

GUIs Part Two

Images

Check Boxes

Radio Buttons

Text Areas

Drawing

MORE COMPONENTS

Images

- Create an ImageIcon object
 - ImageIcon icon = new ImageIcon(fileName);
 - fileName is a String (example: "picture.jpg")
- Put the image inside of a label:
 - JLabel imageLabel = new JLabel(icon);
 - imageLabel.setIcon(image2);
- Note: in eclipse, image files go in the same folder that the src and bin folders are in (so above the src files).
- Note: resize images *outside* of Java

Images with Labels

- `JLabel labelPicText = new JLabel(imageIcon);`
- `labelPicText.setText("the text");`
- To orient text to images
 - `setHorizontalTextPosition(int)`
 - `SwingConstants.LEFT` `RIGHT` `CENTER`
 - `setVerticalTextPosition(int)`
 - `SwingConstants.TOP` `BOTTOM` `CENTER`

Label Alignment

**Horizontal: Left
Vertical: Top**

**Horizontal: Left
Vertical: Center**

**Horizontal: Left
Vertical: Bottom**

**Horizontal: Center
Vertical: Top**



**Horizontal: Center
Vertical: Bottom**

**Horizontal: Right
Vertical: Top**

**Horizontal: Right
Vertical: Center**

**Horizontal: Right
Vertical: Bottom**

Practice

- Create a GUI that displays a picture.
 - Add a label that displays a description.
 - Make the label show up centered below the picture.

JCheckBox

- A component
- A *check box* is a button that can be toggled on or off (yes or no, selected or unselected, etc.)
- The constructor takes just a String or a String and boolean (if you want the box to be selected when created)
- Links to an ActionListener (with actionPerformed method)
- Similar to a text field, we can register a listener but we can also just always check whether the box is checked or not
- Methods
 - isSelected
 - setSelected(boolean)

Practice

- Modify the picture program so that the user can decide whether or not to display the picture label.

JRadioButton

- A component
- Radio buttons represent a set of mutually exclusive options (like multiple choice)
 - Each choice is a radio button
- Buttons are linked together into a ButtonGroup
 - Not visual- used just to group together the options- when one in the group is on, all others are off
- The constructor takes just a String or a String and boolean (if you want one choice to be selected when created)
- Links to an ActionListener (with actionPerformed method)
- We can register a listener or just check whether a button is selected
- Methods
 - isSelected
 - setSelected(boolean)

JRadioButton

- Create each button
- Add **each** button to a ButtonGroup (not visual)
- Add **each** button to the panel (visual)
- Link a register to **each** button

Practice

- Modify the picture program so that the user can display either the dog picture or the cat picture.

JTextArea

- A component
- Text fields takes a single line; text areas have multiple lines
- Can be used for input or output
- Links to an ActionListener (with actionPerformed method)
- To make scrollable, create a JScrollPane from the area and add that to the panel

```
textArea = new JTextArea(5, 10); // 5 rows, 10 columns
```

```
JScrollPane scrollPane = new JScrollPane(textArea);
```

```
mainPanel.add(scrollPane);
```

- **Methods**
 - getText()
 - setText(String)
 - append(String) // can use with “\n”
 - isEditable()
 - setEditable(boolean)

Practice

- Write a program to allow a user to enter in a series of names.
 - Store the names in an ArrayList.
 - Display the names in a text area.
 - Add buttons so that the user can sort or clear the list.
 - Update the display accordingly.
- Write a program to allow the user to input the information listed below.
 - Use labels to provide instructions.
 - Include a “submit” button. When the user clicks the button, display a summary of their information in a text area.
 - User inputs:
 - Name (use a text field)
 - Age (use a text field)
 - Gender (use radio buttons)
 - Whether the person is a student (use a checkbox)

