EECE.3220: Data Structures
Spring 2017

Lecture 22, Key Questions
March 22, 2017

1. Describe how a stack data structure can be used to evaluate postfix expressions.

2. Describe the general design of a queue data structure, as well as some basic applications in which it is useful.

3. Describe how an array can be used to implement a queue class.

4. Write definitions for each function below, assuming an array-based queue with data members
   `QueueElement myArray[CAPACITY]`, `int myFront`, and `int myBack`:

```
// Default constructor
Queue::Queue()
{

}

// True if list is empty
bool Queue::isEmpty() {



}

// Add new value to back of queue
void Queue::enqueue(const QueueElement &val) {








}

// Remove element at front of Queue
void Queue::dequeue() {







}

// Retrieve value of element at top of Queue
QueueElement Queue::front() {




}
```

5. Explain how a linked queue is implemented.

6. Write (in code or pseudo-code) definitions for each function below, assuming a linked Queue with data members `Node *myFront` and `Node *myBack`:

```
// Default constructor
Queue::Queue()
{}

// True if list is empty
bool Queue::isEmpty() {



}

// Add new value to back of Queue
void Queue::enqueue(const QueueElement &val) {




}
```

5 (continued)
```
// Remove element at front of Queue
void Queue::dequeue() {




}

// Retrieve value of element at top of Queue
QueueElement Queue::front() {




}
```

7. Describe how a circular linked list could be used to implement a linked queue.