

CS 1110, LAB 5: OBJECTS AND CONDITIONALS

<http://www.cs.cornell.edu/courses/cs1110/2015fa/labs/lab05.pdf>

First Name: _____ Last Name: _____ NetID: _____

You have now had an extensive assignment that made you an expert of string slicing. However, none of the functions in that assignment required conditionals. This lab builds upon those skills, and gives you experience writing more complex functions involving conditionals.

This lab also have some questions regarding objects. You are going to get a lot more experience with them on Assignment 3. To prepare you for that assignment, we want you to write some functions for a new class that you have never seen before.

The coding exercises in this lab are prime exam questions. In fact, both the Pig Latin question and the Time object question come from Prelim 1 in a previous semester of CS 1110.

Lab Materials. We have created several Python files for this lab. You can download all of the from the Labs section of the course web page.

<http://www.cs.cornell.edu/courses/cs1110/2015fa/labs>

For today's lab you will notice two files.

- `lab05.py` (a module functions for you to implement)
- `time.py` (a module with the `Time` class)

Once again you should create a *new* directory on your hard drive and download all of the files into that directory. Alternatively, you can get all of the files bundled in a single ZIP file called `lab04.zip` from the Labs section of the course web page.

This lab will require that you modify this file, as well as answer some questions on this worksheet. You need to show both to your instructor to get credit for the lab.

Getting Credit for the Lab. When you are done, show both this handout, and `lab05.py` to your instructor. Your instructor will then swipe your ID card to record your success. You do not need to submit the paper with your answers, and you do not need to submit the computer file.

As with all previous labs, if you do not finish during the lab, you have **until the beginning of lab next week to finish it**. You should always do your best to finish during lab hours. Remember that labs are graded on effort, not correctness.

1. PIG LATIN

Pig Latin is childish encoding of English that adheres to the following rules:

- (1) The vowels are ‘a’, ‘e’, ‘i’, ‘o’, ‘u’. A ‘y’ that is *not* the first letter of a word is also considered a vowel. All other letters are consonants. For example, “yearly” has three vowels (‘e’, ‘a’, and the last ‘y’) and three consonants (the first ‘y’, ‘r’, and ‘l’).
- (2) If the English word begins with a vowel, append “hay” to the end of the word to get the Pig Latin equivalent. For example, “ask” becomes “askhay,” “use” becomes “usehay.”
- (3) If the English word starts with ‘q’, assume it is followed by ‘u’; move “qu” to the end of the word, and append “ay”. Hence “quiet” becomes “ietquay,” “quay” becomes “ayquay.”
- (4) If the English word begins with a consonant, all the consonant letters up to the first vowel (if any) are moved to the end of the word, and “ay” is appended to get the Pig Latin equivalent. For example, “tomato” becomes “omatotay,” “school” becomes “oolschay,” “you” becomes “ouyay,” “my” becomes “ymay,” and “ssssh” becomes “sssshay.”

Your goal is to write a function `pigify` that takes a single English word (e.g. a string with only letters and no spaces), and converts it into Pig Latin.

1.1. **The Function `first_vowel()`.** To aid with our Pig Latin conversion, we have provided a helper function `first_vowel(w)`, with the following specification:

```
def first_vowel(w):  
    """Returns: position of the first vowel; -1 if no vowels.  
    Parameter w: the word to check  
    Precondition: w is a nonempty string with only lowercase letters"""
```

We hope that this helper function is correct. To verify this, write down at least 5 key test cases (each should be testing a different “idea”, so do not just say ‘a’, ‘e’, ‘i’, ‘o’, and ‘u’). **We do not want you to write a test script or try to fix any bugs.** Just write down the test cases.

1.2. **The Function `pigify`.** The function `pigify()` has a short-and-simple specification:

```
def pigify(w):  
    """Returns: copy of w converted to Pig Latin.  
    Parameter w: the word to convert to Pig Latin  
    Precondition: w is a nonempty string with only lowercase letters"""
```

This specification assumes that you have read the definition of Pig Latin on the previous page. **Implement this function** in `lab05.py`

When you are done, you will want to test your answer. Instead of creating a unit test, we only want you to write down a list of test cases to verify that your implementation is correct.

2. THE TIME CLASS

The objects of class `Time` have exactly two attributes, `minutes` and `hours`. Both of these attributes are integers, though there is an important restriction on `minutes`: it must be between 0 and 59. If you go over 59 minutes, you are supposed to increase the `hours` attribute instead.

Open up the Python interactive shell and type in the following two statements:

```
>>> import time
>>> t = time.Time(2,30)
```

Using the interactive shell, you will fill out the table below, just as you did in labs 1 and 2. The first column contains *either* an expression or a command. If it is an expression, write the value. If it is a command, you should just write “None”.

Statement or Expression	Expected Value	Calculated Value	Reason for Calculated Value
<code>t.minutes</code>			
<code>t.hours</code>			
<code>t.days</code>			
<code>s = t</code>			
<code>s.minutes</code>			
<code>t.minutes = 20</code>			
<code>s.minutes</code>			
<code>s.minutes = 60</code>			

At the end of the file `lab05.py`, you will see a stub for a function called `add_time`. This function has the following specification:

```
def add_time(time1, time2):  
    """Returns: The sum of time1 and time2 as a new Time object  
    Example: Sum of 1 hr 59 min and 1 hr 2 min is 3 hr 1 min  
    DO NOT ALTER time1 or time2, even though they are mutable  
    Parameter time1: The starting time  
    Precondition: time1 is a Time object"""  
    Parameter time2: The time to add  
    Precondition: time2 is a Time object"""
```

Because you are creating a new `Time` object, this function will need to call the constructor for `Time`. Implement this function inside `lab05.py`. This time we do not need you to list any test cases. However, it might be a good idea to test the function before showing it to your instructor. Once you have implemented this function, you are done with the lab.