

## newPlayer.java

```
1
2 import java.awt.event.KeyEvent;
6
8  * To change this license header, choose License Headers in Project
   Properties.
12
13 /**
14  *
15  * @author Tanim
16  */
17 public class newPlayer extends javax.swing.JFrame {
18     Player[] list = new Player[30];
19     /**
20      * Creates new form newPlayer
21      */
22     public newPlayer() {
23         initComponents();
24     }
25     public newPlayer(Player[] pl){
26         initComponents();
27         list = pl;
28     }
29
30     /**
31      * This method is called from within the constructor to initialize the
   form.
32      * WARNING: Do NOT modify this code. The content of this method is
   always
33      * regenerated by the Form Editor.
34      */
35     @SuppressWarnings("unchecked")
36     // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
   BEGIN: initComponents
37     private void initComponents() {
38
39         GRU = new javax.swing.JLabel();
40         BBallTeam = new javax.swing.JLabel();
41         FName = new javax.swing.JLabel();
42         LName = new javax.swing.JLabel();
43         GmsPlayed = new javax.swing.JLabel();
44         P0sition = new javax.swing.JLabel();
45         fname = new javax.swing.JTextField();
```

newPlayer.java

```
46     lname = new javax.swing.JTextField();
47     gamesPlayed = new javax.swing.JTextField();
48     Rebounds = new javax.swing.JLabel();
49     Assists = new javax.swing.JLabel();
50     Number = new javax.swing.JLabel();
51     Points = new javax.swing.JLabel();
52     rebounds = new javax.swing.JTextField();
53     assists = new javax.swing.JTextField();
54     number = new javax.swing.JTextField();
55     points = new javax.swing.JTextField();
56     createPlayer = new javax.swing.JButton();
57     Cancel = new javax.swing.JButton();
58     pos = new javax.swing.JComboBox<>();
59     jLabel1 = new javax.swing.JLabel();
60
61     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
62     setMinimumSize(new java.awt.Dimension(700, 400));
63     setResizable(false);
64     setSize(new java.awt.Dimension(700, 400));
65     getContentPane().setLayout(new
org.netbeans.lib.awtextra.AbsoluteLayout());
66
67     GRU.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N
68     GRU.setForeground(new java.awt.Color(255, 255, 255));
69     GRU.setText("Grand Ridge University ");
70     getContentPane().add(GRU, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 20, -1, -1));
71
72     BBallTeam.setFont(new java.awt.Font("Times New Roman", 1, 36)); //
NOI18N
73     BBallTeam.setForeground(new java.awt.Color(255, 255, 255));
74     BBallTeam.setText("Basketball Team");
75     getContentPane().add(BBallTeam, new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 80, -1, -1));
76
77     FName.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
78     FName.setForeground(new java.awt.Color(255, 255, 255));
79     FName.setText("First Name:");
80     FName.setToolTipText("");
81     getContentPane().add(FName, new
```

## newPlayer.java

```
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 140, -1, -1));
82
83     LName.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
84     LName.setForeground(new java.awt.Color(255, 255, 255));
85     LName.setText("Last Name:");
86     getContentPane().add(LName, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 190, -1, -1));
87
88     GmsPlayed.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
89     GmsPlayed.setForeground(new java.awt.Color(255, 255, 255));
90     GmsPlayed.setText("Games Played:");
91     getContentPane().add(GmsPlayed, new
org.netbeans.lib.awtextra.AbsoluteConstraints(340, 140, -1, -1));
92
93     P0sition.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
94     P0sition.setForeground(new java.awt.Color(255, 255, 255));
95     P0sition.setText("Position:");
96     P0sition.setToolTipText("");
97     getContentPane().add(P0sition, new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 240, -1, -1));
98
99     fname.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
100     fname.addActionListener(new java.awt.event.ActionListener() {
101         public void actionPerformed(java.awt.event.ActionEvent evt) {
102             fnameActionPerformed(evt);
103         }
104     });
105     fname.addKeyListener(new java.awt.event.KeyAdapter() {
106         public void keyPressed(java.awt.event.KeyEvent evt) {
107             fnameKeyPressed(evt);
108         }
109     });
110     getContentPane().add(fname, new
org.netbeans.lib.awtextra.AbsoluteConstraints(170, 140, 150, 30));
111
112     lname.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
113     lname.addKeyListener(new java.awt.event.KeyAdapter() {
```

