Juhwan Lee

CS-163

July 13th 2020

Program #2 Design Write Up

The purpose of the second programming assignment is to experience stacks, queues, and new data structures. The data structures for this assignment are linear linked list of arrays and a circular linked list. With this setting, the second programming assignment is going to be about keeping track of everyone in line for their appointments, for example, store, restaurant, or doctor's office, and letting people know when it is their turn. For this program, there is going to be two abstract data types, one is queues, and the other is stacks. Queue is first in first out structure and stack is last in first out structure. A queue would be great to keep track of everyone in line because it is first in first out. Then a stack would hold the alerts as they are sent out from the store, restaurant, or doctor's office to let people know when it is their turn. The stack is used to hold the alerts because if a person does not respond within a given amount of time, then the next person should be alerted and they will get priority. In order to have two abstract data types, each needs to be implemented as a separate class. The class interfaces can be in one .h file or two .h files. I will decide whether to have one .h file or two .h files as I program. The information kept in the queue of people in line is going to have person's full name and contact information, for example, cell phone number. And the information kept in the stack of people alerted as to their turn is going to have person's full name and the information about when the alert took place. And just like first programming assignment, modular design is going to be used, separating the .h files from the .cpp files. And also, we are creating abstract data types and so the user of the program must not be aware that stacks and queues are being used. However, the client program knows that we are using these abstract data types. Main.cpp file is going to be a test program also called client and .h files will have all the structs and class interfaces. In

class interfaces, there should be public section and private section and under public section, there will be all the function prototypes that the abstract data type performs and under private section, there is going to be all the required data members. Other .cpp file is going to have all the codes about function prototypes and the return types will be integer type, it will return 0, 1, or 2, in order to communicate with the test program. The queue will be implemented using a circular linked list where the rear pointer points to the last person in line, and rear->next point to the first. For queue ADT, there is going to be implementation of enqueue, dequeue, peek, and display. On the other hand, the stack will be implemented using a linear linked list of arrays, where each element in the array is a person being alerted. The array will be dynamically allocated and each array will be the same size. Stack ADT will include push, pop, peek, and display.