**CPSC1012 Advanced Portfolio 2 – Multidimensional Arrays**

**Weight: 5% of your final mark**

**Tic-Tac-Toe Game**

In a game of tic-tac-toe, two players take turns marking an available cell in a 3 x 3 grid with their respective tokens (either X or O). When one player has placed three tokens in a horizontal, vertical, or diagonal row on the grid, the game is over, and that player has won. A draw (no winner) occurs when all the cells on the grid have been filled with tokens and neither player has achieved a win. Create a program for playing a tic-tac-toe game.

The program prompts the two players to alternately enter an X token and O token. Whenever a token is entered, the program redisplays the board on the console and determines the status of the game (win, draw, or continue). Here is a sample run:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Tie-Tac-Toe Game \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The cell numbers for the game is shown below.

-------------

| 7 | 8 | 9 |

-------------

| 4 | 5 | 6 |

-------------

| 1 | 2 | 3 |

-------------

Enter cell number (1-9) for player X: 5

-------------

| | | |

-------------

| | X | |

-------------

| | | |

-------------

Enter cell number (1-9) for player O: 7

-------------

| O | | |

-------------

| | X | |

-------------

| | | |

-------------

Enter cell number (1-9) for player X:

...

-------------

| O | | X |

-------------

| O | X | |

-------------

| X | | |

-------------

Player X wins!

Would you like to play again (y/n)? n

Good-bye and thanks for playing.

**Marking Guide**

|  |  |  |
| --- | --- | --- |
| **Description** | **Marks Possible** | **Marks Earned** |
| Correctness   * Correct cell number is displayed on board * Logic to determine winner * Logic to determine tie game * Logic to determine if cell is already taken * Can play again and board is cleared | 5 |  |
| Structure   * Code to create and initialize array of cells * Code to display empty tic-tac-toe board * Code to display board cell selection | 3 |  |
| Style and Readability   * Horizontal and vertical white space * Meaningful identifiers | 1 |  |
| Documentation   * Opening documentation * Source code comments | 1 |  |
| **Total:** | **10** |  |

**Coding Requirements**

The following coding standards must be followed when developing your program:

* Your C# Console App project must be named as **AdvancedPortfolio02-*YourFullName*** (eg: AdvancedPortfolio02-CodeGuru)
* Opening documentation at the beginning of the source file describing the **purpose**, **input**, **process**, **output, author, last modified date** of the program.
* Write only one statement per line.
* Write only one declaration per line.
* Use camelCase for local variable names and method parameter name.
* Use PascalCase for method names and constant variable names.
* If continuation lines are not indented automatically, indent them one tab stop (four spaces).
* Do NOT use the goto statement.
* There can only be one exit point for a loop, do not use the break statement inside a loop
* Do NOT use static variables.

**Demonstration and Submission Requirements**

* Demonstrate to your instructor your working program before submitting to Moodle. Be prepared to answer questions about your code after the demonstration. **No marks will be given** if you are unable to explain your code or if you submit your project without a demonstration of your working program to your instructor.