Even more TDD

Announcements

- Quiz 3 and Lab 6 will be given back Friday

- Email your top-down design to mauskop@cs.swarthmore.edu

- The final exam is on May 6, a Saturday (ugh)

Today's plan

- Review file I/O

- Talk about Lab 7

  - `break` statements

  - `.index()` method for sequences

  - Easy-to-miss requirements

  - More benefits of top-down design

Today's plan

- Finish cs21-quiz.py

    - I'll post link to solution in lecture notes

- Write mastermind.py program, using top-down design

File I/O

- Give the `open` function a file name and a mode of interaction: 'r' for 'read', 'w' for 'write', 'a' for 'append'

- To read text from a file use file object methods `.readline(), .readlines()` or put the file object in a `for` loop (file objects are sequences)

- To write text into a file use `.write()` method

break statements

- You can exit out of a `for` or `while` loop with a `break` statement

- They are sometimes convenient, but never necessary (can be replaced by boolean flag)

```
while [PUZZLE NOT SOLVED]:

    ...
    if [USER QUITS]:
        break

    ...
# At this point (outside while loop)
# Either the puzzle was solved or the user quit
```

.index() method

- A method for sequences that takes any value as input and returns the index of the first occurrence of that value in the sequence

```
$ python
>>> letters = ['a', 'b', 'c']
>>> letters.index('c')
2
```

Easy-to-miss requirements

- Convert capital letters in the quote to lowercase before encoding

- The user needs to be able to change their guessed decodings

- The user needs to be able to quit by entering 'quit'

- Program should validate that single letters are being entered by user

More benefits of TDD

- You won't waste time implementing functions that you don't actually need

- It's good (in CS and in life) to start with a plan, understanding that the plan may need to change

Practice problems

- cs21-quiz.py link:

www.cs.swarthmore.edu/~mauskop/cs21/s17/practice/cs21-quiz.html

- mastermind.py