CS 349
Layout

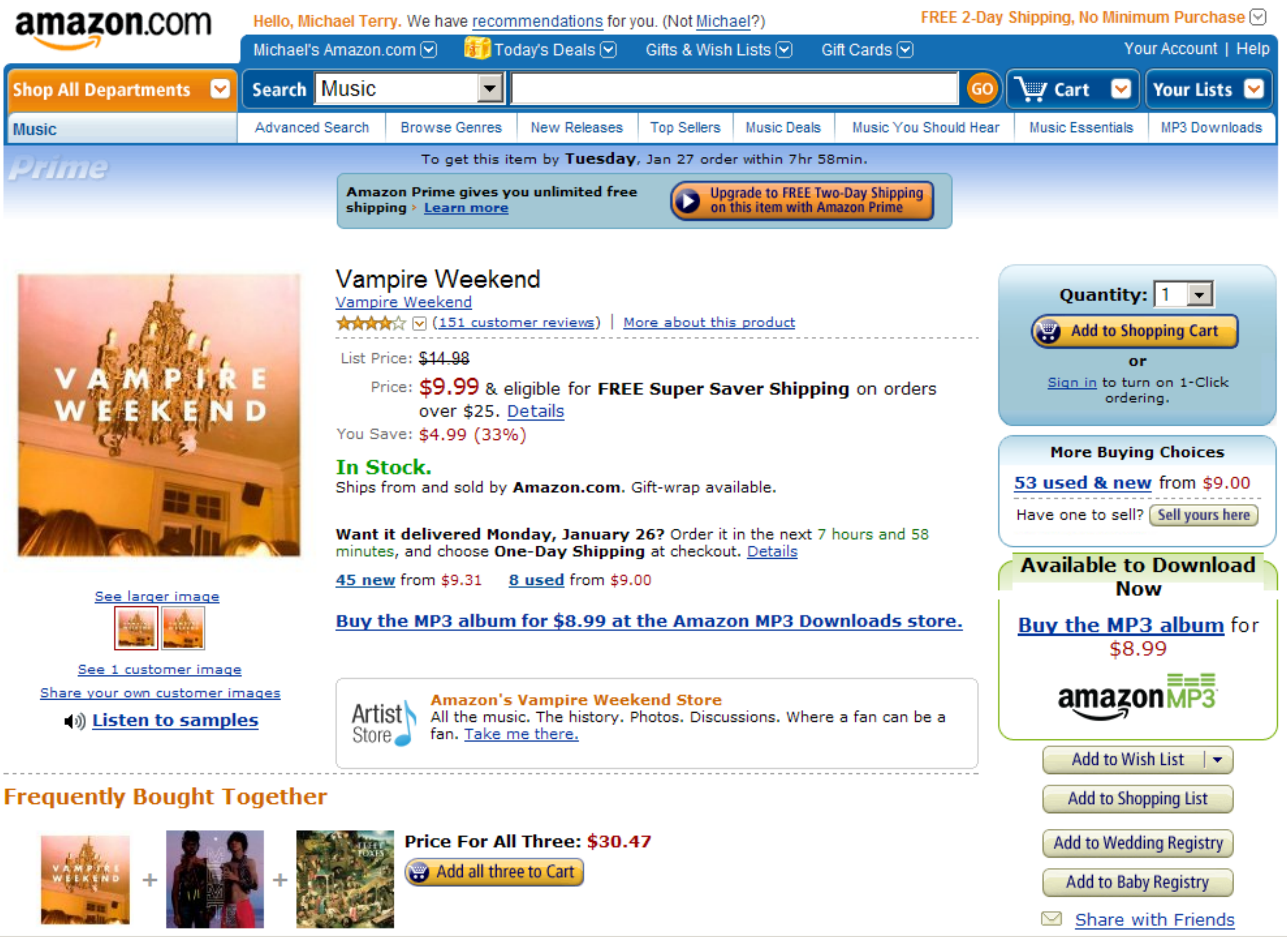
Byron Weber Becker
Spring 2009

Slides mostly by Michael Terry

University of Waterloo

Interface Layout

- Layout of components can be thought of as two processes:
  - Determining an optimal visual layout (ie, applying principles of good graphic design)
  - Applying algorithms that maintain that desired visual layout through resizes of window
- Demo: Amazon page
- This lecture focuses on the latter



amazon.com

Hello, Michael Terry. We have recommendations for you. (Not Michael?)