newPlayer.java

```
114         public void keyTyped(java.awt.event.KeyEvent evt) {
115             lnameKeyTyped(evt);
116         }
117     });
118     getContentPane().add(lname, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(170, 190, 150, 30));
119
120     gamesPlayed.setFont(new java.awt.Font("Times New Roman", 1,
    14)); // NOI18N
121     gamesPlayed.addKeyListener(new java.awt.event.KeyAdapter() {
122         public void keyTyped(java.awt.event.KeyEvent evt) {
123             gamesPlayedKeyTyped(evt);
124         }
125     });
126     getContentPane().add(gamesPlayed, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(500, 140, 150, 30));
127
128     Rebounds.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
    NOI18N
129     Rebounds.setForeground(new java.awt.Color(255, 255, 255));
130     Rebounds.setText("Rebounds:");
131     getContentPane().add(Rebounds, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(360, 290, -1, -1));
132
133     Assists.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
    NOI18N
134     Assists.setForeground(new java.awt.Color(255, 255, 255));
135     Assists.setText("Assists:");
136     getContentPane().add(Assists, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(380, 240, -1, -1));
137
138     Number.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
    NOI18N
139     Number.setForeground(new java.awt.Color(255, 255, 255));
140     Number.setText("Number:");
141     getContentPane().add(Number, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(50, 290, -1, -1));
142
143     Points.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
    NOI18N
144     Points.setForeground(new java.awt.Color(255, 255, 255));
145     Points.setText("Points:");
```

newPlayer.java

```
146         getContentPane().add(Points, new
org.netbeans.lib.awtextra.AbsoluteConstraints(380, 190, 80, 30));
147
148         rebounds.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
149         rebounds.addKeyListener(new java.awt.event.KeyAdapter() {
150             public void keyTyped(java.awt.event.KeyEvent evt) {
151                 reboundsKeyTyped(evt);
152             }
153         });
154         getContentPane().add(rebounds, new
org.netbeans.lib.awtextra.AbsoluteConstraints(500, 290, 150, 30));
155
156         assists.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
157         assists.addKeyListener(new java.awt.event.KeyAdapter() {
158             public void keyTyped(java.awt.event.KeyEvent evt) {
159                 assistsKeyTyped(evt);
160             }
161         });
162         getContentPane().add(assists, new
org.netbeans.lib.awtextra.AbsoluteConstraints(500, 240, 150, 30));
163
164         number.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
165         number.addKeyListener(new java.awt.event.KeyAdapter() {
166             public void keyTyped(java.awt.event.KeyEvent evt) {
167                 numberKeyTyped(evt);
168             }
169         });
170         getContentPane().add(number, new
org.netbeans.lib.awtextra.AbsoluteConstraints(170, 290, 150, 30));
171
172         points.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
173         points.addKeyListener(new java.awt.event.KeyAdapter() {
174             public void keyTyped(java.awt.event.KeyEvent evt) {
175                 pointsKeyTyped(evt);
176             }
177         });
178         getContentPane().add(points, new
org.netbeans.lib.awtextra.AbsoluteConstraints(500, 190, 150, 30));
```

newPlayer.java

```
179
180     createPlayer.setFont(new java.awt.Font("Times New Roman", 1,
181 14)); // NOI18N
182     createPlayer.setText("Create Player");
183     createPlayer.addActionListener(new java.awt.event.ActionListener()
184     {
185         public void actionPerformed(java.awt.event.ActionEvent evt) {
186             createPlayerActionPerformed(evt);
187         }
188     });
189     getContentPane().add(createPlayer, new
190 org.netbeans.lib.awtextra.AbsoluteConstraints(220, 340, 120, 30));
191
192     Cancel.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
193 NOI18N
194     Cancel.setText("Cancel");
195     Cancel.addActionListener(new java.awt.event.ActionListener() {
196         public void actionPerformed(java.awt.event.ActionEvent evt) {
197             CancelActionPerformed(evt);
198         }
199     });
200     getContentPane().add(Cancel, new
201 org.netbeans.lib.awtextra.AbsoluteConstraints(370, 340, 120, 30));
202
203     pos.setModel(new javax.swing.DefaultComboBoxModel<(new String[]
204 { "N/A", "Forward", "Guard" }));
205     getContentPane().add(pos, new
206 org.netbeans.lib.awtextra.AbsoluteConstraints(170, 240, 150, 30));
207
208     jLabel1.setIcon(new
209 javax.swing.ImageIcon(getClass().getResource("/basketball/1.jpg"))); //
210 NOI18N
211     jLabel1.setText("jLabel1");
212     jLabel1.setMaximumSize(new java.awt.Dimension(700, 400));
213     jLabel1.setMinimumSize(new java.awt.Dimension(700, 400));
214     jLabel1.setPreferredSize(new java.awt.Dimension(770, 440));
215     getContentPane().add(jLabel1, new
216 org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));
217
218     setSize(new java.awt.Dimension(716, 439));
219     setLocationRelativeTo(null);
220 } // </editor-fold> // GEN-END: initComponents
```