DRAWING

Drawing Shapes

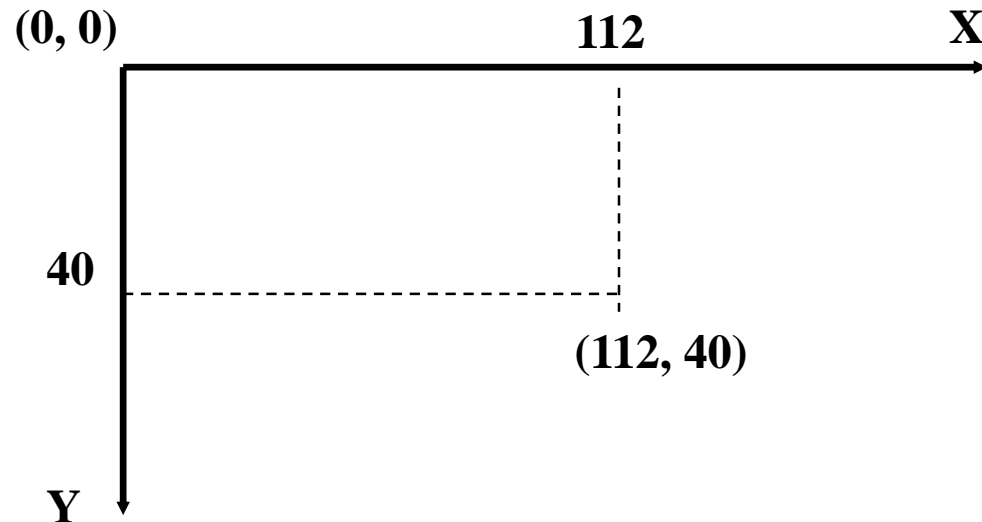
- The `Graphics` and `Graphics2D` classes allow you to draw shapes using various methods.
 - The method parameters specify coordinates and sizes.
- Shapes with curves are usually drawn by specifying that shape's *bounding rectangle*.
- Every drawing surface has a *background color* and every graphics object has a *foreground color*.

Drawing Shapes

- A shape can be filled or unfilled.
- Shapes with curves are usually drawn by specifying that shape's *bounding rectangle*.

Java Coordinate Space

- Use a coordinate space
 - Origin is in the top-left corner.
 - x-values increase to the right
 - y-values increase down



