**Lab 5 – Queues and Simulations**

**CSC 3302**

|  |  |
| --- | --- |
|  |  |
| **Introduction:** | Queues are useful for running simulations. Simulations allow us to gather data on real world events. This usually is a result of time constraints or event driven data. With our simulations we can get the data of many events and condense it to a short time. Most of the time with events we monitor the actions as they happen. A queue is the perfect data structure for monitoring such events. By placing the events in the queue we can maintain the order they happen while other events are being processed. |
|  |  |
| **Description:** | The students in a class would really like it if Professor NotHearingIt would move an exam back a week. None of the students wish to confront the professor and so decided to get in a circle and remove every Nth student until only one is left. This student is the one who must go ask for the extension.  One student, **Joseph**, does not actually wish to move the test back. At the same time he does not want the other students to know that he is the only one who feels this way. He does not know what number N will be decided for deciding speaker until the process begins. He wants to know where to stand in the circle so that he will be chosen to go speak to the professor so the exam will not change.  You are to build the simulation to help Joseph decided where to stand for any given N value. |
|  |  |
| **Program:** | You are to create a file called **CircleQueue**.**java**. You will create two class **CircleQueue** and **Node.** The queue will be a circular queue with the nodes storing the students’ names and starting position. This queue must be reference based since you will not know how many students to have at the beginning.  As with last time, your Node should be generic but have two generic values. The students name will be a string while starting position will be an integer, but the node should be generic and accept any two values.  You are given a file called **p5.dat** in the folder on the server. This file contains the name of each student on a line. You must read in each student and add him or her to the queue. The node created will contain the student’s name as well as some integer that keeps track of the order added to the queue as the starting position.  NOTE: this starting position will not change will NOT change after set. The first position will be 0 and the last will be size – 1.  Once the queue is filled you will go through and start removing students. First ask the user to enter the N value to be the counter for removing students (for example an N of two will remove person 2,4,6,etc…)  Start with the first person and dequeue him or her. If the person does not meet the order corresponding to the number, add the student back to the queue. If the student does correspond to the N value, excuse the student and do not add him or her back.  Do this until you are left with one student, and display the name and starting position of the student who must go talk to the professor. |
|  |  |
| **Input:** | There is an input file called **p5.dat** were each line contains a name. |

|  |  |
| --- | --- |
| **Output:** | Enter the number used to remove people from list: 2  Sammy is excused.  Susie is excused.  Angela S. is excused.  George is excused.  Thomas the Great is excused.  Jack Spratt is excused.  Steven is excused.  Chuckles the Clown is excused.  Joseph is excused.  Jim Bob is excused.  What's His Name is excused.  Eileen is excused.  Elizabeth Anne is excused.  Kenny is excused.  Daniel is excused.  Mr. Know-it-All is excused.  Ruth is excused.  Robert : 4 must go talk to the instructor. |

|  |  |
| --- | --- |
|  |  |
| **Hints:** | Use appropriate software design techniques, and implement the class methods with Java constructs for I/O, declarations, and calculations.  Build your program in steps (i.e., get the input and output working, then add the functions, etc.). Emphasize functionality first, then add the advanced features. Work on this program over time, DO NOT wait until the last minute (i.e., the day before it is due) to start! |
|  |  |
| **Data:** | On the server, you will need to use the following absolute path and append the filename to it. "/home/courses/csci3302-002/datafiles/" |
|  |  |
|  | Remember that you must pass the data file name in as a command line argument. |
|  |  |
|  |  |