CS180 - List Class

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
  my_list[]
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 it
How to use: (based on STL)

```cpp
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:

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

cout << endl;
```

5 6 7 9
Functions from last time

- Constructor (blank one)

- `begin` - returns iterator to front of list

- operator * - return contents of iterator

- operator ++ - go to next object in list
In list class:

```cpp
iterator begin() const {
    iterator i = this->begin;
    return iterator(--front);
}
```

```
pointer1 = &var1
```

```
++i++;
++i;
```

This constructor will be off limits to main.