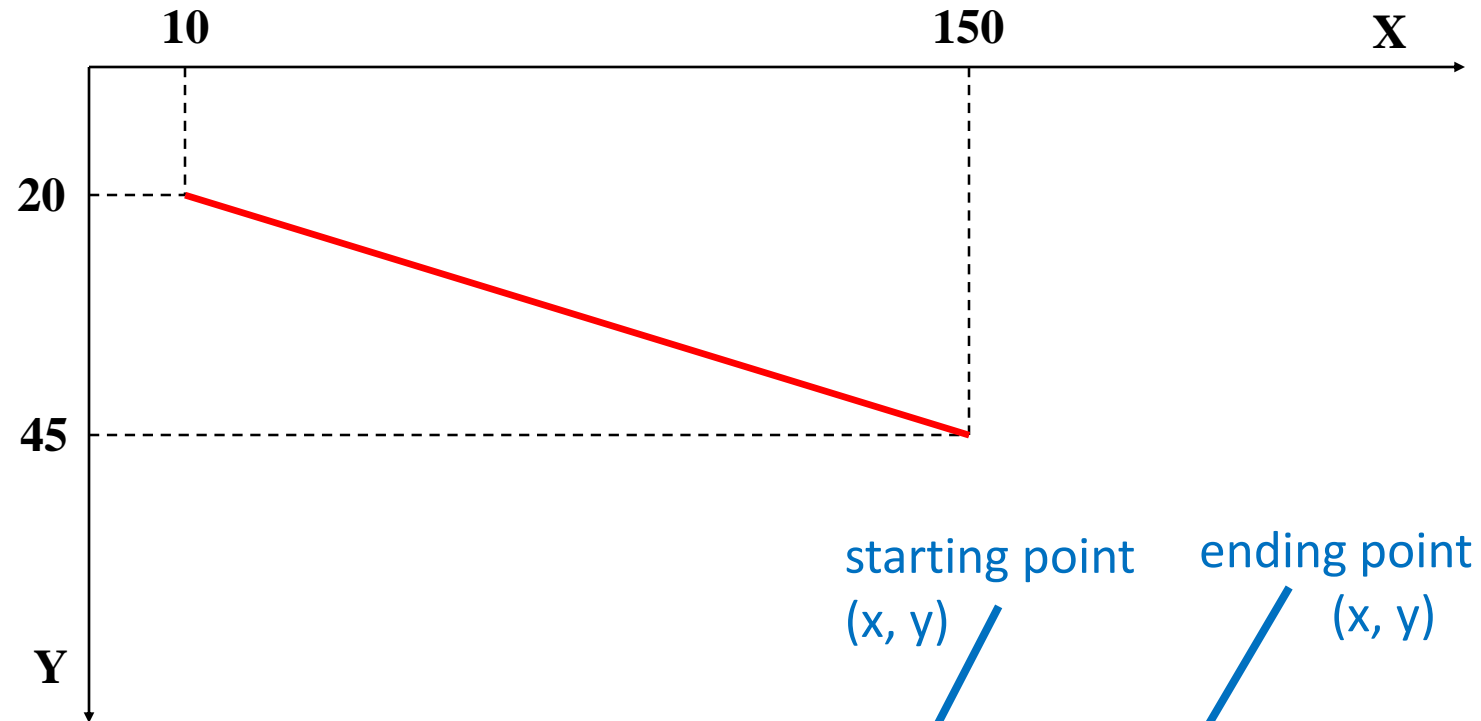
Java 2D Library

- Use an object of the Graphics2D class:

```
Graphics2D g2 = (Graphics2D) g;
```

- Useful classes:
 - Point2D
 - Line2D
 - RectangularShape
 - Rectangle2D
 - Ellipse2D
- g2.draw(...)

Drawing a Line



starting point
(x, y)

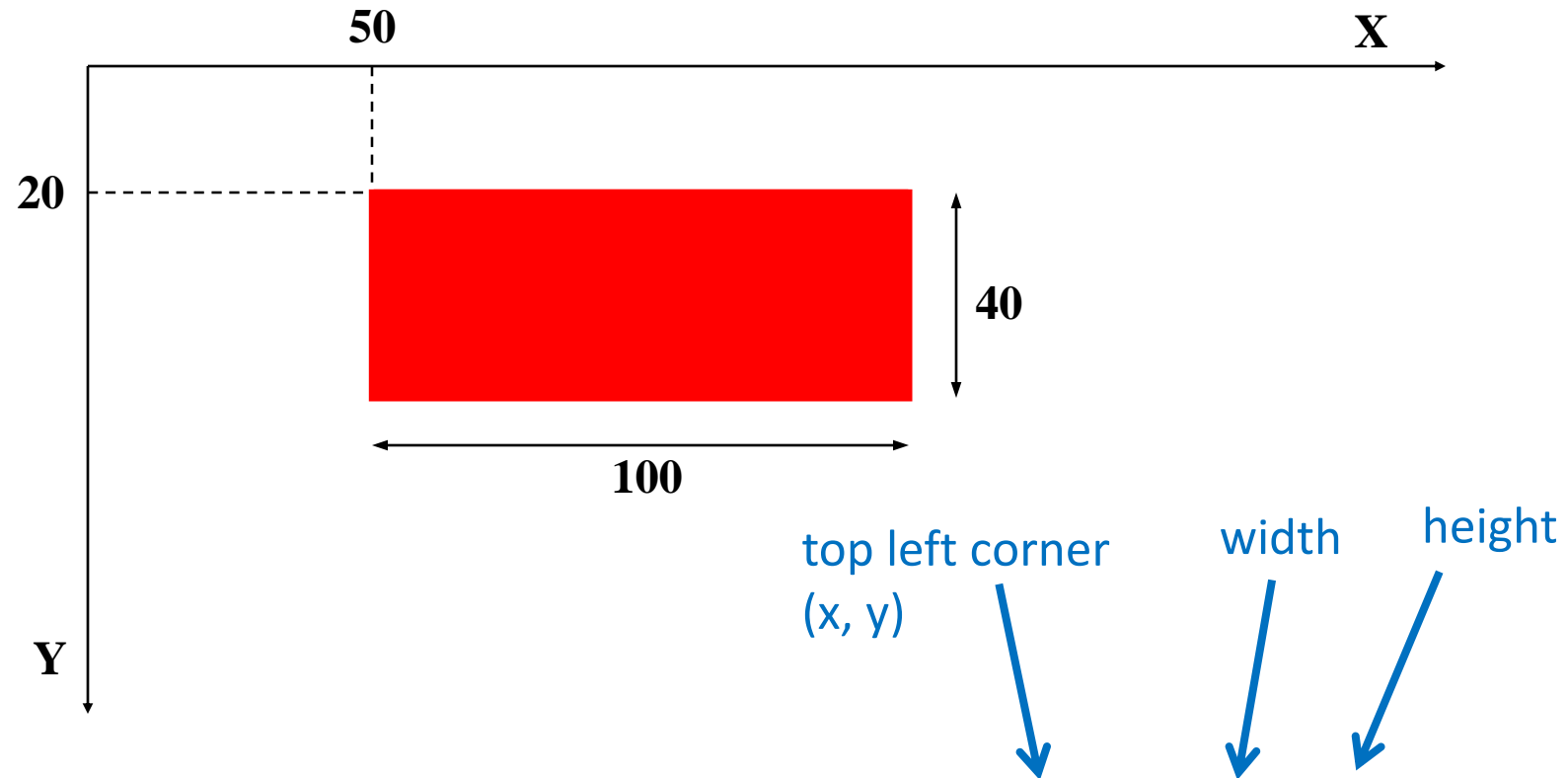
ending point
(x, y)

```
g2.draw(new Line2D.Double(10, 20, 150, 45));
```

