CS367 Announcements
Monday, July 1, 2013

- H1 due Mon 6pm
- P1 due Wed 11:59pm
- no class July 4th
- common h1 errors

Last Time

- Complexity
- Primitive vs Reference

Today

- Primitive vs Reference (cont.)
- Linked Lists

Primitive vs Reference Types: Parameter Passing

Primitive

Given:

```
void mod1(int x) {
    x = 7;
}
```

Consider (assume in main):

```
int x = 1;
int[] y = {1, 2, 3};
mod1(x); mod1(y[2]);
```

Reference

Given:

```
void mod2(int[] x) {
    x[0] = 7;
}

void mod3(int[] x) {
    x = new int[x.length];
    x[0] = 14;
}
```

Consider (assume in main):

```
int[] a = {1, 2, 3};
mod2(a);
mod3(a);
```

ADTs vs. Data Structures

abstract data type (ADT):

data structure:

Chain of Linked Nodes

Conceptual picture:

Goal:

Java Visibility Modifiers

public	`public class ArrayList`
private	`private Object[] items`
protected	`protected String name`
package	`int studentID`

Listnode Class

```
class Listnode<E> {
    private E data;
    private Listnode<E> next;

    public Listnode(E d) {
        this(d, null);
    }

    public Listnode(E d, Listnode<E> n) {
        data = d;
        next = n;
    }

    public E getData() { return data; }

    public Listnode<E> getNext() { return next; }

    public void setData(E ob) { data = ob; }

    public void setNext(Listnode<E> n) { next = n; }
}
```

Practice: Making a Chain

Create a chain containing the strings "yipie","ki", and "yay" (in that order)

Practice: Traversing a Chain

Assume `head` points to the first node in a chain of nodes containing `Integers`.

Write a code fragment that counts the number of items in the chain of nodes.

Practice: Adding a Node at End

Assume `head` **points to the first node in a chain of nodes containing** `Integers`**.**

Write a code fragment that adds 48 to the end of the chain of nodes.