Lab 03 - Abstraction & Encapsulation

Instructions:

- **In the game Connect 4, two players take turns dropping colored discs down columns of a 6×7 grid, aiming to connect four discs in a row – horizontally, vertically, or diagonally. The grid can be represented as a string of length 42, where players' discs and empty spaces are denoted by the characters `'O'`, `'X'`, and `'*'`, respectively. For example, an empty board is shown on the left.**

	0	1	2	3	4	5	6
0	*	*	*	*	*	*	*
1	*	*	*	*	*	*	*
2	*	*	*	*	*	*	*
3	*	*	*	*	*	*	*
4	*	*	*	*	*	*	*
5	*	*	*	*	*	*	*

Empty Board Display

	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	7	8	9	10	11	12	13
2	14	15	16	17	18	19	20
3	21	22	23	24	25	26	27
4	28	29	30	31	32	33	34
5	35	36	37	38	39	40	41

String Indices Correlation

with the indices of the string elements on the right. Your objective is to define a class named *C4Board* using the above information in a header file named `'Connect4.h'` except it should represent a three-player game with the third disc being `'S'`.

- **`'Connect4.h'` must contain a header guard.**
- **The class and function must be defined within a namespace named `'oopl'`.**
- **`'Connect4.h'` can only include the libraries *iostream*, *string*, *sstream*, and *iomanip*.**
- **Each function or method excluding special member functions must include pseudocode as a comment above it to receive any credit.**
- **Your submissions must be submitted to the GitHub repository in the Lab03 directory.**
- **Cheating of any kind is prohibited and will not be tolerated.**
- **Violating or failing to follow any of the rules above will result in an automatic zero (0) for the lab.**

Grading

Task	Maximum Points	Points Earned
1	5	
Total	5	

Note: solutions will be provided for tasks colored blue only.

Task 1

- In a header file named `"Connect4.h"` within the namespace *oopl*, define a class named *C4Board* that needs to contain
  - ☐ a private string field named *grid*.
  - ☐ a private int field named *player*.
  - ☐ a public static constant string field named *discs* that is initialized to `"OXS"`.
  - ☐ a public default constructor that assigns a string of 42 asterisks and 0 to *grid* and *player*, respectively.
  - ☐ a public copy constructor.
  - ☐ a public assignment operator.
  - ☐ a public empty destructor.
  - ☐ a public integer constant method named `current()` that takes no parameters and returns the current player (range [1,3]).
  - ☐ a public character constant method named `value()` that takes an integer parameter and returns the *grid* element with grid index (the parameter) if the parameter is within the valid range [0-41]; otherwise, it returns an asterisk.
  - ☐ a public Boolean constant method named `full()` that takes no parameters and returns true if *grid* contains no blank characters (asterisks); otherwise, it returns false.
  - ☐ a public Boolean constant method named `empty()` that takes no parameters and returns true if *grid* contains only blank characters (asterisks); otherwise, it returns false.
  - ☐ a public Boolean constant method named `space()` that takes an integer parameter and returns true if the column index (the parameter) is within the valid range [0–6] and the column has available space; otherwise, it returns false.
  - ☐ a public void method named `next()` that takes no parameters and switches the turn to the next player.
  - ☐ a public char constant method named `disc()` that takes no parameters and returns the disc of the current player.
  - ☐ a public Boolean method named `set()` that takes an integer parameter. It inserts the current player's disc from *discs* into *grid* and returns true if the column index (the parameter) is within the valid range [0–6] and the column has available space. Otherwise, it returns false.
  - ☐ a public void method named `reset()` that takes no parameters and assigns a string of 42 asterisks and 0 to *grid* and *player*, respectively.
  - ☐ a public string constant method named `toString()` that takes no parameters and returns a string of *grid* in the same format as the example in the above instructions.
  - ☐ a friend ostream operator that produces the same display as the `toString()` method.

Extra Credit

- Create a cpp file named '`extra.cpp`' that defines
  - ☐ a Boolean function named `ColumnWin()` that takes a constant *C4Board* reference parameter and returns true if the board (the parameter) has a winner in a column; otherwise, it returns false.
  - ☐ a Boolean function named `RowWin()` that takes a constant *C4Board* reference parameter and returns true if the board (the parameter) has a winner in a row; otherwise, it returns false.