(could also take two Point2D objects)

```
pen.drawLine(10, 20, 150, 45);
```

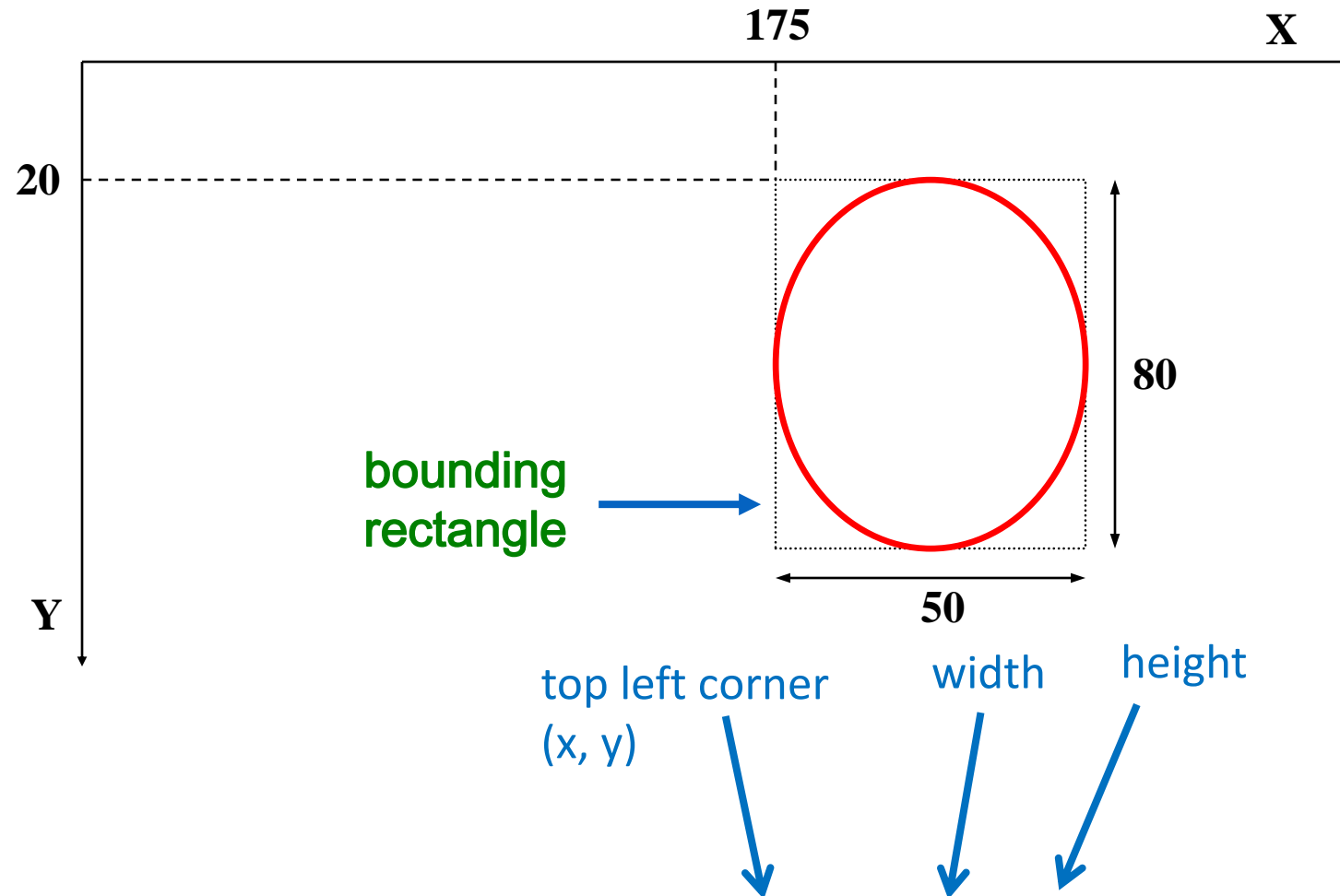
Drawing a Rectangle



```
g2.draw(new Rectangle2D.Double(50, 20, 100, 40));
```

