CSE 143
Lecture 9

References and Linked Nodes

reading: 3.3; 16.1

slides created by Marty Stepp
http://www.cs.washington.edu/143/

A swap method?

- Does the following swap method work? Why or why not?

```
public static void main(String[] args) {
    int a = 7;
    int b = 35;

    // swap a with b
    swap(a, b);

    System.out.println(a + " " + b);
}

public static void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}
```

Value semantics

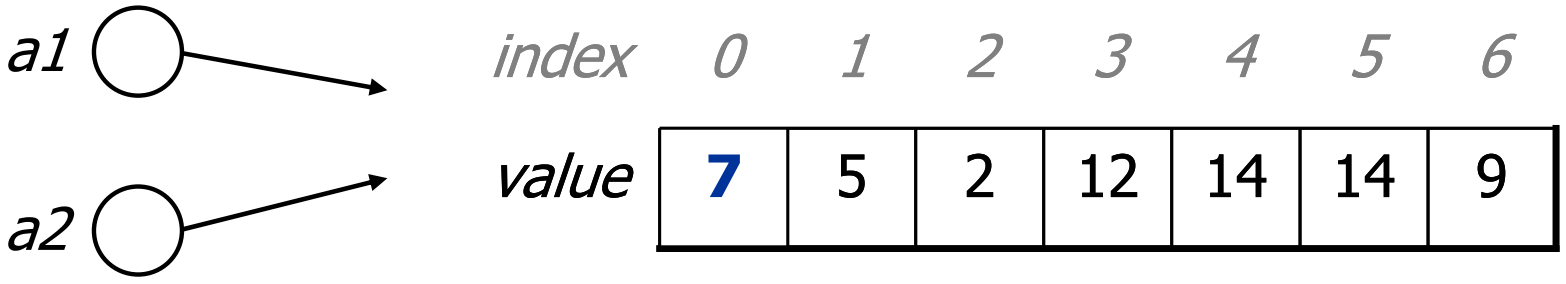
- **value semantics**: Behavior where values are copied when assigned to each other or passed as parameters.
  - When one primitive is assigned to another, its value is copied.
  - Modifying the value of one variable does not affect others.

```
int x = 5;
int y = x;      // x = 5, y = 5
y = 17;         // x = 5, y = 17
x = 8;          // x = 8, y = 17
```

Reference semantics

- **reference semantics**: Behavior where variables actually store the address of an object in memory.
  - When one reference variable is assigned to another, the object is *not* copied; both variables refer to the *same object*.

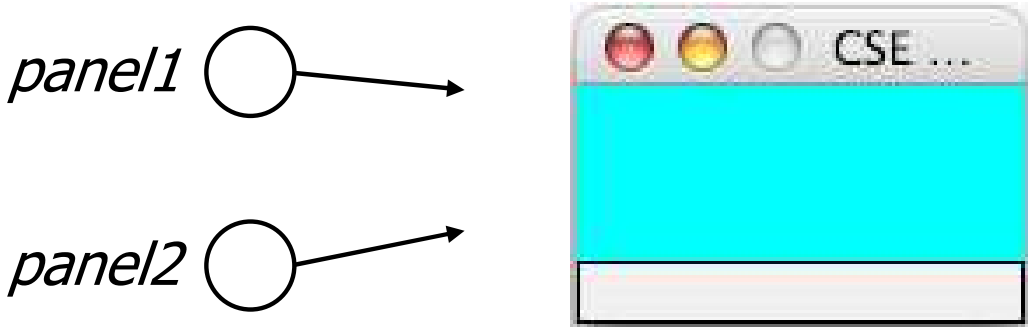
```
int[] a1 = {4, 5, 2, 12, 14, 14, 9};
int[] a2 = a1;      // refers to same array as a1
a2[0] = 7;
System.out.println(a1[0]);   // 7
```



References and objects

- In Java, objects and arrays use reference semantics. Why?
  - *efficiency.* Copying large objects slows down a program.
  - *sharing.* It's useful to share an object's data among methods.

