

CmpE 150 Introduction to Computing, Fall 2018

Project 2—Due: 07/12/2018, 23:00

You will write an interactive Java program to play the tic-tac-toe game. The board size is 4 x 4 and is made of cells that contain E symbols, X's or O's. Below is the initial configuration of the board:

```
| E | E | E | E |
| E | E | E | E |
| E | E | E | E |
| E | E | E | E |
```

Empty cells are denoted with the E symbol. Each cell of the board can be identified with coordinates as first coordinate is the row number and second coordinate is the column number. The cell in the top left corner has coordinate (1, 1) whereas the lower right corner has coordinates (4, 4). For example, bottom left corner has coordinate (4, 1). When printing the board to the screen, there should be only one space before and after the symbols.

There are two players in the game. One player will be the user and the other player will be the computer. Firstly, user will be asked if he/she wants to load the board from a file or create a new empty board. If the user selects to load, then the user will enter the file name where the board configuration is written and the state of the board will be read from that file. Then, program will ask the user if he/she wants to play as 'X' or 'O' and the computer will play as the opposite side. Choosing the starting player will be decided **randomly**. Game will continue until one player wins. Winning conditions are having **three** consecutive X's or O's horizontally, vertically, or diagonally. The game may also end in a tie, which is the condition that neither of the players win. When the game ends, you should ask the user if he/she wants to start over and continue playing. If the user decides to continue, then the game starts from the beginning (skipping the load file/create new board question). If the user decides to end the game, then the number of games the user won and the number of games the computer won should be printed. (Ties should be ignored). See the example run below:

Welcome to the XOX Game.

Would you like to load the board from file or create a new one? (L or C) C

Enter your symbol: (X or O) X

You are player X and the computer is player O.

You will start:

```
| E | E | E | E |
| E | E | E | E |
| E | E | E | E |
| E | E | E | E |
```

Enter coordinates: 1 2

```
| E | X | E | E |
| E | E | E | E |
| E | E | E | O |
| E | E | E | E |
```

Enter coordinates: 2 2

| E | X | E | E |

| E | X | E | E |

| E | O | E | O |

| E | E | E | E |

Enter coordinates: 3 3

| E | X | E | E |

| E | X | O | E |

| E | O | X | O |

| E | E | E | E |

Enter coordinates: 4 4

| E | X | E | E |

| E | X | O | E |

| E | O | X | O |

| E | E | E | X |

You win! Do you want to play again? (Y or N) Y

Computer will start:

| E | E | E | E |

| E | E | E | E |

| E | E | O | E |

| E | E | E | E |

Enter coordinates: 2 2

| E | E | E | E |

| E | X | E | E |

| E | O | O | E |

| E | E | E | E |

Enter coordinates: 2 3

| E | E | E | E |

| E | X | X | E |

| O | O | O | E |

| E | E | E | E |

Computer wins! Do you want to play again? (Y or N) N

You: 1 Computer: 1

Thanks for playing!

Also, if the user enters an invalid coordinate, which is outside the boundaries of the board or a non-empty cell in the board, program should warn the user and the user should enter coordinates again.

For example:

.....

.....

.....

| E | X | E | E |

| E | X | E | E |

| E | O | E | O |

| E | E | E | E |

Enter coordinates: 3 2

Wrong input! Try again: 3 3

| E | X | E | E |

| E | X | O | E |

| E | O | X | O |

| E | E | E | E |

.....

.....

.....

Example Run 2 (with loading from file):

Welcome to the XOX Game.

Would you like to load the board from file or create a new one? (L or C) L

Please enter the file name: board.txt

Load successful.

Enter your symbol: (X or O) O

You are player O and the computer is player X.

| E | E | E | O |

| E | X | E | E |

| E | O | E | E |

| E | X | E | E |

Enter coordinates:

.....

.....

.....

Bonus points: You will get bonus points if the computer plays competitively. It means that computer should make logical decisions and should have a good chance to win or the game always ends in a tie.

Please make sure you follow these rules in your implementation:

1. Your program should have at least two static methods in addition to your main method. Try

to write your program as modular as possible (without overusing methods).

2. You are not allowed to use statements that we have not learned in class as of **25/11/18 (such as arrays, recursion, and so on)**.
3. Your program **should not give runtime errors in any situation**.
4. You are **not allowed** to use **Arrays** for this assignment. You can use the String methods.

Submission: You will submit a project report and your code over Moodle.

Project report should consist of five sections. These are:

1. Problem Description: In this section, you should describe the problem in your words.
2. Problem Solution: In this section, you should specify the concepts (methods, for loop, etc.) that you use in your program. Explain each one (i.e. why you need it, what you accomplish by using it, so on.). Report how many for loops you use.
3. Implementation: This section will include your whole code with comments. You need to pay attention to indentation in order to improve readability.
 - Do not forget to explain each variable that you use (i.e. `int count=0; // count is the number of items`).
 - Before each method, specify what the method does (i.e. `/E This method . E/`)
4. Output of the program: A screen-shot of your program output should be put in this section. Two example runs are enough.
5. Conclusion: You should evaluate your work here. State whether you have solved the problem correctly. If not, state what is missing, what could have been improved, and so on.

Your .java file should be named with your initials and your student number together (e.g., OS2013800027). If you have Turkish characters as your initials, please change them to non-Turkish. (Example: `ÖS2013800027` should be `OS2013800027`) You will submit these over Moodle as a single zip file where the file name is your student number. Your zip file should consist of your .java file and your report in .doc or .pdf format. **Do not use any Turkish characters in your code, class/variable names, or .java file names.**

Partial Submission: If you cannot do everything you are asked in this project, you should still submit your code as well as your report. Try to do as much as you can. In your report, explain which parts you can implement and which parts you cannot.

Late Submission: Late submission formula is $-10x^2$ where x is the number of days of late submission.