Name:_____ **Partner:** _____

Python Activity 20: Dictionaries

Learning Objectives
Students will be able to:
Content:
- Define a **dictionary**
- Identify the **key** and **value** pair of a dictionary

Process:
- Write code that accesses the keys and values of a dictionary
- Write code that iterates through the dictionary's keys, values, items.
- Write code to append an item to a list value in a dictionary

Prior Knowledge
- Python concepts from Activities 1-19.

Folks, this is a brand new activity. If you encounter any issues/typos, please let Iris know

Critical Thinking Questions:

1. Examine the sample code defining a list of lists, below.

Sample Code

```
dog2owner =
[['pixel','iris'],['wally','steve'],['tally','duane']]
```

a. What's stored at `dog2owner[0][0]`?_____

b. What's stored at `dog2owner[0][1]`?_____

c. Write a line of code to print the name of Wally's owner using list indexing:

_____

d. Write a line of code to access and print the name of Duane's dog via list indexing:

_____

FYI: A ***dictionary*** is a data structure that is similar to a list, but instead of storing values at numerical indices, ***values*** are mapped to ***keys***, which must be an immutable data type.

FYI: Dictionaries are ***unordered***. Keys are not necessarily sorted in any particular order, and so you cannot rely on Dictionary keys to appear in the same order every time. This has implications for when you iterative over a dictionary. We talk about this in class.

2. The following code occurs in interactive Python:

```
>>> d = {'pixel':'iris','wally':'steve','tally':'duane'}
>>> d['wally']
'steve'
```

a. What does `d['wally']` do?

b. In the line, `d['wally']`, what does 'wally' represent?

c. Write a couple lines of code to print the name of your CS134 instructor and their dog's name, accessed via the dictionary, `d`:

3. Examine the following code from interactive Python:

```
>>> d = {'pixel':'iris','wally':'steve','tally':'duane'}
>>> d['linus'] = 'jeannie'
>>> d
{'pixel':'iris','wally','steve','tally':'duane', 'linus':'jeannie'}
```

a. What does the line `d['linus'] = 'jeannie'` do?

b. How does this indicate to us that dictionaries are mutable objects?

c. Write a line of code to add Bill and his dog, Annie, to our dictionary.

4. Examine the following example code:

```
>>> d = dict()      # can also do: d = {}
>>> d
```

a. If we wrote a third line of code, `len(d)`, what would be the output? _____

b. If instead our third line of code was `d['colleges'] = 'williams'`, what would `len(d)` return?

c. Write some code to create a new dictionary, then place `month, day, year` keys, mapped to today's date values, into the dictionary:

5. Examine the following example code:

```
>>> d = {}           # can also do: d = dict()
>>> d['colleges'] = 'williams'
>>> d['colleges'] = 'amherst'
```

a. If we wrote a fourth line of code, `print(d)`, what might be the output?

_____

b. At the end of this code execution, `d` only has: `{'colleges': 'amherst'}` Why might this be?

_____

FYI: Dictionaries can only have one key of its value, any replicated key:value mappings added will simply overwrite the existing ones!

c. Write a function that checks if `d` has a value mapped to `key`. If it doesn't, create a new list at `key` with the given `value` as its only element. If it does already have the `key`, append `value` to the existing list mapped to `key`.

```
def appendDictList(d, key, value):
```

_____

_____

_____

_____

_____

_____

_____

6. Examine the following example code:

```
0 >>> d = {'colleges':['williams'],'univ':['umass']}
1 >>> collist = d.get('colleges', [])
2 >>> collist
3 ['williams']
4 >>> collist.append('amherst')
5 >>> d
6 {'colleges':['williams','amherst'],'univ':['umass']}
```

a. What is the type of the value mapped to '`colleges`' at line 0? _____

b. At line 0, what is the value associated with '`colleges`'? _____

c. How does this value change, between line 0 and line 6?

_____

7. Examine the following example code which continues from the previous question:

```
7 >>> instlist = d.get('inst', [])
8 >>> instlist
9 []
10 >>> instlist.append('rpi')
11 >>> d['inst'] = instlist
12 >>> d
13 {'colleges':['williams','amherst'],'univ':['umass'],
'inst':['rpi']}
```

a. What is added to our dictionary on line 10? Examine lines 6 and 13 for differences.

_____

b. What does the first parameter passed to the `.get(..)` method on lines 1 & 7 represent?

_____

c. What does the second parameter passed to the `.get(..)` method on lines 1 & 7 do?

_____

d. Rewrite your `appendDictList(k,v)` function from the previous section to use the `.get(..)` method:

```
def appendDictList(d, key, value):
```

_____

_____

_____

_____

_____

_____

_____

h. How might lines 1-4 and 7-10 change if our values were strings instead of lists?

_____

8. Examine the following example code:

```
1 >>> d = {'pixel':'iris','wally':'steve','tally':'duane'}
2 >>> for mykey in d:
3 ...          print("{}'s dog is {}.".format(d[mykey], mykey)

4 >>> for k,v in d.items():
5 ...          print("{}'s dog is {}: ".format(v, k)
```

a. What does the programmer hope the output will be on line 3?

_____

b. For the first item of our dict, d, what is `mykey` and what is `d[mykey]`?

key:_____ d[mykey]:_____

c. What might line 2, `for mykey in d:` do?

_____

d. Write some code that will iterate over the items in `d` and print just the values:

_____

_____

_____

e. The output from lines 4 and 5 are identical to the output from lines 2 and 3. Explain why this might be the case:

_____

_____

f. For the lines 4 & 5, what might `k` and `v` represent?
k:_____v:_____

g. Write some lines of code to iterate through this dictionary of hockey team rankings and print the team name and current ranking:
`ranks = {'Amherst':18, 'Williams':7, 'Middlebury': 9}`

_____

_____

_____

_____

9. Examine the following example code:

```
0 >>> d = {'pixel':'iris','wally':'steve','tally':'duane'}
1 >>> for val in d.values():
2 ...          print("The value is: " + val)
```

a. What might the line `for val in d.values:` do?

_____

b. What would you guess the output of this code to look like?

_____

Application Questions: Use the Python Interpreter to check your work

1. Dictionaries make it easier to access values mapped to a given key, but it is trickier to find keys mapped to a given value. Write a function, `findKeys`, that, given a `value`, returns a list of all the keys mapped to that value.

```
def findKeys(value):
```

_____