GUIs
Graphics
Event Handling
Part One

INTRODUCTION

Containers and Components

- Components
  - Display information to the user
  - Interact with the user
  - Examples: button, text field, label, image, drop-down menu
- Containers
  - Hold components
  - Hold other containers
  - Examples: frames and panels

JFrame

- A container
- The top-level window that opens
- Frame hold other components- they are just a shell
  - To add things to the frame, add to the frame's *content pane*
- Often, your GUI program *is a* frame- it's a special kind of frame that contains certain containers and components.
- To display your GUI, you create your frame, add things to it, and make it visible
- Methods you can use:
  - setTitle(String)
  - setSize(int, int) // pixels
  - getContentPane()

JPanel

- A container
- The actual space on which we add components or draw things
  - Think of the frame as the window pane and the panel as the actual glass
- We often don't *see* the panel, we just see the component we add to the panel
  - A panel cannot be displayed on its own- it must be added to a frame
- To use a panel, create it, add it to the frame's content pane, then add things to the panel
- Methods you can use:
  - setBackground(Color)
  - add(component)

JLabel

- A component
- Displays text, and image, or both
- Methods
  - setText(String)
  - setForeground(Color)
  - setIcon(ImageIcon)

Color

- Use the constants
  - Example: Color.RED, Color.BLUE
- Create your own color:
  - Example: Color myColor = new Color(5, 194, 37);
  - The three parameters are the Red, Green, Blue components between 0 and 255 (inclusive)

Practice

- Use the GUITemplate file to create a Hello World program.

EVENT HANDLING

Event Handling in Java

- Event Source
  - Where the user's action comes from
  - We create this (often as a component)
  - Examples: a button, a text field
- Event Object
  - The action taken by the user
  - Automatically create by Java
  - Examples: a mouse click, a button press
- Event Listener
  - The response to the user's action
  - We create this (as a class)
  - Examples: get the user's text and display it back to them, perform a calculation

Event Handling in Java

- When the user interacts with an *Event Source*, an *Event Object* is automatically created and passed to the registered *Event Listener*, which takes the appropriate action.

Event Sources

1. Declare a component as instance data
2. Initialize the component in the constructor
3. Customize the look of the component
4. Register a listener to the component
5. Add the component to the panel

```
private JButton submitButton; // instance data

// in constructor:
submitButton = new JButton(“Submit”);
submitButton.setEnabled(false);
submitButton.addActionListener(new SubmitButtonListener());
mainPanel.add(submitButton);
```

Event Objects

- Automatically generated when the user acts
- Contain information about the action
  - Example: the source of the click, the location of the mouse

Event Listeners

- A class that implements a listener interface
  - The method in the class is sent an Event Object as a parameter
  - That object can be used to determine the source (if necessary)
- Executes code to respond to the user's action

```
private class ButtonListener implements ActionListener {
     public void actionPerformed(ActionEvent event) {
          if(event.getSource()==submitButton) {
               // take some action here!
          }
     }
}
```

JButton

- A component
- Can display text and/or an image
- Links to an ActionListener that implements this method:
  - public void actionPerformed(ActionEvent event)
- Methods
  - setText(String)
  - setEnabled(boolean)
  - isEnabled()

Practice

- Modify the Hello World program to add a button.
  - When the user clicks the button, change the text displayed on the label.
  - Disable the button after it is clicked.

JTextField

- A component
- Can display a single line of text
- The user can enter text in as a single line
- Create with either default text and/or a width (in columns)
  - The user can enter longer text, it just won't be visible
- The event object is generated when the user clicks "enter"
  - But you can always retrieve the text at any time!
- Links to an ActionListener that implements this method:
  - public void actionPerformed(ActionEvent event)
- Methods
  - getText() // returns String only!! Use the Integer.parseInt or Double.parseDouble methods to convert to numbers
  - setText(String)
  - isEditable()
  - setEditable(boolean)

Practice

- Modify the Hello World program to add a text field.
  - The user can enter their name.
  - When they click the button, the name is displayed as part of the label.
  - Then modify so that the display is changed when either the button is clicked *or* the user clicks enter in the text field.

Summary: Creating GUI Components

1. Declare the component as instance data
2. In the constructor:
   a. Initialize the component
   b. Customize it (optional)
   c. Add the component to a panel
   d. Register a listener (only if the user interacts)
3. Write your listener class (if needed)

Practice

- Write a GUI that keeps track of the number of times a button is clicked.
  - Display the count in a label.
  - Modify the program to change the background color to a new random color with every fifth click.
- Write a GUI to allow the user to click buttons to increment or decrement a counter by 1.
  - Modify the program to allow the user to enter a new interval so that increment and decrement will use that interval (instead of 1).
- Write a GUI to allow the user to enter in a series of integer numbers.
  - A “calculate” button will display the total, min, max, and average of the numbers.
  - A “clear” button will allow the user to start over.
- Write a GUI Hi Lo guessing game.
  - The user enters a number and you display whether it is too high, too low, or correct.
  - Modify the program with a “play again” button.

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 the button?
- Did you accidentally re-declare a component inside the constructor?
- Did you create the correct frame object in main?