

CISC106 Summer 2012 Lab04

- This lab and 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
- 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)

Program (to be graded)

1. Download `zelda_demo.py` and `Frames.py` from the course website. Run `zelda_demo.py` and you should get an error about module `lab04` not existing or some such. We'll come back to this later. For now, make `lab04.py` and `lab04_test.py` files in the same folder where you saved `zelda_demo.py` and `Frames.py`. You'll probably want to use the `lab03_tests.py` in the labs folder on the course site as a template for your `lab04_tests.py`.
2. Implement a function called `get_direction` which, on a particular character, gives the direction corresponding to that character. The correspondences are as follows:
 - The character `'w'` corresponds to the direction `'North'`
 - The character `'a'` corresponds to the direction `'West'`
 - The character `'s'` corresponds to the direction `'South'`
 - The character `'d'` corresponds to the direction `'East'`

On any other input, the behavior of the function is *undefined*² As always *write the tests before you write the code!* There are four cases you need to test for.

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.

²In programming, when behavior is documented as undefined, this means that the behavior could technically be anything. For example, your `get_direction` could deliver a box filled with angry cats to the caller if the caller passed in a `'u'` and this would be correct behavior.

3. Now try running `zelda_demo.py` again. You should get a window with Link from *The Legend of Zelda* somewhere near the middle. Try moving him around on the screen:

- `w` should cause him to move *up*
- `a` should cause him to move *to the left*
- `s` should cause him to move *down*
- `d` should cause him to move *to the right*

If he reaches the edge of the window, he'll wrap around to the other side. **Now**, to prove that what you just did wasn't magic, change the name of `get_direction` to simply `direction`. Of course, you should update the tests to reflect this new name as well. You should also update `zelda_demo.py` so that it works with this change.³

You should submit `lab04.py`, `lab04_tests.py`, and `zelda_demo.py` along with any other docs required by your TA on Sakai.

³**Hint:** Look for the occurrences of `get_direction` in `zelda_demo.py` and replace them with `direction`. Find and replace them individually - if you just do a "replace all" you're ripping yourself off!