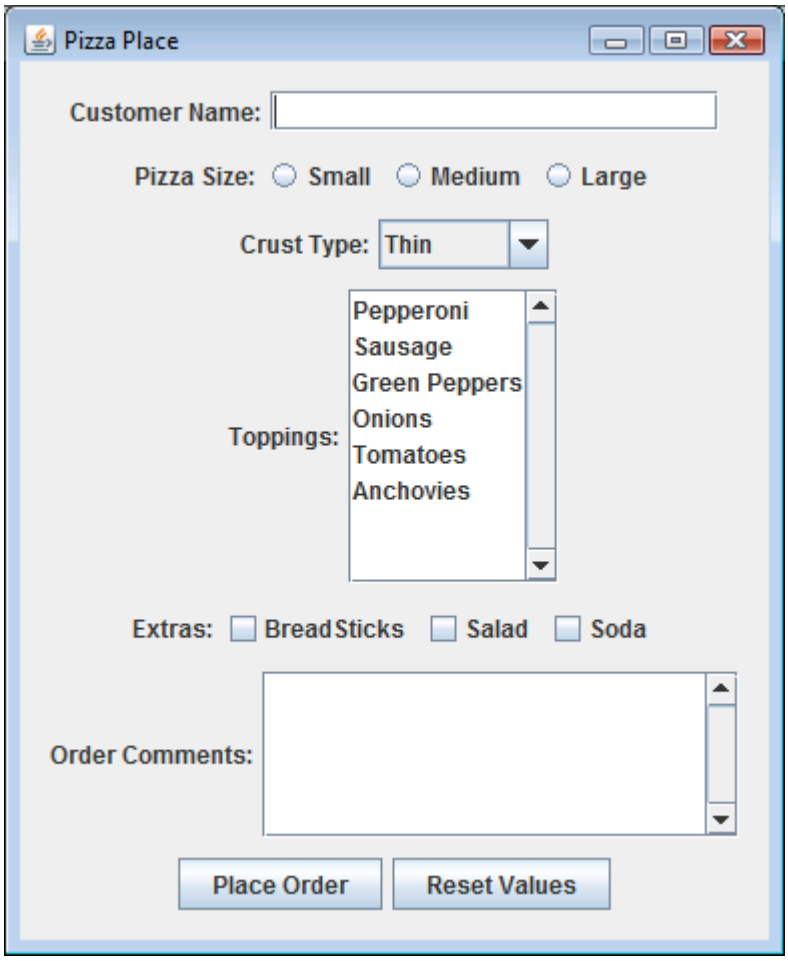
Chapter Twelve Activity (Pizza Place)

In this activity, the student will use their new-found knowledge of the different available input controls to create a pizza ordering screen!

Project Details

The student will create a GUI screen as shown to the right. This program will use a textbox to retrieve the customer's name, radio buttons to determine pizza size, a combo box and list box for crust and topping options, check boxes for any extra items, and a text area for additional comments. The bottom of the screen will have a button that will display a message box with order details, and a button that will reset all of the values on the screen.



This program will use nearly all of the skills they learned in this chapter and it is their largest program so far! The student is encouraged to write the program piece-by-piece, making sure one piece works before going on to the next one.

The final code for this program is listed below. The comments in the code will demonstrate the different pieces for the different steps in the activity.

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

// PizzaPlace is the main program class and it implements
// the ActionListener interface in order to receive button
// click events.
public class PizzaPlace implements ActionListener
```

```
{
	// The static main method will run when the program starts
	public static void main(String[] args)
	{
		// Create a new instance of the PizzaPlace class.
		// The constructor will handle all of the GUI
		// initialization and display.
		new PizzaPlace();
	}

	// PizzaPlace member variables

	JTextField nameText;	// JTextField will hold the customer's name

	// These JRadioButtons will allow the customer to pick a size
	JRadioButton smallRadio;
	JRadioButton mediumRadio;
	JRadioButton largeRadio;

	JComboBox crustCombo;	// JComboBox will hold the crust options
	JList toppingList;	// JList for holding the selected toppings

	JButton orderButton;	// JButton for placing an order
	JButton resetButton;	// JButton for resetting the controls

	// These JCheckBoxes will allow the customer to select order add-ons
	JCheckBox breadCheck;
	JCheckBox saladCheck;
	JCheckBox sodaCheck;

	JTextArea commentArea;	// JTextArea will hold the customer's comments

	// PizzaPlace constructor will do all the GUI construction work
	public PizzaPlace()
	{
		// STEP ONE:
		// Create the main JFrame and set the default close operation
		JFrame myFrame = new JFrame();
		myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		myFrame.setTitle("Pizza Place");
```

```
        JPanel mainPanel = (JPanel)myFrame.getContentPane();

