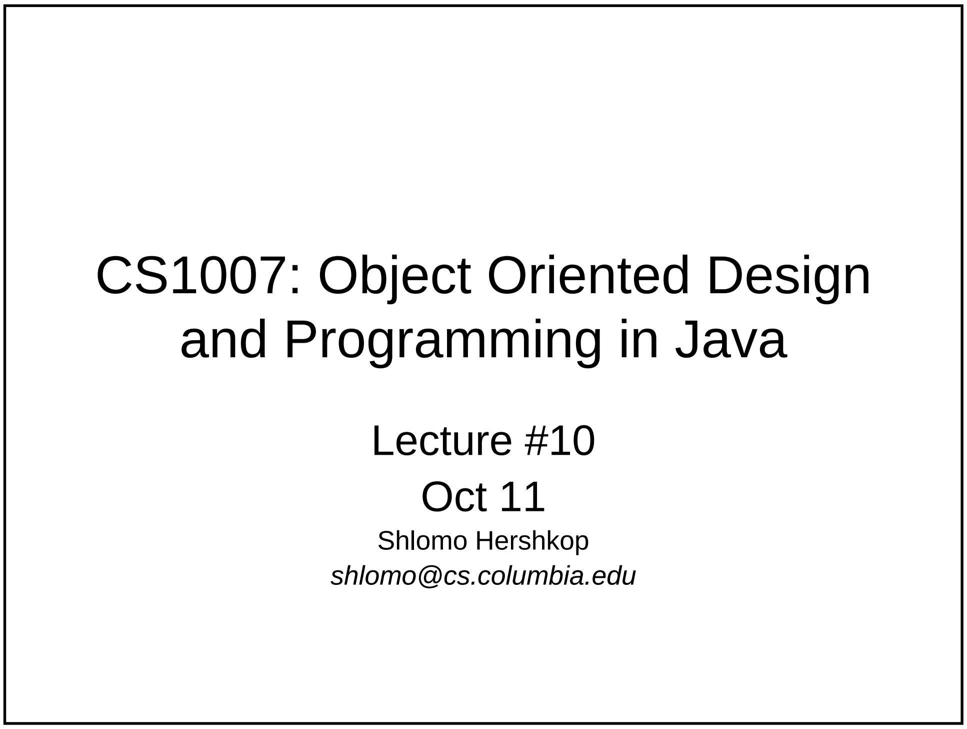


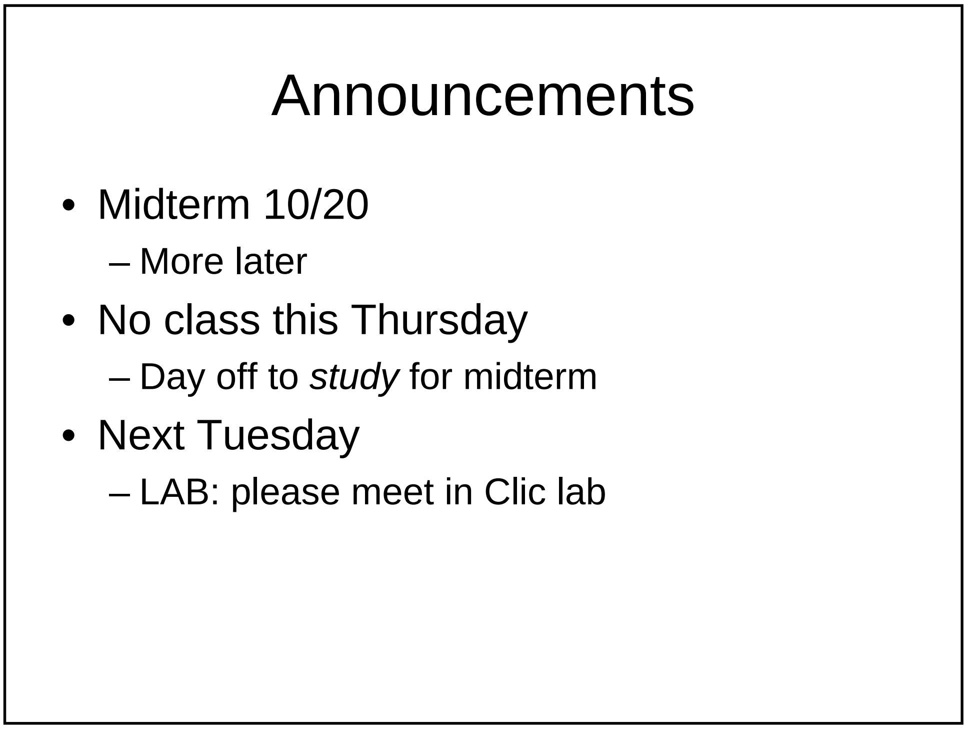
CS1007: Object Oriented Design and Programming in Java