FREE 2-Day Shipping, No Minimum Purchase

To get this item by **Tuesday**, Jan 27 order within 7hr 58min.

Amazon Prime gives you unlimited free shipping › Learn more

Upgrade to FREE Two-Day Shipping on this item with Amazon Prime

Vampire Weekend

Vampire Weekend

★★★★☆ (151 customer reviews) | More about this product

List Price: ~~$14.98~~

Price: $9.99 & eligible for **FREE Super Saver Shipping** on orders over $25. Details

You Save: $4.99 (33%)

In Stock.

Ships from and sold by **Amazon.com**. Gift-wrap available.

Want it delivered Monday, January 26? Order it in the next 7 hours and 58 minutes, and choose **One-Day Shipping** at checkout. Details

45 new from $9.31 8 used from $9.00

Buy the MP3 album for $8.99 at the Amazon MP3 Downloads store.

See larger image

See 1 customer image

Share your own customer images

Listen to samples

Artist Store

Amazon's Vampire Weekend Store

All the music. The history. Photos. Discussions. Where a fan can be a fan. Take me there.

Quantity: 1

Add to Shopping Cart

or

Sign in to turn on 1-Click ordering.

More Buying Choices

53 used & new from $9.00

Have one to sell? Sell yours here

Available to Download Now

Buy the MP3 album for $8.99

amazon MP3

Add to Wish List

Add to Shopping List

Add to Wedding Registry

Add to Baby Registry

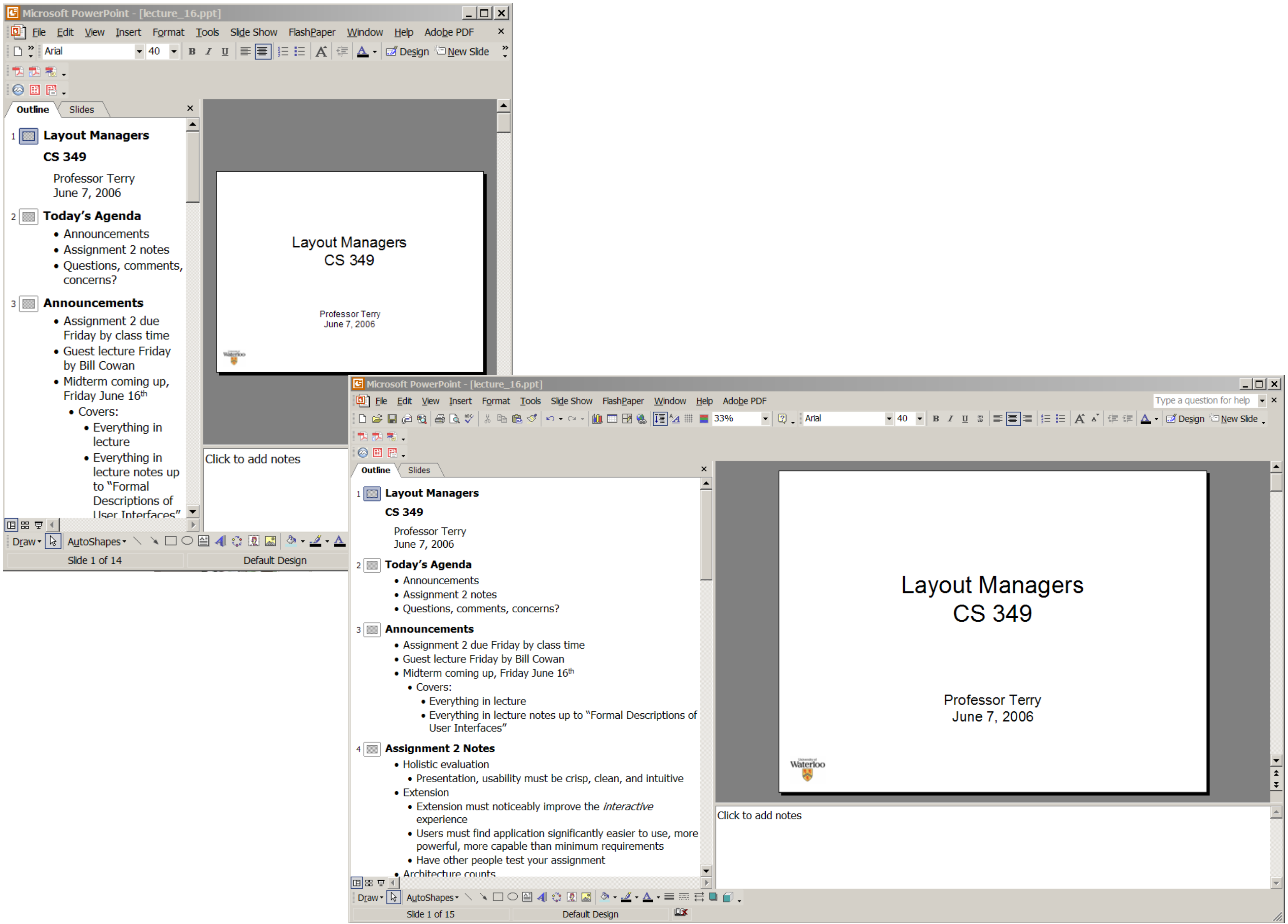
Share with Friends

Frequently Bought Together

Price For All Three: $30.47

Add all three to Cart

University of Waterloo



Microsoft PowerPoint - [lecture_16.ppt]
File Edit View Insert Format Tools Slide Show FlashPaper Window Help Adobe PDF
Arial 40
Design New Slide
Outline Slides

1 **Layout Managers**

CS 349

Professor Terry
June 7, 2006

2 **Today's Agenda**

- Announcements
- Assignment 2 notes
- Questions, comments, concerns?