# newPlayer.java

```

211
212     private void createPlayerActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_createPlayerActionPerformed
213         String name1 = fname.getText();
214         String name2 = lname.getText();
215         int gp = 0;
216         int reb = 0;
217         int as = 0;
218         int n = 0;
219         int p = 0;
220         if(!gamesPlayed.getText().equals("")){
221             if(!rebounds.getText().equals("")){
222                 if(!assists.getText().equals("")){
223                     if(!number.getText().equals("")){
224                         if(!points.getText().equals("")){
225                             gp = Integer.parseInt(gamesPlayed.getText());
226                             reb = Integer.parseInt(rebounds.getText());
227                             as = Integer.parseInt(assists.getText());
228                             n = Integer.parseInt(number.getText());
229                             p = Integer.parseInt(points.getText());
230                         }
231                     }
232                 }
233             }
234         }
235         String position = (String) pos.getSelectedItem();
236         boolean player = false;
237         if(fname.getText().equals("")||lname.getText().equals("")||
    gamesPlayed.getText().equals("")||rebounds.getText().equals("")||
    assists.getText().equals("")|| number.getText().equals("")||
    points.getText().equals("")|| position.equals("N/A")){
238             JOptionPane.showMessageDialog(null, "Please fill in all
    required fields.");
239             player = false;
240         }else{
241             player = true;
242         }
243         if(player == true){
244             /*
245             list.add(new Player(fname.getText(), lname.getText(), gp, n,
    reb, as, position, p ));
246             JOptionPane.showMessageDialog(null, "The player " +

```

newPlayer.java

```
    fname.getText() + " " + lname.getText()+ " was added to the database.");
247        updateFile(list);*/
248        boolean found = false; //control variable for the next empty
    slot
249        int index = 0; //has to be 0 to 29
250        while(found == false){
251            if(index >= 0 && index <= 29 ){
252                if(list[index] == null){
253                    list[index] = new Player(fname.getText(),
    lname.getText(), gp, n, reb, as, position, p );
254                    JOptionPane.showMessageDialog(null, "A new
    Player was successfully added.");
255                    found = true;
256                }
257                index++;
258            }
259        }
260
261
262        updateFile(list);
263
264
265        this.setVisible(false);
266        StartFrame st = new StartFrame();
267        st.setVisible(true);
268    }
269
270 }//GEN-LAST:event_createPlayerActionPerformed
271
272 private boolean isEmpty(Player[] ab){
273     boolean isit = true;
274     for(int i = 0; i< ab.length; i++ ){//if there is even a single
    non-null, the list isn't empty
275         if(ab[i] != null){
276             isit = false;
277         }
278     }
279     return isit;
280 }
281 private void updateFile(Player[] lt){
282     try{
283         FileOutputStream fos= new FileOutputStream("Players");
```



newPlayer.java

```
284         ObjectOutputStream oos= new ObjectOutputStream(fos);
285         oos.writeObject(lt); //this is a way to store the
    arraylist in a file
286         oos.close();
287         fos.close();
288     }catch(IOException ioe){
289         ioe.printStackTrace();
290     }
291 }
292 private void CancelActionPerformed(java.awt.event.ActionEvent evt) {//
GEN-FIRST:event_CancelActionPerformed
293     // TODO add your handling code here:
294     StartFrame sf = new StartFrame();
295     this.setVisible(false);
296     sf.setVisible(true);
297 }//GEN-LAST:event_CancelActionPerformed
298
299 private void fnameKeyTyped(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_fnameKeyTyped
300     // TODO add your handling code here:
301     char c=evt.getKeyChar();
302     if(!(Character.isAlphabetic(c) || (c==KeyEvent.VK_BACK_SPACE)||
c==KeyEvent.VK_DELETE ))
303         evt.consume();
304 }//GEN-LAST:event_fnameKeyTyped
305
306 private void lnameKeyTyped(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_lnameKeyTyped
307     // TODO add your handling code here:
308     char c=evt.getKeyChar();
309     if(!(Character.isAlphabetic(c) || (c==KeyEvent.VK_BACK_SPACE)||
c==KeyEvent.VK_DELETE ))
310         evt.consume();
311 }//GEN-LAST:event_lnameKeyTyped
312
313 private void gamesPlayedKeyTyped(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_gamesPlayedKeyTyped
314     // TODO add your handling code here:
315     char c = evt.getKeyChar();
316     if(!(Character.isDigit(c) || c == KeyEvent.VK_BACK_SPACE) || c ==
KeyEvent.VK_DELETE){
317         evt.consume();
```