Lecture #10

Oct 11

Shlomo Hershkop

shlomo@cs.columbia.edu



Announcements

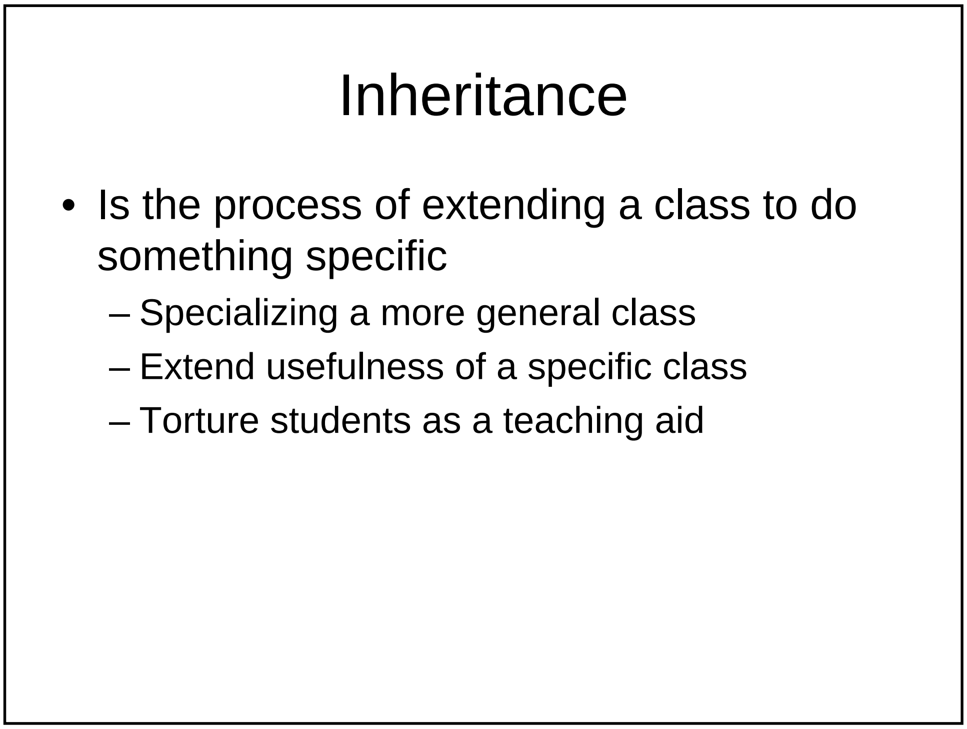
- Midterm 10/20
  - More later
- No class this Thursday
  - Day off to *study* for midterm
- Next Tuesday
  - LAB: please meet in Clic lab

