CS 132 Lecture Topic ?? Debugger

Lonnie Heinke

- Todays Agenda
  - Introduction to debugging using the Debugger

- Start up Visual Studio 2015
  - Create a new project called "Sandbox??" that you will use for your activities.
  - Keep this project so you can use them as code examples in the future or you could expand on them.

- ….

Why use a Debugger?

- Without information from your program, it is hard or impossible to find your bugs

- Low tech solution
- You can just **cout** some information. This works but…
  - Takes time to set up and then clean up afterwards
    (adding the cout code and then removing)
  - Sometimes you can be overwhelmed with all the output

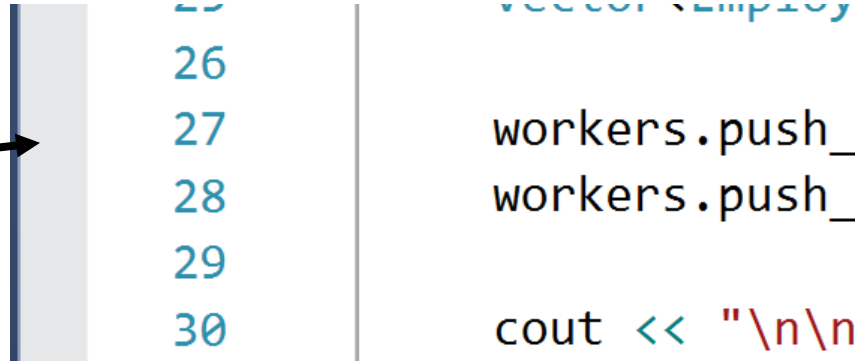
Why use a Debugger?

- Without information from your program, it is hard or impossible to find your bugs

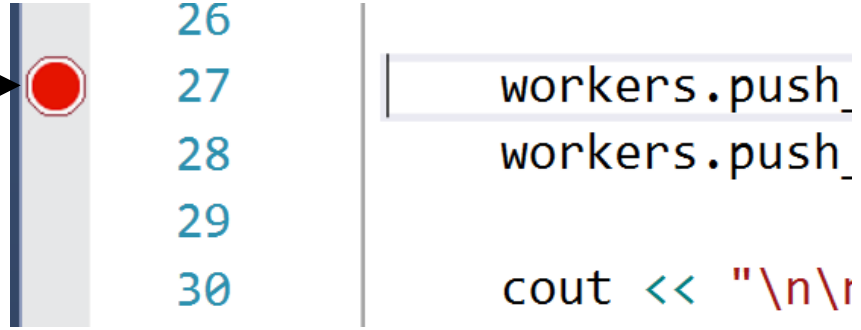
- High tech solution : Debugger
- Advantages:
  - Allows you to see and change data while the program is running
  - Allows you to execute the program line by line so you can see the behavior
    - See what is happening in your if statements, or loops (infinite)

Using the Debugger

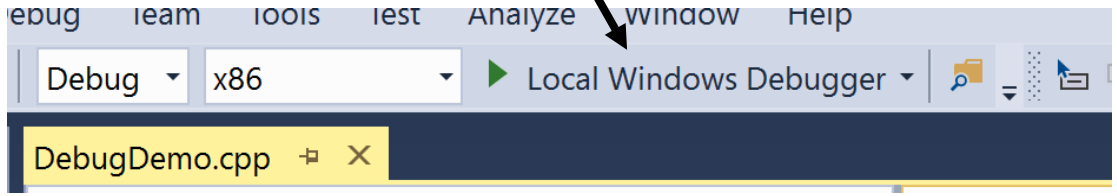
- Place a breakpoint on a line of executable code
  - click on the grey boarder of the code window



  - this should give you a breakpoint which will show as a red stop sign'ish circle

