COSC 010 Project 1

I. Overview for Project 1

- For Project 1, you will create a (primitive) conversational agent using Python. The purpose of the conversational agent (i.e., what kind of conversation it will have with the user) is up to you, but you must use specific programming concepts (detailed in Section III).
- Here's a short description of this topic: https://www.chatbots.org/conversational_agent/
- Note that, while modern conversational agents involve a great deal of sophistication, at their core they require a program to a.) take in user input, b.) make decisions based on that input and c.) provide user-friendly output. You thus already have the tools to create some sort of conversational agent. To be clear, your conversational agent does not need to be anywhere near as complicated as the examples provided at the above website (for one thing, yours will just be text – no picture or animation), although the examples there might serve as inspiration.
- **Python Topics:** user input/output, data types, conditionals, string methods
- **Resources**: Remember that the Python for Everyone textbook (Chapters 1-3 and 6) and Dr. Gates's textbook are both great resources for the various Python topics.
- You must complete this project **individually** and submit it to Canvas.

II. What You Will Submit

Part 1: The Python Program
You will submit your program as a .py file. See the next section below for the specifications and required elements. ***Take careful note of the following:***

- Projects must run using Python 3.
- The file must run and function to gain credit. Programs that do not run will receive zero credit.
- If the graders need to email you or "work" with you after submission to see that your project runs, the penalty is -30%.
- Double check your work.

Part 2: The Project Report
You will also submit a Project Report as a .pdf file.

1. The first page will contain an explanation of what your project does and how it works, including:
   - Overview of your conversational agent, including its purpose and function
   - Expected inputs and outputs
   - Detailed description of how you used each of the required computing concepts
   - Limitations (e.g., do certain user inputs mess up or "break" your program?)
   - Future expansion options.

   All this will help a person to understand your project. You may use screenshots in the explanations.
2. The second (and more as needed) page(s) will contain three (3) examples of program use – including input and output. You can screenshot these if you wish.
3. The remaining pages will contain a copy of the Python code (***You will also submit the code as .py***).

III. Project Specifications and Required Elements

1) While you must meet the specific requirements, there is room for creativity in this project. Indeed, part of coding is the creativity of understanding a problem itself and thinking how code can solve it. Each project will be unique (like a fingerprint).
   - **DO NOT COPY anyone else's code.**
   - **DO NOT SHARE** your ideas or code with other students (until after project grading is complete).
   - You can use the Internet and all class resources *to learn about syntax*. **However, you must write your own program from start to finish. DO NOT COPY** code from the Internet.

2) **Good code is readable code**. Comments are required in your code. Be sure to submit clear code.

3) **Good code is user-friendly code**. Design your code with the user in mind, with appropriate outputs. Consider using the **time** library (as in Lab 4) to make the timing of your conversational agent's outputs more "natural".

4) **Your project must include the following elements:**
   - User input (via **input()** function)
     - At least one user input must be a number that the program than uses in some numeric context (e.g., arithmetic)
   - Output (via **print()** function)
   - At least 5 variables
   - At least three conditional structures (i.e., beginning with lines that start with the keyword **if**)
   - At least one conditional structure must include an **else** clause
   - At least one conditional structure must include at least one **elif** clause
   - At least one conditional structure must be nested in another one
   - At least one example of exception handling (using **try**/**except**)
   - At least 1 string method (https://docs.python.org/3.8/library/stdtypes.html#string-methods)
     - Read Chapter 6 of Python for Everyone: https://www.py4e.com/html3/06-strings

Final Notes:
- Do not copy or repurpose code from the book or the Internet. Be original and creative. Use the book to learn. Close the book. Then create.
- This does not need to be a huge project. However, you are not limited – feel free to get creative and expansive.
- Start with an idea and plan it out on paper.
- Begin small. Then test the small bit of code to ensure correctness. Then add some more code. Then test the expanded program. Then add some more code – and so on until completion of your program.

IV. How to submit

- Enter Canvas, go to COSC-010, then Assignments and choose Project 1
- Upload your Python Program as ***P1_LastName.py.*** (Replace "LastName" with your actual last name)
- Upload your Project Report as ***P1Report_LastName.pdf***
- ** DOUBLE CHECK your submission. If the grader needs to email you, the penalty is -30%.