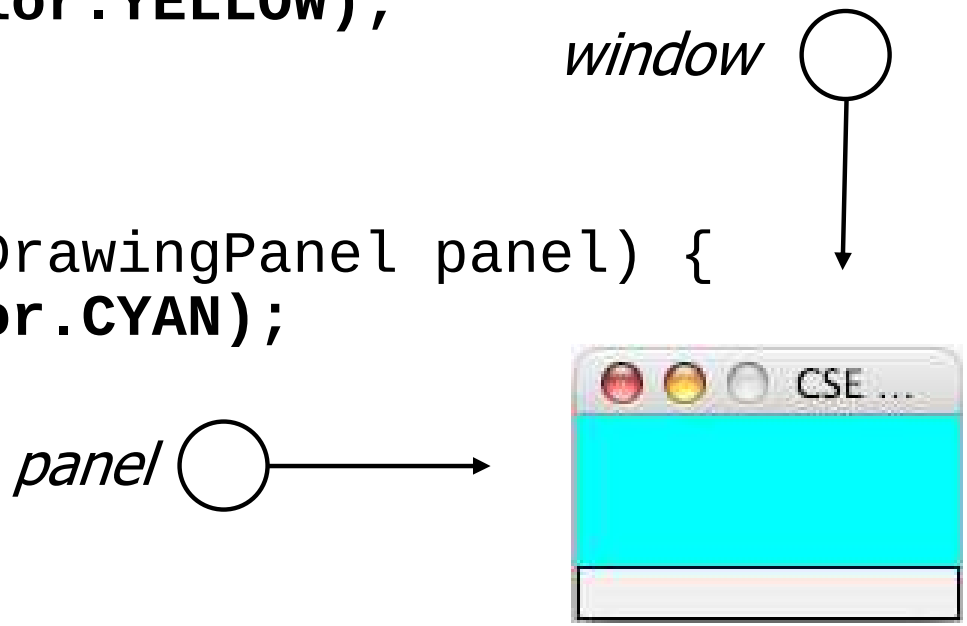
```
DrawingPanel panel1 = new DrawingPanel(80, 50);
DrawingPanel panel2 = panel1;   // same window
panel2.setBackground(Color.CYAN);
```



Objects as parameters

- When an object is passed as a parameter, the object is *not* copied. The parameter refers to the same object.
  - If the parameter is modified, it *will* affect the original object.

```
public static void main(String[] args) {
    DrawingPanel window = new DrawingPanel(80, 50);
    window.setBackground(Color.YELLOW);
    example(window);
}

public static void example(DrawingPanel panel) {
    panel.setBackground(Color.CYAN);
}
```