newPlayer.java

```
318     }
319     }//GEN-LAST:event_gamesPlayedKeyTyped
320
321     private void reboundsKeyTyped(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_reboundsKeyTyped
322         // TODO add your handling code here:
323         char c = evt.getKeyChar();
324         if(!(Character.isDigit(c) || c == KeyEvent.VK_BACK_SPACE) || c ==
KeyEvent.VK_DELETE){
325             evt.consume();
326         }
327     }//GEN-LAST:event_reboundsKeyTyped
328
329     private void assistsKeyTyped(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_assistsKeyTyped
330         // TODO add your handling code here:
331         char c = evt.getKeyChar();
332         if(!(Character.isDigit(c) || c == KeyEvent.VK_BACK_SPACE) || c ==
KeyEvent.VK_DELETE){
333             evt.consume();
334         }
335     }//GEN-LAST:event_assistsKeyTyped
336
337     private void numberKeyTyped(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_numberKeyTyped
338         // TODO add your handling code here:
339         char c = evt.getKeyChar();
340         if(!(Character.isDigit(c) || c == KeyEvent.VK_BACK_SPACE) || c ==
KeyEvent.VK_DELETE){
341             evt.consume();
342         }
343     }//GEN-LAST:event_numberKeyTyped
344
345     private void pointsKeyTyped(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_pointsKeyTyped
346         // TODO add your handling code here:
347         char c = evt.getKeyChar();
348         if(!(Character.isDigit(c) || c == KeyEvent.VK_BACK_SPACE) || c ==
KeyEvent.VK_DELETE){
349             evt.consume();
350         }
351     }//GEN-LAST:event_pointsKeyTyped
```

## newPlayer.java

```
352
353     private void fnameActionPerformed(java.awt.event.ActionEvent evt) {//
GEN-FIRST:event_fnameActionPerformed
354         // TODO add your handling code here:
355     }//GEN-LAST:event_fnameActionPerformed
356
357     /**
358      * @param args the command line arguments
359      */
360     public static void main(String args[]) {
361         /* Set the Nimbus look and feel */
362         //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
363         /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
364          * For details see http://download.oracle.com/javase/tutorial/
uiswing/lookandfeel/plaf.html
365          */
366         try {
367             for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
368                 if ("Nimbus".equals(info.getName())) {
369                     javax.swing.UIManager.setLookAndFeel(info.getClassName());
370                     break;
371                 }
372             }
373         } catch (ClassNotFoundException ex) {
374             java.util.logging.Logger.getLogger(newPlayer.class.getName()).log(java.util
l.logging.Level.SEVERE, null, ex);
375         } catch (InstantiationException ex) {
376             java.util.logging.Logger.getLogger(newPlayer.class.getName()).log(java.util
l.logging.Level.SEVERE, null, ex);
377         } catch (IllegalAccessException ex) {
378             java.util.logging.Logger.getLogger(newPlayer.class.getName()).log(java.util
l.logging.Level.SEVERE, null, ex);
379         } catch (javax.swing.UnsupportedLookAndFeelException ex) {
380             java.util.logging.Logger.getLogger(newPlayer.class.getName()).log(java.util
```

newPlayer.java

```
l.logging.Level.SEVERE, null, ex);
381     }
382     //</editor-fold>
383
384     /* Create and display the form */
385     java.awt.EventQueue.invokeLater(new Runnable() {
386         public void run() {
387             new newPlayer().setVisible(true);
388         }
389     });
390 }
391
392 // Variables declaration - do not modify//GEN-BEGIN:variables
393 private javax.swing.JLabel Assists;
394 private javax.swing.JLabel BBallTeam;
395 private javax.swing.JButton Cancel;
396 private javax.swing.JLabel FName;
397 private javax.swing.JLabel GRU;
398 private javax.swing.JLabel GmsPlayed;
399 private javax.swing.JLabel LName;
400 private javax.swing.JLabel Number;
401 private javax.swing.JLabel P0sition;
402 private javax.swing.JLabel Points;
403 private javax.swing.JLabel Rebounds;
404 private javax.swing.JTextField assists;
405 private javax.swing.JButton createPlayer;
406 private javax.swing.JTextField fname;
407 private javax.swing.JTextField gamesPlayed;
408 private javax.swing.JLabel jLabel1;
409 private javax.swing.JTextField lname;
410 private javax.swing.JTextField number;
411 private javax.swing.JTextField points;
412 private javax.swing.JComboBox<String> pos;
413 private javax.swing.JTextField rebounds;
414 // End of variables declaration//GEN-END:variables
415 }
416
```

