Java Chapter 13 – Introduction to Swing Components

**SWING COMPONENTS:**

\*Computer programs are more user friendly when they contain GUI components

* GUI Components 🡪 buttons, text fields, & other components in which the user can interact
* AKA controls or widgets
* Java contains 2 sets of prewritten GUI Components:
* Abstract Window Toolkit (AWT):
* Older, less portable, does not have a consistent appearance
* Simple names: frame, button
* Swing:
* Used more frequently in new program development
* Added J in front of class names: JFrame, JButton
* Swing Components:
* Descendants of JComponent 🡪 inherits from java.awt.Container class

\*import.javax.swing.\*; 🡪 use of all Swing GUI Components & their methods

\*Almost all Swing components 🡪 lightweight 🡪 written completely in Java & don’t rely on the local OS code\*

* Usually placed in containers 🡪 Containers are defined in the Container class
* Object class 🡪 Component class 🡪 Container class
* Containers the Take Form of:
* Window: can drag, resize, minimize, restore, & close; class is child of the Container class
* Frame: often preferred; child of Window
* Java.lang.Object
* 🡪 Component
* 🡪 🡪 Container
* 🡪 🡪 🡪 Container
* 🡪 🡪 🡪 🡪 Window
* 🡪 🡪 🡪 🡪 🡪 Frame
* 🡪 🡪 🡪 🡪 🡪 🡪 JFrame – Swing Object that is a child of Frame

\*Object’s descendants are not automatically imported w/ java.lang package

**JFRAME CLASS:**

\*Usually you create a JFrame so that you can place other objects within it for display

* Four Constructors
* Constructing:
* JFrame firstFrame = new JFrame(“Hello”); 🡪 w/ title “Hello”
* JFrame secondFrame = new JFrame(); 🡪 w/ no title
* firstFrame.setSize(250, 100); 🡪 250px horizontally, 100px vertically
* firstFrame.setVisible(true); 🡪 won’t be visible w/out this statement
* Close button 🡪 default is JFrame to be hidden & for the app to keep running
* Can change w/ setDefaultCloseOperation() method
* JFrame Appearance:

\*Is provided by the OS in which the program is running

* Window Decorations (icons, buttons): Windows default 🡪 supplied by OS
* Look & Feel: can request Java’s looks & feel provide the decorations for a frame (Metal)
* Set w/ setDefaultLookAndFeelDecorated() method

**JLABEL CLASS:**

\*Built in Java Swing class that allows you to create a label that you can display in a Jframe

* Available Constructors
* Create a JLabel named greeting that holds the words ‘Good day’:

JLabel greeting = new JLabel(“Good day”);

* Add the greeting object to the Jframe object named firstFrame using the add() method:

firstFrame.add(greeting);

* Changing a JLabel’s font 🡪 w/ setFont() method
* Requires a Font object argument
* Typeface argument
* Style argument
* Point size argument

USING A LAYOUT MANAGER:

\*When you want to add multiple components to a Jframe or other container, you usually need to provide instructions for the layout of the components

* \*To place multiple components at specified positions in a container so they do not hide each other, you must explicitily use a layout manager
* The normal default behavior of a Jframe is to use a border layout manager – divides a container into regions
* BorderLayout class: The Java class that provides the layout of a border layout manager
* When using this w/out specifiying a region in which to place a component, all the components are placed int the same region, obscuring each other

EXTENDING THE JFRAME CLASS

ADDING JTEXTFIELDS, JBUTTONS, AND TOOL TIPS TO A JFRAME: