Programming Project I

PROG1700 Logic and Programming

Evaluation : 10% of final grade

Due Date : Nov 1 @ 8:30am

All students will submit their finished project folder to Brightspace (Projects / Programming Project I) on or before the deadline.

The project will then be marked "face to face" with each student during code review during the lesson times for that week. All students must be present for code review. Your project will NOT be marked if you are not present for code review, regardless if you submitted it to Brightspace. No code review = 0 on your project.

Assignment Description

Programming Project I will be to design and develop a python application that generates insults. This "generator" will take a victim's name as input and generate random insults about that victim consisting of 1 to 3 adjectives and 1 noun.

The requirements are:

1. The application must get the following input from the user:
   - The number of insults to generate
   - The victim's name
   - The number of adjectives to include in the generated insults
2. All input must be error checked as follows:
   - The number of insults must be an integer of one or greater
   - The victim's name can be anything except ""
   - The number of adjectives must only be 1, 2, or 3

   If an error is detected, a detailed message describing the specific error must be displayed to the user. The program will then require all input to be entered again
3. If all input passes error checking, the application generates a series of insults consisting of randomly selected adjectives and a randomly selected noun. The number of insults generated depends on the input provided. The number of adjectives in each insult depends on the input provided.
4. The application contains a total of 10 possible adjectives and 6 possible nouns to randomly select.
5. A generated insult follows the following format (depending on number of adjectives): [victim's name] is a [adjective1] [adjective2] [adjective3] [noun]!
6. Repeated adjectives are allowed in a single insult.
7. After the insults are generated, the program will automatically start over again asking the user to enter in the input once again.
8. Your application should include a main() user-defined function as well as other user-defined functions where appropriate much like our lesson samples
9. You should not have any large blocks of code that are repeated. This is known as redundant code...however, handling multiple insult adjectives in this project tends to take beginners down this route. Can you use loops or user functions to avoid this?
10. You MUST be able to loosely explain your code / logic (blindly copying and pasting code is not programming). Failure to do so will result in a zero on the project. Essentially, if you wrote it, you will understand it.
11. Be sure to include informative comments in your code

Requirements (Marks breakdown)

Application Test Cases	
Application test cases *A series of test cases that test all requirements listed above*	10
Code Requirements and Design	
All code organized in user functions	1
Redundant code minimized with loops and/or user functions	2
Internal Commenting	1
TOTAL MARK	14