## Player.java

```
2  * To change this license header, choose License Headers in Project
   Properties.
11 import java.io.*;
12 public class Player implements Serializable{
13
14     String fName;
15     String lName;
16     //int height;
17     int totalGames;
18     int jerseyNumber;
19     int rebounds;
20     int assist;
21     String position;
22     int points;
23
24     public Player(String f, String l, int totalg, int jN, int re, int as,
   String pos, int pt){
25         fName = f;
26         lName = l;
27         totalGames = totalg;
28         jerseyNumber = jN;
29         rebounds = re;
30         assist = as;
31         position = pos;
32         points = pt;
33     }
34
35 }
36
```

## StartFrame.java

```
1 import java.io.*;
2
3 public class StartFrame extends javax.swing.JFrame {
4     Player[] pl = new Player[30]; //max allowed is 30 ppl
5     /**
6      * Creates new form StartFrame
7      */
8     public StartFrame() {
9         initComponents();
10        File(); //read from file
11    }
12
13    /**
14     * This method is called from within the constructor to initialize the
15     * form.
16     * WARNING: Do NOT modify this code. The content of this method is
17     * always
18     * regenerated by the Form Editor.
19     */
20    @SuppressWarnings("unchecked")
21    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
22    BEGIN:initComponents
23    private void initComponents() {
24
25        GRU = new javax.swing.JLabel();
26        ViewPlayers = new javax.swing.JButton();
27        AddPlayer = new javax.swing.JButton();
28        Exit = new javax.swing.JButton();
29        BBallTeam = new javax.swing.JLabel();
30        Logo = new javax.swing.JLabel();
31        HomeOfTheBears = new javax.swing.JLabel();
32        jLabel2 = new javax.swing.JLabel();
33
34        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
35        setMinimumSize(new java.awt.Dimension(700, 400));
36        getContentPane().setLayout(new
37        org.netbeans.lib.awtextra.AbsoluteLayout());
38
39        GRU.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N
40        GRU.setForeground(new java.awt.Color(255, 255, 255));
41        GRU.setText("Grand Ridge University ");
42        getContentPane().add(GRU, new
```

## StartFrame.java

```
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 20, -1, -1));
39
40    ViewPlayers.setFont(new java.awt.Font("Times New Roman", 1,
18)); // NOI18N
41    ViewPlayers.setText("View Players");
42    ViewPlayers.addActionListener(new java.awt.event.ActionListener()
{
43        public void actionPerformed(java.awt.event.ActionEvent evt) {
44            ViewPlayersActionPerformed(evt);
45        }
46    });
47    getContentPane().add(ViewPlayers, new
org.netbeans.lib.awtextra.AbsoluteConstraints(410, 220, 200, 40));
48
49    AddPlayer.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
NOI18N
50    AddPlayer.setText("Add Players");
51    AddPlayer.addActionListener(new java.awt.event.ActionListener() {
52        public void actionPerformed(java.awt.event.ActionEvent evt) {
53            AddPlayerActionPerformed(evt);
54        }
55    });
56    getContentPane().add(AddPlayer, new
org.netbeans.lib.awtextra.AbsoluteConstraints(410, 160, 200, 40));
57
58    Exit.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
NOI18N
59    Exit.setText("Exit");
60    Exit.setCursor(new
java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
61    Exit.addActionListener(new java.awt.event.ActionListener() {
62        public void actionPerformed(java.awt.event.ActionEvent evt) {
63            ExitActionPerformed(evt);
64        }
65    });
66    getContentPane().add(Exit, new
org.netbeans.lib.awtextra.AbsoluteConstraints(410, 280, 200, 40));
67
68    BBallTeam.setFont(new java.awt.Font("Times New Roman", 1, 36)); //
NOI18N
69    BBallTeam.setForeground(new java.awt.Color(255, 255, 255));
70    BBallTeam.setText("Basketball Team");
```

## StartFrame.java

```
71         getContentPane().add(BBallTeam, new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 80, -1, -1));
72
73         Logo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/
basketball/Bear Basketball IA.png"))); // NOI18N
74         getContentPane().add(Logo, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 160, -1, 210));
75
76         HomeOfTheBears.setFont(new java.awt.Font("Times New Roman", 1,
24)); // NOI18N
77         HomeOfTheBears.setForeground(new java.awt.Color(204, 153, 0));
78         HomeOfTheBears.setText("Home of the Bears");
79         getContentPane().add(HomeOfTheBears, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 130, -1, -1));
80
81         jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/basketball/1.jpg"))); //
NOI18N
82         jLabel2.setMaximumSize(new java.awt.Dimension(700, 400));
83         jLabel2.setMinimumSize(new java.awt.Dimension(700, 400));
84         jLabel2.setPreferredSize(new java.awt.Dimension(700, 400));
85         getContentPane().add(jLabel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 700, -1));
86
87         setSize(new java.awt.Dimension(716, 439));
88         setLocationRelativeTo(null);
89     } // </editor-fold> // GEN-END: initComponents
90
91     public void File(){
92         File temp = new File("Players");
93         if(temp.exists()== true){ //if the file is there
94             //if the file exists, read it it into the arraylist of users
95             try
96             {
97                 FileInputStream fis = new FileInputStream("Players");
98                 ObjectInputStream ois = new ObjectInputStream(fis);
99                 pl = (Player[]) ois.readObject();
100                 ois.close();
101                 fis.close();
102             } catch(IOException ioe){
103                 ioe.printStackTrace();
104                 return;
```



