Circularly Linked Lists

Lecture 20
Section 11.2

Robb T. Koether

Hampden-Sydney College

Wed, Mar 4, 2009
Outline

1. Circularly Linked Lists
2. Benefits
3. Examples
4. Assignment
Definition (Circularly Linked List)

A **circularly linked list** is a doubly linked list in which one additional (dummy) node is allocated whose pointers serve as the head and tail pointers. The dummy node’s data member is not used.
CircLinkedList Data Members

- **int mSize** - The number of elements in the list.
- **DoublyLinkedListNode* list** - A pointer to the dummy node.
A `CircularlyLinkedList` uses `DoublyLinkedListNode`s.

The dummy node is always allocated—even in an empty list.
Benefits of this Implementation

- The `next` pointer of the last node points to the dummy node, so it is not null.
- The `prev` pointer of the first node points to the dummy node, so it is not null.
- In fact, none of the pointers in the structure is null.
- Since there are no null pointers, the code in the member functions contains no special cases!
Implementing Member Functions

- Write the `insert()` function.
The CircLinkedList Class

- Download `doublylinkedlistnode.h`.
- Download `circlinkedlist.h`.
- Download and run `ListTest.cpp`.
Assignment

Homework

- Read Section 11.2, pages 589 - 591.