**Data Structures (2080C) – Lab02**

***Topics covered: Stacks and Queues***

**Objective:**

The objective of this Lab is to examine array based stacks and queues built using C++. We are going to create a stack and a queue to be used in the Towers of Hanoi game.

Read the introduction to the game at <https://en.wikipedia.org/wiki/Tower_of_Hanoi>.

Task 0: Make the additions required to get the starter project to compile.

**Task 1:** Create a stack that will be used as the basis for the remainder of the lab.

1. Design a stack using an array, using methods defined in the inteface.

**Task 2:** Create a queue to track steps.

1. Create a queue. Design the queue class using an array to store data. Include the standard functions for a queue specified in the header file.

**Task 3:** Create a single person version of the “Towers of Hanoi” game.

Four disks, three towers

Modify the code from Tasks 1 and 2 to store any moves made in the queue class. When the game is won, the queue should be used to print out all the moves made in order. Store the values as strings in this format - <disk>,<column from>, <column to>

1. There will be four disks, and three poles. Represent the disk by a number, 1 for smallest disk, and so on ….
2. It is expected that the students will create their own class to represent the Towers of Hanoi game. Hint use the stack to represent the towers and keep track of moves in the Queue. (you can even use that queue to generate a tests, **by outputting it to a file**)
3. Using the queue output at the end of the execution to a file (say temp.txt) all the moves. This can be then used to test your program as well (see note below about indirection).
4. Create a main program that uses stacks to allows the user to interactively play the game.
   1. It needs to provide a way for the user to indicate the tower to move from and to.
   2. It needs to prevent illegal moves as defined in the Wikipedia rules.
   3. It needs to provide the user with a way to see their progress on the screen.

Note : it is easier to output this horizontally.

<start>

Tower 1: 4 3 2 1

Tower 2:

Tower 3:

<some moves>

Tower 1: 4 3

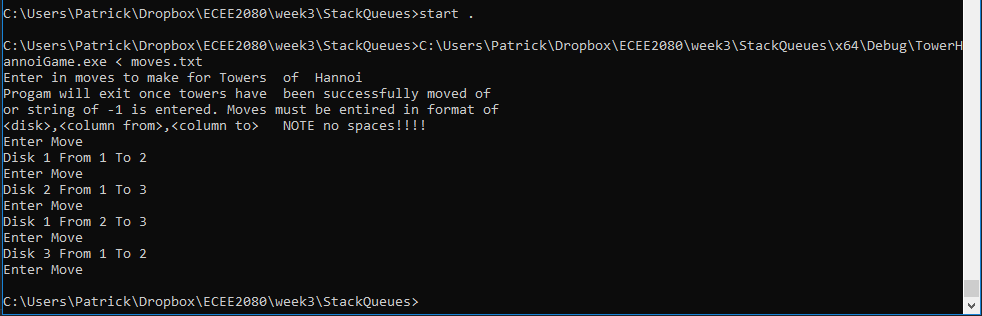
Tower 2: 1

Tower 3: 2

1. Test your program to ensure it is working correctly. This should test both expected success conditions and expected error conditions.
2. You program should be able to read input from std in, by using indirection. For example:

ucstudent@ucstudent-VirtualBox:~/tt/Lab02$ **build/runMain < moves.txt**

(as discussed in class)



1. Include in the lab report a screen shot(s) of the output of your tests.

**Lab Submission:**

* lab report doc/pdf of screenshots of the Hanoi game running, show errors being handled and game being done to completion (don’t need to see exhaustive moves, just good sample)
* main.cpp
* StackQueues.cpp
* StackQueues.h
* moves.txt (this should be the a solution of the game from start to end)

**Lab Grading:**

1. 20% - Lab attendance
2. 15% - Task 1 has been correctly implemented and meets all requirements (Unit Test Passes).
3. 20% - Task 2 has been correctly implemented and meets all requirements (Unit Test Passes).
4. 25% - Task 3 has been correctly implemented and meets all requirements.
5. 20% - Code style

If program fails to compile, 0% will be given for that Task.