Homework 1: Exploring the basics of Python

In this homework, you will apply what we have learned about the basic object classes of Python.

The assignment is split into two parts.

- **Part 1** should be completed by both **undergraduates (UG) and graduates (G)**.
- **Part 2** should be completed by (but is not restricted to) **graduates**

Part 1 (UG/G)

1. **Define a variable `sentence` and assign it the following value: `"Wherever you go, there you are."`**

```
#Defining variable
sentence="Wherever you go, there you are."
```

2. **Use 2 slicing espressions (via the `[]` notation) to isolate the word `there`.**

```
#first slicing expression
slice_exp1=sentence[17:22]
slice_exp1
```

```
'there'
```

```
#second slicing expression
slice_exp2=slice(17,22)
print(sentence[slice_exp2])
```

```
there
```

3. **Using only the method `.replace()`, change `sentence` to say "Wherever you go, you are there."**

`str.replace(x, y)` works as follows:

```
"string to change".replace(" to", "s")
```

```
'strings change'
```

```
#replacing "there you are" with "you are there"
sentence.replace("there you are","you are there")
```

```
'Wherever you go, you are there.'
```

4. **Now use `.replace()` to remove punctuation ("," and "."). Save the result as `clean_sentence`.**

```
#removing punctuation using replace command
clean_sentence=sentence.replace(".","")
clean_sentence=clean_sentence.replace("go,","go")
clean_sentence
```

```
'Wherever you go there you are'
```

5. **Split `clean_sentence` into a list of words. Save the result as `split_sentence` and print.**

```
# Splitting string
clean_sentence=clean_sentence.split()
clean_sentence
```

```
['Wherever', 'you', 'go', 'there', 'you', 'are']
```

5. **Write a loop that stores each unique word (key) and each length of the word (value) from `clean_sentence` as a dictionary `length_dictionary`.**

- Don't forget to start with an empty dictionary
- You can get each unique word from a list by using `set()`, as follows:

```
sample = ['a', 'b', 'c', 'a', 'a', 'c']
unique_from_list = set(sample)
print(unique_from_list)
```

```
{'b', 'c', 'a'}
```

... then you can loop over the set as if it were a list:

```
for word in unique_from_list:...
```

```
#creating dictionary
length_dictionary={}
```

```
for word in set(clean_sentence):
    print(word)
```

```
go
you
are
Wherever
there
```

```
for word in set(clean_sentence):
    length_dictionary[word]=len(word)
length_dictionary
```

```
{'go': 2, 'you': 3, 'are': 3, 'Wherever': 8, 'there': 5}
```

```
for word in unique_from_list:
    print(word)
```

```
b
c
a
```

6. **Now write a loop that prints each word in `length_dictionary` if it is 4 or more characters in length**

```
for l,w in length_dictionary.items():
    if w>=4:
        print(l)
```

```
Wherever
there
```

Part 2 (G)

1. **Define a variable `lyrics` using the following song lyrics.**

```
lyrics =  "White on white translucent black capes Back on the rack Bela Lugosi's dead The bat
```

2. **Write a loop that counts how many times each word occurs and stores it in a dictionary `word_count` . Print the result.**

Split the string into a list.

Open an empty dictionary.

The keys should be words, and the values should be frequencies.

Don't worry about case.

Remember that you can use `+=` to both add and replace previously defined terms:

- `word_count[key]` = 8
- `word_count[key]` += 1 --> 9

(you can add one to the value of a key each time you see that key)

You can only add to keys that you have seen before! **You must use a condition to check if the word you are seeing is in your dictionary or not!**

```
open an empty dictionary
if a word is in your dictionary keys
  add one to the count
otherwise
  define a new key with a value of one
```

```
# Spliting string
lyrics.split()
```

```
         'Bela',
        "Lugosi's",
        'dead',
        'Bela',
        "Lugosi's",
        'dead',
        'Undead',
        'undead',
        'undead',
        'Undead',
        'undead',
        'undead',
        'Undead',
        'Oh',
        'Bela',
        "Bela's",
        'undead',
        'Oh',
        'Bela',
        "Bela's",
        'undead',
        "Bela's",
        'undead',
        'Oh',
        'Bela',
        "Bela's",
        'undead',
        'Oh',
        'Bela',
        'Undead']

string=lyrics.lower()
string=string.split(" ")


#frequency dictionary
word_freq_dict={}
for i in string:
    if i in word_freq_dict:
        word_freq_dict[i]+=1
    else:
        word_freq_dict[i]=1
word_freq_dict

     {'white': 2,
      'on': 2,
      'translucent': 1,
      'black': 2,
      'capes': 1,
      'back': 1,
      'the': 7,
      'rack': 1,
      'bela': 10,
      "lugosi's": 6,
      'dead': 7,
```

```
 'bats': 1,
 'have': 2,
 'left': 1,
 'bell': 1,
 'tower': 1,
 'victims': 1,
 'been': 1,
 'bled': 1,
 'red': 1,
 'velvet': 1,
 'lines': 1,
 'box': 1,
 'undead': 18,
 'virginal': 1,
 'brides': 1,
 'file': 1,
 'past': 1,
 'his': 1,
 'tomb': 1,
 'strewn': 1,
 'with': 1,
 "time's": 1,
 'flowers': 1,
 'bereft': 1,
 'in': 2,
 'deathly': 1,
 'bloom': 1,
 'alone': 1,
 'a': 1,
 'darkened': 1,
 'room': 1,
 'count': 1,
 'oh': 4,
 "bela's": 4}
```

Try the string method `.lower()`.

```
"CAT".lower()
```

```
'cat'
```

```
lyrics.lower()
```

```
'white on white translucent black capes back on the rack bela lugosi's dead the bats hav
have been bled red velvet lines the black box bela lugosi's dead bela lugosi's dead unde
ead the virginal brides file past his tomb strewn with time's dead flowers bereft in dea
m the count bela lugosi's dead bela lugosi's dead bela lugosi's dead undead undead undea
ela bela's undead oh bela bela's undead bela's undead oh bela bela's undead oh bela unde
```

Can you improve your counts from above?

```
#improving counts
def wordFrequency(string):
    string=string.lower()
    string=string.split(" ")
    word_frequency={}
    for i in string:
        if i in word_frequency:
            word_frequency[i]+=1
        else:
            word_frequency[i]=1
    return(word_frequency)

print(wordFrequency(lyrics))
```

```
{'white': 2, 'on': 2, 'translucent': 1, 'black': 2, 'capes': 1, 'back': 1, 'the': 7, 'ra
```