3 **Announcements**

- Assignment 2 due Friday by class time
- Guest lecture Friday by Bill Cowan
- Midterm coming up, Friday June 16th
  - Covers:
    - Everything in lecture
    - Everything in lecture notes up to "Formal Descriptions of User Interfaces"

Layout Managers
CS 349

Professor Terry
June 7, 2006

Click to add notes

Draw AutoShapes
Slide 1 of 14
Default Design
Microsoft PowerPoint - [lecture_16.ppt]
File Edit View Insert Format Tools Slide Show FlashPaper Window Help Adobe PDF
Type a question for help
33% Arial 40
Design New Slide
Outline Slides

1 **Layout Managers**

CS 349

Professor Terry
June 7, 2006

2 **Today's Agenda**

- Announcements
- Assignment 2 notes
- Questions, comments, concerns?

3 **Announcements**

- Assignment 2 due Friday by class time
- Guest lecture Friday by Bill Cowan
- Midterm coming up, Friday June 16th
  - Covers:
    - Everything in lecture
    - Everything in lecture notes up to "Formal Descriptions of User Interfaces"

4 **Assignment 2 Notes**

- Holistic evaluation
  - Presentation, usability must be crisp, clean, and intuitive
- Extension
  - Extension must noticeably improve the *interactive* experience
  - Users must find application significantly easier to use, more powerful, more capable than minimum requirements
  - Have other people test your assignment
- Architecture counts

Layout Managers
CS 349

Professor Terry
June 7, 2006

Click to add notes

Dynamic Layout

- Windows are dynamic, can be resized
- Through any resize, we wish to:
  - Maintain *consistency* in interface's presentation
  - Preserve *affordances* communicated through interface's layout
  - Preserve overall visual layout found to be ideal in user testing
- Need to dynamically modify allocation of space, locations of objects in interface