        // Set the main panel's vertical Box layout and border
        mainPanel.setLayout(new BoxLayout(mainPanel,BoxLayout.Y_AXIS));
        mainPanel.setBorder(BorderFactory.createEmptyBorder(10,10,10,10));

        // STEP TWO:
        // Create the JPanel holding a JLabel and JTextField to receive
        // the customer's name.  Then add the name panel to the main panel.
        JPanel namePanel = new JPanel();
        nameText = new JTextField(20);
        namePanel.add(new JLabel("Customer Name:"));
        namePanel.add(nameText);
        // add this panel to the main panel
        mainPanel.add(namePanel);

        // STEP THREE:
        // Create the JPanel holding the radio buttons for the pizza size.
        JPanel sizePanel = new JPanel();
        sizePanel.add(new JLabel("Pizza Size:"));
        smallRadio = new JRadioButton("Small");
        mediumRadio = new JRadioButton("Medium");
        largeRadio = new JRadioButton("Large");
        sizePanel.add(smallRadio);
        sizePanel.add(mediumRadio);
        sizePanel.add(largeRadio);
        // Put the buttons in a ButtonGroup so they will work together
        ButtonGroup sizeGroup = new ButtonGroup();
        sizeGroup.add(smallRadio);
        sizeGroup.add(mediumRadio);
        sizeGroup.add(largeRadio);
        // add this panel to the main panel
        mainPanel.add(sizePanel);

        // STEP FOUR:
        // Create the JPanel holding the JComboBox for crust type.  Add
        // a descriptive JLabel and add several selections to the combo box.
        JPanel crustPanel = new JPanel();
        crustPanel.add(new JLabel("Crust Type:"));
        crustCombo = new JComboBox();
```

```
            crustCombo.addItem("Thin");
            crustCombo.addItem("Thick");
            crustCombo.addItem("Deep Dish");
            crustPanel.add(crustCombo);
            // add this panel to the main panel
            mainPanel.add(crustPanel);

            // STEP FIVE:
            // Create the JPanel holding the JList for topping selections.  Add
            // a descriptive label, initialize an array of toppings, and create
            // the new JList
            JPanel toppingPanel = new JPanel();
            JLabel toppingLabel = new JLabel("Toppings:");
            String[] toppings = {"Pepperoni", "Sausage", "Green Peppers", "Onions",
"Tomatoes", "Anchovies"};
            toppingList = new JList(toppings);
            // Allow the user to select more than one topping at a time.

      toppingList.setSelectionMode(ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);
            // Wrap the JList in a JScrollPane
            JScrollPane topScroll = new JScrollPane(toppingList,
JScrollPane.VERTICAL_SCROLLBAR_ALWAYS, JScrollPane.HORIZONTAL_SCROLLBAR_NEVER);
            // add the label and scroll pane to the topping panel
            toppingPanel.add(toppingLabel);
            toppingPanel.add(topScroll);
            // add this panel to the main panel
            mainPanel.add(toppingPanel);

            // STEP SIX:
            // Create the JPanel for order extras.  Each extra will have a
            // JCheckBox allowing the user to pick more than one extra.
            JPanel extraPanel = new JPanel();
            extraPanel.add(new JLabel("Extras:"));
            breadCheck = new JCheckBox("BreadSticks");
            saladCheck = new JCheckBox("Salad");
            sodaCheck = new JCheckBox("Soda");
            extraPanel.add(breadCheck);
            extraPanel.add(saladCheck);
            extraPanel.add(sodaCheck);
            // add this panel to the main panel
```

```
            mainPanel.add(extraPanel);

            // STEP SEVEN:
            // Create the JPanel holding the JTextArea for customer comments.
            // Add a descriptive JLabel and wrap the JTextArea in a JScrollPane.
            JPanel commentPanel = new JPanel();
            commentPanel.add(new JLabel("Order Comments:"));
            commentArea = new JTextArea(5,20);
            commentArea.setLineWrap(true);
            JScrollPane commentScroll = new JScrollPane(commentArea,
JScrollPane.VERTICAL_SCROLLBAR_ALWAYS, JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
            // add the controls to this comment panel
            commentPanel.add(commentScroll);
            // add this panel to the main panel
            mainPanel.add(commentPanel);

            // STEP EIGHT:
            // Create the JPanel holding the JButtons to place order and reset
values
            JPanel buttonPanel = new JPanel();
            orderButton = new JButton("Place Order");
            resetButton = new JButton("Reset Values");
            // Set this PizzaPlace instance as the ActionListener to receive button
clicks
            orderButton.addActionListener(this);
            resetButton.addActionListener(this);
            // Add the buttons to the button panel
            buttonPanel.add(orderButton);
            buttonPanel.add(resetButton);
            // add this panel to the main panel
            mainPanel.add(buttonPanel);