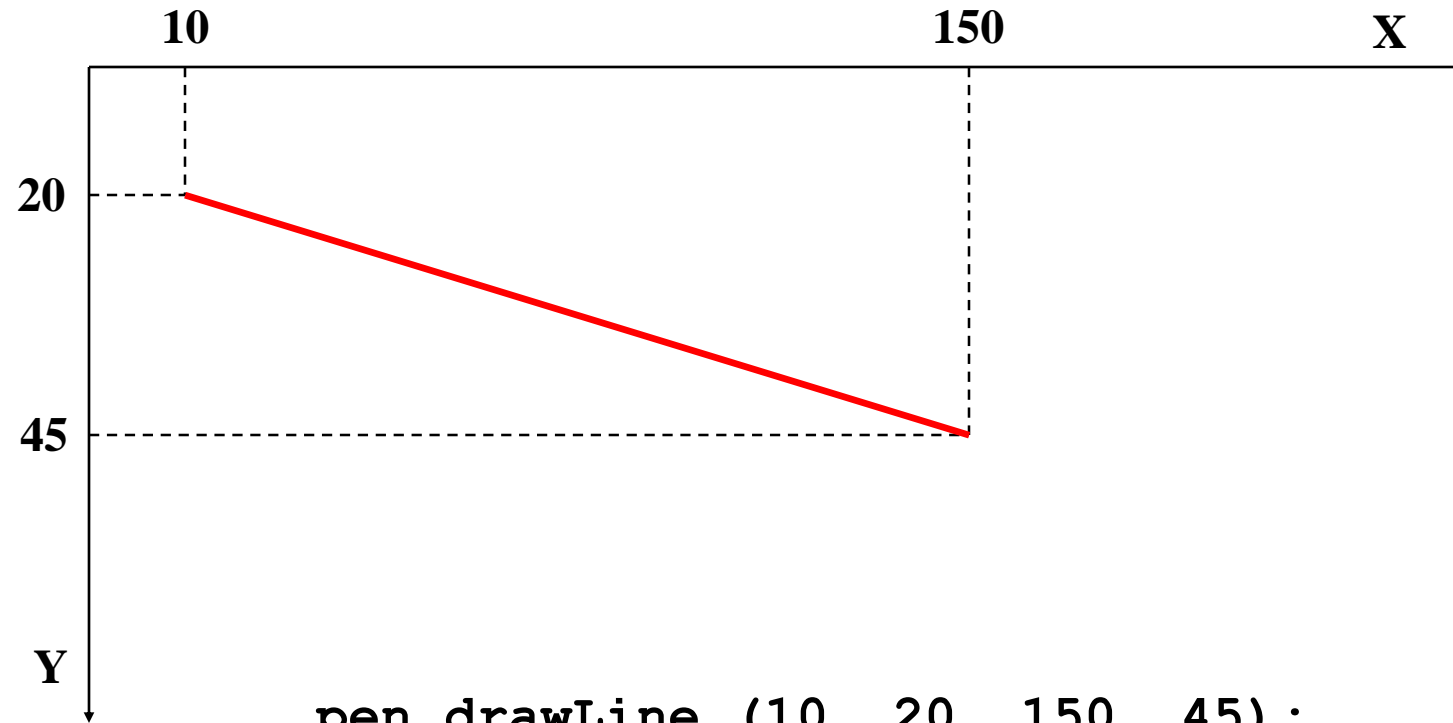
```
pen.fillRect(50, 20, 100, 40);  
pen.drawRect(50, 20, 100, 40);
```