Dynamic Layout

- Dynamic layout a process of:
  - Specifying components
  - Specifying desired *constraints* for the components and their relationships with respect to one another
  - Attempting to satisfy those constraints
- Dynamic layout has applications in:
  - User interface design
  - Document layout (eg, TeX)
  - Information visualization

Layout in Java

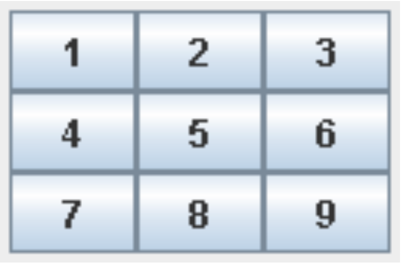
- *Containers* maintain collection of components
- Container (eg, JPanel) can utilize a *LayoutManager*

  ```
  myPanel.setLayout(new GridLayout(1,1))
  ```

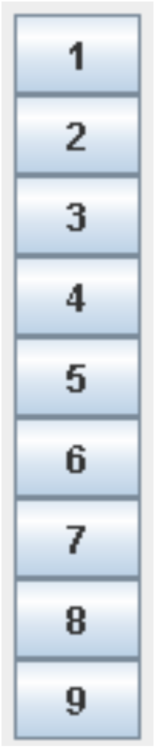
- LayoutManager a *strategy pattern* that factors out process of positioning, sizing components within that container
- Can vary LayoutManager independently of container, components

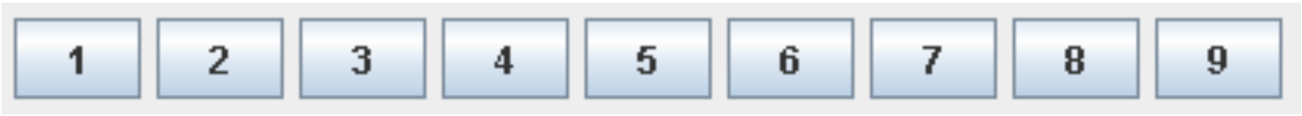
Layout Example

Grid Layout



Box Layout (y axis)





Flow Layout

Java Layout Demo

- LayoutDemo.java
- Available on CS349 Resources page.

General Layout Strategies

- Fixed layout
- Intrinsic size
- Struts and springs
- Variable intrinsic size
- Constraints

Fixed Layout

- Components are of a fixed size, position
- In Java, achieved by setting LayoutManager to *null*
- Where/when is this practical?
- How can it break down even when windows aren't resized?

Intrinsic Size

- Query each item for its preferred size
- Grow the component to perfectly contain each item
- A bottom-up approach where top-level component’s size completely dependent on its contained components
- Example LayoutManagers in Java that use this strategy
  - BoxLayout, FlowLayout
- Examples of use in interface design?
- Special needs?

Intrinsic Size Uses and Needs

- A list of items that grows to accommodate every item
  - Menus
  - Address book list
- Text documents
  - Text panel continually grows to contain text
- If can grow arbitrarily large, could overflow bounds of screen
  - Scrollbars, cascading menus help address these problems

Struts and Springs

- Layout specified by marking aspects of components that are fixed vs. those that can “stretch”
- Strut defines a fixed length (width/height)
  - Specifies invariant relationships in a layout
- Spring defines a space that “pushes” on nearby edges
  - Specifies variable relationships
- Example LayoutManagers in Java
  - SpringLayout

Struts and Springs Uses

- One of the most common strategies, especially in user interface builders
- Provides easily accessible metaphors for people performing layout
- Cocoa demo…

