CISC106 Summer 2012 Lab06

- This lab an all subsequent labs will be due Sunday at 11:55 PM EDT on Sakai.
- The preparation problems below are to develop your understanding without creating extra work for you or the TA; these problems will not be graded. Be sure to read and understand them - they will help with the problems you must submit for grading
- Review the code examples from your notes in class.
- You may work in pairs on your lab. If you do, **one** of you should be designated to submit the assignment on Sakai. **Both of your names** should appear on code that you develop together[1].
- Whom do you think deducts more points: a happy TA, or a frustrated TA? Make your work easy to read! It isn't just good software engineering, it is good for your grade!
- EVERY python program/function must include header, doc string that contains a human-readable desciption of what the function does, and must be followed by a good series of tests, as discussed in class. Always test boundaries. Do not test erroneous input (e.g. a factorial function does not need to correctly handle strings).
- EVERY .py file must have a comment line at the very top containing your name(s), lab section, and a brief description of what the file is.
- Write the tests first! Real software engineers do this for very good reasons - so should you!
- Create lab06_tests.py (for tests) and lab06.py (for code) files. Use one of your old test files as a reference for how to start lab06_tests.py.

Preparation (do not submit for grading)

1. Run IDLE and at the prompt in the shell window enter the following commands:

```
>>> new_file = open('new_file.txt', 'w')
>>> new_file.write('this\n')
>>> new_file.write('that\n')
>>> new_file.write('the other thing\n')
>>> new_file.close()
```

2. Now enter the following:

```
>>> new_file = open('new_file.txt', 'r')
>>> for line in new_file:
         print line
>>> new_file.close()
```

[1] If you would like to work with someone but don't know whom, your TA may be able to help connect you to other students looking for lab partners.

Note what this outputs.

3. Now import the random package:

```
>>> import random
```

Run this line a few different times and see what happens:

```
>>> random.choice(['one', 'two', 'three', 'four', 'five'])
```

Problems (to be graded)

1. Write a function `insert_in_order` which takes a list of strings *items* and a string *item*. You will assume *items* is already sorted in alphabetical order and must insert *item* into the correct position in *items*.

2. Write a function `remove` which takes a list *items* and a string *item*. Your function should remove the first occurence of *item* in *items*. If *item* does not occur at all in *items*, the function should simply leave *items* unchanged.

3. Write a function `read_list` which takes a string *fname*. Your function should open the file at *fname* and build a list such that each line in the file is an element of the list.[2] The list should be in alphabetical order.

4. Write a function `write_list` which takes a string *fname* and a list *items*. This function should open the file at *fname* for writing and write each element *items* to the file so that each element is on its own line. **Hint:** If you have an open file *fin*, you can get the entire contents of the file as a string with `fin.read()`. Use this fact when you write your test for this function.

5. Now download the file *goldlib.py* from the course website. Run it and get familiar with all of its different options. You will need to modify goldlib.py for each of the following options so that they work as prescribed: Read List, Write List to file, Add verb to List and Remove Verb from List. Use your functions from 1 thru 4 where appropriate.

You should submit your lab06.py, lab06_tests.py, goldlib.py and any other docs required by your TA on Sakai.

[2] When you iterate over the lines in a file, each line will include the `'\n'` at the end of it. Make sure you get rid of it before appending the line to the list.