CS112: Data Structures

Instructor: Prof. Louis Steinberg

- **office: Hill 401**
- **email: lou@cs.rutgers.edu**
- **phone: 848-445-7289**

CS112: Data Structures

TAs:

- **To be announced**

CS112: Data Structures

- **Today's topics:**
  - **Intro to Course**
  - **Intro to Linked lists**

Class Web Site

- **http://sakai.rutgers.edu**
  - **Log in using Rutgers NetID & password, click on "CS112 – Spring 2015" tab**
- **You are assumed to know anything posted.**

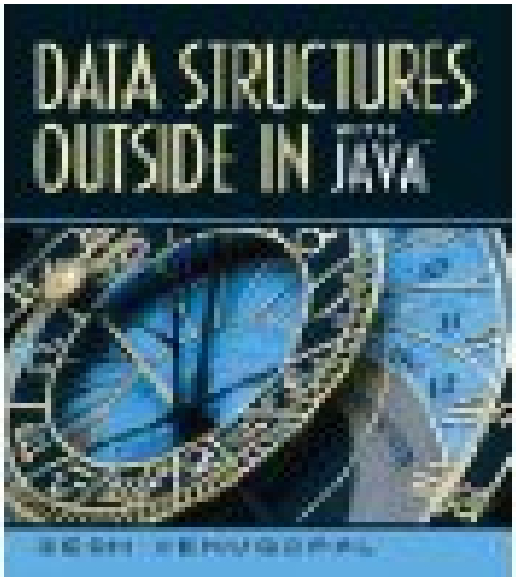
Textbook

- ***Data Structures Outside In with Java, 1st Edition.***

  by Sesh Venugopal

  Prentice-Hall, 2006.

  ISBN 978-0131986190.



Three lecture sections

Same syllabus, same assignments, same exams

- **Prof. Venugopal**
  - **Tusday/Thursday 3:20 pm, TIL 254**
- **Prof. Tjang**
  - **Tuesday/Thursday 5:00 pm, Ph 115**
- **Prof. Steinberg**
  - **Monday/Thursday Noon, TIL 257**

Prerequisite

- **CS 111 or equivalent**
  - **Comfortable writing and debugging Java programs 2 - 3 pages long**
  - **Basic Java (types, control flow, etc.)**
  - **Strings, Arrays**
  - **Searching and sorting arrays**
  - **Recursion**
  - **Using objects (not designing classes)**
  - **Big-O worst case analysis**
- **We will NOT assume exceptions or linked lists**

What You (Should Have) Learned in 111

(Graphic Art from wordle.net)

Slide from Sesh Venugopal

To Review Objects, Big-O

- **Read DSOI**
  - **Sections 1.1, 1.2**
  - **All of Chapter 3**

Grading

- **Assignments (5): 35%**
- **Midterm 1 (Written): 15%**
- **Midterm 2 (Written): 15%**
- **Final (Written): 30%**
- **Recitation problems (1 per recitation): 5%**

No clicker quizzes

How to succeed in 112

- **COME TO LECTURE and PAY ATTENTION**
- **SPEND TIME outside of class, reviewing concepts and doing problems. Think through the problems before recitation.**
- **READ the textbook and WATCH You Tube videos**
  - **See http://www.cs.rutgers.edu/~venugopa for a complete list**
- **Study with FRIENDS**

What is a data structure

- A representation scheme that stores
  - Multiple pieces of data
  - Relationships between pieces of data
- E.g,
  - Object
  - Array

What you will learn in 112

Specialized Data Structures:

- Linear
  - Array
  - Linked list
  - Stack
  - Queue
- Trees
  - Binary Tree
  - Binary Search Tree
  - AVL Tree
  - Heap
- Graphs
  - Undirected
  - Directed
  - Weighted
- Hash Tables

What you will learn in 112

- **Algorithms**
  - **Searching**
  - **Sorting**
  - **Graph Algorithms**
- **Analysis of space & time taken by algorithms:**
  - **Best case**
  - **Average case**
  - **Worst case**

What you will learn in 112

- **Programming in Java using Eclipse**

Our first data structure

- **Linked Lists**
  - **Like an array, a linked list stores an ordered list of data items, all of the same type**
  - **But has different time and space costs**

Linked Lists

- When you have data in order in an array

```
  0    1    2    3
 75   100  110  ...
```

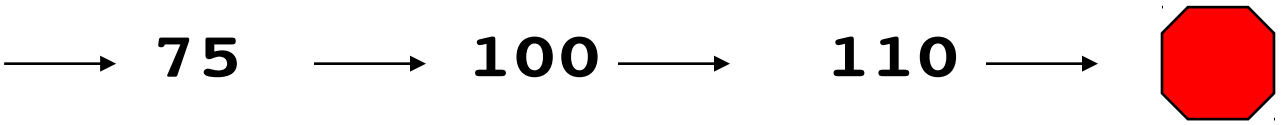
- You are storing who is 1st, 2nd, etc
  - Which largely changes when you insert

```
  0    1    2    3
 75   80   100  110   ...
```

