LAB #10 – Object Oriented Programming

Pre-Lab:
For those of you who didn’t finish lab #9, finish this lab and make sure you understand it!!!

OOP Understanding:
As a part of this lab, you will practice making a class.

You will learn to write an object-oriented program (OOP) with a class acting as a blueprint for playing Tic-Tac-Toe. In this program, you will need to think about the states and behavior for playing tic-tac-toe. You will write the TicTacToe class and create an instance of the class, i.e. a TicTacToe object. Write a TicTacToe class containing the constructors, member variables, and member functions for playing tic-tac-toe.

1. First, create a tictactoe.cpp file containing the proper states for playing the game. For example, you will need two players and a board.
   
   class TicTacToe {
   public:
       char board[3][3];
       char player1;
       char player2;
       ...
   };

2. Now, create two constructors, one which is a default constructor, TicTacToe(), and another constructor that takes the input for player 1 and player 2 choices as its parameters, TicTacToe(char p1, char p2). As you can see in this example, you will need to have some method for initializing the board to spaces to start, and you will need to fill in the details of the TicTacToe constructors.
   
   TicTacToe::TicTacToe() {
       player1='x';
       player2='o';
       initializeBoard();
       ...
   }

   TicTacToe::TicTacToe(char p1, char p2) {
       ...
   }

3. Now, you need to make the board and player pieces private to provide encapsulation. Once these are made private, write accessor and mutator functions to get and set these member variables, i.e. setPlayer1(char p1), setPlayer2(char p2), getPlayer1(), getPlayer2(), etc.
4. Now, declare other member functions for the tic-tac-toe that allow the players to pick their pieces, i.e. ‘X’ or ‘O’, `determinePlayerChoice()`, the board to be filled with the players’ choice, `fillBoard()`, the board to be printed to the screen after each user’s turn, `printBoard()`, the board to be checked if it is full, `isFull()`, a check to determine a winner, `checkForWinner()`, and the results of the game to be printed to the screen, `printWinnerResults()`. Determine which of these functions need to be private and which will be public.

Implement the `determinePlayerChoice()`, `fillBoard()`, and `printBoard()` member functions.

5. After getting the class setup to use, then create a TicTacToe object, determine which piece each player wants to be, i.e. ‘X’ vs. ‘O’, print the board, ask player 1 to choose a spot on the board, print the board again, ask player 2 to choose a spot, and then print the board again.

**Extras:**
You can finish implementing Tic-Tac-Toe, including handling errors such as the user entering an invalid spot on the board, e.g. 5 2, and selecting a spot already taken!!!

**Show your program to a lab TA for lab credit, and enjoy your winter break!!!**