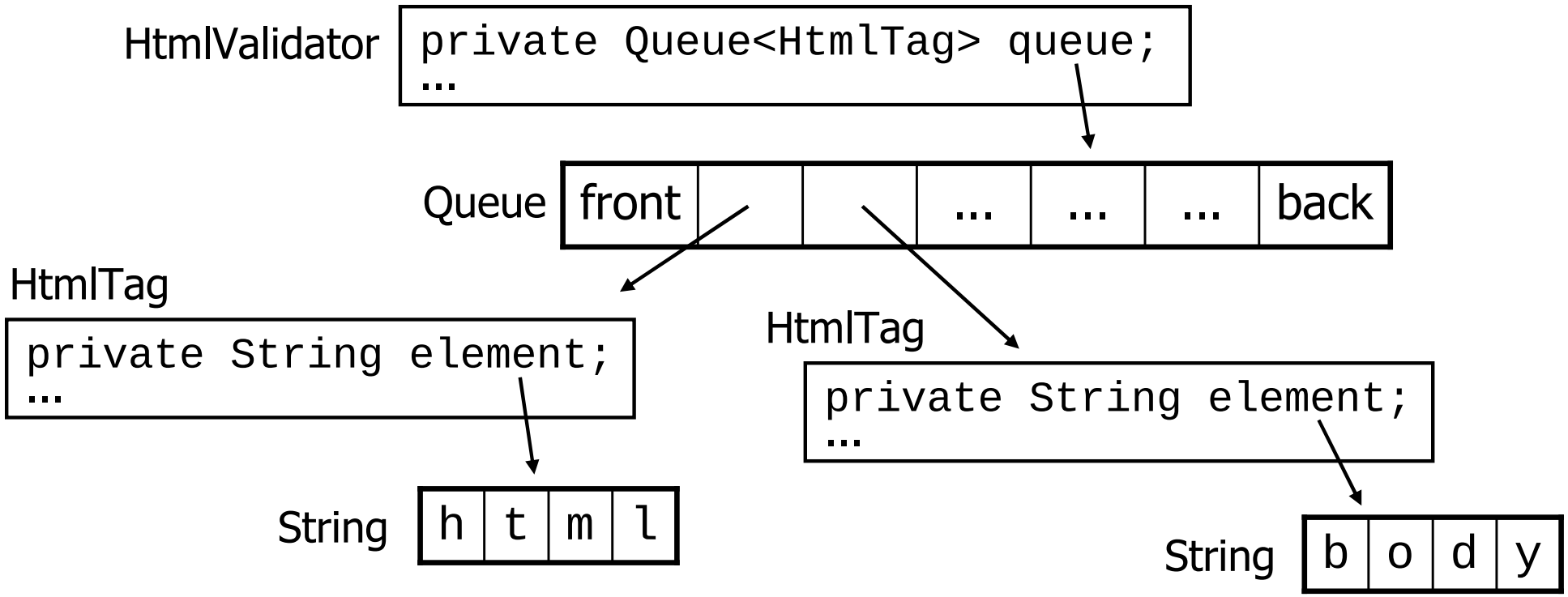


References as fields

- Objects can store references to other objects as fields.

  Example: Homework 2 (HTML Validator)

  - `HtmlValidator` stores a reference to a `Queue`
  - the `Queue` stores many references to `HtmlTag` objects
  - each `HtmlTag` object stores a reference to its element `String`



References to same type

- What would happen if we had a class that declared one of its own type as a field?

```
public class StrangeObject {
    String name;
    StrangeObject other;
}
```

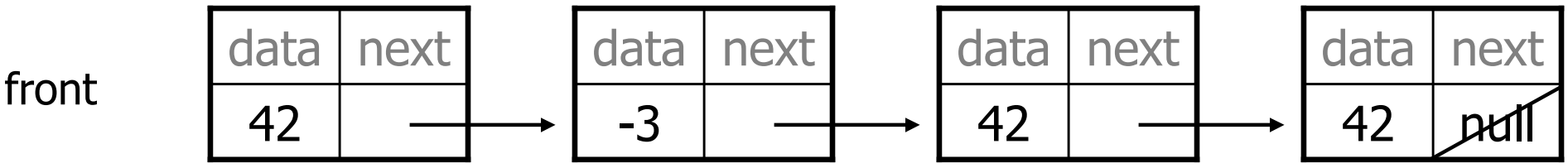
  - Will this compile?
    - If so, what is the behavior of the `other` field? What can it do?
    - If not, why not? What is the error and the reasoning behind it?

Linked data structures

- All of the collections we will use and implement in this course use one of the following two underlying data structures:
  - an **array** to store all elements of the collection
    - `ArrayList`, `Stack`, `HashSet`, `HashMap`

index	0	1	2	3
value	42	-3	17	9

  - a set of **linked objects**, each storing one element, that contain references to each other
    - `LinkedList`, `TreeSet`, `TreeMap`

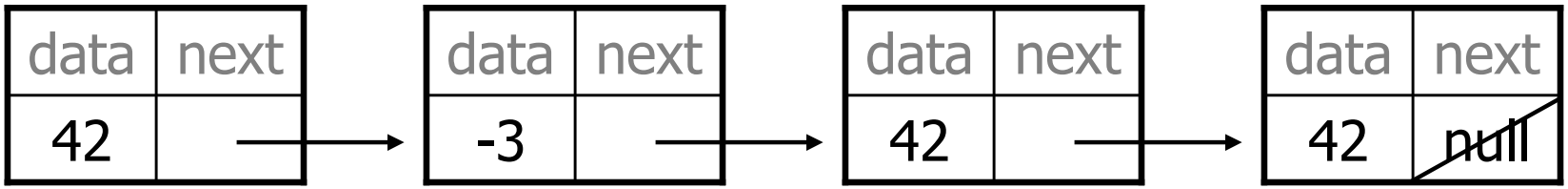


A list node class

```
public class ListNode {
    int data;
    ListNode next;
}
```