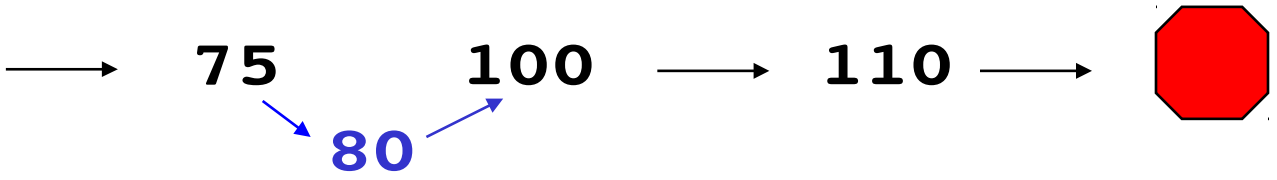
- What if all you care about is what is after what?

Linked Lists

- **Suppose what you store is "what comes next"**



- **When you insert, there is less to change**

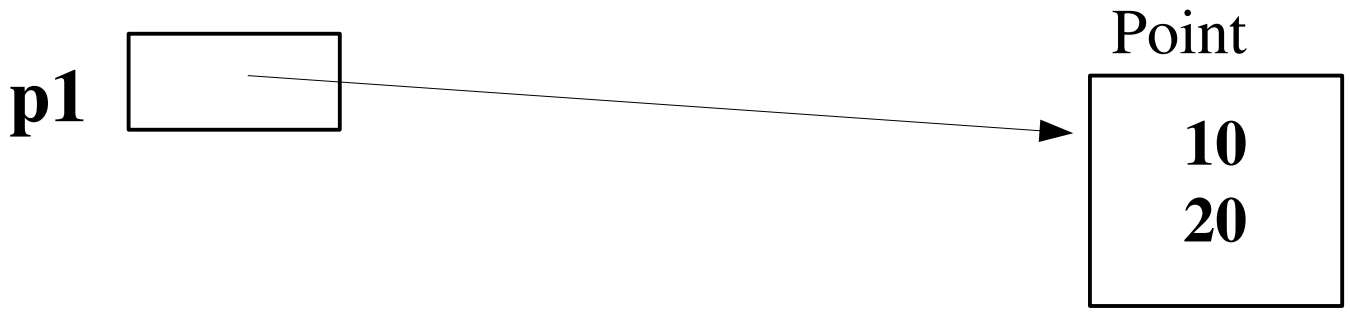


IntNodes

- **A node is an object that has**
  - **a field for data**
  - **a field to refer to the next node in the linked list**
- **data is an int → IntNode**
- **An IntNode object has an instance variable that contains a *reference to* an IntNode object**

Method variable holds a reference to an object

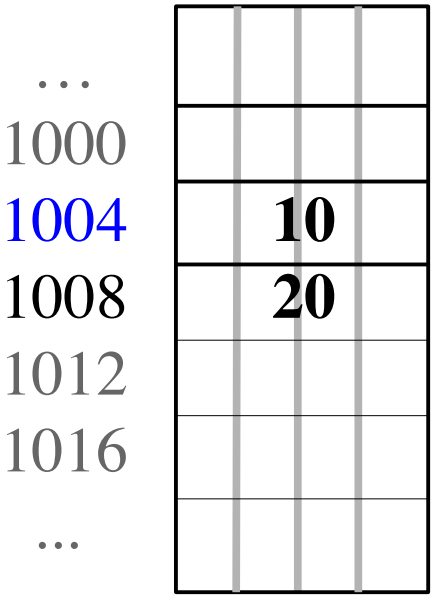
Point p1 = new Point(10, 20);



Method variable holds a reference to an object

Point p1 = new Point(10, 20);

p1 **1004**



Consequences of Representing an Object by a Reference

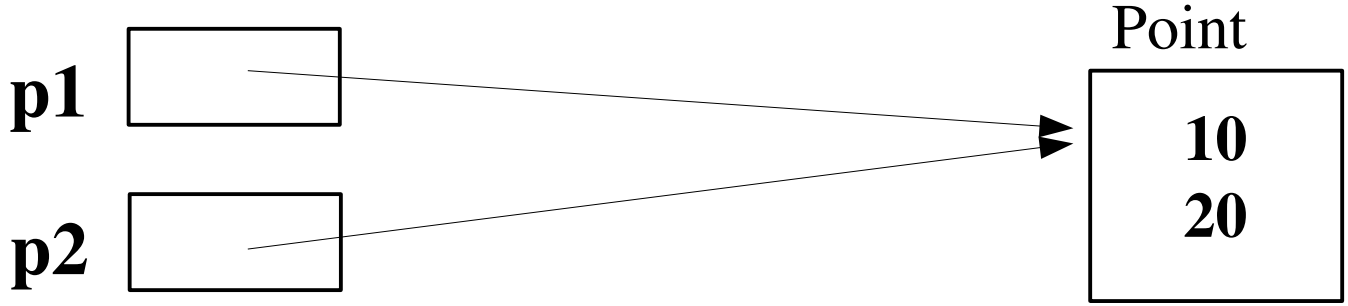
- **Assignment copies the reference, not the object**

```
Point p1 = new Point(10, 20);
Point p2 = p1;
```

