CSSE 220 Day 7
GUI programming in Java Using Swing
JUnit Testing exercise is due now.
Turn in your written problems from Assignment 6.
Everything for BigRational is due tomorrow at 8:05.
"Not answering mail at night" does not mean "Please don't send me mail". It's about timing, not about lack of desire to help you.
Your questions about ...

- Java
- Reading from the textbook
- Homework
- etc.
Python provides an extensive GUI toolkit called Tkinter, which is built on top of the (not specific to Python) Tk/Tk framework.

- [www.tcl.tk/software/tcltk/](www.tcl.tk/software/tcltk/)

In CSSE120, we did not use Tkinter directly

Instead used ZelleGraphics

- a simplified collection of classes for drawing on the screen.
- Hides details what would be confusing to beginners in OOP.

In Java, we'll see "the real thing": Swing
**Swing resources**

- Appendices in the Weiss book (sketchy).
  - This is the best Swing resource, in my opinion (for both learning and reference):
  - Available for you to read on Safari Tech Books Online
    - [http://proquest.safaribooksonline.com/?uicode=rosehulman](http://proquest.safaribooksonline.com/?uicode=rosehulman)
      Then find the Java Swing book
    - If that link does not work for you, go to the Logan Library page, and choose Safari from the Databases drop-down list near the top of the page, then click Go.
  - This one has great examples, but it tends to assume a deeper familiarity with Java than most of you have now.
Java GUI toolkits

- AWT (Abstract Windowing Toolkit) was part of the original Java release
  - Many features are still used
  - But users were dissatisfied …

- Swing was standardized with Java 2 (1999)
  - The most widely used Java GUI toolkit.
  - The one we will study

- SWT (Standard Widget Toolkit) was developed by IBM for Eclipse
  - Simpler to get started with than Swing.
  - Has some limitations.
What is a GUI toolkit?

- A collection of *widgets* and ways to control their interaction with the user and with each others

- Examples of widgets
  - window
  - menu
  - button
  - text area
  - slider
  - scroll bar
# Some Classes That We will be Using

<table>
<thead>
<tr>
<th>Class</th>
<th>What it is</th>
</tr>
</thead>
<tbody>
<tr>
<td>JFrame</td>
<td>a top-level window</td>
</tr>
<tr>
<td>JComponent</td>
<td>a region where we can draw; also parent of many other widget classes</td>
</tr>
<tr>
<td>JButton</td>
<td>a JComponent representing a button. When clicked, an action can happen</td>
</tr>
<tr>
<td>JLabel</td>
<td>a place to put text in a window</td>
</tr>
<tr>
<td>JTextField</td>
<td>a place for the user to enter text</td>
</tr>
<tr>
<td>JPanel</td>
<td>a JComponent that can be used as a container for organizing other widgets</td>
</tr>
<tr>
<td>Graphics</td>
<td>an object that can draw things on a JComponent. We never have to create</td>
</tr>
<tr>
<td></td>
<td>this object; it is provided to us by the system</td>
</tr>
<tr>
<td>Graphics2D</td>
<td>a more &quot;object-oriented&quot; graphics object</td>
</tr>
<tr>
<td>JOptionPane</td>
<td>Request a single line of input from the user,</td>
</tr>
</tbody>
</table>
Two Classes Needed for a Simple Application That Draws Things

- A JFrame in which to put our component(s)
- A JComponent in which to draw things
  - We need to extend JComponent
  - The extension class must provide a `paintComponent()` method that does the actual drawing
  - Sometimes we will extend JPanel, which extends JComponent.
- `paintComponent()` is automatically called when the program starts, and when the window is resized or unhidden.
Live Demo

- We will learn by doing.
- After class, I will post my notes for this live session, so you will have them for reference.
  - 220-Day07_200820-script.docx
- Many of the examples I use are based on Cay Horstmann's examples in *Big Java*.

- This should be a lot of fun!

- Ask for help from the assistants if something does not work for you.
To do before Session 8

- The next reading assignment.
- No ANGEL quiz today.
- No Written problems today.
- Finish BigRational.
- Experiment with some of the things we did in class.
- Read documentation, etc. Especially Shape classes.