- Each list node object stores:
  - one piece of integer data
  - a reference to another list node

- `ListNodes` can be "linked" into chains to store a list of values:



List node w/ constructor

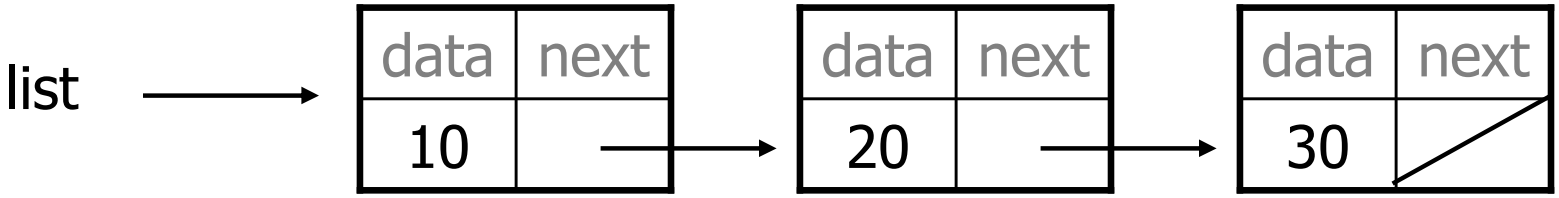
```
public class ListNode {
    int data;
    ListNode next;

    public ListNode(int data) {
        this.data = data;
        this.next = null;
    }

    public ListNode(int data, ListNode next) {
        this.data = data;
        this.next = next;
    }
}
```

– Exercise: Write the code to produce the list on the previous slide.

Linked node problem 1

- What set of statements turns this picture:

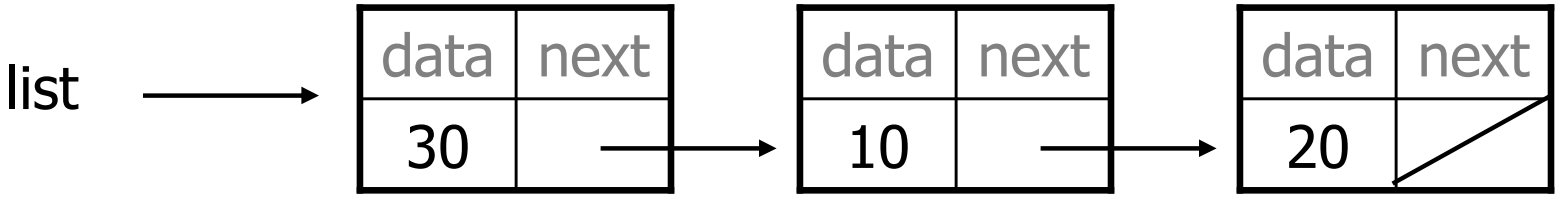
- Into this?



Linked node problem 2

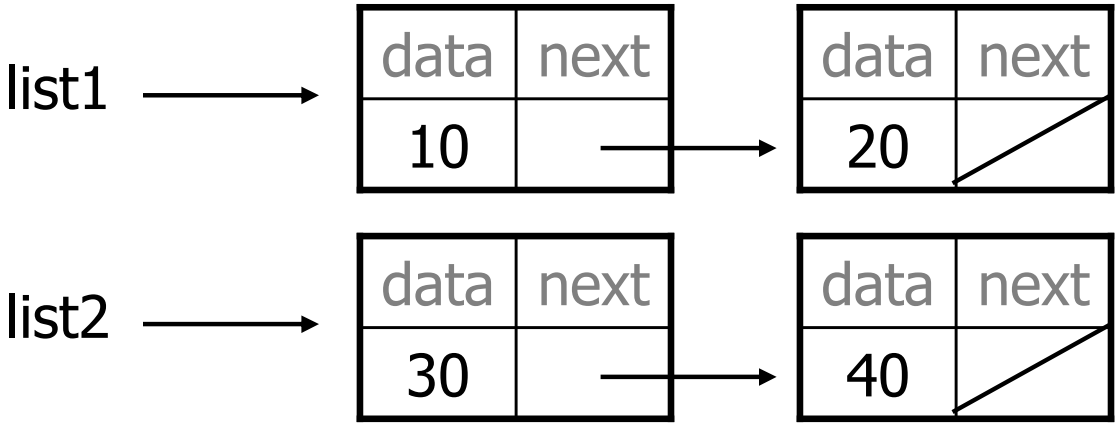
- What set of statements turns this picture:

