6.1 – GUI Elements

- Command-line applications, interact with the user through simple prompts and feedback.
- ... but they lack the rich user experience.
- With GUI, the user is not limited to responding to prompts in a particular order and receiving feedback in one place.
- Three kinds of objects are needed to create a GUI in Java:
  - components
  - events
  - listeners

Simple GUI

- Composed of simple JComponents:
  - JLabel
  - JButton
  - JRadioButton
  - JCheckbox
  - JTextField
  - JComboBox
  - JTextArea

Interactive Components

- buttons – of course...
- text fields – input from keyboard
- check boxes – a button that can be toggled on or off using the mouse (indicates a boolean value is set or unset)
- radio buttons – to provide a set of mutually exclusive options
- sliders – specify a numeric value within a bounded range
- combo boxes – select one of several options from a “drop down” menu

Java GUI Event Loop
6.1 – Components, Events, Listeners

- **Component** – defines a screen element used to display information or allow the user to interact with the program
  - A *container* is a special type of component that is used to hold and organize other components
- **Event** – an object that represents some occurrence in which we may be interested
  - Often correspond to user actions (mouse button press, key press)
  - Most GUI components generate events to indicate a user action related to that component
- **Listener** – an object that "waits" for an event to occur and responds in way when it does (aka: event handler)
  - In designing a GUI-based program we need to establish the relationships between the listener, the event it listens for, and the component that generates the event

6.1 – GUI Elements

- To create a Java program that uses a GUI, we must:
  - set up ("instantiate") the necessary components,
  - implement listener classes that define what happens when particular events occur, and
  - Establish relationship between listeners and components that generate the events of interest
- Java components and other GUI-related classes are defined primarily in two packages
  - java.awt
  - javax.swing

GUI Program: PushCounter

```java
import javax.swing.JFrame;

public class PushCounter {
    // Create and displays the main program frame.
    public static void main(String[] args) {
        JFrame frame = new JFrame("Push Counter");
        // Set initial window size and position.
        frame.setSize(250, 100);
        frame.setLocationRelativeTo(null);
        // Create the PushCounterPanel and add it to the frame's content pane.
        PushCounterPanel panel = new PushCounterPanel();
        frame.getContentPane().add(panel);
        // Set frame visibility.
        frame.setVisible(true);
    }
}
```
PushCounterPanel.java

```java
public class PushCounterPanel {
    private int count;

    public PushCounterPanel() {
        count = 0;
    }

    public void pushButtonClicked() {
        count += 1;
        updateCountLabel();
    }

    private void updateCountLabel() {
        // Update the label with the current count
    }
}
```

More samples

- LeftRight.java
- Fahrenheit.java
- StyleOptions.java
- QuoteOptions.java