ASSIGNMENT 1

Contents

[Contents 1](#_Toc467838104)

[Purpose 2](#_Toc467838105)

[Assignment Background 2](#_Toc467838106)

[Assignment Statement 2](#_Toc467838107)

[Requirements: 3](#_Toc467838108)

[Code/Comment Format 3](#_Toc467838109)

[What to Deliver 4](#_Toc467838110)

[Notes 4](#_Toc467838111)

[Grading 5](#_Toc467838112)

# Purpose

In this assignment, you will be demonstrating your understanding and use of python constructs, as covered in module 1.

# Assignment Background

This assignment involves the game Mad Libs. Mad Libs is a game which has text missing some types of words such as verbs, nouns, adjectives etc. Here is a basic example of a multiple line mad lib:

Man verb on a adjective noun.

Noun verb to the adjective ground.

All the king’s adjective horses and all the king’s dainty noun could not verb scrambled egg man back together again.

Example word lists:

NOUNS = ["time", "year", "people", "way", "day", "man", "thing", "woman"]

VERBS = ["pay", "put", "read", "run", "say", "see"]

ADJECTIVES = ["other", "new", "good", "high", "old"]

# Assignment Statement

* You can either choose to use the example or create your own mab lib.
* We provide you with examples of lists that you use or you may create four lists of your choice.
  + 1 for nouns
  + 1 for verbs
  + 1 for adjectives
  + 1 for sentences (with missing words)
* The lists must be of different lengths. For example, nouns list has 4 items, other lists have more or less than 4 items. Only one list can have a given length.
* The lists should be < 10 items long.
* Each sentence should be missing 1 element of each type (noun, verb, adjective).

The file*\_\_init\_\_.py* is blank and should not be modified. You are required to supply *Last Name\_First Name\_Assignment1.py.* In this assignment, you will accomplish the following:

# Requirements:

1. Prompt user for a number.
2. Validate user input for the following:
   1. Numbers are positive
   2. Numbers are integers (Not floats)
3. The user input will be used to retrieve from each of your lists.
   1. Get 1 sentence from the sentence list.
   2. Get 1 noun from the noun list.
   3. Get 1 verb from the verb list.
   4. Get 1 adjective from the adjective list

*Hint: Watch out! The user input could be larger than the number of elements in the list*

1. Insert the selected noun, verb and adjective into the selected sentence.
2. Check that the resulting sentence is unique (has not been used before).
   1. If it has not been used before, save it in the list of completed sentences (your mad lib)
   2. If it has been used before, do not save it and print message letting the user know.
3. Print your current mad lib to the user (list of all completed sentences so far)
4. Ask user if he/she wants to keep playing (ans: y or n)
   1. Validate user input
   2. If “y” go back to 1
   3. If “n” exit

# Code/Comment Format

Good code includes well named variables that are consistent from the beginning to the end of the program. Naming of objects should be self-explanatory. For instance, iterator\_for\_noun\_list is much better than i.

Every program consists of a sequence of paragraphs, each of which has objectives, and which builds on the previous paragraphs. We are mostly interested in objectives that are valid at the end of the program so we can verify the program's design. The following is a preferred form for such paragraph headings. The # sign is adequate when the comment is a single line.

#This is an in-line comment – used to document the code for you, or anyone else, that intends

#To extend the code

In-line comments are helpful when one has to go back to the code 6 months later to make changes.

For doc strings, python allows the use of triple quotes. The triple quotes can be either single or double quotes. A doc sting is generally used as user documentation. It does not need to include details of the implementation of the program, but instead it provides documentation as how to use the API for the program (input, output etc.)

For example:

“””

This is an example of a doc string

It allows multiple lines within the string.

“””

‘’’

This is an example of a doc string

It allows multiple lines within the string.

‘’’

This becomes significant when using functions, classes etc. as the triple quotes help to self-document the parameters and return values of the function.

Provides a good guide to code with in Python. You are not required to use it, but whatever style you choose must be consistent throughout. <https://www.python.org/dev/peps/pep-0008/>

# What to Deliver

Supply

1. The *Last Name\_First Name\_Assignment1.py* file (the *\_\_init\_\_.py* file is attached and should be left as is).

# Notes

* Assignments can be submitted once. If extenuating circumstances exist, contact your facilitator.
* Note the statement in the syllabus on timeliness of submissions (the gist being that all assignments must observe the deadlines).
* Start by identifying and ordering the objectives.
* There are no testing requirements for this assignment. However, it would be prudent to make sure your program does not crash and all input validation is performed correctly.

# Grading

Assigned point values for Assignment 1

Step 1 – 10 points

Step 2 – 10 points

Step 3 – 20 points

Step 4 – 10 points

Step 5 – 25 points

Step 6 – 10 points

Step 7 – 15 points