Lecture 19
Midterm 2 Curve

• If raw score greater than 61:
  – $C = ((R - 61) / 13) \times 0.9 + 3.3$

• If raw score less than 61:
  – $C = ((R - 38) / 23) \times 2.3 + 1$
Java UI’s so far

• Create a top level window.
• Create a JPanel to be content pane.
• Fill JPanel with other components
  – Buttons, sliders, etc.
  – Containers with components arranged in them.
    • And so forth.
• Connect UI events with “listeners”
  – Listeners take action as response.
  – Action may change/update UI
• Make top level window visible.
Swing Component Class Hierarchy

java.awt.Component
  java.awt.Container
    javax.swing.JComponent
      JPanel
      JScrollPane
      JSplitPane
      JTabbedPane
      JToolBar
      JComboBox
      JLabel
      JList
      JProgressBar
      JSeparator
      JSlider
      JToolTip
      JTree

AbstractButton
  JButton
  JToggleButton
    JCheckBox
    JRadioButton
  JMenuItem
    JMenu
Listener Types

• Supported at `awt.Component`
  – `ComponentListener`
  – `FocusListener`
  – `KeyListener`
  – `MouseListener`
  – `MouseMotionListener`
  – `MouseWheelListener`

• Supported at `awt.Container`
  – `ContainerListener`

• Supported at `AbstractButton`
  – `ActionListener`
  – `ChangeListener`
  – `ItemListener`

• Supported by individual components
  – `ActionListener` : `JComboBox`
  – `ChangeListener` : `JSlider`, `JProgressBar`
Keyboard Concepts

• Keyboard events go to component that currently has “focus”.
  – setFocusable(boolean status)
  – isFocusable()
  – requestFocusInWindow()

• Java provides a framework for managing focus in more sophisticated ways if necessary.
  – http://docs.oracle.com/javase/tutorial/uiswing/misc/focus.html#api

• KeyListener interface
  – keyTyped(KeyEvent e)
  – keyPressed(KeyEvent e)
  – keyReleased(KeyEvent e)

• Typing of character vs. pressing of key
  – Basic sequence of events generated: key press, key typed, key released

• Important KeyEvent methods
  – char getKeyChar()
    • This is only valid for key typed events.
  – int getKeyCode()
  – isAltDown(), isControlDown(), isShiftDown(), isMetaDown()
Mouse Concepts

- **MouseListener**
  - mouseClicked(MouseEvent e)
  - mouseEntered(MouseEvent e)
  - mouseExited(MouseEvent e)
  - mousePressed(MouseEvent e)
  - mouseReleased(MouseEvent e)

- **MouseMotionListener**
  - mouseDragged(MouseEvent e)
  - mouseMoved(MouseEvent e)

- **MouseEvent**
  - Position info
    - Relative to component: getX(), getY(), getPoint()
    - Absolute position: getXOnScreen(), getYOnScreen()
  - Click info: getClickCount(), getButton()
  - Modifiers: isAltDown(), isShiftDown(), etc...
Composing Widgets

• A set of related UI elements that act as a unit within your UI.

• Basic idea:
  – Subclass JPanel
  – In constructor, create and arrange UI components.
  – Provide methods for attaching listeners, requesting current state of UI, etc.

  • Could be straight delegation or could be mediated by more sophisticated logic.
Model – View Pattern

• Keep object representing the data separate from widget providing the UI
  – Model: Object representing logical entity within your application.
    • Assignment example: Plate, Sushi, Belt, etc.
  – View: Object (widget) representing UI to model.
    • Assignment example: BeltUI, WorkspaceUI