CSE241 – OOP (Fall 2025)

Homework #3

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_H3.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 write a game (ConnectFour program) in C++ using dynamic memory and object-oriented programming techniques. You will not use any vectors or constant sized arrays.

The game of Connect Four is played by two players (computer-user or user1-user2) on a two dimensional board (2D array) with rectangular cells. Each cell is either computer, user, or empty. The game starts with the following board for an 8x8 game

```
abcdefgh
........
........
........
........
........
........
........
........
```

where '.' represents the empty cells, 'X' represents the computer(user1) cells and 'O' represents the user (user2) cells. The players take turns to play the game. For the above board, the user can pick any letter to put the first piece. For example, if the user1 plays to letter C and user2 plays to letter B then the board becomes

```
abcdefgh
........
........
........
........
........
........
........
.XO.....
```

The game continues with the objective of "Connect four of your checkers in a row while preventing your opponent from doing the same". The rows can be diagonal, vertical or horizontal. For the detailed rules of the game see https://www.hasbro.com/common/instruct/ConnectFour.PDF

Your main class for this homework will be `ConnectFour`. This class will have a private inner class named `Cell` to represent a Connect Four cell. The class `Cell` will hold the position of the cell (A, B, C, etc) and the row number (1, 2, 3, etc). This class will also include all necessary setters/getters, constructors etc. Remember a cell can be empty, user (user1, user2) or computer. You will overload the following operators for the class cell.

- Operator == to compare two cells.
- Pre- and post- increment/decrement operators for cell. The ++ operator changes the cell state from empty to user1, from user1 to user2, from user2 to computer, from computer back to empty.
- Stream insertion and extraction operators for cell.

Next, write a class named `ConnectFour` to represent and play the game. The class `ConnectFour` will hold a `Cell ** gameBoard` data member to represent the game board as a 2D dynamic array.

The class `ConnectFour` will also have the following features and methods:

- There is no limit for the board dimensions. Your game will resize according to the parameter for the constructor.
- There should be at least 3 constructors.
    - The no parameter constructor creates a 5x5 game.
    - Another contructor takes number of rows and columns as parameters.
    - Another constructor takes a text file as a parameter. This text file will contain the board shape marked with * characters. For example, the following is a board configuration: it has 7 rows and 17 columns. Some of the columns are shorter than the others.

```
   ***
  *****
 *******
*********
 ***********
  ***********
***************
```

- The class will have functions to read and write from files. You will decide on the file format.
- The class will have functions to print the current configuration of the board to the screen.
- The class will have two functions named play that plays the game for a single step. First function does not take a parameter, and it plays as the computer. The second function takes a cell position, and it plays as 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 file name first then asks the user to play, and the computer plays, etc.
- The class will overload operator == and operator !=
- You can add any other functions (public or private) needed.

Write your main function to test both classes. Make at least 5 objects of class `ConnectFour` and play the games at the same time.

Important Considerations:

- Do not use any functions from the standard C library (like **printf**), you will use << and >> operators to print and write strings.
- 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.
- Submit at least 3 sample board configuration files.