Homework #1

Introduction to Engineering Computing

Due at 5:00 PM on February 7, 2020

Homework Description

The show *The Big Bang Theory* features four socially challenged friends: three physicists and an engineer. In one episode, one of the physicists, Dr. Sheldon Cooper, creates an algorithm to make friends. Go here to see the clip: http://youtu.be/k0xgjUhEG3U.

You are to design and implement your own friendship algorithm to determine if someone could be your friend. Your algorithm should have **at least six questions in total**, with at least **three of them nested** (i.e. a question that is only asked when another question is answered a certain way). Also, each possible answer to each question should **have a given number of points** associated with it.

As an example, here is a possible question you could ask:

What do you think of the Big Bang Theory? (1 – Love it; 2 – It's alright; 3 – Can't stand it)

For this example question, you may decide to assign points to the user's answer as follows:

- If the user enters 1, then 100 points will be added to their score *and* a nested question would then appear, asking "**Who is your favorite character on the show?**" (The nested question would again have multiple answer choices, with each answer resulting in some change in the user's total point score).
- If the user enters 2, then 10 points will be added to their score (with no nested question given).
- If the user enters 3, then 15 points are subtracted from their score (again, with no nested question issued).

When the user has been asked all applicable questions, **their score should be printed out** along with a **message informing the user about their friendship compatibility**. The questions asked, the points assigned, and the scoring system are all completely up to you.

As an example:

- a score of 300 or more could trigger the program to output a message like "**Wow, you are PERFECTLY matched with me!**"
- a score of 200 or more could instead say "**You are well-matched with me**,"
- a score between 50 and 200 might say "**We'll probably get along OK**,"
- while a score of less than 50 (including a negative score) could say "**You are not likely to become my friend.**"

Your scoring system and your messages to the user are completely up to you.

Note: In the clip Sheldon has a loop in his algorithm (which turns out to be an infinite loop -- oops). We have not covered loops yet (e.g., `while` and `for` loops) so you are not expected to use any loops, but if you are feeling ambitious, you are welcome to try it. (However, if you do include one, make sure that it won't be an infinite loop.)

Some food for thought

Before you jump right in and start programming, you should probably take some time and plan your algorithm just like Sheldon did in the clip. You might want to use a flow chart (like Sheldon) or pseudo-code to help you out.

Grading Rubric

80 points for working code

- 25 points for correct logic for at least six total questions
- 25 points for correct logic for at least three nested questions
- 20 points for correctly giving and taking points away for each answer and correctly computing the final score
- 10 points for printing friend status

20 points for correct style (see Style Guide on ICON for correct style)

- 5 points for comment block above top of program
- 5 points for meaningful variable names
- 5 points for proper indentation
- 5 points for in-line comments

NOTE: If your program does not compile on an IEC machine or if you do not submit a .cpp file by the deadline, you will receive a zero on your homework. Late homework will not be accepted – please do NOT wait until the last minute and submit at 4:59pm the day the assignment is due (excuses such as "the internet went down at my house just as I tried to submit it" will not be accepted). Also, as a reminder, make sure you have read over the homework cheating policy and submitted the cheating policy "assignment" before submitting homework 1. Any evidence of working together on this assignment will result in **dropping a whole letter grade** for the course and a letter being sent to the Dean's Office.