Reading

- Chapter 1-4

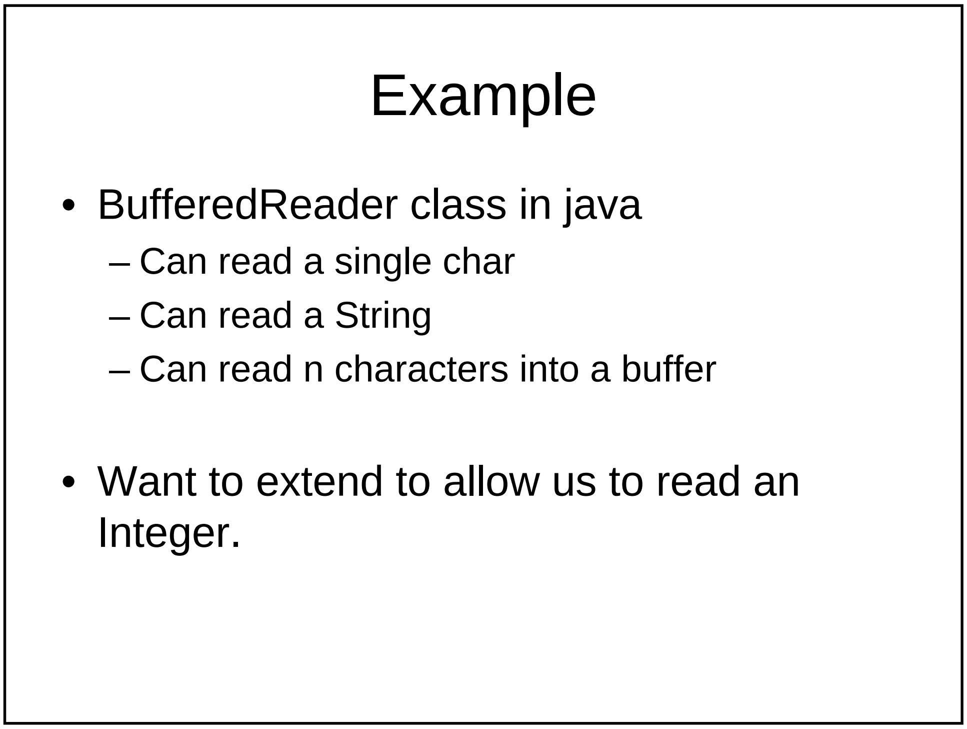
Fundamentals again

- Inheritance
- Polymorphism
- Functional inheritance
- Encapsulation



Inheritance

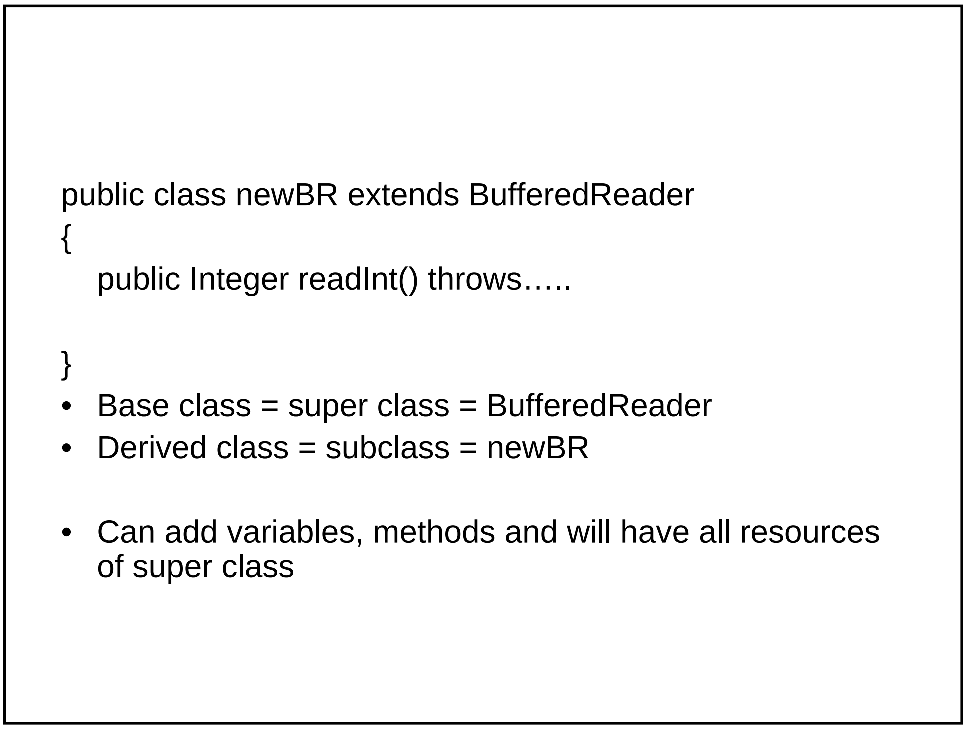
- Is the process of extending a class to do something specific
  - Specializing a more general class
  - Extend usefulness of a specific class
  - Torture students as a teaching aid



Example

- BufferedReader class in java
  - Can read a single char
  - Can read a String
  - Can read n characters into a buffer

- Want to extend to allow us to read an Integer.

