**Foundations of Computer Science Lab 6**

We want to build a connect 4 game that runs in our command line interface (terminal). We have just recently learned about 2D Lists and how to manipulate them. Additionally, we learned how to score a tic tac toe game for a 3x3 grid. The natural extension of this is connect 4.

In this lab you are given four function headers:

* drawBoard(): prints out the connect 4 board in a nice, clean way.
* switchPlayer(): Switches the current player from X to O or O to X
* dropPiece(): Drops the piece in the specified column
* checkWinner(): Checks if the current player won with either a horizontal victory, vertical victory, or left/right diagonal victories.

You are to implement these four functions in the lab template provided at [this link.](https://github.com/cookiebouquets/CHC-275-class-notes/blob/main/Lab%20Templates/lab_6_template.py)

Some hints:

* You need to be very careful with bounds on your Board. A player should not be able to drop a piece into a column that is already filled
* Python will throw a runtime error if you are not careful with your indices on your game board.
* **For loops are your friend**

Lab Writeup is required. Main explanatory points include:

* How you determined what indices to check for in checkWinner()
* How you got drawBoard() to look clean.
* How you got dropPiece() to check if a column is full.

Examples of core functions on next page:

**Draw Board**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Drop Piece**

**A black screen with white text

AI-generated content may be incorrect.**

**Check Win (Vertical)**

**A black background with white text

AI-generated content may be incorrect.**

**Check Win (Horizontal)**

**A screen shot of a computer screen

AI-generated content may be incorrect.**

**Check Win (Diagonal)**

**A black screen with white text

AI-generated content may be incorrect.**