CMSC 132: OBJECT-ORIENTED PROGRAMMING II

Simplified List Implementation

Department of Computer Science
University of Maryland, College Park
Typical List Implementation

• Class List {
      Node head;
   }

• Class Node {
      Object value;
      Node next;
   }
Insert in Front of i’th Element

```java
void insertInFrontOf(int pos, Object value) {
    if (pos == 0) {
        Node newNode = new Node(value, head);
        head = newNode;
    } else {
        Node after = head;
        for(int i = 1; i < pos; i++) after = after.next;
        Node newNode = new Node(value, after.next);
        after.next = newNode;
    }
}
```
List Implementation Trick

• You must practice this technique if you expect to use it
• head is never null, even for an empty list
• head is set to first node when list is created
  • head is never changed
  • the value of the first node isn’t ever looked at
• Class List {
    // value of first Node isn’t part of list
    final Node head = new Node(null);
}
Insert in Front of i’th Element

```java
void insertInFrontOf(int pos, Object value) {
    Node after = head;
    for(int i = 0; i < pos; i++)
        after = after.next;
    Node newNode = new Node(value, after.next);
    after.next = newNode;
}
```