behavior

**ISSUED:** Friday, 20 April 2012  
**DUE:** Wednesday, 25 April 2012, by the end of lab  
**POINTS POSSIBLE:** 1  
**WEIGHT:** 1% of total class grade

#### Overview

There is no programming for you in this lab. Your job is to figure out what's going on in the following program:

```
~gfisher/classes/101/examples/simple-stuff/precision.c
```

and add comments that describe the program's behavior. When you're done, you'll submit your commented version of `precision.c`.

#### Choosing a Debugger

For this lab, you can use the C debugger of your choice. The two most likely candidates are `gdb` and `jGrasp`.

There are two handouts that describe the `gdb` command-line debugger:

- `gdb basics` [ `.html` , `.pdf` ]
- `gdb commands` [ `.html` , `.pdf` ]

If you want to use `gdb`, read through these handouts and give it a try. If you want to use another debugger, such as the one in `jGrasp` or other lab IDE, you may do so. Read the appropriate documentation, and ask your instructor for help during lab.

#### Using a Debugger to 'Figure things out'

Two of the most fundamental commands in a debugger are *setting breakpoints* and *single stepping* in a program. For this lab, your specific assignment is as follows:

1. make a copy of the program `~gfisher/classes/101/examples/simple-stuff/precision.c` into a working directory of your choice
2. compile the program
3. start the compiled program in a debugger
4. set a breakpoint at line 15
5. look at the statement on line 15 and predict what you think the value of `i` will be when the line gets executed
6. single step over line 15
7. print the value of the variable `i` to see what it is (in a graphical debugger like `jGrasp`, you can use the variable inspector instead of direct printing)
8. for lines 16 through 22, proceed as you did above for line 15; that is, before each line is executed predict what the value of the variable will be, then step through the line and print the variable to see its actual value.

You can repeat this debugging procedure as many times as you need to figure things out. If you want, make another copy of the program that you can modify experimentally as necessary.

Once you think you've figured out the behavior of the program, add a comment above each of lines 15 through 22 that explains what each line is doing. As a concrete example and hint, here's my comment for line 15:

```
/* Assigns the value 0 to i, the remainder of 0.5 is lost */  
i = 1/2;
```

You can use the comment as is. Then for lines 16-22, add similar comments. Your comments may need to be more than one line long to fully explain things.

**Submitting Your Work**

Sometime before end of lab on Monday, demonstrate in person to your instructor that you know how to set a breakpoint and print a variable when stopped at a breakpoint. During your demonstration, your instructor will ask some additional questions about the use of a debugger.

To verify that you've completed the lab, submit your commented version of `precision.c` on `unix1`:

```
handin gfisher 101_lab4 precision.c
```

Be sure to submit a version that has the same code as the original, with only your comments added. If you modify the code to play around with it, do the code modifications in another copy of the program.