CSCI 2170 Spring 2006 Lecture Notes (11)
Recursion with linked list

Applying recursion on linked list

<u>Example 1:</u>

```
//Write the content of a linked list using recursion
void WriteString(ptrType StringPtr)

{
  if (StringPtr != NULL)
  {  // write the first character
     cout << StringPtr->Item;


     // write the string minus its first character
     WriteString(StringPtr->Next);
  }  // end if
}  // end WriteString
```

<u>Example 2:</u>

```
//Write the content of a linked list of items backwards
void WriteBackward(ptrType StringPtr)

{
  if (StringPtr != NULL)
  {
     // write the string minus its first character backward
     WriteBackward(StringPtr->Next);

     // write the first character
     cout << StringPtr->Item;
  }  // end if
}  // end WriteBackward
```

<u>Example 3:</u>

```
// insert an item into linked list
void LinkedListInsert(ptrType& HeadPtr, itemType NewItem, bool& Success)
{
  if ((HeadPtr == NULL) || (NewItem < HeadPtr->Item))
  {  // base case: insert NewItem at beginning of the linked list to which HeadPtr points
     ptrType NewPtr = new node;
     Success = bool(NewPtr != NULL);
     if (Success)
     {
        NewPtr->Item = NewItem;
        NewPtr->Next = HeadPtr;
        HeadPtr = NewPtr;
```

```
    }  // end if
  }
  else
    LinkedListInsert(HeadPtr->Next, NewItem, Success);

}  // end LinkedListInsert
```

Recursive function as a member of listClass
!! head pointer is private data in listClass !!

```
class listClass
{
public:
        …
        void ListInsert(itemType newItem, bool & Success);
        …
private:
        …
        ptrType head;
        void LinkedListInsert(ptrType& HeadPtr, itemType NewItem, bool& Success);
};
void listClass itemType newItem, bool & Success)
{
        LinkedListInsert(head, newItem, Success);
}
// same implementation as above
void listClass::LinkedListInsert(ptrType& HeadPtr, itemType NewItem, bool& Success)
{
  if ((HeadPtr == NULL) || (NewItem < HeadPtr->Item))
  {  // base case: insert NewItem at beginning of the linked list to which HeadPtr points
    ptrType NewPtr = new node;
    Success = bool(NewPtr != NULL);
    if (Success)
    {
      NewPtr->Item = NewItem;
      NewPtr->Next = HeadPtr;
      HeadPtr = NewPtr;
    }  // end if
  }
  else
    LinkedListInsert(HeadPtr->Next, NewItem, Success);

}  // end LinkedListInsert
```