CS 225 - Lecture 3

Scribe : Harsha Srimath Tirumala

1 Learning Goals

↪ Templates

↪ Functions/operations of list ADT

↪ List implementation strategies

↪ Linked Lists

↪ Practice C++ fundamentals in context of lists

2 Templates

A generic code whose type is determined during compilation. Templates are a code recipe using generic types. Compiler uses templates to generate C++ code *only when needed.* They are particularly useful when there is code overlap. For example, Sum(int, int) and Sum(float, float) can instead be templated using T sum(T a, T b).

3 List ADT

↪ Lists are an ordered collection of items. Items can be homogeneous or heterogeneous. Lists can have a fixed size or be re-sizable.

↪ Minimal Set of operations for a list :
1. Insert 2. Delete 3. isEmpty 4. getData 5. Create an Empty list

4 List Implementations

↪ List can be implemented using two strategies : *Linked Lists* or *Array Lists.*

5 Linked Lists

↪ List implemented using List Nodes.

↪ **Design Choices**
- Data by reference : Can't change what we are pointing at + can't be nullptr + no pointer overhead
- Next by pointer : Want to be able to change what it points at + can be nullptr (default)

```
class ListNode {
    T & data;
    ListNode * next;
    ListNode(T & data) : data(data), next(NULL) {}
 };
```