Working with Vectors
- Vector v = new Vector();
- v.addElement(new Song("Jingle Bells", 80));
- (v.firstElement()).print();
- Java looks for an Object.print()
- You must cast the Object as a Song:
- ((Song)(v.firstElement())).print();

Other Java Hints
- Be sure to delete your *.class files periodically. It doesn't always get the dependencies right.

Iterators
- Problem:
  - You can't access an element without moving the iterator.
- Solution: Use a ListIterator and follow every next() access with a previous().

Events
- Change in status that can initiate a response from the computer.
- Examples:
  - Click a mouse button --> Mouse event
  - Press a key --> Keyboard event
  - Move, hide, click in title bar of window --> Window event
- For your program to respond to events
  - Define a listener class
  - Attach an instance of that class (a listener) to a component

A Listener Class
public class myWindowListener extends WindowAdapter {
  public void windowClosing (WindowEvent e) {
    System.exit(0); return;
  }
}

A Listener Instance
public class App {
  public ... main (String argv []) {
    JFrame f = new JFrame("Window");
    f.setSize(300, 100);
    f.addWindowListener(new myWindowListener());
    f.show();
  }
}

Another Listener Instance
// from Sun's SwingApplication.java
frame.addWindowListener(new WindowAdapter() {
  public void windowClosing(WindowEvent e) {
    System.exit(0);
  }
});

// Anonymous inner class
+ Listeners Create "Delegation"
  - Components delegates responsibility to attached listeners.
+ Benefits of listeners:
  - Refine pre-existing classes, reuse behavior.
  - Can be attached to and deleted from components dynamically.
  - Keep track of state, such as number of times an event occurred (as in the Sun demo).

+ The Model-View-Controller (MVC) Architecture
  - Model: The data
  - View: The display of the data
  - Controller: How the UI reacts to events
+ Example: scroll bar
  - Model: min, max, current position.
  - View: What it looks like.
  - Controller: Drag and move, click on the ends.
+ Benefits: increased flexibility and reuse.
+ Where are they in your Java Swing classes?

+ Swing Simplifies the MVC Architecture
  - Just "Model-View" or "Component and UI Delegate"
  - Model: Info about the component's state
  - UI Delegate: How to draw the component and how to react to events.
+ Example: JTable
  + Model: JTable(data, columnNames), getModel(), getValueAt(i, j)
  + UI Delegate: Handle all GUI responsibilities, including displaying and handling events
    - addMouseListener()

+ Layout Managers
  - Organize your components within a logical hierarchy of containers.
  - Nest components into subsets and manage the organization and arrangement of each subset individually.
  + Layout managers are responsible for two tasks:
    - Arrange the components in a container
    - Calculate the sizes of containers

+ BorderLayout
  - From Sun’s Java Tutorial
  - Components can only be added to one of five regions.
  - The components are often other containers, such as JPanels