```
/*********************************************************
 * Program Name: Contractor.java
 * Name: Barbara Williams
 * Program Description: This program is called by the
 *                      PoolVolumeCalculator program and allows
 *                      contractors to be added or deleted
 *                      within any program.
 *********************************************************/

package Project;

import java.awt.Color;
import java.awt.Container;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;


class Contractor extends JFrame
{
      private String[] states = { "AL", "AK", "AZ", "AR", "CA", "CO", "CT",
"DE",
                  "FL", "GA", "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA",
"ME",
                  "MD", "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH",
"NJ",
                  "NM", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
"SD",
                  "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY" };
      private JComboBox StateList = new JComboBox(states);
      private JTextField NameText = new JTextField(20);
      private JTextField AddressText = new JTextField(26);
      private JTextField CityText = new JTextField(28);
      private JTextField ZipText = new JTextField(5);
      private JTextField PhoneText = new JTextField(10);
      private JTextField PopMessageText = new JTextField(30);
      private AddContButtonHandler addConHandler = new
AddContButtonHandler();

      public Contractor(String who)
      {
            popUpWindow(who);
      }

      // Customer and Contractor Pop Up Window
      public void popUpWindow(final String who)
      {
```

```
            // add components to contractor Pop up window
            final JFrame popWindow;
            popWindow = new JFrame(who);
            popWindow.setSize(370, 300);
            popWindow.setLocation(100, 100);
            popWindow.setVisible(true);
            setDefaultCloseOperation(EXIT_ON_CLOSE);

            Container c = new Container();
            popWindow.add(c);
            c.setLayout(new FlowLayout());
            c.add(new JLabel(who + " Name "));
            c.add(NameText);
            c.add(new JLabel("Address "));
            c.add(AddressText);
            c.add(new JLabel("City "));
            c.add(CityText);
            c.add(new JLabel("State "));
            StateList.setSelectedIndex(0);
            c.add(StateList);
            c.add(new JLabel("ZIP"));
            c.add(ZipText);
            c.add(new JLabel("Phone"));
            c.add(PhoneText);

            JButton addwho = new JButton("Add " + who);
            addwho.setMnemonic('A');

            JButton close = new JButton("Close");
            close.setMnemonic('C');

            JButton deleteFile = new JButton("Delete File");
            deleteFile.setMnemonic('D');

            c.add(addwho);
            c.add(close);
            c.add(deleteFile);
            PopMessageText.setEditable(false);
            PopMessageText.setHorizontalAlignment(JTextField.CENTER);
            PopMessageText.setForeground(Color.blue);
            PopMessageText.setBackground(Color.white);
            PopMessageText.setText("File contractor.txt does not exist yet -
will be created when you add contractors!");

            // message.setOpaque(false);
            c.add(PopMessageText);

            deleteFile.setToolTipText("Delete File");
            addwho.setToolTipText("Add " + who);
            close.setToolTipText("Close");
            if (who == "Contractor")
            {
                  addwho.addActionListener(addConHandler); // registers
listener
            }

            close.addActionListener(new ActionListener()
```

```
		{
			public void actionPerformed(ActionEvent e)
			{
				popWindow.dispose();
			}
		});

		deleteFile.addActionListener(new ActionListener()
		{
			public void actionPerformed(ActionEvent e)
			{
				PopMessageText.setText("");
				if (who == "Contractor")
				{
					File file = new File("contractor.txt");
					boolean cusFileDeleted = file.delete();
					if (cusFileDeleted)
					{
						PopMessageText.setText("File
contractor.text deleted!");
						PopMessageText.setForeground(Color.blue);
					}
					else
					{
						PopMessageText.setText("There was an
error in deleting file");
					}
				}
			}
		});
	} // end Pop up window

	// Class handler to add customer information to a file
	class AddContButtonHandler implements ActionListener
	{
		public void actionPerformed(ActionEvent addCusHandler)
		{
			int StateIndex;
			try
			{
				File file = new File("contractor.txt");
				boolean success = file.createNewFile();
				if (success)
				{
					PopMessageText
							.setText("Contractor added!");