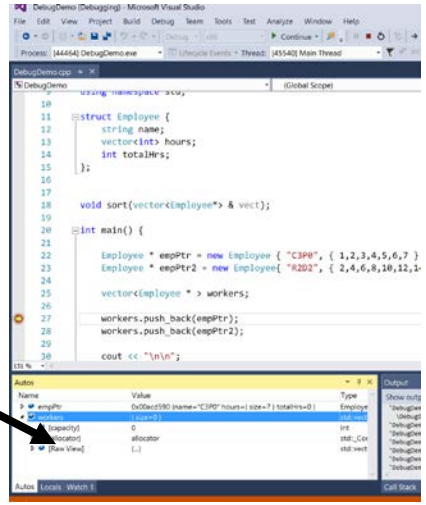


- Now you can start your program by clicking the Local Windows Debugger



Viewing your data

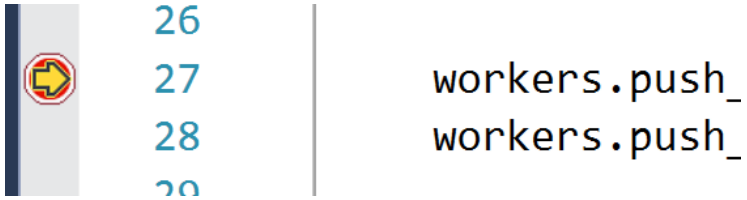
- Your variable that are in scope can be viewed in two ways
  - In the Variable window at the bottom of the program



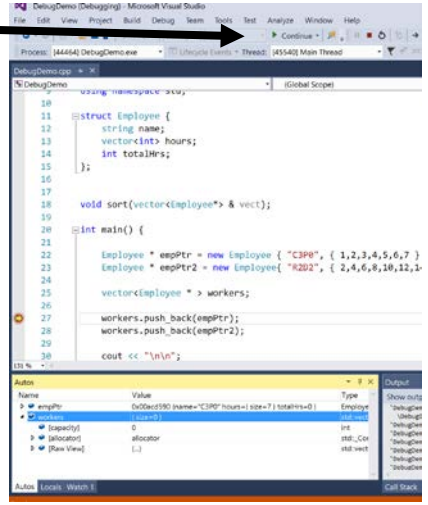
  - Or by hovering your mouse over a variable (very hard to get a screen capture of)

Walking through your code

- When you start the Debugger it will execute code automatically until it reaches a breakpoint where it will wait for you to tell it what to do next.



- Your options for going forward are:



Step Into: if this line of code contains a function call, it will take the execution into the function

Step Out: if the execution is inside a function other than main, it will execute all code in the function and then return from the function, where it will stop at the next line of code

Step Over: execute the current line of code (if it contains a function, then it will complete the function) and advance to the next line of code.

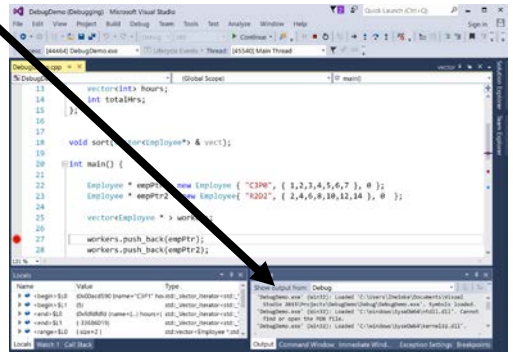
Continue: Execute code until the next breakpoint

Special Breakpoints

- Right clicking on a breakpoint allows you to designate the breakpoint as a special breakpoint

- Types of Special Breakpoints
  - **Conditional:** Allow you to give a boolean expression for when execution will stop at the breakpoint (for example: (x >= 5))

  - **Action:** Allows you to have information sent to the *output window*



Sharpening your Debugging skills

- Please start to use the Debugger to help understand what your program is doing.

- The Debugger is another tool that you will have to practice using to get better at using it effectively

Testable Information

- For quizzes and exams, I expect you to understand the concepts or usage of the following:

  - Step Into
  - Step Out
  - Step Over
  - Conditional Breakpoint
  - Action Breakpoint