

CISC106 Spring 2013 Lab06

- This lab and all subsequent labs will be due Thursday 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 together¹.
- 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 description 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!

Preparation (do not submit for grading)

1. Run IDLE and at the prompt in the shell window import the random package:

```
>>> import random
```

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

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

2. Now enter the following lines:

```
>>> str1 = 'apple'
>>> str2 = 'Zebra'
>>> str1 < str2
```

¹If 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.

The result of that boolean expression on the last line should surprise you! The string 'apple' should come before the string 'Zebra' if we're comparing them by alphabetical order. This has to do with the fact that lowercase letters are considered 'less than' capital letters.² Try the following boolean expression instead:

```
>>> str1.lower() < str2.lower()
```

As you may guess, this converts both string to lower case and then compares them. Now the string that comes first in alphabetical order will be less.

Problems (to be graded)

1. Download the files *lab06.py*, *lab06_tests.py*, and *lab06ReadTest.txt* from the course website. Open *lab06_tests.py* and run the tests. You should see that they're failing.³
2. 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*.
3. Write a function `remove` which takes a list *items* and a string *item*. Your function should remove the first occurrence of *item* in *items*. If *item* does not occur at all in *items*, the function should simply leave *items* unchanged. **NB** when you finish this function and the one from 2, the tests for the code I gave you in *lab06.py* should be passing.
4. 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 both Add verb to List and Remove Verb so that they work as prescribed. Use your functions from 2 and 3 where appropriate.

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

²This seems like nonsense right now, but will become apparent when we discuss Unicode toward the end of the semester

³As an aside, if you take a look at the code being tested in *lab06.py*, you'll see it involves using files. Take a look at it and see if it makes sense to you. If not, don't worry - we'll cover file I/O in a few weeks.