            // Finally, pack the main frame and make it visible!
            myFrame.pack();
            myFrame.setVisible(true);

      }
```

```
      // STEP NINE:
```

```
	// ActionListener interface method
	public void actionPerformed(ActionEvent e)
	{
		// If the "Place Order" button was clicked
		if (e.getSource() == orderButton)
		{
			// Start building a summary string starting with the customer's
name.
			// We are using the "\n" escape character wherever we want to
move to a new line!
			String pizzaInfo = "PIZZA ORDER FOR: " + nameText.getText() +
"\n";

			// Add size information based on the selected radio button
			pizzaInfo = pizzaInfo + "SIZE:\n";
			if (smallRadio.isSelected() == true)
				pizzaInfo = pizzaInfo + "  Small\n";
			else if (mediumRadio.isSelected() == true)
				pizzaInfo = pizzaInfo + "  Medium\n";
			else
				pizzaInfo = pizzaInfo + "  Large\n";

			// Add crust information based on the selected item in the combo
box
			pizzaInfo = pizzaInfo + "CRUST TYPE:\n";
			pizzaInfo = pizzaInfo + "  " +
(String)crustCombo.getSelectedItem() + "\n";

			// Add topping information based on the selected values in the
list box
			pizzaInfo = pizzaInfo + "TOPPINGS:\n";
			Object[] toppings = toppingList.getSelectedValues();
			for (int i=0; i< toppings.length; i++)    // for each selected
item in the array
			{
				// Add the topping as a separate line to the summary string
				pizzaInfo = pizzaInfo + "  " + (String) toppings[i]+ "\n";
			}
```

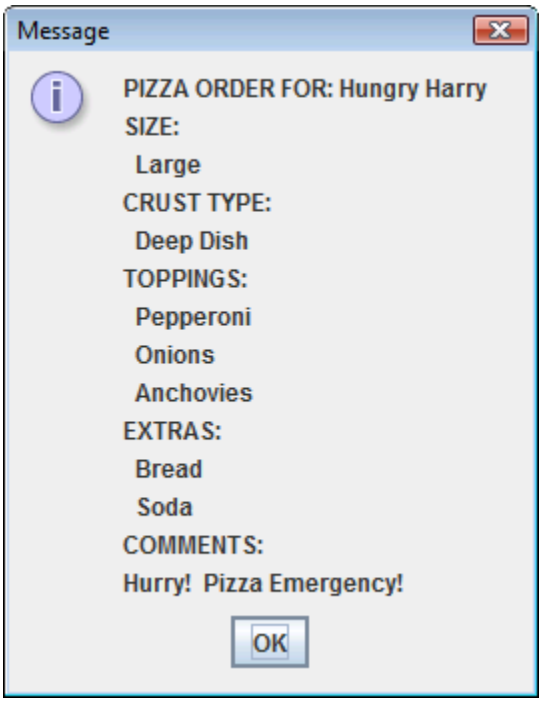
```
			// Add order extras based on the selected checkbox(es)
```

```
            pizzaInfo = pizzaInfo + "EXTRAS:\n";
            if (breadCheck.isSelected() == true)
                  pizzaInfo = pizzaInfo + "  Bread\n";
            if (saladCheck.isSelected() == true)
                  pizzaInfo = pizzaInfo + "  Salad\n";
            if (sodaCheck.isSelected() == true)
                  pizzaInfo = pizzaInfo + "  Soda\n";

            // Add any customer comments that were entered
            pizzaInfo = pizzaInfo + "COMMENTS:\n";
            pizzaInfo = pizzaInfo + commentArea.getText();

            // Finally, show the order summary!
            JOptionPane.showMessageDialog(null,  pizzaInfo);

      }
      // Else if the reset button was clicked
      else if (e.getSource() == resetButton)
      {
            // clear customer name
            nameText.setText("");

            // select the small crust size
            smallRadio.setSelected(true);
            mediumRadio.setSelected(false);
            largeRadio.setSelected(false);

            // select the first entry in the crust type combo box
            crustCombo.setSelectedIndex(0);

            // remove all topping selections from the list box
            toppingList.clearSelection();

            // un-check all order extra checkboxes
            breadCheck.setSelected(false);
            saladCheck.setSelected(false);
            sodaCheck.setSelected(false);
```

```
                // clear the customer comments
                commentArea.setText("");
            }
        }
}
```

Activity Output

When the student completes and runs their code, they should be able to select different options for their pizza. When they click on the "Place Order" button, they should see a message box pop-up and display a summary of their order. Here is an example:



The completed project for this activity is located in the "Activity Solutions\PizzaPlace" folder underneath the Solution Files installation directory.