Programming Project I

PROG1700 Logic and Programming

Evaluation : 10% of final grade

Due Date : Nov 1 (WEB) / Oct 30 (SYSTEMS)

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 one or greater
   - The victim's name must not be ""
   - The number of adjectives must be between 1 and 3
   - Letters, decimal numbers, etc. as input for number of insults / number of adjectives

   If an error is detected an appropriate error message must be displayed to the user and the application will stop
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. Repeated adjectives are allowed.
5. A generated insult follows the following format (depending on number of adjectives): [victim's name] is a [adjective1] [adjective2] [adjective3] [noun]!
6. Your application should include a main() user-defined function as well as other user-defined functions where appropriate much like our lesson samples
7. 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?
8. In the past, some students have explored Lists when developing a solution for this project. Although they do provide an efficient solution to the problem, this project requires that NO LISTS / DICTIONARIES / TUPLES / SETS are used.
9. You must be able to explain your code / logic – blindly copying and pasting code is not programming. Failure to do so could result in lost marks or even a zero on the project. Essentially, if you wrote it you will understand it.
10. 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

Other Notes

This project will be marked face to face with each student during code review on the due date. This means that I will mark the project with you. Remember to bring your finished PY file!