Announcements
- MT average - 58.5
- HW due next Thursday

(Overview of midterm)
Theory of Unit testing

Unit testing is part of top-down design. Idea is to carefully specify requirements and behavior expected. Then run tests often - at completion or change of any included module, at least.

One way - run a script
Our way of unit testing:

Example: File xxx.py

class XXX:
    # code for class

if __name__ == "__main__":
    # test code

This test only runs when we type `python xxx.py` (not when module is imported)
How to use unittest (PyUnit)

Import it:
```
import unittest
```

Unit tests themselves consist of functions the programmer writes to test the code.

E.g.:
```
if __name__ == "__main__":
    unittest.main()
```
Example: a stack
A simple data structure w/ 3 functions.
  - Push(data)
  - Pop()
  - Peek() (If empty, None is returned)

Let's code this, and then test it.
Running PyUnit

coded some example unit tests for our stock.