CSC 150

Prelab #8: Debugging - Continued

Reread PreLab 5 and the relevant text material.

With the addition of functions to your repertoire, debugging requires more attention to what values are passed to a function and what values are returned, if any. You may find it helpful to add output statements to a function displaying the input values and what value it is returning, then in the calling function check that those match what you think it should be getting and giving.

Loops, especially for loops, are another area where you will often find logic errors. The most common, and often the hardest to catch, is the Off By One (OBO) error. This is where one of several conditions exist:

- starting value is incorrect
- ending value incorrect
- use of <= when < test needed (or vice versa)

Note that most example code provided uses for loops in the form

```
for( i = 0; i < limit; i++ )
```

where limit is the number of times you want the loop to execute. So i will run from 0 to limit-1, which is limit times. This model will be important when dealing with arrays.

For interactive debugging, use of the F11 key is handy. This is the Step Into action, which will take you from the current function and enter the code of the function being called. If you use F10 at the line where a function is, the function will just execute and then you go directly to the next line.

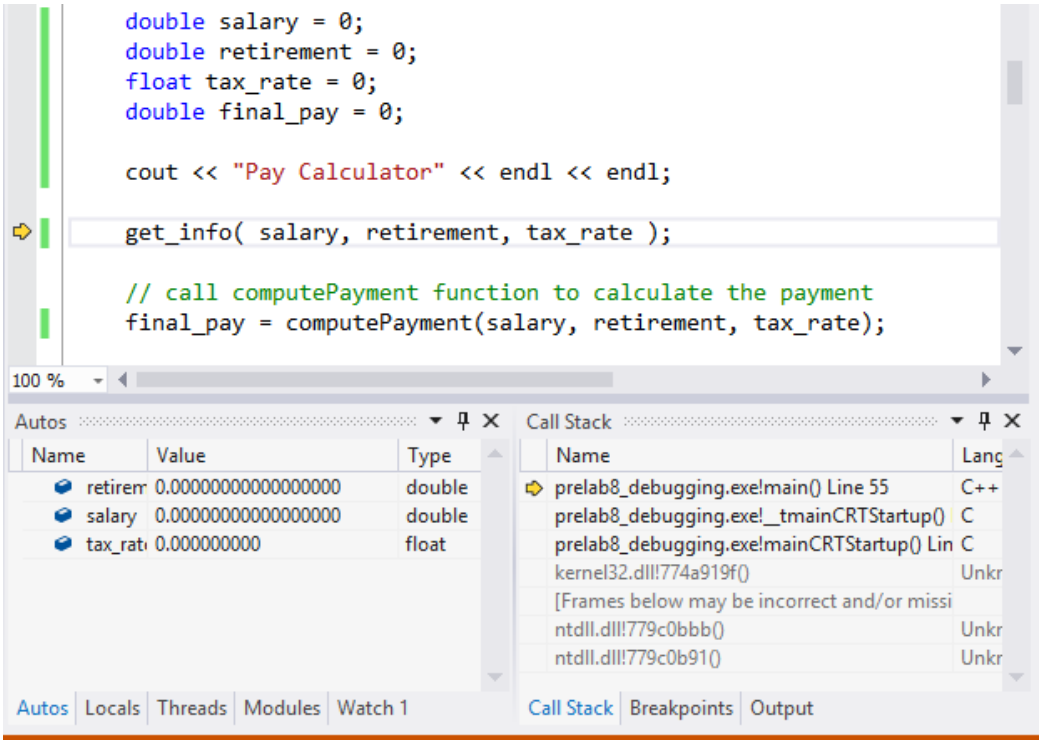
Another handy keyboard shortcut is control-F10, Run To Cursor. This one is not shown on the Debug menu or toolbar. Place your cursor at the point you want to stop, then begin debugging. This is similar to using breakpoints, but is completely interactive.

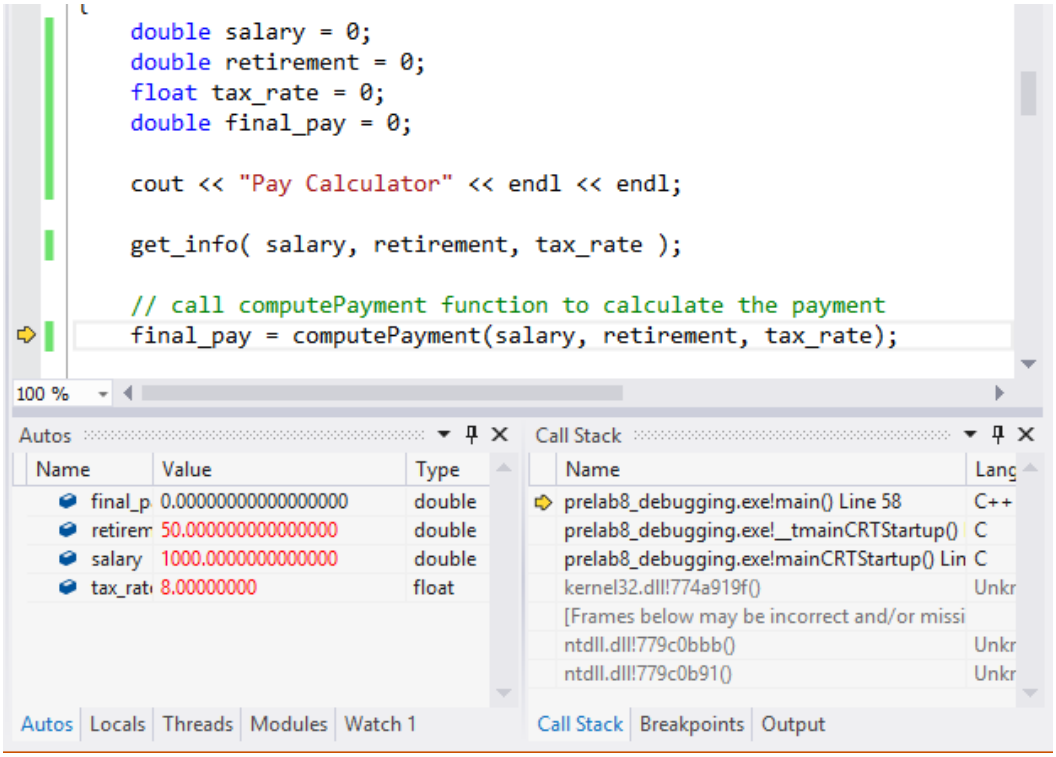
Pay attention to the information in the output area at the bottom of VS. The Autos tab will display the values of variables currently in scope. As you step through the code, you will see them change. You can also hover your mouse over a variable and its value will be displayed.

In the following images, note how the variables are all 0.00 to begin with. After function get_info() executes, note the changes in red indicating variables recently modified. When you press F10 or another debugging action, those in red will change back to black.

Autos will also display the value returned by a function. The Locals tab only displays the variables. Both these windows can change frequently as you move between functions. If there is a particular variable you want to keep a focus on, use the Watch tab. You can type the name of a variable in the tab's area, or right-click a variable and select Add Watch. To remove a variable from the Watch, just click on it and press the delete key.

Experiment with these techniques using the sample code on the class website.





There is nothing to turn in for this prelab.