CSC 1103 – Object Oriented Programming | INDIVIDUAL ASSIGNMENT | Semester 2, 2019/2020

1. INSTRUCTIONS:

- Name, Matric number & Section should be included in the comments of the program file/s, at the beginning.
- Submit your files by uploading them to the google classroom.
- All files must be submitted as .java or .txt.
- This is an **exploratory assignment**. You are free to refer to any online/offline resources as long as you don't plagiarise and refer to your sources. Any form of plagiarism is highly unacceptable.
- **Assignment bears 10 Marks (10% of your Total Course Evaluation).**

2. IMPORTANT DATES

SUBMISSION DEADLINE	**12th July 2020, 11: 50 PM**

- Any assignment submitted by **09th July 2020, 08:00 PM** will be given a **bonus** of 2 Marks. The bonus will be given, if the assignment scores at least **7 Marks** upon checking.

3. EXERCISES

Exercise 1: The Eight Queens problem is to find a solution to place a queen in each row on a chessboard such that no two queens can attack each other. You can use a two-dimensional array to represent a chessboard. However, since each row can have only one queen, it is sufficient to use a one-dimensional array to denote the position of the queen in the row. Thus, you can define the **queens** array as:

```
int[] queens = new int[8];
```

Assign **j** to **queens[i]** to denote that a queen is placed in row **i** and column **j**.

Define a class named EQ to represent eight queens in a chess board with the following members:

1. A private data field queens of the type int[].
2. A no-arg constructor that constructs an object of EQ default queens values of -1s in the array.

3. A constructor named EQ(int[] queens) that constructs an object of EQ with the specified queen placement.
4. A method named get(int i) that returns queens[i].
5. A method named set(int i, int j) that sets queens[i] with j.
6. A method named isSolved() that returns true if all queens are placed in the board correctly.
7. A method named printBoard() that displays the board with the queens like the following:

```
|X| | | | | | | |
| | | | |X| | | |
| | | | | | | |X|
| | | | | |X| | |
| | |X| | | | | |
| | | | | | |X| |
| |X| | | | | | |
| | | |X| | | | |
```

NOTE: DO NOT USE BACKTRACKING FOR THIS EXERCISE

Implement the class and use the following program to test your class.

```
 public static void main(String[] args) {
    EQ board1 = new EQ();
    board1.set(0, 2);
    board1.set(1, 4);
    board1.set(2, 7);
    board1.set(3, 1);
    board1.set(4, 0);
    board1.set(5, 3);
    board1.set(6, 6);
    board1.set(7, 5);
    System.out.println("Is board 1 a correct eight queen placement?
"
        + board1.isSolved());

    EQ board2 = new EQ(new int[] { 0, 4, 7, 5, 2, 6, 1, 3 });
    if (board2.isSolved()) {
      System.out.println("Eight queens are placed correctly "
        + "in board 2");
      board2.printBoard();
    }
    else {
      System.out.println("Eight queens are placed incorrectly "
        + "in board 2"); } }
```

Exercise 2: Arguments for the main method are passed as strings. Strings enclosed in quotation marks are considered as one argument. Write a program to parse arguments from a string. Arguments are separated by spaces. Enclosed strings are considered as one argument. Your program should prompt the user to enter a string and display the arguments, each per line. [Set the name of the class as *parsec* with main()]

*****END OF DOCUMENT*****