```java
/*
    Chapter 6:  Java Calculator
    Programmer: Mike Brown
    Date:       April 6, 2009
    Filename:   Calculator.java
    Purpose:    This program creates a calculator with a menu.
*/

import java.awt.*;
import java.awt.event.*;
import java.awt.datatransfer.*;
import java.text.DecimalFormat;
import javax.swing.JOptionPane;

public class Calculator extends Frame implements ActionListener
{
    private Button keys[];
    private Panel keypad;
    private TextField lcd;
    private double op1;
    private boolean first;
    private boolean foundKey;
    private boolean clearText;
    private int lastOp;
    private DecimalFormat calcPattern;

    public Calculator()
    {
        //create an instance of the menu
        MenuBar mnuBar = new MenuBar();
        setMenuBar(mnuBar);

        //construct and populate the File Menu
        Menu mnuFile = new Menu("File", true);
        mnuBar.add(mnuFile);
            MenuItem mnuFileExit = new MenuItem("Exit");
            mnuFile.add(mnuFileExit);

        //construct and populate the Edit menu
        Menu mnuEdit = new Menu("Edit", true);
        mnuBar.add(mnuEdit);
            MenuItem mnuEditClear = new MenuItem("Clear");
            mnuEdit.add(mnuEditClear);
            mnuEdit.insertSeparator(1);
            MenuItem mnuEditCopy = new MenuItem("Copy");
```

```
46              mnuEdit.add(mnuEditCopy);
47              MenuItem mnuEditPaste = new MenuItem("Paste");
48              mnuEdit.add(mnuEditPaste);
49
50          //construct and populate the About menu
51          Menu mnuAbout = new Menu("About", true);
52              mnuBar.add(mnuAbout);
53              MenuItem mnuAboutCalculator = new MenuItem("About
  calculator");
54              mnuAbout.add(mnuAboutCalculator);
55
56          //add the ActionListener to each menu item
57          mnuFileExit.addActionListener(this);
58          mnuEditClear.addActionListener(this);
59          mnuEditCopy.addActionListener(this);
60          mnuEditPaste.addActionListener(this);
61          mnuAboutCalculator.addActionListener(this);
62
63          //assign an ActionCommand to each menu item
64          mnuFileExit.setActionCommand("Exit");
65          mnuEditClear.setActionCommand("Clear");
66          mnuEditCopy.setActionCommand("Copy");
67          mnuEditPaste.setActionCommand("Paste");
68          mnuAboutCalculator.setActionCommand("About");
69
70          //construct components and initialize beginning values
71          lcd = new TextField(20);
72              lcd.setEditable(false);
73          keypad = new Panel();
74          keys = new Button[16];
75          first = true;
76          op1 = 0.0;
77          clearText = true;
78          lastOp = 0;
79          calcPattern = new DecimalFormat("########.########");
80
81          //construct and assign captions to the Buttons
82          for(int i=0; i<=9; i++)
83              keys[i] = new Button(String.valueOf(i));
84
85          keys[10] = new Button("/");
86          keys[11] = new Button("*");
87          keys[12] = new Button("-");
88          keys[13] = new Button("+");
89          keys[14] = new Button("=");
```

```
 90          keys[15] = new Button(".");
 91
 92          //set Frame and keypad layout to grid layout
 93          setLayout(new BorderLayout());
 94          keypad.setLayout(new GridLayout(4,4,10,10));
 95
 96          for (int i=7; i<=10; i++)//7,8,9 divide
 97              keypad.add(keys[i]);
 98
 99          for(int i=4; i<=6; i++)//4,5,6
100              keypad.add(keys[i]);
101
102          keypad.add(keys[11]);//multiply
103
104          for(int i=1; i<=3; i++)//1,2,3
105              keypad.add(keys[i]);
106
107          keypad.add(keys[12]);//subtract
108
109          keypad.add(keys[0]);//0 key
110
111          for(int i=15; i >=13; i--)
112              keypad.add(keys[i]);//decimal point, =, add (+) keys
113
114          for(int i=0; i<keys.length; i++)
115              keys[i].addActionListener(this);
116
117          add(lcd, BorderLayout.NORTH);
118          add(keypad,BorderLayout.CENTER);
119
120          addWindowListener(
121                  new WindowAdapter()
122                  {
123                      public void windowClosing(WindowEvent e)
124                      {
125                          System.exit(0);
126                      }
127                  }
128          );
129
130      }//end of constructor method
131
132      public void actionPerformed(ActionEvent e)
133      {
134          //test for menu clicks
```

```
135         String arg = e.getActionCommand();
136         if(arg == "Exit")
137             System.exit(0);
138 
139         if(arg == "Clear")
