**C++ 2018B - MTA - Exercises**

**Requirements and Guidelines**

The exercises in the course would require you to implement an advanced [rock–paper–scissors](https://en.wikipedia.org/wiki/Rock%E2%80%93paper%E2%80%93scissors) game.

Note: you will get submission and environment instructions separately. In short: the exercise should be implemented in Visual Studio 2015 or later, with standard C++ libraries and run on Windows with Console screen of standard size (80\*25), using gotoxy for printing in (X, Y).

**Exercise 2**

Topics: inheritance, polymorphism and operators overloading

In this exercise, you should support input from the console as well as from the file. Meaning, all input that was given in the file in the first exercise, should also be inserted from the console:

1- No need to support a file vs. console game, only console vs. console and keep supporting the file option.

2- In the console mode first the user would be asked what are the dimensions of the board and how many tools of each type exists.

Then the game would ask each player to position the pieces in the board: one piece for each player each round (player 1 positions one piece, then player 2, then player 1 again etc.).

3- The input from the console should ask for the piece type and the coordinates to locate the piece on the board.

4- When positioning a Joker, user would be asked to give the initial representation of the Joker.

5-Moves: the user would be asked in his turn which tool to move, and to where. Then he would be asked whether he wants to change Joker representation (only if there is at least one Joker alive) and if he wants to, which Joker (only if there is more than 1) and to which representation.

6-Game would present the data to screen as known to the player who is currently playing (we assume that two players are playing and each time one of them closes his eyes).

6- No need to create output file in console mode.

The decision whether to get input from the configuration file or from the console should be passed as a command line parameter: -console would mean console game

Code requirements:

1. Use inheritance where it allows to reuse code.
2. In order to practice operators overloading, implement in the exercise 3 operators of your choice that would be used in your code. For example: operator< can be implemented for the pieces, by their strength.

Note: (a) the operators should be intuitive - don’t implement operator! for moving a piece  
 (b) operator= is not counted in this request for 3 operators