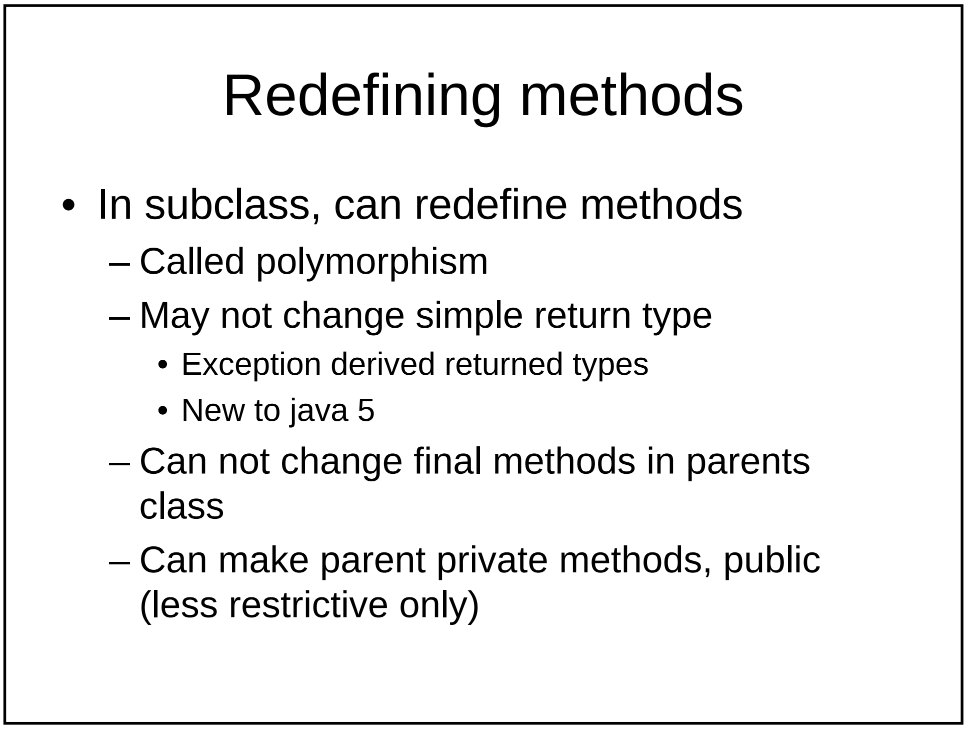


```
public class newBR extends BufferedReader
{
   public Integer readInt() throws…..

}
```

- Base class = super class = BufferedReader
- Derived class = subclass = newBR

- Can add variables, methods and will have all resources of super class



Redefining methods

- In subclass, can redefine methods
  - Called polymorphism
  - May not change simple return type
    - Exception derived returned types
    - New to java 5
  - Can not change final methods in parents class
  - Can make parent private methods, public (less restrictive only)

Overriding vs overloading

- Overriding:
  - When redefine method with exact arguments and return type in subclass
- Overloading:
  - Adding a method with the same name but new number of arguments
  - Result in 2 methods available in the subclass

Access rules

- Private variable in the base class are not accessible in the derived class.

- So how do we manipulate them?

Encapsulation

- `protected` modifier:
  - – Allows access to variable/method only in same, derived and package classes.
  - – Weak protection compared to public/private modifiers

Iterators

- Iterator is an object which allows you to step through (and maybe modify) a collection of objects in some order.
- Simplest example:
  - – Step through an array with a counter

Iterator<T> interface

- standard Iterator interface type

```
public interface
 Iterator<LineItem>
{
    boolean hasNext();
    LineItem next();
    void remove();
}
```

Danger

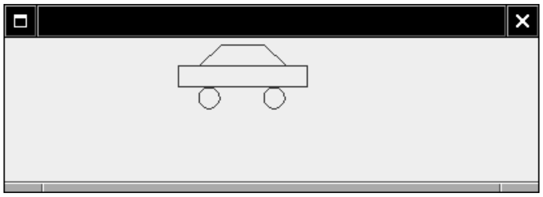
- Should next() return
  - Value
  - Reference

Back to chapter 5

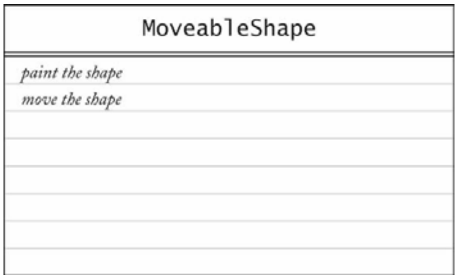
- We would like to create an animation sequence with an OO approach.

