CS180- List Class

Note Title 10/4/2010

Announcements

- Program due next Tuesday

Lists:

Motivation: If we know where something should go in the list, it should be a fast operation.

Trade-off with vectors:

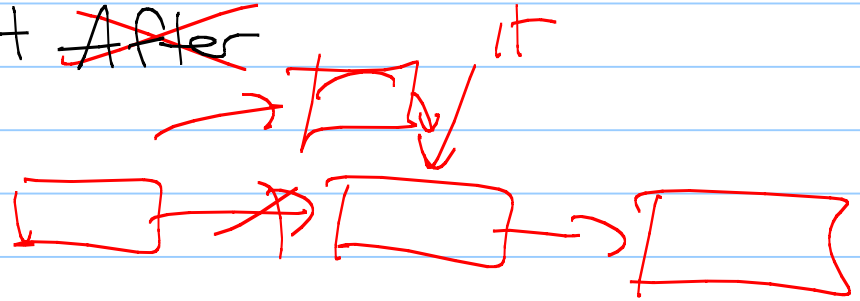
- insert is fast
- less memory
- don't have [] notation
  mylist[7]

Iterators

Essentially, an iterator is a user-safe pointer.

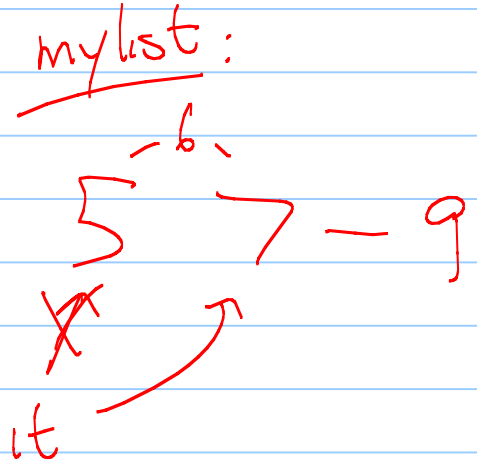
We write the iterator class to give the user a way to select an item in the list.

Ex: function insert ~~After~~

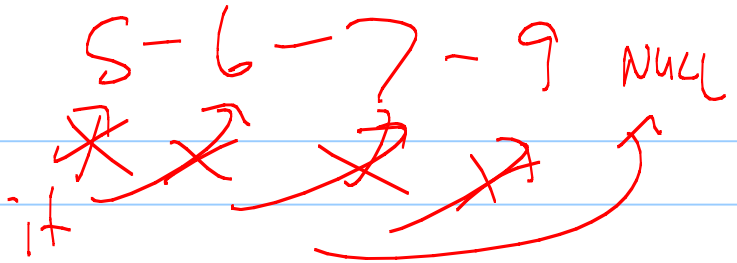


How to use: (based on STL)

```
List<int>  mylist;

List<int>::iterator  it;

mylist.push_back(5);
mylist.push_back(7);
mylist.push_back(9);

it = mylist.begin();

it++;
mylist.insert(it, 6);  //insert 6 before
                       //the iterator
```



More:




```
for (it = mylist.begin(); it != mylist.end(); it++)
    cout << " " << *it;
cout << endl;
```

-5-6-7-9

Functors from last time

- Constructor (blank one)
- ~~front~~ begin - returns iterator to front of list

belongs in List, not iterator

- operator * - return contents of iterator
- operator ++ - go to next object in list

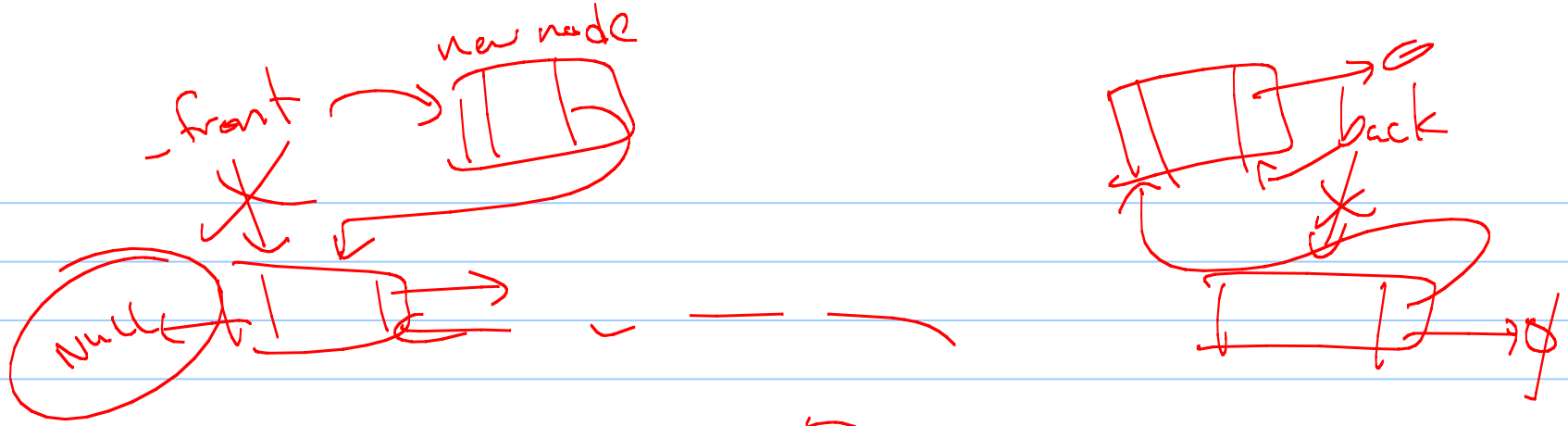
in List class

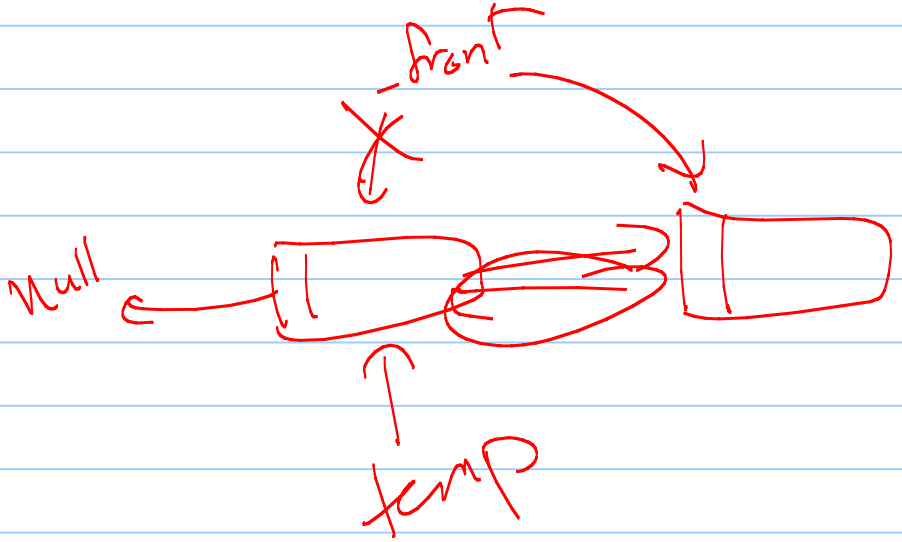
```
iterator begin() const {
    return iterator(_front);
}
```

this construction will be off limits to main

pointer1 = &variable;

it++;
++it;


new node
front
Null
back


front
Null
temp