

Lab 12 - CMPS 1044 Computer Science I Two-Dimensional Arrays

Objective: Demonstrate understanding and use of 2D arrays.

2D array: An array of data organized into rows and columns, like a table.

Examples of declarations:

```
int scores[10][15]; // Ten rows of 15 scores each
char gameboard[3][3]; // 3 rows of characters with 3 in each row
double temps[7][24]; // holds hourly temperatures for 7 days
```

Important concepts

- The first number indicates rows, 2nd number is number of columns
 - Indexing starts with zero
 - Input, output, and traversing of 2D arrays is usually accomplished with nested loops
1. Download the file "**ticLab12.txt**". It contains 3 x 3 representations of tic-tac-toe boards. Every space is filled with O, X, or a period (.) indicating an open space. Also, download the file **tictactoeLab12.cpp**. Create a new project using tictactoeLab12.cpp as your source file. Make sure ticLab12.txt is in the same directory. Build and run the project. Verify that the values in the input file are correctly displayed as a tic-tac-toe card on the screen.

For the rest of the lab, you will work with the GA to incrementally complete the program so that for each game card read in, you will determine if there is a winner, and if so, who won.

2. How would you determine whether 'X' has won the game? If there is an 'X' in every spot on any row, column, or diagonal, then 'X' won the game. To see if all the entries on the top row are X's, you could do this:

```
if ((board[0][0] == 'X') && (board[0][1] == 'X') && (board[0][2] == 'X'))
{
    winnerFound = true;
    winner = 'X';
}
```

But that only checks the top row. Modify the code so that it will check **every** row using a loop. Build/run and verify that for the **first** card you get the message: **Game won by X**. Since there are all X's on the bottom row.

3. Now extend the code so that you also check **every column** to see if 'X' is the winner. Build/run and verify that for the **2nd** card you get the message: **Game won by X**. Since there are all X's in the first column.

4. Now extend the code so that you also check **the major diagonal** to see if 'X' is the winner. Build/run and verify that for the **3rd** card you get the message: **Game won by X**. Since there are all X's along the major diagonal.
5. Now extend the code so that you also check **the minor diagonal** to see if 'X' is the winner. Build/run and verify that for the **4th** card you get the message: **Game won by X**. Since there are all X's along the minor diagonal.
6. What about O??? Now we need to check to see if O won the game. You can write a loop that will loop from the first element to the last element of the token array. Put it around the other loops you have written. Build/run and verify that for the **5th through 8th** cards, you get the message: **Game won by O**.
7. What if there's no winner? Then we say the Cat won the game. To print an appropriate message if there is no winner, simply add another if statement at the end. After you complete this step, Build/run and verify that for **the 9th and 10th** cards, you get the message: **Cat won the game**.
8. Once your code displays all ten games correctly, change your `cout` statements to print to a file. Include a full header at the top of your program and print that header at the top of the file containing your results. Turn in a print out of your source code and output file.