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

@SuppressWarnings("serial")
public class CalculatorInterface extends JFrame implements ActionListener
{
    private CalculatorFunctions calculator;

    private JMenu arithmeticMenu;
        private JMenuItem addMenuItem, subtractMenuItem, multiplyMenuItem, divideMenuItem, modMenuItem, exitMenuItem;
    private JMenu exponentialMenu;
        private JMenuItem powerMenuItem, squareRootMenuItem, logMenuItem,lnMenuItem;
    private JMenu trigMenu;
        private JMenuItem trigInstructionMenuItem, sinMenuItem, cosMenuItem, tanMenuItem, cotMenuItem, secMenuItem, cscMenuItem;
    private JMenu probabilityMenu;
        private JMenuItem factorialMenuItem, nCrMenuItem, nPrMenuItem;
    private JMenu exitMenu;

    public CalculatorInterface()
    {
        super("Calculator");

        arithmeticMenu = new JMenu("Arithmetic");
        exponentialMenu = new JMenu("Exponential Functions");
        trigMenu = new JMenu("Trigonometric Functions");
        probabilityMenu = new JMenu("Probability");
        exitMenu = new JMenu("Exit");

        addMenuItem = new JMenuItem("Add");
        subtractMenuItem = new JMenuItem("Subtract");
        multiplyMenuItem = new JMenuItem("Multiply");
        divideMenuItem = new JMenuItem("Divide");
        modMenuItem = new JMenuItem("Modular (Number1 mod Number2)");

        powerMenuItem = new JMenuItem("Power (Number1^Number2)");
```

```
36          squareRootMenuItem = new JMenuItem("Square Root (Number 1 is
    argument)");
37          logMenuItem = new JMenuItem("Logarithm (Number 1 is
    argument, Number 2 is base)");
38          lnMenuItem = new JMenuItem("Natural Logarithm (Number 1 is
    argument)");
39
40          trigInstructionMenuItem = new JMenuItem("Note: Number 1 is
    argument");
41          sinMenuItem = new JMenuItem("Sine");
42          cosMenuItem = new JMenuItem("Cosine");
43          tanMenuItem = new JMenuItem("Tangent");
44          cotMenuItem = new JMenuItem("Cotangent");
45          secMenuItem = new JMenuItem("Secant");
46          cscMenuItem = new JMenuItem("Cosecant");
47
48          factorialMenuItem = new JMenuItem("Factorial");
49          nPrMenuItem = new JMenuItem("Permutations");
50          nCrMenuItem = new JMenuItem("Combinations");
51
52          exitMenuItem = new JMenuItem("Exit Calculator");
53
54          arithmeticMenu.add(addMenuItem);
55          arithmeticMenu.add(subtractMenuItem);
56          arithmeticMenu.add(multiplyMenuItem);
57          arithmeticMenu.add(divideMenuItem);
58          arithmeticMenu.add(modMenuItem);
59
60          exponentialMenu.add(powerMenuItem);
61          exponentialMenu.add(squareRootMenuItem);
62          exponentialMenu.add(logMenuItem);
63          exponentialMenu.add(lnMenuItem);
64
65          trigMenu.add(trigInstructionMenuItem);
66          trigMenu.addSeparator();
67          trigMenu.add(sinMenuItem);
68          trigMenu.add(cosMenuItem);
69          trigMenu.add(tanMenuItem);
70          trigMenu.add(cotMenuItem);
71          trigMenu.add(secMenuItem);
72          trigMenu.add(cscMenuItem);
```

```
73
74          probabilityMenu.add(factorialMenuItem);
75          probabilityMenu.add(nPrMenuItem);
76          probabilityMenu.add(nCrMenuItem);
77
78          exitMenu.add(exitMenuItem);
79
80          JMenuBar menuBar = new JMenuBar();
81          menuBar.add(arithmeticMenu);
82          menuBar.add(exponentialMenu);
83          menuBar.add(trigMenu);
84          menuBar.add(probabilityMenu);
85          menuBar.add(exitMenu);
86
87          for(int i = 0; i<menuBar.getMenuCount();i++)
88          {
89              JMenu currentMenu1 = menuBar.getMenu(i);
90              for(int j = 0; j<currentMenu1.getItemCount();j++)
91              {
92                  JMenuItem currentItem = currentMenu1.getItem(j);
93                  if(currentItem != null)
94                  {
95                      currentItem.addActionListener(this);
96                  }
97              }
98          }
99
100         setJMenuBar(menuBar);
101
102         calculator = new CalculatorFunctions();
103         getContentPane().add(calculator);
104         setDefaultCloseOperation(EXIT_ON_CLOSE);
105         pack();
106     }
107
108     public void actionPerformed(ActionEvent e)
109     {
110         if(e.getSource() == addMenuItem)
111         {
112             calculator.add();
113         }
```

```
114
115         if(e.getSource() == subtractMenuItem)
116         {
117             calculator.subtract();
118         }
119
120         if(e.getSource() == multiplyMenuItem)
121         {
122             calculator.multiply();
123         }
124
125         if(e.getSource() == divideMenuItem)
126         {
127             calculator.divide();
128         }
129
130         if(e.getSource() == modMenuItem)
131         {
132             calculator.mod();
133         }
134
135         if(e.getSource() == powerMenuItem)
136         {
137             calculator.power();
138         }
139
140         if(e.getSource() == squareRootMenuItem)
141         {
142             calculator.squareRoot();
143         }
144
145         if(e.getSource() == logMenuItem)
146         {
147             calculator.log();
148         }
149
150         if(e.getSource() == lnMenuItem)
151         {
152             calculator.ln();
153         }
154
```

```
155         if(e.getSource() == sinMenuItem)
156         {
157             calculator.sin();
158         }
159 
160         if(e.getSource() == cosMenuItem)
161         {
162             calculator.cos();
163         }
164 
165         if(e.getSource() == tanMenuItem)
166         {
167             calculator.tan();
168         }
169 
170         if(e.getSource() == cotMenuItem)
171         {
172             calculator.cot();
173         }
174 
175         if(e.getSource() == secMenuItem)
176         {
177             calculator.sec();
178         }
179 
180         if(e.getSource() == cscMenuItem)
181         {
182             calculator.csc();
183         }
184 
185         if(e.getSource() == factorialMenuItem)
186         {
187             calculator.factorial();
188         }
189 
190         if(e.getSource() == nPrMenuItem)
191         {
192             calculator.nPr();
193         }
194 
195         if (e.getSource() == nCrMenuItem)
```

```
196         {
197             calculator.nCr();
198         }
199 
200         if(e.getSource() == exitMenuItem)
201         {
202             dispose();
203             System.exit(0);
204         }
205     }
206 
207     @SuppressWarnings("deprecation")
208     public static void main(String[] args)
209     {
210         CalculatorInterface cf = new CalculatorInterface();
211         cf.show();
212     }
213 
214 }
215 
```