Consequences of Representing an Object by a Reference

- **Assignment copies the reference, not the object**

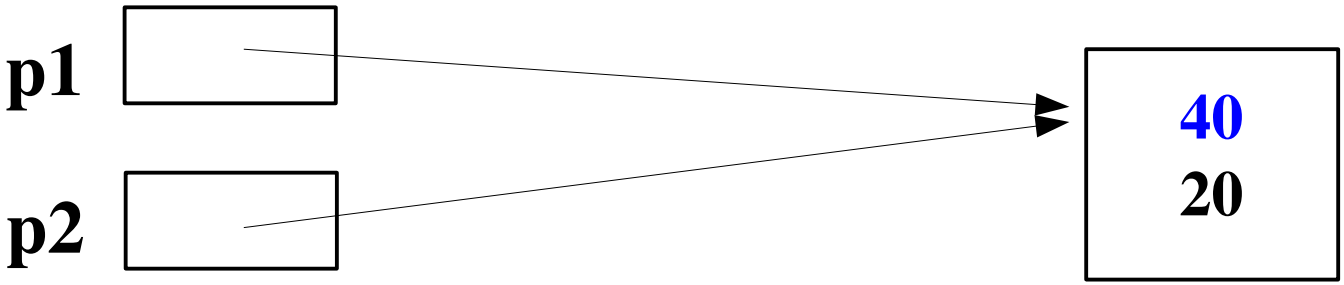
```
Point p1 = new Point(10, 20);
Point p2 = p1;
```



Consequences of Representing an Object by a Reference

- **Change via copied reference is also seen via original reference**

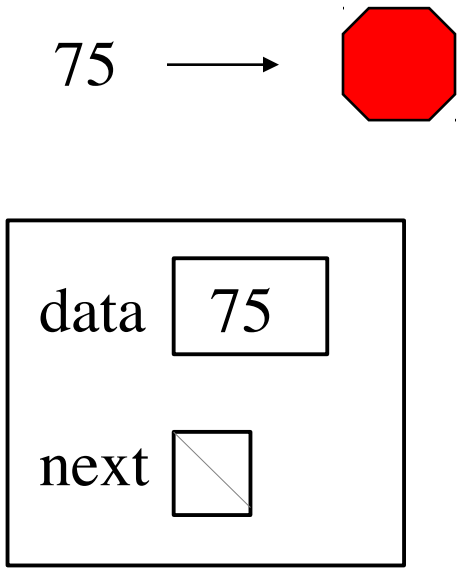
```
Point p1 = new Point(10, 20);
Point p2 = p1;
p2. x = 40;
System. out.println(p1. x); // prints 40
```