				}
				else if (file.canWrite())
				{
					PopMessageText
							.setText("Writing data to
Contractor.txt, file added");
				}
				else
				{
```

```
                        PopMessageText.setText("Cannot create file:
Contractor.txt");
                    }
                    try
                    {
                        FileWriter fileW = new
FileWriter("Contractor.txt", true);
                        fileW.write("Name: " + NameText.getText());
                        fileW.write("\n");
                        fileW.write("Address: " +
AddressText.getText());
                        fileW.write("\n");
                        fileW.write("City: " + CityText.getText());
                        fileW.write("\n");
                        StateIndex = StateList.getSelectedIndex();
                        fileW.write("State: " + states[StateIndex]);
                        fileW.write("\n");
                        fileW.write("Zip: " + ZipText.getText());
                        fileW.write("\n");
                        fileW.write("Phone: " + PhoneText.getText());
                        fileW.write("\r\n\n");
                        fileW.close();
                        PopMessageText.setText("Contractor added!");
                    }
                    catch (IOException e1)
                    {
                        JOptionPane.showMessageDialog(null,
e1.getMessage(),
                                "ERROR", 2); // Will display error
message if unable
                        // to write to file
                    }
                    NameText.setText("");
                    AddressText.setText("");
                    CityText.setText("");
                    ZipText.setText("");
                    PhoneText.setText("");
                }
                catch (IOException e1)
                {
                }
            }
        }

}
```

```
/*********************************************************
 * Program Name: Customer.java
 * Name: Barbara Williams
 * Program Description: This program is called by the
 *                      PoolVolumeCalculator program and allows
 *                      customers to be added or deleted within
 *                      any program.
 *********************************************************/

package Project;

import java.awt.Color;
import java.awt.Container;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;

class Customer extends JFrame
{
      private String[] states = { "AL", "AK", "AZ", "AR", "CA", "CO", "CT",
"DE",
                  "FL", "GA", "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA",
"ME",
                  "MD", "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH",
"NJ",
                  "NM", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
"SD",
                  "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY" };
      private JComboBox StateList = new JComboBox(states);
      private JTextField NameText = new JTextField(20);
      private JTextField AddressText = new JTextField(26);
      private JTextField CityText = new JTextField(28);
      private JTextField ZipText = new JTextField(5);
      private JTextField PhoneText = new JTextField(10);
      private JTextField PopMessageText = new JTextField(30);
      private AddCustButtonHandler addCusHandler = new
AddCustButtonHandler();

      public Customer(String who)
      {
            popUpWindow(who);
      }

      // Customer and Contractor Pop Up Window
      public void popUpWindow(final String who)
      {
```

```
            // add components to contractor Pop up window
            final JFrame popWindow;
            popWindow = new JFrame(who);
            popWindow.setSize(370, 250);
            popWindow.setLocation(100, 100);
            popWindow.setVisible(true);
            setDefaultCloseOperation(EXIT_ON_CLOSE);

            Container c = new Container();
            popWindow.add(c);
            c.setLayout(new FlowLayout());
            c.add(new JLabel(who + " Name "));
            c.add(NameText);
            c.add(new JLabel("Address "));
            c.add(AddressText);
            c.add(new JLabel("City "));
            c.add(CityText);
            c.add(new JLabel("State "));
            StateList.setSelectedIndex(0);
            c.add(StateList);
            c.add(new JLabel("ZIP"));
            c.add(ZipText);
            c.add(new JLabel("Phone"));
            c.add(PhoneText);

            JButton addwho = new JButton("Add " + who);
            addwho.setMnemonic('A');

            JButton close = new JButton("Close");
            close.setMnemonic('C');

            JButton deleteFile = new JButton("Delete File");
            deleteFile.setMnemonic('D');

