Discussion 12: More Python

Planning Your Phase II

(a) In the table below, write out Python code to execute the following commands on `my_dict`.

```
my_dict = {'Math':'1A', 'English':'R1A'}
```

Add the key 'CS' with the value '10'	`my_dict['CS'] = '10'`
Access the value of 'Math'	`my_dict['Math']`
Change the value of 'Math' to '1B'	`my_dict['Math'] = '1B'`
Check if 'UGBA' is a key in my_dict	`'UGBA' in my_dict`
Check if '10' is a value in my_dict	`'10' in my_dict.values()`
Get a list of the keys in my_dict	`list(my_dict.keys())`

(b) Can you access a key, value pair by its index in a dictionary?

No, dictionary values can only be accessed by key

(c) Are keys or values in a dictionary returned in any particular order?

No, dictionaries are unordered

Iterating over Dictionaries

```
fav_numbers = {'Yifat': 20, 'Mansi': 7, 'Jobel': 120, 'Schuyler': 10,
'Jessica': 16}
```

(a) Increment each person's favorite number by the length of their name.

```
for name in fav_numbers:
    fav_numbers[name] += len(name)
```

```
nums = [10, 20]
```

(b) Use a list comprehension to return the names of individuals whose favorite numbers are in `nums`.

```
[name for name in fav_numbers if fav_numbers[name] in nums]
```

Find the Index

(a) The following function takes in an item and a list and returns the index of the item in the list, but it's buggy. Mark the fixes it needs.

```
1 def find_index(item, lst):
2     i = 1   Since python 0 indexes, we should set i to 0 initially.
3     while i <= len(lst):   This line should be i < len(lst). The last index in a list is the length of the list - 1, so i should never be greater than that.
4         if item = lst[i]:   if item == lst[i]:
5             return i
6         i += 1
```

(b) Now, write the same function recursively.

```
def find_index(item, lst):
    if item == lst[0]:
        return 0
    else:
        return 1 + find_index(item, lst[1:])
```

(c) Now, write the same function with a list comprehension (hint: it may help to first think of what the function would look like with a for loop, then condense it into a list comprehension)

```
[x for x in range(len(lst)) if lst[x] == item]
```

Lambdas/Higher Order Functions

(a) Write a lambda function called f that takes in a number and outputs that number squared.

```
f = lambda x: x ** 2
```

(b) Now, use a list comprehension and your lambda function f to return a list the squares of all numbers between 1-5.

```
[f(x) for x in range(1, 6)]
```

(c) What would the interpreter display for the following lines of code?

```
>>> S = “Berkeley”
>>> S[1:3]
 “er”
```

```
>>> [x * 2 for x in range(4) if x % 2 == 1]
 [2, 6]
```

```
>>> "".join([word[0] for word in "Univ of Calif at
Berkeley".split(“ “) if not(len(word) == 2)])
 “UCB”
```

```
>>> f1 = lambda x: x + x
>>> f2 = lambda x: x > 9
>>> [f(10) for f in [f1, f2]]
 [20, True]
```
