

# Lecture 13 Notes

Derek Harter

2015-04-21

## 1 First Session (11 - 11:40)

### 1.1 Data Structures

- Arrays are fixed-size data structures.
- Dynamic data structures can grow and shrink during execution.
- Are widely needed and useful.
- We will look at linked lists, stacks, queues and trees.

### 1.2 Self-Referential Structures

- A self-referential structure (or class) contains a pointer member that points to a `struct` of the

same type.

- For example, a structure that holds an integer element, and can point to other structures

```
struct Node
{
    int data;
    Node* nextPtr;
};
```

- This user defined type has 2 members
  - A field to hold some data (could be arbitrary, even another struct)
  - A `nextPtr` field, which is a pointer to a `Node`. Can be used to point to next item in a list, stack, queue or tree.

- Self-referential class objects can be linked together to form useful data structures such as lists, queues, stacks and trees.
- By convention, we use a NULL pointer to indicate the end of a list or chain of such items (similar to null character in a C

char array).

## **2 Second Session (11:45 - 12:30)**

### **2.1 Dynamic Memory Allocation**

- Creating and maintaining data structures that can grow or shrink requires dynamic memory allocation.
- New operator takes an argument of the type being allocated, and returns a pointer to the new items allocated address.
- If no memory available, an exception occurs.
- We can dynamically deallocate memory with the corresponding delete operator on a pointer to a dynamically allocated object.

## **3 Third Session (12:40 - 1:40)**

### **3.1 Linked Lists**

- A linked list is a linear collection of self-referential user defined types (Node).
- The nodes are connected by pointer links.
- There is a linear ordering. There is a first item in a list, which points to the next item, and so on.
- The last item in the list (by convention) points to the NULL pointer. We use the NULL pointer when processing such a list to determine when we have reached the end of the list. Forgetting to set the NULL pointer for the last item will cause incorrect behavior.

### **3.2 Circular linked lists**

### **3.3 Doubly linked list**