# StartFrame.java

```

105         }catch(ClassNotFoundException c){
106             System.out.println("Class not found");
107             c.printStackTrace();
108             return;
109         }
110         //for(int i = 0; i< pl.length; i++)
111             //System.out.println(pl[i]);//works
112     }else{
113         pl[0] = new Player("Anthony", "Davis", 45, 23, 12, 17,
114 "Forward", 36); //if the file isn't there
115
116         /*boolean found = false;
117         int index = 0; //has to be 0 to 29
118         if(isEmpty(pl) == false){
119             while(found == false){
120                 if(index >= 0 && index <= 29 ){
121                     if(pl[index] == null){
122                         pl[index] = new Player("Anthony", "Davis", 45,
123 23, 12, 17, "Forward", 36); //if the file isn't there
124                         found = true;
125                     }
126                 }
127             }
128         }
129
130     private void AddPlayerActionPerformed(java.awt.event.ActionEvent evt)
131     {//GEN-FIRST:event_AddPlayerActionPerformed
132         // TODO add your handling code here:
133         this.setVisible(false);
134         newPlayer ne = new newPlayer(pl);
135         ne.setVisible(true);
136     }//GEN-LAST:event_AddPlayerActionPerformed
137
138     private void ViewPlayersActionPerformed(java.awt.event.ActionEvent
139     evt) {//GEN-FIRST:event_ViewPlayersActionPerformed
140         // TODO add your handling code here:
141         this.setVisible(false);
142         ViewPlayers vp = new ViewPlayers(pl);
143         vp.setVisible(true);
144     }//GEN-LAST:event_ViewPlayersActionPerformed

```

## StartFrame.java

```
143
144     private void ExitActionPerformed(java.awt.event.ActionEvent evt) {//
GEN-FIRST:event_ExitActionPerformed
145         // TODO add your handling code here:
146
147         //write to file and then close
148         System.exit(0);
149     }//GEN-LAST:event_ExitActionPerformed
150
151     /**
152      * @param args the command line arguments
153      */
154     public static void main(String args[]) {
155         /* Set the Nimbus look and feel */
156         //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
157         /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
158          * For details see http://download.oracle.com/javase/tutorial/
uiswing/lookandfeel/plaf.html
159          */
160         try {
161             for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
162                 if ("Nimbus".equals(info.getName())) {
163
javax.swing.UIManager.setLookAndFeel(info.getClassName());
164                 break;
165             }
166         }
167     } catch (ClassNotFoundException ex) {
168
java.util.logging.Logger.getLogger(StartFrame.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
169     } catch (InstantiationException ex) {
170
java.util.logging.Logger.getLogger(StartFrame.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
171     } catch (IllegalAccessException ex) {
172
java.util.logging.Logger.getLogger(StartFrame.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
```

## StartFrame.java

```
173         } catch (javax.swing.UnsupportedLookAndFeelException ex) {
174
175             java.util.logging.Logger.getLogger(StartFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
176         }
177         //</editor-fold>
178         /* Create and display the form */
179         java.awt.EventQueue.invokeLater(new Runnable() {
180             public void run() {
181                 new StartFrame().setVisible(true);
182             }
183         });
184     }
185
186     // Variables declaration - do not modify//GEN-BEGIN:variables
187     private javax.swing.JButton AddPlayer;
188     private javax.swing.JLabel BBallTeam;
189     private javax.swing.JButton Exit;
190     private javax.swing.JLabel GRU;
191     private javax.swing.JLabel HomeOfTheBears;
192     private javax.swing.JLabel Logo;
193     private javax.swing.JButton ViewPlayers;
194     private javax.swing.JLabel jLabel2;
195     // End of variables declaration//GEN-END:variables
196 }
197
```

