# CST8110 Practical Assessment #2 [15%]

**This assignment relates to the following Course Learning Requirements:**

CLR 1: Define, analyze, and document the logic of a solution to a given problem.

CLR 2: Implement the solution to a given problem by writing the appropriate code in a high-level language (Java).

CLR 4: Install and use the Java Development, Runtime Environment, and documentation libraries.

CLR 7: Create documentation and a Java solution for programming problems that adhere to the submission standard identified within the timeframe given in the problem description.

Objective of this Assignment:

The following is an exercise to help you understand what you have learned in this module. It will allow you to familiarize yourself with the module content as well as practice the skills required to develop software using the Java programming language. The objective of this practical assignment is to design and implement a basic version of the “Connect Four” board game.

Connect Four is a two-player connection game in which the players first choose a color and then take turns dropping one colored disc from the top into a seven-column, six-row vertically suspended grid. The pieces fall straight down, occupying the lowest available space within the column. The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of four of one's own discs.

More information about the game can be found here: <https://en.wikipedia.org/wiki/Connect_Four>.

There are several ways to design the logic for this assignment, however if the logic receives the correct result, it is correct!

# Pre-Assignment Instructions:

1. To prepare you for this assignment, please complete the content associated to this module.
2. Complete all module reading and learning activities
3. Complete all reference video tutorials
4. Access this assignment in Brightspace. Download and unzip the starter code files provided. The instructions for requirements are within the files as guidelines.
5. Develop this program locally on your machine in a plain text editor or other IDE.

**Assignment Tasks/Steps:**

1. Download the ZIP file from Brightspace containing starter code for your Practical Assessment
2. Extract the Java starter code files and import them into a new Project in Eclipse
   1. Most of your effort will be around updating the program using the comments within the starter code files as guidelines
      1. Use these as instructions for the requirements
   2. Please take note that some comments will appear in your Eclipse Tasks list (// TODO’s enable this)

**Submit Your Assignment**

Your assignment is to be submitted on Brightspace as Java files. It should be submitted with the following guidelines:

1. Include the file header using the template provided in Assignment 1, in every java file submitted
2. Follow expected style guidelines:
   * 1. Use “Egyptian” style braces for all classes and methods
     2. Indent your code using 4 spaces (no tabs)
     3. Follow naming conventions for all class, variable, and method identifiers
     4. Use appropriate whitespace for readability
     5. Comment your code (no less than one comment per class, and one per method)
3. Files must be named  as **Ordinal.java, Column.java, Board.java, Player.java, NibbleNabble.java, Main.java**
4. Method names, return types, and parameters must match those in the requirements of the guidelines where applicable.

Feel free to make any assumption you need to make to implement a working code. Make sure the code delivers expected successful output. Highlight any assumption you may make while submitting the assessment.

**Assignment Grading Rubric (5%)**

|  |  |
| --- | --- |
| Criteria / Marks | Points |
| All programs compile | /2 |
| All programs execute and output as expected from the updates performed by the learner  (i.e.: refer to section #2 to verify output from the requirements) | /30 |
| Header file is inserted at top of file and all variables and place holder and information replaced with learner’s information | /2 |
| The style guidelines are followed:  i. Use “Egyptian” style braces for all classes and methods  ii. Indent your code using 4 spaces (no tabs)  iii. Follow naming conventions for all class, variable, and method identifiers  iv. Use appropriate whitespace for readability  v. Comment your code (no less than one comment per class, and one comment per method) | /10 |
| Method names match those in the requirements exactly | /4 |
| Assignment uploaded / pushed | /2 |
| Comments |  |
| Total Points | /50 |