CS 232 – Programming in Python

Assignment #3

Deadlines

This assignment is due on **Thursday, April 12, 2016 @ 5:00 PM (beginning of class)**

How to submit your work on the assignment:

- Your assignment will be turned in as a computer file (the "module") containing your Python code. The module will contain only the code that satisfies the problems' requirements. **There is NO NEED to submit a testing module! I will run your code with my own testing module.**

The File Naming Convention

- Properly name the file that contains your work. The file's name will contain the following, with dashes in between each part listed below and no spaces:

  * The course name, **`CS232`**

  * Then a dash and the assignment number as a two-digit number, in this case **`03`**

  * Then a dash and your HSU account username with your initials and a number – in this example, I'll use **`jqs123`** for a typical student named John Q. Smith

  * Windows should automatically add a **`.py`** at the end of the file. This may be invisible, depending on how Windows is configured on the computer you're using. If you're using a Mac, you'll likely need to add a **`.py`** to the end of the filename.

  Put all the rules together, and John Q. Smith's Assignment 1 file in CS 232 would be named

  `CS232-03-jqs123.py`

  Your file name will be different from this because your HSU username is different, but that's the **only** difference!

Submitting your file electronically

For this assignment, you will email your file to the instructor as a file attachment. Future assignments may change the way you submit your work, but email will do for now.

Send your email to: **`david.tuttle@humboldt.edu`**

Type the Subject line of the email as: **`CS 232 - Assignment #3`**

When I receive your file in the correct way, I will send you an acknowledgment by reply email. If there's a problem, I will let you know by reply email that you need to submit the file again.

If you do NOT get an email reply within 12 hours, try sending the file again. If you wait until too close to the deadline, you run the risk that an improper submission may result in a late penalty!

Documentation of your Python code

For this assignment, please place all functions within a single Python module (file).

Begin your **Python module** (file) that you submit with at least the following opening comments:

* a comment containing the name of the file,
* a comment containing your name, and
* a comment containing the date that your module was last modified

For example:

```
# CS232-03-jqs123.py
# John Q. Smith
# Last modified: February 30, 2099
```

In this homework, the documentation at the beginning of each function has already been written:

For example:

```
# Problem 1
# add_them: float float → float
# purpose:  expects two numeric (float) values
#           returns the sum of the two values
# side effects: prints a message containing the sum of the two values
```

PROJECT – THE GAME OF HANGMAN

Using the supplied files **`hangman_template.py`** and **`words.txt`**, complete the coding necessary to play the game Hangman. They need to be kept in the same local folder when testing your code!

Rename the **`hangman_template.py`** file to the requested filename used for CS 232 homework submissions (explained in page 1). Place them into the folder you will use for this assignment, and uncomment and type the full folder path into the **`os.chdir()`** statement near the top of the file. When submitting the assignment, you need only supply the one Python file – I don't need extra copies of the **`words.txt`** file!

It will help you greatly to carefully look over the code already written in **`hangman_template.py`** – familiarize yourself with the supplied functions and especially the global variables defined to contain items such as the **`secret_word`** to be guessed and the list of **`letters_guessed`**. Your **`play_hangman()`** function should also use a local variable **`mistakes_made`** to keep track of the number of wrong letter guesses. These three variables are all the information Hangman needs to keep track of during game play.

PROBLEM 1 – THE `word_guessed()` FUNCTION (15 points)

First, write the function **`word_guessed()`** and test it thoroughly by supplying it with various values for secret_word and letters_guessed. For example, once your function is written you can test it by doing the following:

```
Loading word list from file...
   55900 words loaded.
Enter play_hangman() to play a game of hangman!
>>> secret_word = "arf"
>>> letters_guessed = ['a', 'r', 'f']
>>> word_guessed()
True
>>> letters_guessed = ['a', 'r']
>>> word_guessed()
False
```

Test it thoroughly with various words and letter lists.

HINT: You can test to see if something is NOT in a list by using the following syntax:

```
if item_variable not in list_variable:
```

PROBLEM 2 – THE `print_guessed()` FUNCTION (15 points)

Now write the print_guessed() function and test it thoroughly using the same method as in Problem 1.

```
Loading word list from file...
   55900 words loaded.
Enter play_hangman() to play a game of hangman!
>>> secret_word = "arf"
>>> letters_guessed = ['a', 'r', 'f']
>>> print_guessed()
a r f
>>> letters_guessed = ['a', 'r']
>>> print_guessed()
a r _
```

Test it thoroughly with various words and letter lists.

PROBLEM 3 – THE `play_hangman()` FUNCTION (70 points)

This problem will take by far the longest time, so be sure to make time for it!

This function will control the play of the game by choosing a secret word then performing a loop where it will:

- Ask the player for a letter
- Determine if the word has been completely guessed
- If the word isn't guessed yet, find out if the guessed letter is in the word
- Draw the updated Hangman image to the screen and print the guessed letters in the word
- If the player's out of guesses and is "hanged", then exit the function
- If the player still has guesses left, go back to the top and ask for another letter

For example, if the secret word is "claptrap", then game play would look like this:

```
Loading word list from file...
   55900 words loaded.
Enter play_hangman() to play a game of hangman!
>>> play_hangman()
The secret word is ready!
Guess a letter: a
     _________
     |    |
     |
     |
     |
     |
     |_________

     _ _ a _ _ _ a _
     LETTERS GUESSED: a

Guess a letter: e
     _________
     |    |
     |    O
     |
     |
     |
     |_________

    _ _ a _ _ _ a _
    LETTERS GUESSED: a e
```

And so on. Test it thoroughly, and have fun playing Hangman!