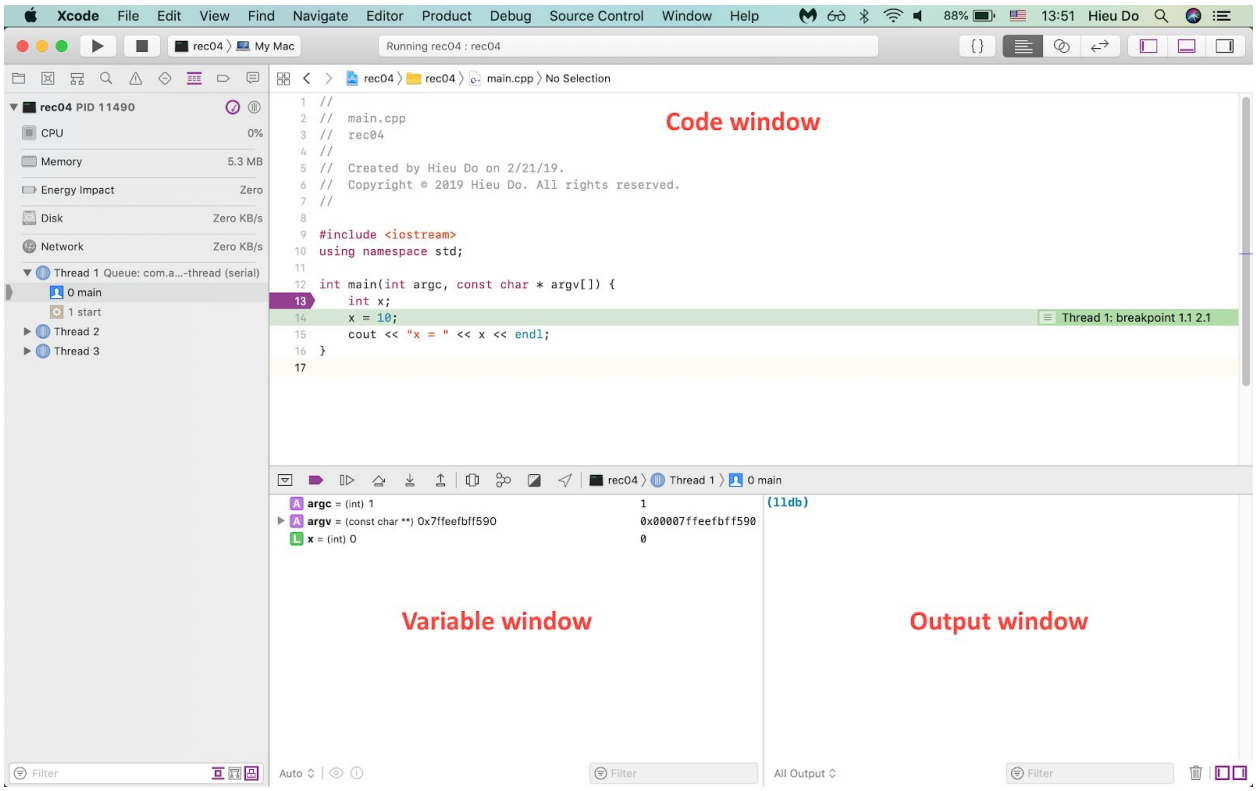
Rec04: Intro to Pointers - Additional Info for Xcode

This document is useful for Task 2 - 5 (introducing the IDE debugging environment). For the remaining tasks, whenever the instruction refers to Local or Watch window, think of Variable or Output window in Xcode; F10 means Step Over, and F11 means Step Into.

Set a breakpoint at the beginning of main by clicking on the line number. Once you have set the breakpoint and click Run, Xcode will automatically enter the debugging environment that looks like in the screenshot. The 3 windows (labelled below) we care about are:

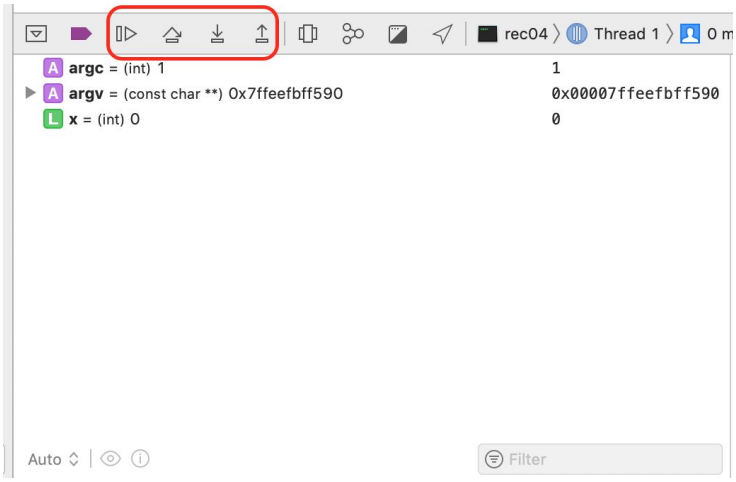
- Code window: show your C++ source code
- Variable window: show all the variables defined in your program
- Output window: show the output of your program and the debugger (lldb)

When you're in debugging environment, one entire line in your source code is highlighted (mine is green, but you might get a different color). After we click Run, the line should be the assignment statement `x = 10;`. That is the *next* statement to be executed. If you look at the Variable window, there should be one variable displayed in the variables window: `x`, which should have a "garbage" value. Garbage is a technical computer science term for a value that was *not* placed into a variable under the programmer's control and must never be used under any circumstances. The value displayed here is 0, but this might be different on your machine since this value is just whatever bits happened to be turned on in that memory location when the program used it for `x`.



Your program is waiting for you. In debug mode you are stepping through the execution of the program using one of buttons in the Variable window (you can hover over each button to see its label).

- Continue program execution: run the program till the next breakpoint or the end
- Step over: go over each line in your code
- Step into: if you're debugging a function call, step into will enter that function definition
- Step out: opposite of step into



Execute the next statement by clicking on the *Step Over* button. The Variables window will change and display `x` with a value of 10. You see in the Code window that the next line (cout << …) is now highlighted.

In the Variables window, right click on the variable `x` to see more options (left screenshot). What we want to show you are the *Watch "x"* and the *Add Experession* option. When you select the Watch "x" option and the value of `x` gets modified, you will see print statements in the Output window to the right, showing the Before and After values of `x`. The *Add Experession* option let you type in your own expression (like x + 2) to see the value of that expression AT THIS MOMENT in the program's execution. Type in an expression you want, like x + 1, and see the expression evaluated to 11. You can type &x to see the memory address of `x` (right screenshot).