```
package Info;
import java.awt.SystemColor;
import java.awt.Color;
import javax.swing.GroupLayout.Alignment;
import javax.swing.GroupLayout;

/**
 * this is a frame to show the instructions for the user
 */
public class Instruction extends javax.swing.JFrame {

    /**
     * Creates new form Instruction
     */
    public Instruction() {
        initComponents();
        setLocationRelativeTo(null);
        jTextArea1.setCaretPosition(0);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();
        jTextArea1.setForeground(Color.YELLOW);
        jTextArea1.setBackground(SystemColor.inactiveCaptionText);

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("User  Manual");
        setAlwaysOnTop(true);
        setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        setResizable(false);

        jScrollPane1.setFocusCycleRoot(true);
        jScrollPane1.setHorizontalScrollBar(null);

        jTextArea1.setEditable(false);
        jTextArea1.setColumns(20);
        jTextArea1.setFont(new java.awt.Font("Monospaced", 0, 14)); // NOI18N
        jTextArea1.setRows(5);
```

```
jTextArea1.setText("#OPERATION MODES\r\n----------\r\n-Fixed mode\r\n It works by setting the initial jobs data for the program\r\n to work on. \r\n\r\n-Random mode\r\n The program sets the initial data itself randomly.\r\n\r\n----------\r\n#CONTROLS \r\n----------\r\n- (num of process) \r\n set the number of wanted processes for the simulation\r\n and it can vary from 1 to 10 jobs.\r\n\r\n- (Algorithm)\r\n Choose the wanted algorithm for the simulation.\r\n 1) FCFS -> First Come First Serve\r\n 2) SJF -> Shortest Job First\r\n 3) STRF -> Shortest Time Remaining First\r\n 4) RR -> Round Robin \r\n 5) priority1 -> Priority Not Preemptive\r\n 6) priority2 -> Priority Preemptive\r\n\r\n- (Speed)\r\n control the speed of the automatic simulation, \r\n 1 is the fastest, and 7 is the slowest.\r\n\r\n- (Time Slice) \r\n select the desired quantum time for the Round \r\n Robin algorithm.\r\n\r\n- (Simulate button)\r\n .start the automatic simulation by the speed selected\r\n at \u00C3\u00A2\u00E2\u201A\u00AC\u00C5\u201Csim speed\u00C3\u00A2\u00E2\u201A\u00AC\u00EF\u00BF\u00BD combo box.\r\n .resume the simulation after pausing.\r\n\r\n- (Stop button) \r\n pause the automatic simulation.\r\n\r\n- (Next Step button)\r\n used for running the simulation manually step by step after\r\n every click.\r\n\r\n- (Restart button) \r\n restart the simulation from the beginning with the same \r\n data and the same number of jobs (reset all the data\r\n to the start point).\r\n\r\n- (Start another simulation button)\r\n restart the simulation with new random data.\r\n \r\n- (Finish button)\r\n reach the end of the simulation immediately.\r\n----------\r\n#INDIVIDUAL PROCESSES\r\n----------\r\n the table shows all the data of all the jobs in the \r\n simulation.\r\n \r\n -# : job number.\r\n -arrive : job arrival time.\r\n -burst : job burst time.\r\n -priority : job priority.\r\n -start : job start execution time.\r\n -wait : all job wait time (so far).\r\n -remain : all job remain time (so far).\r\n -finish : job finish execution time.\r\n -turn : job turnaround time (so far).\r\n -% : the finished percent of the job.\r\n----------\r\n#READY QUEUE\r\n----------\r\n show the jobs in the ready queue in specific time.\r\n note : if the ready queue is empty and the cpu is idle\r\n the new job won't appear in the ready queue and will be\r\n executed on the cpu immediately.\r\n----------\r\n#UTILIZATION\r\n----------\r\n show the utilization of the cpu (so far).\r\n ( the percent of working time of the cpu to the idle time)\r\n----------\r\n#AVERAGE\r\n--------\r\n - waiting time\r\n show the average wait time of all the jobs (so far).\r\n \r\n - turnaround time\r\n show the average turnaround time of all the jobs (so far).\r\n----------\r\n#Gantt chart\r\n----------\r\n visualization of the gantt chart of all the executed jobs\r\n in the simulation so far.\r\n----------");
        jScrollPane1.setViewportView(jTextArea1);

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        layout.setHorizontalGroup(
            layout.createParallelGroup(Alignment.LEADING)
                    .addGroup(Alignment.TRAILING, layout.createSequentialGroup()
                            .addContainerGap()
                            .addComponent(jScrollPane1, GroupLayout.DEFAULT_SIZE, 517,
Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(Alignment.LEADING)
                    .addGroup(Alignment.TRAILING, layout.createSequentialGroup()
                            .addContainerGap()
                            .addComponent(jScrollPane1, GroupLayout.DEFAULT_SIZE, 350,
Short.MAX_VALUE))
```

```
        );
        getContentPane().setLayout(layout);

        pack();
    }// </editor-fold>

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(Instruction.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(Instruction.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(Instruction.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(Instruction.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Instruction().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextArea jTextArea1;
```

```
    // End of variables declaration
}
```