**CS 210: Algorithms and Data Structures   
Maze Project (100 points) – Part 2 Of 3  
Due: Thursday, March 6, 2014**

**Maze-o-Rama:**

All classes for this project are in package maze. You are welcome to use your previous codes as well as incorporating Java library and of course writing several new ones. Note that this project does not explicitly list all of the classes you'll have to write---part of that will be design decisions left up to you. In addition, you will now be working in a team.

**The maze itself**

As agreed there is a **MazeReader** class that stores the maze from a text file. Before you get to the solving part, this project requires a Maze that can store enumerated Square values. You may assume that any well-formed maze will have exactly one start and zero or many exits. You may assume that all valid mazes will be entirely enclosed with walls (unless the exit is in the outer wall).

**The maze solver**

The main component of the project will be the writing of a **MazeSolver** class (and all other associated classes), which will bundle up the functionality of determining whether a maze has a solution---that is, whether you can get from the start to the finish (without jumping over any walls). The algorithm usually goes something like this: start at the start location, and trace along all possible paths to (eventually) all reachable open squares. If at some point you run across the finish, it was reachable. If not, it wasn't.

**Project Requirement**:

In creating the maze solver, your team will be required to ensure that the algorithm is capable of handling the type of maze presented in Module 1. You are allowed to traverse the path multiple times as long as the final decision is accurate. Your algorithm also need to track the number of spaces explored before coming to the conclusion and the number of spaces needed to get to the exit.

**Individual Assessment**: To avoid potentially uneven workload distribution, each team member must complete an individual assessment every week. The individual assessment will require each member to discuss their workload and the progress of the project.

**Assessment Due**: Personal assessment will be due at the end of the project.

**Submission**: \\citecs\cs210\yourFolder.