Defining a New Interface Type

- Use timer to move car shapes
- Draw car with CarShape
- Two responsibilities:
  - Draw shape
  - Move shape
- Define new interface type MoveableShape



CRC



MoveableShape

paint the shape

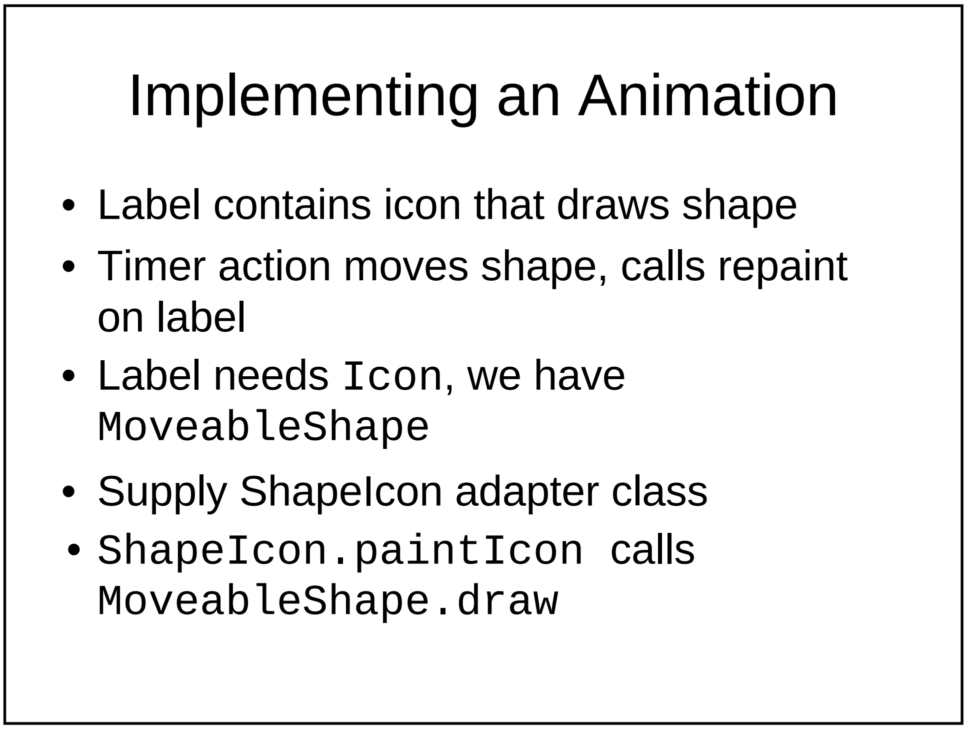
move the shape

- Name the methods to conform to standard library

```
public interface MoveableShape
{
   void draw(Graphics2D g2);
   void translate(int dx, int dy);
}
```