140         {
141             clearText = true;
142             first = true;
143             op1 = 0.0;
144             lcd.setText("");
145             lcd.requestFocus();
146         }
147 
148         if(arg == "Copy")
149         {
150             Clipboard cb =
    Toolkit.getDefaultToolkit().getSystemClipboard();
151             StringSelection contents = new
    StringSelection(lcd.getText());
152             cb.setContents(contents, null);
153         }
154 
155         if (arg == "Paste")
156         {
157             Clipboard cb =
    Toolkit.getDefaultToolkit().getSystemClipboard();
158             Transferable content = cb.getContents(this);
159             try
160             {
161                 String s =
    (String)content.getTransferData(DataFlavor.stringFlavor);
162                 lcd.setText(calcPattern.format(Double.parseDoubleu
   s)));
163             }
164             catch(Throwable exc)
165             {
166                 lcd.setText("");
167             }
168         }
169 
170         if(arg == "About")
171         {
172             String message = "Calculator ver. 1.0 \n The Molex
   Group Software LLC.\n Copyright 2009\n All rights reserved";
173             JOptionPane.showMessageDialog(null,message, "About
```

```
   Calculator", JOptionPane.INFORMATION_MESSAGE);
174         }
175
176         //test for button clicks
177         foundKey = false;
178
179         //search for the clicked key
180         for (int i=0; i<keys.length && !foundKey; i++)
181         {
182             if(e.getSource() == keys[i])
183             {
184                 foundKey = true;
185                 switch(i)
186                 {
187                     //number and decimal point buttons
188                 case 0: case 1: case 2: case 3: case 4:
189                 case 5: case 6: case 7: case 8: case 9: case 15:
190                     if(clearText)
191                     {
192                         lcd.setText("");
193                         clearText = false; //!clearText;
194                     }
195                     lcd.setText(lcd.getText() +
    keys[i].getLabel());
196                     break;
197                 //operator buttons
198                 case 10: case 11: case 12: case 13: case 14:
199                     clearText = true;
200
201                     if(first) //first operand
202                     {
203                         if(lcd.getText().length() == 0) op1 =
    0.0;
204                         else op1 =
    Double.parseDouble(lcd.getText());
205
206                         first = false;
207                         clearText = true;
208                         lastOp = i; //save last operator
209                     }
210                     else //second operand
211                     {
212                         switch(lastOp)
213                         {
214                         case 10: //divide button
```

```
215                                 op1 /=
    Double.parseDouble(lcd.getText());
216                                 break;
217                             case 11: //multiply button
218                                 op1 *=
    Double.parseDouble(lcd.getText());
219                                 break;
220                             case 12: //minus button
221                                 op1 -=
    Double.parseDouble(lcd.getText());
222                                 break;
223                             case 13: //plus button
224                                 op1 +=
    Double.parseDouble(lcd.getText());
225                                 break;
226                             } //end of switch(lastOp)
227                             lcd.setText(calcPattern.format(op1));
228                             clearText = true;
229
230                             if(i == 14) first = true; //equal button
231                             else lastOp = i; //save last operator
232                         } //end else
233                         break;
234                     }//end switch(i)
235                 }//end of if
236             }//end of for
237         }//end of actionPerformed
238
239         public static void main(String args[])
240         {
241             //set frame properties
242             Calculator f = new Calculator();
243             f.setTitle("Calculator Application");
244             f.setBounds(200,200,300,300);
245             f.setVisible(true);
246
247             //set image properties and add to frame
248             Image icon =
    Toolkit.getDefaultToolkit().getImage("calcImage.gif");
249             f.setIconImage(icon);
250
251         }//end of main
252 }//end of class
```