2 & 4 hours

The basic idea of this assignment is to design and implement a `Sentence` class.

The first method that all classes provide is the constructor `__init__`.

The constructor will then create instance variable to hold the values. Each instance variables holds a reference to an object.

The extra parameter self, is a special parameter that will always refer to the object that is being constructed, it must always be the first parameter in the list.

Then we write the accessor methods to access the instance variables of the object. Also called the "getter" methods. Typically, each instance variable might have an associated accessor method.

```
>>> #PPiC 10.4 Create a new class called Sentence. The constructor should accept a single parameter that is a string.
#Create an instance variable that stores the sentence as a string. Assume the sentence has no punctuation.
>>> #PPiC 10.5 Write the following accessor methods for the sentence class created in Exercise 10.4
>>> import requests
>>> class Sentence: #create a class called Sentence
        def __init__(self, istring): #10.4 constructor __init__ parameter istring as a string
                self.string = istring
        def getSentence(self): #10.5(a) getSentence: Return the sentence as a string.
                return self.string
        def getWords(self): #10.5(b) getWords: Return the list of words in the sentence.
                return self.getSentence().split() #split() function to split a string by whitespace since no seperator defined.
        def getLength(self): #10.5(c) getLength: Return the number of characters in the sentence.
                return len(self.getSentence()) #len() function returns the number of characters in a string.
        def getNumWords(self): #10.5(d) getNumWords: Return the number of words in the sentence.
                return len(self.getWords()) #len() function returns the number of elements in a list.
        #PPiC 10.10 Write a mutator method for the Sentence class that allows you to capitalize all the words in a sentence.
        def getCapitalizeWords(self):
                newWords = [] #create an empty list called newWords for words to capitalize.
                for word in self.getWords(): #for each word in the list of words in the sentence,
                        newWords.append(word[0].upper() + word[1:]) #word[0].upper() capitalize the first letter of the word
                        #combine it with the rest part of the word, and append to the empty list.
                self.string = " ".join(newWords) #join() returns a string in which the string elements of newWords
                                                 #have been joined by " " separator.
        def __str__(self): #PPiC 10.13 Add a __str__ method to the Sentence class you started in Exercise 10.4 or 10.6.
                return self.getSentence()
        def __len__(self): #PPiC 10.15 Implement the __len__ method so that you can use the len operator.
                return self.getNumWords()


>>> mysentence = Sentence("i love python") #test by taking "i love python" as the string, and use get methods.
>>> mysentence.getSentence()
'i love python'
>>> mysentence.getWords()
['i', 'love', 'python']
>>> mysentence.getLength()
13
>>> mysentence.getNumWords()
3
>>> mysentence.getCapitalizeWords()
'I Love Python'
>>> mysentence.__str__()
'i love python'
>>> mysentence.__len__()
3
```

```
>>> #PPiC 10.6 Create a variation of the Sentence class, again called Sentence.
#The constructor should accept a single parameter that is a string.
#This time create an instance variable that stores the sentence as a list of words.
>>> class Sentence: #create a class called Sentence
        def __init__(self, istring): #constructor __init__ parameter istring as a string
                self.wordlist = istring.split() #get a list of words using split() to split a string
                                                #by whitespace since no seperator defined.


>>> mysentence = Sentence("i love python")
>>> mysentence.wordlist
['i', 'love', 'python']
>>> #PPiC 10.7 Write the following accessor methods for the new class created in Exercise 10.6
>>> class Sentence: #create a class called Sentence
        def __init__(self, istring): #10.6
                self.wordlist = istring.split()
        def getSentence(self): #10.7(a) getSentence: Return the sentence as a string.
                return " ".join(self.wordlist)
        def getWords(self): #10.7(b) getWords: Return the list of words in the sentence.
                return self.wordlist
        def getLength(self): #10.7(c) getLength: Return the number of characters in the sentence.
                return len(self.getSentence())
        def getNumWords(self): #10.7(d) getNumWords: Return the number of words in the sentence.
                return len(self.getWords())


>>> mysentence = Sentence("i love python")
>>> mysentence.getSentence()
'i love python'
>>> mysentence.getWords()
['i', 'love', 'python']
>>> mysentence.getLength()
13
>>> mysentence.getNumWords()
3
```