CSE241 – OOP (Fall 2025)

Homework #4

Hand-in Policy: Source code and any documentation should be submitted online as a single .zip or .rar file with naming convention STUDENTID_LASTNAME_FIRSTNAME_H4.ZIP via Teams by the submission deadline. No late submissions will be accepted.

Collaboration Policy: No collaboration is permitted. Any cheating (copying someone else's work in any form) will result in a grade of -100 for the first offense and -200 for the subsequent attempts.
Grading: Each homework will be graded on the scale of 100. Unless otherwise noted, the questions/parts will be weighed equal.

In this homework, you will re-implement your ConnectFour program in C++ using hierarchy of `ConncetFour` classes. To start with, you create a cell class (no inner class) which holds the position of the cell (A, B, C, etc.) and the row number (1, 2, 3, etc.). The class `Cell` also includes all necessary setters/getters, constructors etc. Remember a cell can be empty, user (user1, user2) or computer.

Your base class `ConnectFourAbstract` defines the following member functions. It is an abstract base class.

- The class has functions to read and write from files. You will decide on the file format.
- The class has functions to return the current width and height of the board.
- The class has a function that displays the current board on the screen.
- The class has two functions named `play` playing the game for a single time step. First function does not take a parameter, and it plays the computer. The second function takes a cell position, and it plays the user.
- The class should have a function that returns if the game ended.

The class should have a function named `playGame`. This function plays the game by asking the user the board size first then asks the user to play and the computer plays, etc.

The class `ConnectFourPlus` derives from the abstract class. It plays very similar to regular ConnectFour but it only accepts horizontally or vertically connected cells as the goal of the game.

Similarly, the class `ConnectFourDiag` derives from the abstract class. It plays very similar to regular ConnectFour but it will only accept diagonally connected cells as the goal of the game.

The class `ConnectFourPlusUndo` is very similar to `ConnectFourPlus`. It can undo the moves of the user or the computer until the beginning of the game.

Important Considerations:

- Use dynamic allocation to keep your Cell data.
- Separate the interface and the implementation of the classes. Since there are 5 classes, you will have at least 11 source files to be submitted if you include the driver code.
- Use your own namespace to keep all the classes.
- Your program should use object-oriented programming paradigms (i.e., user classes, encapsulate your data and algorithms as needed).

- Do not forget to “indent” your code and provide comments.
- Do not use anything that we did not learn in the lectures.
- Check the validity of the user input.
- Test your programs very carefully and see several different runs. Submit at least two saved files with your HW.