Data Structures 10/19/2016

0145-343-001

Note Taker: Jai Punjwani

ANNOUNCEMENTS

10/21 – Midterm (updated topic list):

HW#7 – postponed till next week

1. Write a class in C++
2. Write an implementation for stacks and/or queue
3. Trace a program using a stack, queue and/or linked list operations
4. Write and/or trace recursive methods
5. Convert between infix, prefix, postfix

Review:

* Linked List implementation of queues: remove() works in the same fashion as pop()
* freeNode() is essentially ‘delete node’ in C++
* note that we create a node to a struct by declaring a pointer to an instance of the struct itself. This creates a pointer within an object X to another object of type X.
* syntax reminder: void push(NodePtr &list, int x) – the ‘&’ signifies that this method receives a reference to
* in implementation, every node has to have a pointer pointing to it.
* Also, when you perform operations on your linked list, check whether your list is empty FIRST!
* CAVEAT: Objects in C++ are passed by VALUE. Thus, to manage our memory, and pass references, we will have to do so manually.