Programming Assignment #2

Due Oct 8 by 3pm **Points** 100 **Submitting** a file upload
Available Oct 1 at 8pm - Dec 5 at 12pm 2 months

This assignment was locked Dec 5 at 12pm.

Assignment #2: Structured Programming and Top-Down Design

For this assignment, create a robust C# console application that implements the game "Connect 4." Here is a video of the original, physical game in action:

Connect Four - "Pretty Sneaky, Sis..." (Commercial, 1981) (https://www.youtube.com/watch?time_continue=5&v=KN3nohBw_CE)

(https://www.youtube.com/watch?time_continue=5&v=KN3nohBw_CE)

"Pretty sneaky, sis!" - lol

Implement a computer version of Connect 4 that allows a human to play a computer. For each turn, prompt the human player for the number of the column that they wish to drop a checker into. Have the computer player select the column for their turn randomly. Your program needs to show the playing grid after each turn. If one of the players gets 4 checkers in a row, your program needs to detect that, end the game and announce the winner.

Click here to see the output from a sample version of this program:
PROGRAMMING ASSIGNMENT #2 - SAMPLE OUTPUT

Please use the Top-Down Design techniques we discussed to first create pseudocode for this program. **Have another student from this class review your** pseudo**code informally before starting to write your actual code! (You can also review their pseudocode.) Include the name of your design reviewer in a comment at the top of your program.**

In addition, DO NOT CHECK FOR DIAGONAL WINS! Just check for 4 checkers in a row either horizontally or vertically.

Please do not use C#'s "Collection" classes during this assignment. I want to make sure you understand how the non-OO features of C# work before we move on to the OO stuff (including collections). You should only need the simpler data types that we have talked about in class. Specifically: strings, chars, ints, longs,

doubles, bools, enums, and arrays. (You can also try using Structs in this assignment if you want, but they are not required.)

We will start using the OO stuff in our next class.

Important Notes:

- Before starting, be sure to review the information in **PREPARING FOR PROGRAMMING ASSIGNMENT #2**
- Be sure to read, understand and follow **[0070] Our C# Coding Standards**
- Do not use any C# language features that have not be covered in the reading or discussed in class.
- Always choose descriptive variable, parameter, and method names. Always strive to make your code self-documenting.
- Make sure all user prompts and error messages are clear and easily understood.
- ***Validate all input. Make sure a user cannot crash the program by entering bad data.***
- When you are finished, please ZIP up the Visual Studio Solution directory for your program and upload it using the link on this page.
  Please read the detailed instructions for submitting your assignment. They are located at the bottom of PREPARING FOR PROGRAMMING ASSIGNMENT #2

Grading Guidelines:

40% Does the program compile and run?
40% Does the program generate correct results for normal user inputs?
10% Does the program have descriptive variable, parameter and method names? Were methods created intelligently?
10% Was the program designed in a top-down manner? Was the pseudo-code reviewed and used in the creation of the final program?