## updateInfo.java

```
1
2 import java.awt.event.KeyEvent;
5
7 * To change this license header, choose License Headers in Project
  Properties.
11
12 /**
13  *
14  * @author Tanim
15  */
16 public class updateInfo extends javax.swing.JFrame {
17
18     /**
19      * Creates new form updateInfo
20      */
21     Player[] ppl;
22     Player temp = null;
23     public updateInfo(){
24         initComponents();
25     }
26     public updateInfo(Player [] p, Player t) {
27         ppl = p;
28         temp = t;
29         initComponents();
30         updateScreen();
31     }
32
33     /**
34      * This method is called from within the constructor to initialize the
  form.
35      * WARNING: Do NOT modify this code. The content of this method is
  always
36      * regenerated by the Form Editor.
37      */
38     @SuppressWarnings("unchecked")
39     // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
  BEGIN:initComponents
40     private void initComponents() {
41
42         GRU = new javax.swing.JLabel();
43         BBallTeam = new javax.swing.JLabel();
44         name1 = new javax.swing.JTextField();
```

## updateInfo.java

```
45     name2 = new javax.swing.JTextField();
46     numb = new javax.swing.JTextField();
47     pt = new javax.swing.JTextField();
48     ast = new javax.swing.JTextField();
49     rb = new javax.swing.JTextField();
50     gms = new javax.swing.JTextField();
51     newGms = new javax.swing.JTextField();
52     FName = new javax.swing.JLabel();
53     LName = new javax.swing.JLabel();
54     Position = new javax.swing.JLabel();
55     Number = new javax.swing.JLabel();
56     GmsPlayed = new javax.swing.JLabel();
57     Points = new javax.swing.JLabel();
58     Assists = new javax.swing.JLabel();
59     Rebounds = new javax.swing.JLabel();
60     newReb = new javax.swing.JTextField();
61     newAst = new javax.swing.JTextField();
62     pts = new javax.swing.JTextField();
63     nu = new javax.swing.JTextField();
64     posit = new javax.swing.JTextField();
65     newName2 = new javax.swing.JTextField();
66     newName1 = new javax.swing.JTextField();
67     jLabel12 = new javax.swing.JLabel();
68     jLabel13 = new javax.swing.JLabel();
69     posCombo = new javax.swing.JComboBox<>();
70     UpdateInfo = new javax.swing.JButton();
71     Cancel = new javax.swing.JButton();
72     jLabel1 = new javax.swing.JLabel();
73
74     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
75     getContentPane().setLayout(new
org.netbeans.lib.awtextra.AbsoluteLayout());
76
77     GRU.setFont(new java.awt.Font("Times New Roman", 1, 30)); //
NOI18N
78     GRU.setForeground(new java.awt.Color(255, 255, 255));
79     GRU.setText("Grand Ridge University ");
80     getContentPane().add(GRU, new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 30, -1, -1));
81
82     BBallTeam.setFont(new java.awt.Font("Times New Roman", 1, 30)); //
```

## updateInfo.java

NOI18N

```
83      BBallTeam.setForeground(new java.awt.Color(255, 255, 255));
84      BBallTeam.setText("Basketball Team");
85      getContentPane().add(BBallTeam, new
org.netbeans.lib.awtextra.AbsoluteConstraints(370, 70, -1, -1));
86
87      name1.setEditable(false);
88      name1.addActionListener(new java.awt.event.ActionListener() {
89          public void actionPerformed(java.awt.event.ActionEvent evt) {
90              name1ActionPerformed(evt);
91          }
92      });
93      getContentPane().add(name1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 140, 100, -1));
94
95      name2.setEditable(false);
96      getContentPane().add(name2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 170, 100, -1));
97
98      numb.setEditable(false);
99      getContentPane().add(numb, new
org.netbeans.lib.awtextra.AbsoluteConstraints(260, 230, 100, -1));
100
101      pt.setEditable(false);
102      getContentPane().add(pt, new
org.netbeans.lib.awtextra.AbsoluteConstraints(310, 260, 100, -1));
103
104      ast.setEditable(false);
105      getContentPane().add(ast, new
org.netbeans.lib.awtextra.AbsoluteConstraints(360, 290, 100, -1));
106
107      rb.setEditable(false);
108      getContentPane().add(rb, new
org.netbeans.lib.awtextra.AbsoluteConstraints(410, 320, 100, -1));
109
110      gms.setEditable(false);
111      getContentPane().add(gms, new
org.netbeans.lib.awtextra.AbsoluteConstraints(460, 350, 100, -1));
112
113      newGms.setBackground(new java.awt.Color(204, 204, 204));
114      newGms.addKeyListener(new java.awt.event.KeyAdapter() {
115          public void keyTyped(java.awt.event.KeyEvent evt) {
```

