CSSE 220 Day 8

Drawing Objects
Event Handling
Layout Managers

CSSE 220 Day 8

- BigRational exercise is due at 8:05 AM.

Your questions about …

- Java
- Reading from the textbook
- Homework
- etc.

Some Classes That We will be Using

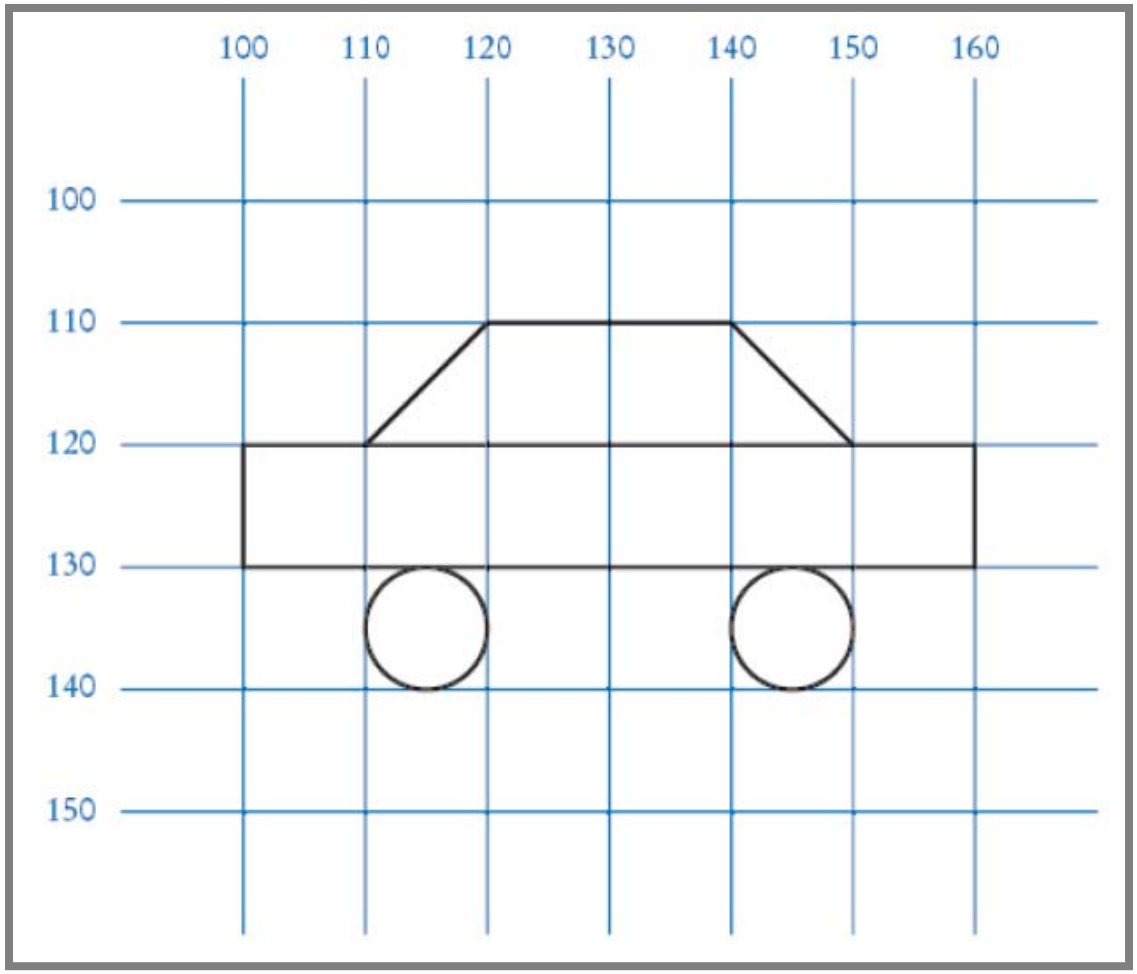
Class	What it is
JFrame	a top-level window
JComponent	a region where we can draw; also parent of many other widget classes
JButton	a JComponent representing a button. When clicked, an action can happen
JLabel	a place to put text in a window
JTextfield	a place for the user to enter text
JPanel	a JComponent that can be used as a container for organizing other widgets
Graphics	an object that can draw things on a JComponent. We never have to create this object; it is provided to us by the system
Graphics2D	a more "object-oriented" graphics object
JOptionPane	Request a single line of input from the user,

GUI recap

- So far we have
  - Created A JFrame to serve as a top level window.
  - Added a subclass of JComponent to the JFrame.
  - Drawn in the component by writing code in the paintComponent() method.
  - Used the Graphics2D object passed to paintComponent by the system.
  - Gotten that object to draw shapes by using Graphics2D's **draw** and **fill** methods.
  - Drawn text and modified colors
  - Constructed colors based on RGB values.

Live Demo – continued

- Do the Car example.


100
110
120
130
140
150
160
100
110
120
130
140
150

Event-driven programming

- The flow of programs we have written so far is controlled by the program itself.
- They only accept input when they ask for it.
- In most modern GUI programs, the user is in control.
  - Once it is initialized, the program does things in response to **events.** Examples:
    - A button or menu item is clicked (ActionEvent)
    - A key is pressed (KeyEvent)
    - The mouse is clicked (MouseEvent)
    - The mouse is moved (MouseMotionEvent)

Why There Are Listeners

- "Most Programs don't want to be flooded by boring events"

  – Cay Horstmann
- If I click the mouse on a button
  - The mouse moves over the button (mouseEntered)
  - Mouse moves within button's borders (mouseMoved)
  - Mouse button is pressed
  - Mouse button is released
- And I don't really care about any of that mouse stuff.
- So I choose not to listen for mouse events.
- I listen for an ActionEvent on the button.

Some demo programs we will write

- ButtonTester/ClickListener
  - About as simple as we can get and still respond to clicks. (from *BigJava*)
  - A separate ActionListener class.
- OneButton
  - Frame is filled with a button that changes colors when clicked.
- FollowTheMouse
  - Draw a small circle where the user clicks.
- OneButton2
  - Make the button smarter …
- ClickCounter
  - Clicking a button causes the contents of a label to change.
  - The Frame is the "boss" and the ActionListener.
- Multiplier
  - Get two numbers from textfields and display their products.

To do before Session 9

- The next reading assignment.
- ANGEL quiz 6 (over Section 5.5)
- Swing Warmup. Three short but not trivial problems.
- Experiment with some of the things we did in class.