

CS120: Programming Graphical User Interfaces (GUI, pronounced gooey)

W. Michael Petullo

University of Wisconsin–La Crosse

As of October 6, 2021



Programming GUIs: Screenshot

CS120 Software Design | +

https://www.flyn.org/courses/cs120-2021-fall/


Flyn Computing Open Source Research System Integration Tests

Syllabus Schedule Resources

CS120 Software Design I

Instructor information

Instructor:	W. Michael Petullo
Office location:	210 Wing Technology Center
Office hours:	2:00 p.m.–3:00 every weekday and by appointment
Telephone:	(608) 785-6617
Email:	wpetullo@uwfwiw.edu



Catalog description

An introduction to the fundamentals of software development; including software classes, objects, inheritance, polymorphism, logic, selection control, repetition control, subprograms, parameter passage, and rudimentary software engineering techniques. Students complete numerous programming projects using a modern programming language.

Prerequisites

MTH151, MTH175, or math placement test scores at or above MTH151

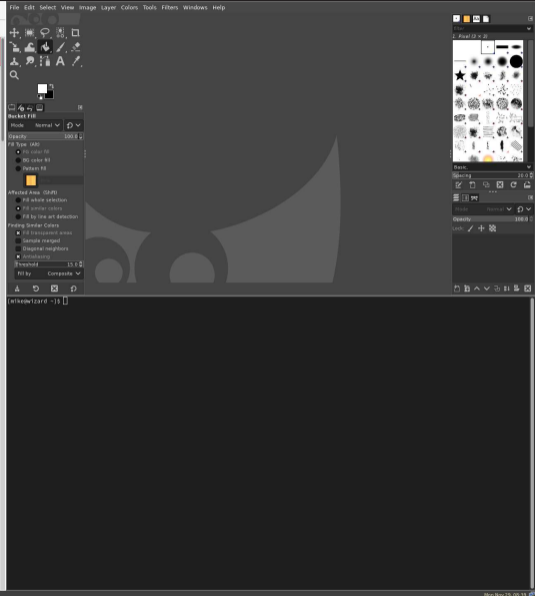
Time and location

Monday, Tuesday, Thursday and Friday at 11:00–11:55 a.m or 12:05–1:00 p.m.
Class meets for lectures in Centennial 2311 on Mondays and Thursdays. Labs take place on Tuesdays and Fridays in Wing 16.

Student learning objectives

This course follows the Computer Science Department's learning objectives for CS120, which are listed below. The [course schedule](#) indicates the objectives covered by each lesson.

1. Write Java programs using non-parallel control instructions, including assignment, method calls, `if`, `while`, `do`, and `for`.
2. Write and evaluate expressions using literals, variables, parenthesis, and the



The screenshot shows a graphics application window with a menu bar (File, Edit, Select, View, Image, Layer, Colors, Tools, Filters, Windows, Help). The toolbar includes various drawing tools like selection, lasso, brush, and eraser. The main canvas displays a large, semi-transparent grey shape. The right-hand panel shows a 'Basic' tool palette with various icons for shapes and patterns. The bottom status bar indicates the user is 'wkeewizard' and the time is 'Mon Nov 25, 09:33'.

Programming GUIs: 1. Declare and Lay Out Widgets



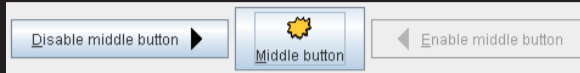
```
JFrame frame = new JFrame("Widgets");
JPanel panel = new JPanel();
JLabel label = new JLabel("A useless ...");
JButton button = new JButton("Click me.");

panel.add(label);
panel.add(button);

frame.add(panel);
```

Programming GUIs: Widgets

Button:



Checkbox:



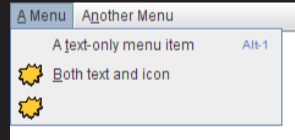
Radio Button:



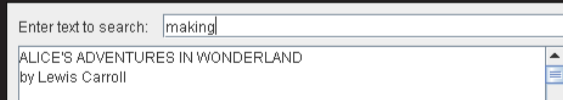
Label:



Menu:



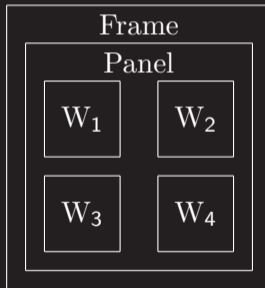
Text Field:



Images © Oracle, e.g., <https://docs.oracle.com/javase/tutorial/ui/swing/components/button.html>

Panels group widgets and lay them out on the screen.

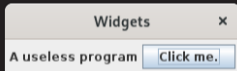
Windows or Frames group containers into an application component.



Programming GUIs: 2. Establish Actions

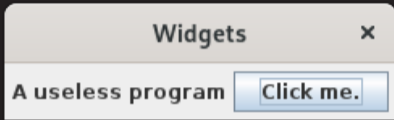
```
class Action implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        System.out.println(" Clicked!");
    }
}

class Widgets {
    public static void main(String[] args) {
        ...
        JButton button = new JButton("...");
        ...
        button.addActionListener(new Action());
    }
}
```



Call of button's `addActionListener` method arranges for Action's `actionPerformed` methods to execute when button clicked. (Event loop will handle this.)

Programming GUIs: 3. Yield Control to Event Loop



```
frame.pack();  
frame.setVisible(true);
```



Programming GUIs: Example

```
import javax.swing.*;
import java.awt.event.*;

class Action implements ActionListener {
    public void
        actionPerformed(ActionEvent e) {
        System.out.println(" Clicked!");
    }
}

public class Widgets {
    public static void main(String [] as) {
        // 1: Declare and lay out.
        JFrame frame = new JFrame("...");
        JPanel panel = new JPanel();
        JLabel label = new JLabel("...");
        JButton button = new JButton("...");
```

```
        // 1 continued.
        panel.add(label);
        panel.add(button);

        frame.add(panel);

        // 2: est. actions.
        Action a = new Action();
        button.addActionListener(a);

        // 3: Yield to event loop.
        frame.pack();
        frame.setVisible(true);
    }
}
```




Graded Homework Aquinas: button in Java

Ungraded Labs Aquinas: hello3 and typist in Java

<https://www.flyn.org/courses/cs120-2021-fall/schedule>