- Into this?

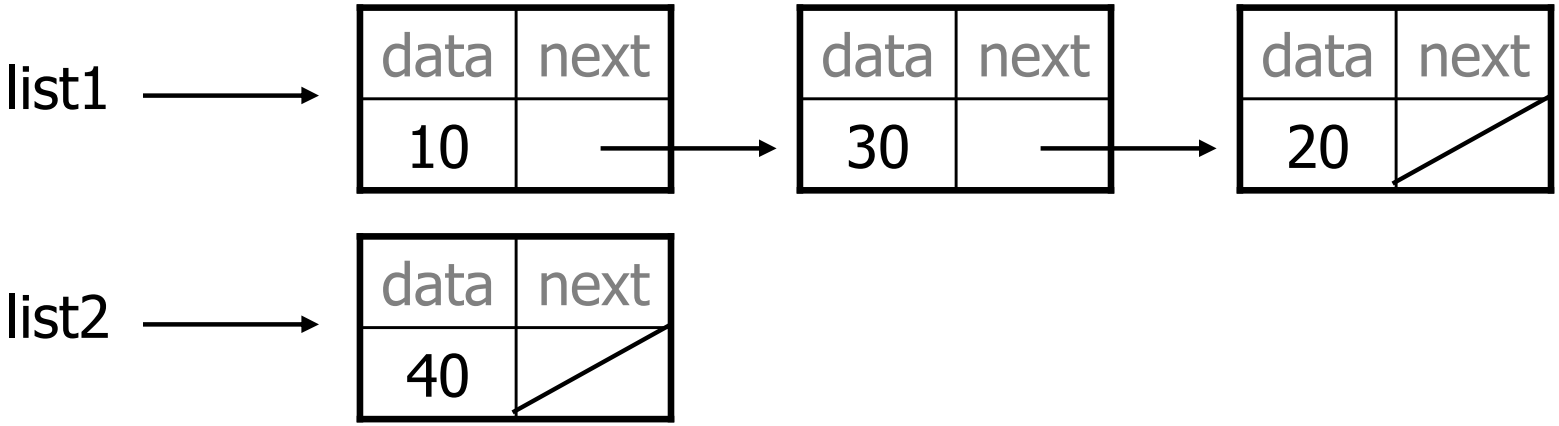


Linked node problem 3

- What set of statements turns this picture:

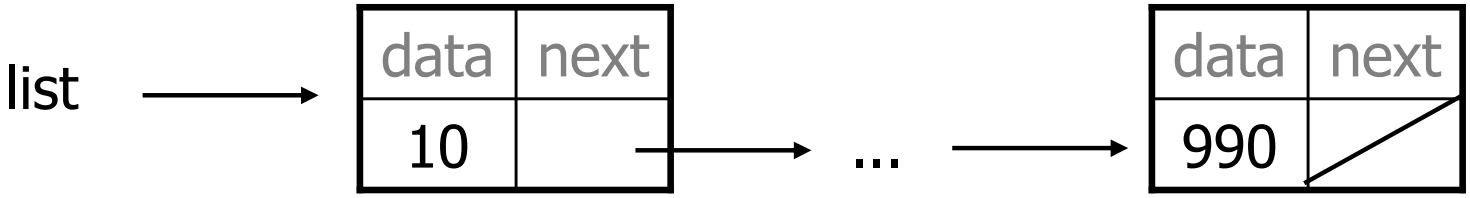


- Into this?



Linked node problem 4

- What set of statements turns this picture:



- Into this?