Variable Intrinsic Size

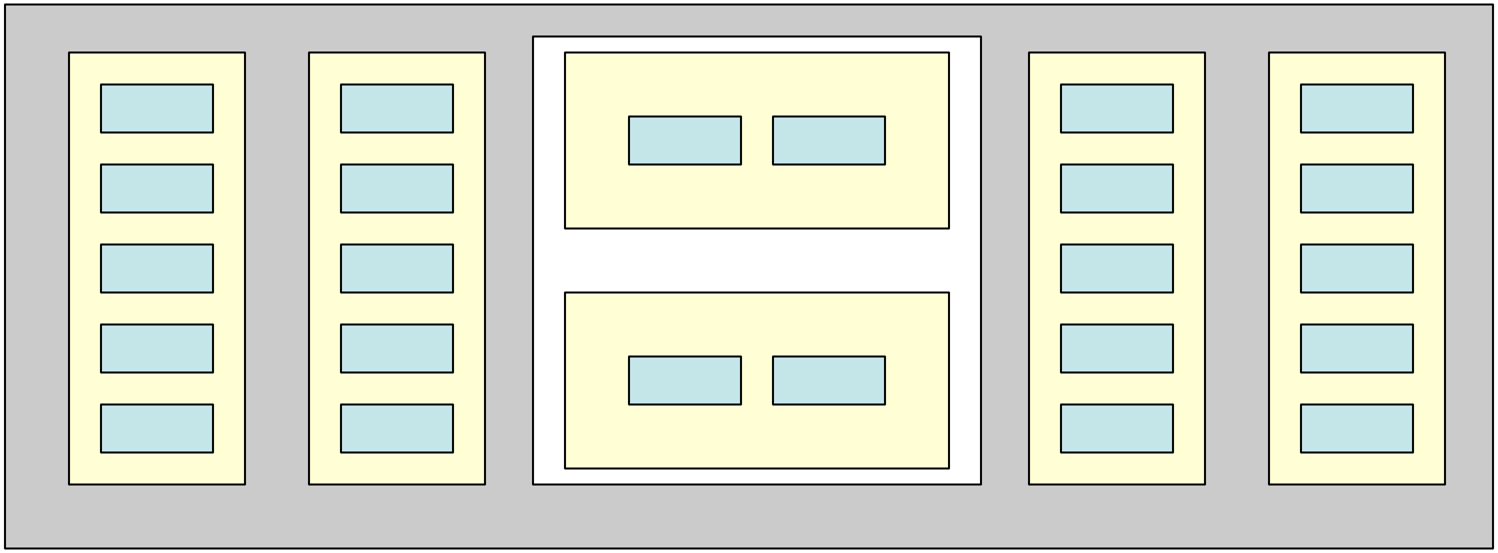
- Layout determined in bottom-up and top-down phases
- Bottom-up phase:
  - Container asks each child for its preferred, minimum, maximum sizes
  - Values used to partition the container's space
- Top-down phase:
  - Children are sized and told to lay themselves out in space specified
- Example LayoutManagers in Java
  - GridBagLayout

Tips and Strategies

- javax.swing.Box has number of useful items that can be used in *any* layout manager
  - “Glue”
    - `Box.createHorizontal/VerticalGlue()`
    - Similar to notion of “springs”: Expands to fill space
  - Struts
    - `Box.createHorizontal/VerticalStrut()`
  - Rigid areas
    - `Box.createRigidArea()`

Tips and Strategies

- Cluster components into panels based on layout needs
- Provide layout manager for each panel



Tips and Strategies

- Define your own layout manager if necessary

```
public interface LayoutManager
{       void addLayoutComponent(String name, Component comp);
        void removeLayoutComponent(Component comp);
        Dimension preferredLayoutSize(Container parent);
        Dimension minimumLayoutSize(Container parent);
        void layoutContainer(Container parent);
}
// LayoutManager2 has methods for specifying constraints
```

Constraints

- Specify the mathematical relationships between components of the interface.
  - All of the layout managers have constraints to some degree.
  - This is meant to be more general.
- Prefuse takes it to a new level
  - Demo
    - AggregateDemo
    - GraphView
    - Fisheye Menu
    - TreeView
  - See prefuse.org for downloads, videos, etc.
    - Importing into Eclipse is particularly straight-forward if you follow the instructions!