

Technical Homework 1: OOP in Python

Python, like Java and many other languages, has object-oriented capabilities. To learn more about object oriented programming in Python, please look at the following slides:

<http://aiti.mit.edu/media/programs/sri-lanka-summer-2012/materials/t-l06.pdf>

and please skim chapters 12-14 of the following book, paying closer attention to chapter 14. You may already have experience with object-oriented programming, so much of this might be familiar. Focus your efforts on understanding the syntax of Python.

<http://www.greenteapress.com/thinkpython/thinkCSpy/thinkCSpy.pdf>

Once you feel comfortable with Classes and Objects in Python, please complete the following programming problems.

Address Book

1.

In the next two problems, you'll be building an address book using object-oriented programming in Python. To begin, you'll define a Person class. You'll use this class in the next problem to build your address book – each entry in your address book will be an instance of the Person class.

To begin, define the `__init__` and `__str__` methods for the Person class. Read the Python documentation if needed to help you understand what `__init__` and `__str__` need to do.

Here's an example interaction in the Python shell demonstrating how your class should work after implementing these two methods:

```
>>> Emily = Person("Zhang", "Emily", "5559358150",
"emzhang@mit.edu")
>>> print Emily
Zhang, Emily -- Phone Number: 5559358150 -- Email
Address(es): emzhang@mit.edu
>>> Wesley = Person("Graybill", "Wesley", "5551940325",
['wgray496@mit.edu', 'wdgraybill@gmail.com'])
>>> print Wesley
```

Graybill, Wesley -- Phone Number: 5551940325 -- Email
Address(es): wgray496@mit.edu, wdgraybill@gmail.com

2.

In this problem, you will use your Person class from the previous problem to create an address book for storing the contacts of your friends and family. It should allow you to search the address book for a friend, and return their contact information.

Begin by defining an `__init__` method. Don't forget about Python's built-in dictionary data type.

Next, write a method `add_contact` that allows you to add a new person to the address book.

Finally, write a method `lookup_contact` that looks up a contact by last name. The method should accept the last name as an argument, and print each contact that matches the last name on a new line. As an additional challenge, extend this method to allow users to optionally specify a first name to narrow down the results when multiple contacts have the same last name. To learn more about optional arguments in Python, search for terms such as "optional arguments in Python" and "default arguments in Python".

For example, suppose the contact book contains entries for both Ali Kamil, and his wife Sama Kamil. The following is an example output:

```
>>> a = AddressBook()
>>> a.add_person(Person("Kamil", "Ali", ... ))
>>> a.add_person(Person("Kamil", "Sama", ... ))
>>> a.lookup_contact("Kamil")
Kamil, Ali -- Phone Number: ...
Kamil, Sama -- Phone Number: ...
>>> a.lookup_contact("Kamil", "Ali")
Kamil, Ali -- Phone Number: ...
```