CS202: PROGRAMMING PARADIGMS & PRAGMATICS

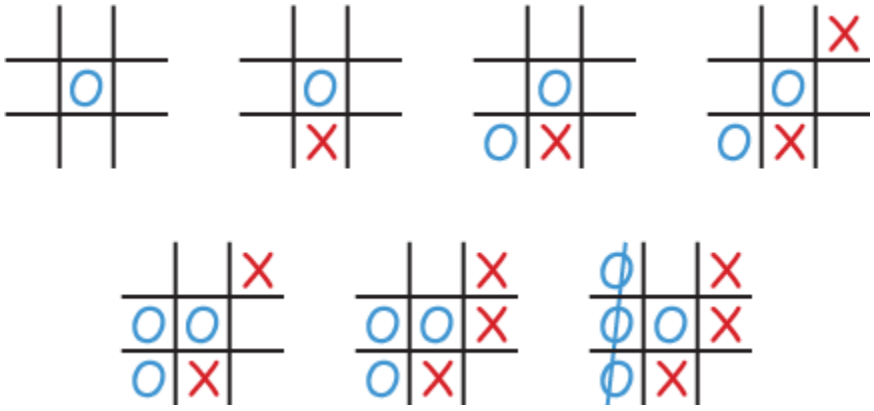
Semester II, 2020 – 2021

Lab 2: Project – Tic-Tac-Toe!

Aim: Write a program to handle the basics of a two-player game of Tic-Tac-Toe.

Requirement: Design the project using OOP in Java!

- **Let's get started!**
  - Create a directory structure to hold your work for this course and all the subsequent labs:
    - Suggestion: `CS202/Lab2`
- **Introduction**
  - Tic-tac-toe is a paper-&-pencil game for two players, X and O, who take turns marking the spaces in a 3×3 grid.
  - The player who succeeds in placing three respective marks in a **horizontal**, **vertical**, or **diagonal** row wins the game.
  - The following example game is won by the second player, O:



- **Game Simplified:**
  - Need to design two versions of the game: Two players or Player Vs. Computer
    - The player always moves first and is X. The computer is O.
    - The only tricky part about the game is determining if a given person has won.
  - Play continues until one player has scored three in a row or all squares have been filled with no winner.
  - The players can play in only those squares that are not already occupied by either player.
  - The program ends when the game is won by either player or the game is a draw.
- **Suggestions:**
  - Whenever/wherever possible, your program should have classes that imitate real world objects
  - Interaction between classes should only occur using the methods!
  - Use Scanner class to Input the user's selection of letters.
  - **First get a simple two player version of your game correct!! Then try the Computer Vs. Player version.**
  - Your main function should be in the TicTacToe class

- **Examples (SAMPLES ONLY – NOT REQUIRED TO REPLICATE!):**

```
Welcome to Tic Tac Toe. Player is X, computer is O.
This is a new game. Board numbers are as follows:
  1 |  2 |  3
  4 |  5 |  6
  7 |  8 |  9
Enter your desired location [1-9] : 5
The computer picked: 3
    |    | O
    |  X |
    |    |
Enter your desired location [1-9]: 8
The computer picked: 2
    |  O |  O
    |  X |
    |  X |
Enter your desired location [1-9]: 1
The computer picked: 7
  X |  O |  O
    |  X |
  O |  X |
Enter your desired location [1-9]: 9
You win!
  X |  O |  O
    |  X |
  O |  X |  X
New game [Y/N]? :
```

```
        Tic Tac Toe
Player 1 (X)  -  Player 2 (O)

  X  |  2  |  3
_____|_____|_____
  4  |  O  |  6
_____|_____|_____
  X  |  8  |  9
     |     |

Player 2, enter a number:
```

```
This is the game of Tic Tac Toe.
You will be playing against the computer.
   |   |
---|---|---
   |   |
---|---|---
   |   |
Enter X,Y coordinates for your move: 1,1
 X | O |
---|---|---
   |   |
---|---|---
   |   |
Enter X,Y coordinates for your move: 2,2
 X | O | O
---|---|---
   | X |
---|---|---
   |   |
Enter X,Y coordinates for your move: 3,3
You won!
 X | O | O
---|---|---
   | X |
---|---|---
   |   | X
```

```
Turn 6
It's turn of Player 2
Line: 2
Column: 3

 X |   | X
 O | O | O
   |   | X
Player 2 won!
BUILD SUCCESSFUL (total time: 34 seconds)
```

```
(...a game already in progress)

      X   O
      O X X
        X O

'O', choose your location (row, column): 0 1

      X O O
      O X X
        X O

'X', choose your location (row, column): 2 0

      X O O
      O X X
      X X O

The game is a tie.
```

- **Submitting your work:**
    - All source files and class files as one tar-gzipped archive.
        - When unzipped, it should create a directory with your ID. Example: **2008CSB1001** (NO OTHER FORMAT IS ACCEPTABLE!!! Case sensitive!!!)
    - Source files should include the following: (Case-Sensitive file names!!)
        - `TicTacToe.java / .class`
        - Any other supporting or required files
    - ***Negative marks for any problems/errors in running your programs***
    - *If any aspect of the game / rules are confusing, make an assumption and state it clearly in your* **README** *file!*
    - **README** file should also have instructions on how to use/run your program!
    - Submit/Upload to Google Classroom
    - Marks Allocation
        - 2-Player Version [10 points]
        - Computer vs. Player Version [10 points]
        - Coding Style [2 points]
        - README [3 points]