Neil Rolf

CSC300

Program 3 – Design (updated)

**Class Declarations**

typedef std::string SElement //quality of life typedef for std::string

Class StaticQueue

*public variables*

* StaticQueue(int size) //default constructor, creates queue of passed in size, sets head and tail position to -1, numItems to 0
* StaticQueue(StaticQueue &) //deep copy constructor
* bool isFull() //returns true if all elements in the queue are in use
* bool isEmpty() //returns true if the queue is empty
* bool enqueue(const SElement, int, SElement) //adds data to the next available location in the queue, returns true if item was stored successfully
* bool dequeue(SElement &, int &, SElement &) //removes the item from the queue and stores its data into reference parameters, returns true if item was removed successfully
* void view() //prints contents of queue (mostly for debug)

*private variables*

* struct PatientInfo //holds data values
  + SElement name //patient name
  + int age //patient age
  + SElement problem //description of patient ailments
* typedef PatientInfo \* Qpointer; //QoL typedef for struct pointer
* Qpointer queue; //pointer to the queue object
* const int Q\_SIZE; //holds passed in size of queue
* int head, tail; //holds value of head and tail positions in queue
* int numItems; //holds value for number of items in the queue

**Driver**

void statusCheck(StaticQueue object);

int(main){

int queueSize = “size of queue”;

StaticQueue testQueueA(queueSize);

nameArray(hardcoded list of names…);

ageArray(hardcoded list of ages…);

problemArray(hardcoded list of problems…);

//test isEmpty

bool testQueue.isEmpty()

//test partial enqueue

for(int i=0, i<”some number”, i++)

bool testQueueA.enqueue(nameArray[i], ageArray[i], problemArray[i]);

void statusCheck(testQueueA);

//test copy constructor

StaticQueue testQueueB(testQueueA);

void statusCheck(testQueueB);

//test partial dequeue

for(int i=0, i<”some number”, i++)

bool testQueueA.dequeue(nameArray[i], ageArray[i], problemArray[i]);

void statusCheck(testQueueA);

//test enqueue past queue size

for(int i=0, i<”queueSize + 1”, i++)

bool testQueueA.enqueue(nameArray[i], ageArray[i], problemArray[i]);

void statusCheck(testQueueA); //isFull tested here

//test dequeue past queue size

for(int i=0, i<”queueSize + 1”, i++)

bool testQueueA.dequeue(nameArray[i], ageArray[i], problemArray[i]);

void statusCheck(testQueueA);

}

//function for periodic status checks

void statusCheck(StaticQueue object)

bool testQueue.isFull()

bool testQueue.isEmpty()

print queue contents

Tests to run:

* enqueue
* dequeue
* isFull
* isEmpty
* enqueue past given array size
* dequeue on empty queue
* copy a queue using copy constructor
  + test integrity of original queue object