CSIS 211 – Data Structures
Spring 2017
Lab #4 – Due Date: 05/08/17 at 11:55 pm
Remember to submit ALL files (.cpp,.h etc.) on Moodle – NO EMAILS – no exceptions
You may work in a group of 1-4 for this assignment
EACH group member should submit just to make sure I don't miss any while grading
IMPORTANT! – put EACH group member's name at the TOP of the CODE in comments
If you are a group of 1 put your name only at the top
Make sure and use good design techniques
This will be worth 50 points

For all Programs:
Do NOT forget to put ALL your groups names at the TOP of EACH FILE in comments.
Example:
//LAB 4
//Group: Kristina Shroyer, John Smith, Jane Doe
Do this EVEN if you have a group of 1
Since this is very important

Lab #4 – Implement the Eight Queens Problem

This is an object oriented program. This is #1 on page 187 of your book (the only difference is I'm requiring you test your code with a client program (a main).

On page 179 of your book is a function placeQueens that solves the Eight Queens problem.
On page 180 of your book are some suggestions to implementing two ADTs to solve the problem in C++. I want you to follow their suggestions to implement the solution.

1. **First create a `Board` class data type.** This class data type should represent the chess board (you can use any implementation you want, a two dimensional array (however this may waste space but not if you're displaying in my opinion), a vector (it could represent only the board places occupied by queens), an array of vectors etc – there is more than one right way!). The board should keep track of the Queens currently placed on it and contain operations (such as placeQueens and displayBoard) to solve the problem and display a solution. You can add other methods as you need them. Note the board will need to contain a Queen (or perhaps an array or vector of queens) – I haven't done this problem yet but I immediately thought composition might be needed.

2. **Next create a Queen data type.** A queen instance should be able to keep track of its current position (row and column) and be able to move to the next row. Again you can add other methods as you see fit.

3. **Add a client program to test your code:**
    - This program should place the first queen anywhere on the board and then call placeQueens to see if a solution can be reached. If the solution can be reached the program should display the board with the queens properly placed.