**CST-201**

**Project 4: Maze Solver**

The purpose of this assignment is to assess your ability to:

* Implement stack and queue abstract data types
* Utilize stack and queue structures in a computational problem.

*This assignment reinforces competency 6.1: Select and utilize data structures appropriate to a given computational problem.*

For this project, implement a stack and a queue data structure. After testing the class, complete the depth-first search method (provided). The method should indicate whether or not a path was found, and, in the case that a path is found, output the sequence of coordinates from start to end.

The following code and related files are provided. Carefully ready the code and documentation before beginning.

* A MazeCell class that models a cell in the maze. Each MazeCell object contains data for the row and column position of the cell, the next direction to check, and a bool variable that indicates whether or not the cell has already been visited.
* A Source file that creates the maze and calls your depth-first search method.
* An input file, maze.in, that may be used for testing.

You will write the following:

* a generic stack class, called MyStack, that supports the provided API (see file stackAPI.png)
* a generic queue class, called MyQueue, that supports the provided API (see file queueAPI.png)
* an implementation for the depth-first algorithm

Create a Loom video with a length of 5 minutes or less in which you run your code and comment on the following:

* Your data structure choices for the stack and queue implementation.
* Your use of MyStack and MyQueue structures in your maze solution
* Your depth first algorithm

Submit the following:

* A zip file with your code
* A link to your Loom video in a separate text file

Review the rubric for this assignment to ensure that you have addressed all assignment criteria.