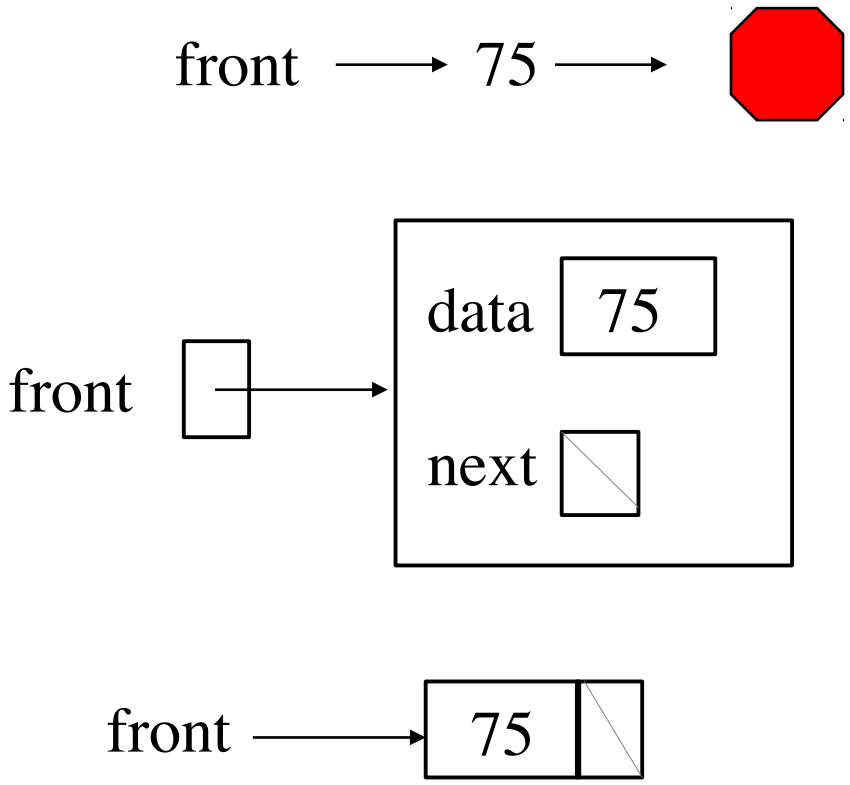
Consequences for method parameters

- **See ParamTest.java**

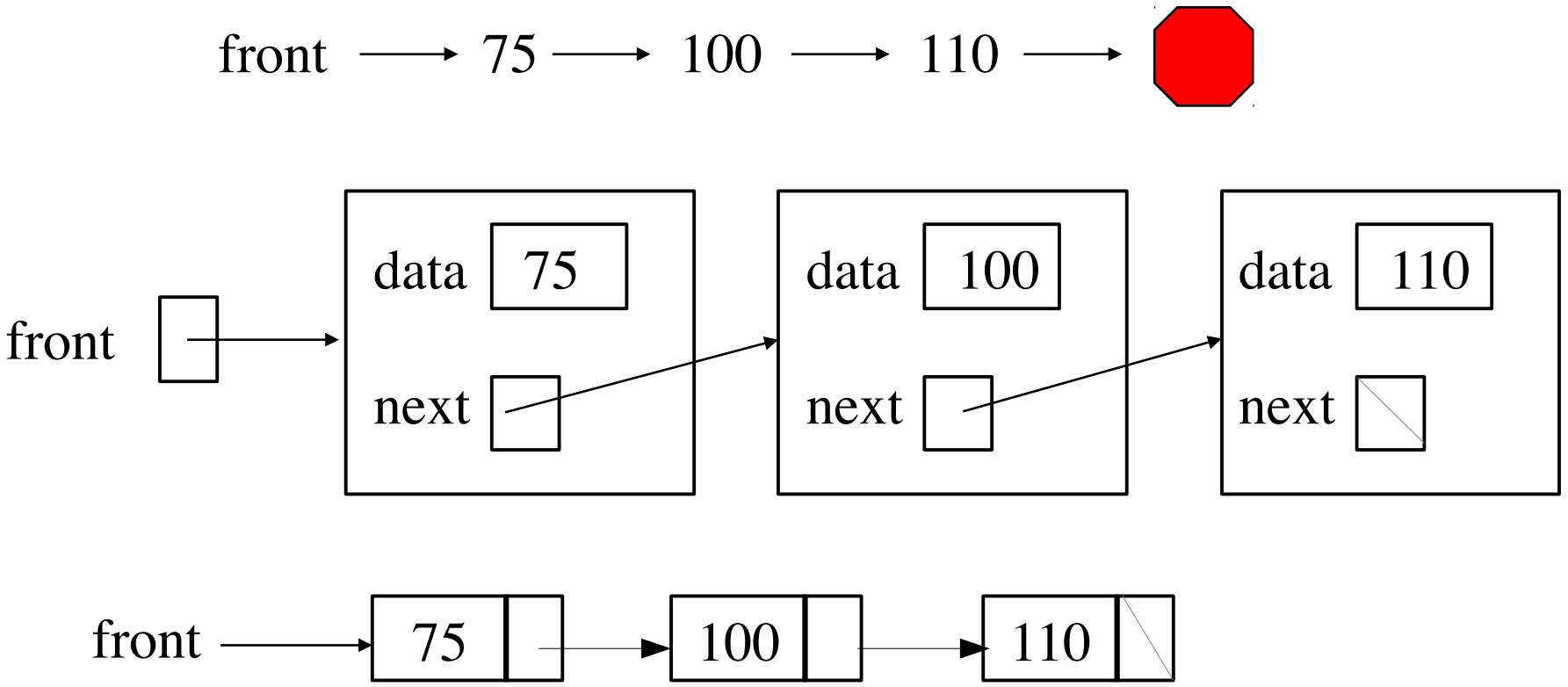
One IntNode



A One-IntNode List



A Three-IntNode List



What is printed?

```
// assume front is declared and set as in
// previous slide
System.out.println(front.data);
System.out.println(front.next.data);
System.out.println(front.next.next.data);
System.out.println(front.next.next.next);
System.out.println(front.next.next.next.data);
```

For Monday

- **Read DSOI**
  - **Sections 1.1, 1.2**
  - **All of Chapter 3**
- **Read the Syllabus on Sakai**
- ~~**Check out Piazza**~~