Drawing an Ellipse



```
g2.draw(new Ellipse2D.Double(175, 20, 50, 80));  
pen.drawOval(175, 20, 50, 80);
```

Drawing a Line



```
pen.drawLine (10, 20, 150, 45) ;
```

or

```
pen.drawLine (150, 45, 10, 20) ;
```

starting point

ending point

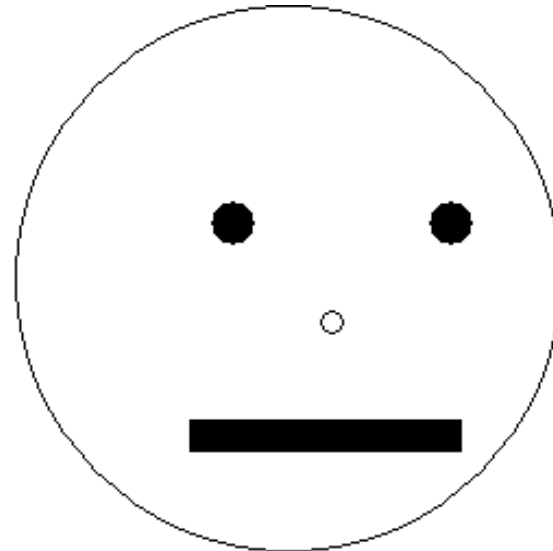
Drawing on Panels

- We can only draw on panels, not on frames.
- Instead of creating a class that is a frame, we create a class that is a panel.
 - Then we put our drawing code in the `paintComponent` method with this **exact** header and first line:

```
public void paintComponent(Graphics pen) {  
    super.paintComponent(pen);
```
- Whenever we want to update the display, we call `repaint()`.
 - Behind the scenes, this calls our `paintComponent` method.
- In main, we create a regular `JFrame` and add our panel to it.

Practice

- Use the GUIDrawingTemplate to create a GUI that draws a face with a happy message.



Have a nice day!

Troubleshooting Checks (List in Progress)

- Did you add the component (button) to the container (panel)?
- Did you add the panel to the frame?
- Did you register a listener to **each** button?
- Did you add **each** radio button to a **ButtonGroup** **and** a panel?
- Did you accidentally re-declare a component inside the constructor?
- Did you create the correct frame object in main?
- **Drawing:**
 - Did you invoke `super.paintComponent(pen)`?
 - Did you invoke `repaint()` when you want to update the display?

Practice

- Modify the face program.
 - Allow the user change the color of the face to a new color.
 - For now, the button will be on the drawing space.
- Create a GUI that draws 50 circles of random color and size.
 - Add a redraw button.
 - Add a checkbox so the user can select filled or unfilled.
 - Add radio buttons for the circles to be drawn in random colors or just a single color.

