
COP 3540 Data Structures with OOP

Chapter 5 - Part 3 Linked Lists

Doubly-Linked Lists and Iterators

Doubly Linked Lists

- ◆ How do we track in singly-linked lists?
current = current.next;
- ◆ Must always keep link from 'last' link for insert() and delete(), etc.....
- ◆ A given link has its data and a forward link!
- ◆ Many applications require both forward and rearward traversal of a linked list.
- ◆ To do this, each link has both forward and rearward (backward) links (pointers).
- ◆ Thus, from any given link, can go either direction.

Code for class Link

```
class Link
{
public long   dData;           // data item
public Link  next;           // next link in list
public link  previous;       // previous link in list
}
```

Downside: each time you insert() a link, you **MUST** account for two pointers: one to the next link and one to the rear link.

Inserting and Deleting can be very complicated.

One **MUST** keep track of the links and change them in a prescribed order (we shall see).