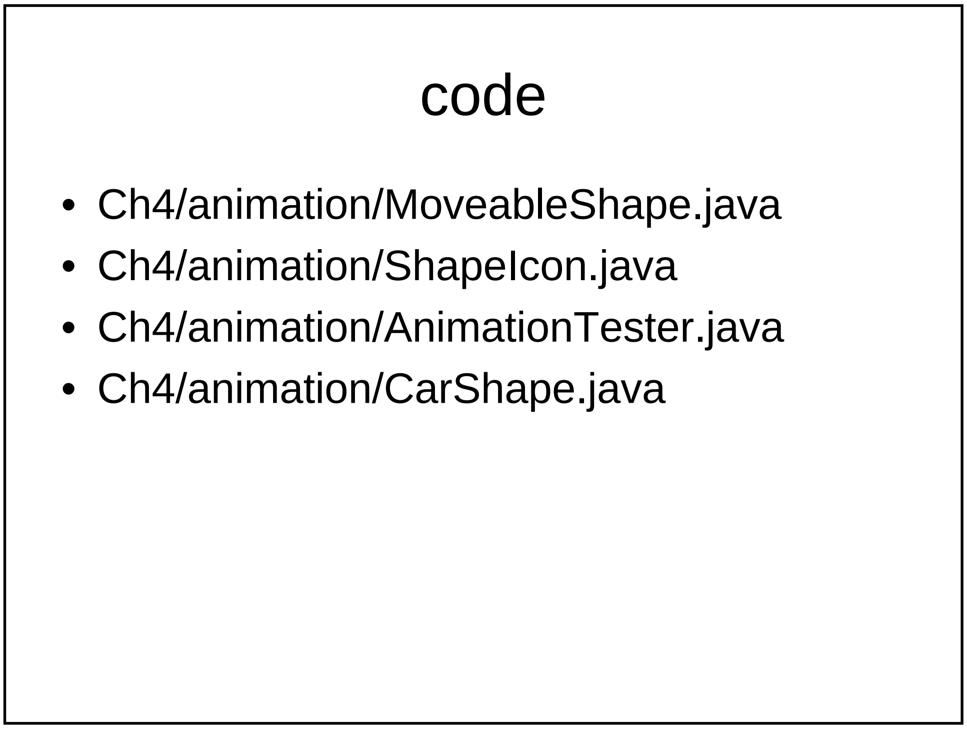
- CarShape class implements MoveableShape

```
public class CarShape implements MoveableShape
{
   public void translate(int dx, int dy)
   { x += dx; y += dy; }
   . . .
}
```



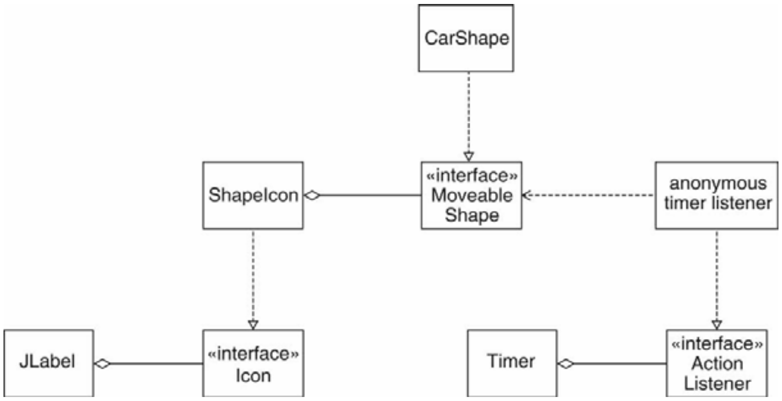
Implementing an Animation

- Label contains icon that draws shape
- Timer action moves shape, calls repaint on label
- Label needs `Icon`, we have `MoveableShape`
- Supply ShapeIcon adapter class
- `ShapeIcon.paintIcon` calls `MoveableShape.draw`



code

- Ch4/animation/MoveableShape.java
- Ch4/animation/ShapeIcon.java
- Ch4/animation/AnimationTester.java
- Ch4/animation/CarShape.java