            c.add(addwho);
            c.add(close);
            c.add(deleteFile);
            PopMessageText.setEditable(false);
            PopMessageText.setHorizontalAlignment(JTextField.CENTER);
            PopMessageText.setForeground(Color.blue);
            PopMessageText.setBackground(Color.white);
            PopMessageText.setText("File customer.txt does not exist yet -
will be created when you add customers!");

            // message.setOpaque(false);
            c.add(PopMessageText);

            deleteFile.setToolTipText("Delete File");
            addwho.setToolTipText("Add " + who);
            close.setToolTipText("Close");
            if (who == "Customer")
            {
                  addwho.addActionListener(addCusHandler); // registers
listener
            }

            close.addActionListener(new ActionListener()
```

```
		{
			public void actionPerformed(ActionEvent e)
			{
				popWindow.dispose();
			}
		});

		deleteFile.addActionListener(new ActionListener()
		{
			public void actionPerformed(ActionEvent e)
			{
				PopMessageText.setText("");
				if (who == "Customer")
				{
					File file = new File("customer.txt");
					boolean cusFileDeleted = file.delete();
					if (cusFileDeleted)
					{
						PopMessageText.setText("File
customer.text deleted!");
						PopMessageText.setForeground(Color.blue);
					}
					else
					{
						PopMessageText.setText("There was an
error in deleting file");
					}
				}
			}
		});
	} // end Pop up window

	// Class handler to add customer information to a file
	class AddCustButtonHandler implements ActionListener
	{
		public void actionPerformed(ActionEvent addCusHandler)
		{
			int StateIndex;
			try
			{
				File file = new File("customer.txt");
				boolean success = file.createNewFile();
				if (success)
				{
					PopMessageText
							.setText("Customer added!");
				}
				else if (file.canWrite())
				{
					PopMessageText
							.setText("Writing data to
Customer.txt, file added");
				}
				else
				{
```

```
                        PopMessageText.setText("Cannot create file:
Customer.txt");
                    }
                    try
                    {
                        FileWriter fileW = new
FileWriter("Customer.txt", true);
                        fileW.write("Name: " + NameText.getText());
                        fileW.write("\n");
                        fileW.write("Address: " +
AddressText.getText());
                        fileW.write("\n");
                        fileW.write("City: " + CityText.getText());
                        fileW.write("\n");
                        StateIndex = StateList.getSelectedIndex();
                        fileW.write("State: " + states[StateIndex]);
                        fileW.write("\n");
                        fileW.write("Zip: " + ZipText.getText());
                        fileW.write("\n");
                        fileW.write("Phone: " + PhoneText.getText());
                        fileW.write("\r\n\n");
                        fileW.close();
                        PopMessageText.setText("Customer added!");
                    }
                    catch (IOException e1)
                    {
                        JOptionPane.showMessageDialog(null,
e1.getMessage(),
                            "ERROR", 2); // Will display error
message if unable
                        // to write to file
                    }
                    NameText.setText("");
                    AddressText.setText("");
                    CityText.setText("");
                    ZipText.setText("");
                    PhoneText.setText("");
                }
                catch (IOException e1)
                {
                }
            }
        }

}
```

```
/*************************************************************
 * Program Name: ExitButton.java
 * Name: Barbara Williams
 * Program Description: This program is called by the
 *                      PoolVolumeCalculator program and allows
 *                      the ExitButton to be added and operational
 *                      within any program.
 *************************************************************/

package Project;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;


public class ExitButton
{
      private static JButton exitButton;

      public JButton getExitButton()
      {
            exitButton = new JButton("Exit");
            exitButton.setMnemonic('x');
            exitButton.addActionListener(new ActionListener()
            {
                  public void actionPerformed(ActionEvent e)
                  {
                        System.exit(0);
                  }
            });

            return exitButton;
      }
}
```