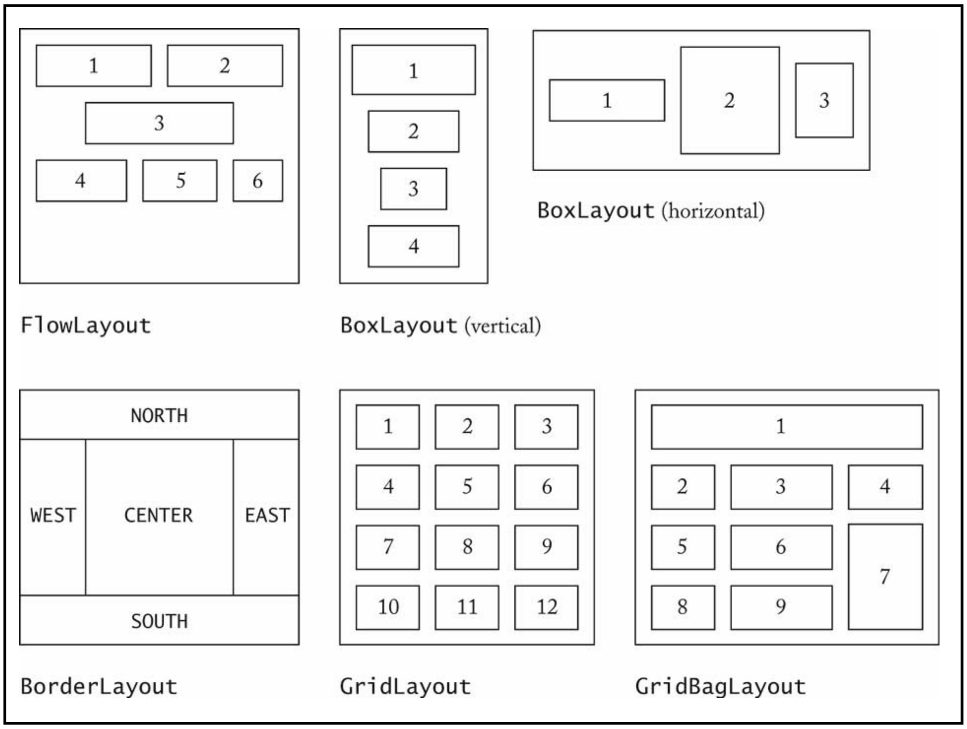


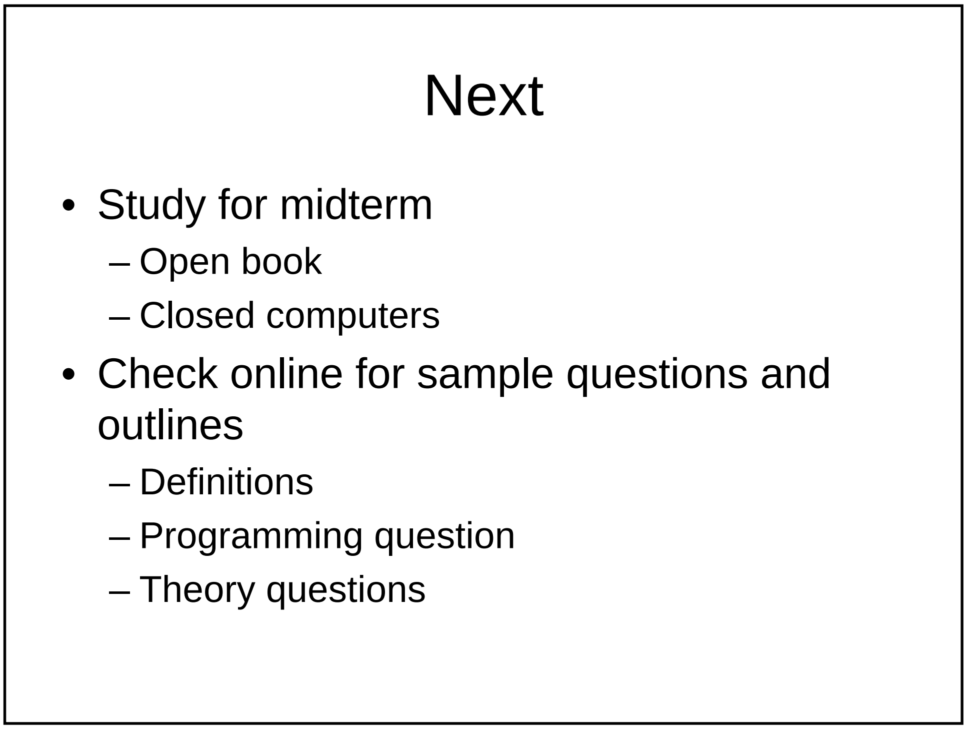
Layout Managers

- User interfaces made up of components
- Components placed in containers
- Container needs to arrange components
- Swing doesn't use hard-coded pixel coordinates
- Advantages:
  - Can switch "look and feel"
  - Can internationalize strings
- Layout manager controls arrangement

Layout Managers

- `FlowLayout`:
  - left to right, start new row when full
- `BoxLayout`:
  - left to right or top to bottom
- `BorderLayout`:
  - 5 areas, Center, North, South, East, West
- `GridLayout`:
  - grid, all components have same size
- `GridBagLayout`:
  - complex, like HTML table





Next

- Study for midterm
  - Open book
  - Closed computers
- Check online for sample questions and outlines
  - Definitions
  - Programming question
  - Theory questions