## updateInfo.java

```
116         newGmsKeyTyped(evt);
117     }
118 });
119 getContentPane().add(newGms, new
org.netbeans.lib.awtextra.AbsoluteConstraints(570, 350, 100, -1));
120
121 FName.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
NOI18N
122 FName.setForeground(new java.awt.Color(255, 255, 255));
123 FName.setText("First Name:");
124 getContentPane().add(FName, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 130, 100, 30));
125
126 LName.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
NOI18N
127 LName.setForeground(new java.awt.Color(255, 255, 255));
128 LName.setText("Last Name:");
129 getContentPane().add(LName, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 170, -1, -1));
130
131 Position.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
NOI18N
132 Position.setForeground(new java.awt.Color(255, 255, 255));
133 Position.setText("Position:");
134 getContentPane().add(Position, new
org.netbeans.lib.awtextra.AbsoluteConstraints(130, 200, -1, 20));
135
136 Number.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
NOI18N
137 Number.setForeground(new java.awt.Color(255, 255, 255));
138 Number.setText("Number:");
139 getContentPane().add(Number, new
org.netbeans.lib.awtextra.AbsoluteConstraints(180, 230, 70, 20));
140
141 GmsPlayed.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
NOI18N
142 GmsPlayed.setForeground(new java.awt.Color(255, 255, 255));
143 GmsPlayed.setText("Gms Played:");
144 getContentPane().add(GmsPlayed, new
org.netbeans.lib.awtextra.AbsoluteConstraints(350, 350, -1, -1));
145
146 Points.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
```

## updateInfo.java

```
NOI18N
147     Points.setForeground(new java.awt.Color(255, 255, 255));
148     Points.setText("Points:");
149     getContentPane().add(Points, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 260, -1, -1));
150
151     Assists.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
NOI18N
152     Assists.setForeground(new java.awt.Color(255, 255, 255));
153     Assists.setText("Assists:");
154     getContentPane().add(Assists, new
org.netbeans.lib.awtextra.AbsoluteConstraints(290, 290, -1, -1));
155
156     Rebounds.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
NOI18N
157     Rebounds.setForeground(new java.awt.Color(255, 255, 255));
158     Rebounds.setText("Rebounds:");
159     getContentPane().add(Rebounds, new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 320, -1, -1));
160
161     newReb.setBackground(new java.awt.Color(204, 204, 204));
162     newReb.addKeyListener(new java.awt.event.KeyAdapter() {
163         public void keyTyped(java.awt.event.KeyEvent evt) {
164             newRebKeyTyped(evt);
165         }
166     });
167     getContentPane().add(newReb, new
org.netbeans.lib.awtextra.AbsoluteConstraints(520, 320, 100, -1));
168
169     newAst.setBackground(new java.awt.Color(204, 204, 204));
170     newAst.addKeyListener(new java.awt.event.KeyAdapter() {
171         public void keyTyped(java.awt.event.KeyEvent evt) {
172             newAstKeyTyped(evt);
173         }
174     });
175     getContentPane().add(newAst, new
org.netbeans.lib.awtextra.AbsoluteConstraints(470, 290, 100, -1));
176
177     pts.setBackground(new java.awt.Color(204, 204, 204));
178     pts.addKeyListener(new java.awt.event.KeyAdapter() {
179         public void keyTyped(java.awt.event.KeyEvent evt) {
180             ptsKeyTyped(evt);
```



## updateInfo.java

```
181         }
182     });
183     getContentPane().add(pts, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(420, 260, 100, -1));
184
185     nu.setBackground(new java.awt.Color(204, 204, 204));
186     nu.addKeyListener(new java.awt.event.KeyAdapter() {
187         public void keyTyped(java.awt.event.KeyEvent evt) {
188             nuKeyTyped(evt);
189         }
190     });
191     getContentPane().add(nu, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(370, 230, 100, -1));
192
193     posit.setEditable(false);
194     getContentPane().add(posit, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(210, 200, 100, -1));
195
196     newName2.setBackground(new java.awt.Color(204, 204, 204));
197     newName2.addKeyListener(new java.awt.event.KeyAdapter() {
198         public void keyTyped(java.awt.event.KeyEvent evt) {
199             newName2KeyTyped(evt);
200         }
201     });
202     getContentPane().add(newName2, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(270, 170, 100, -1));
203
204     newName1.setBackground(new java.awt.Color(204, 204, 204));
205     newName1.addKeyListener(new java.awt.event.KeyAdapter() {
206         public void keyTyped(java.awt.event.KeyEvent evt) {
207             newName1KeyTyped(evt);
208         }
209     });
210     getContentPane().add(newName1, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(230, 140, 100, -1));
211
212     jLabel12.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
    NOI18N
213     jLabel12.setForeground(new java.awt.Color(255, 255, 255));
214     jLabel12.setText("Old Info");
215     getContentPane().add(jLabel12, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(140, 110, -1, -1));
```

## updateInfo.java

```
216
217     jLabel13.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
NOI18N
218     jLabel13.setForeground(new java.awt.Color(255, 255, 255));
219     jLabel13.setText("New Info");
220     getContentPane().add(jLabel13, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(240, 110, -1, -1));
221
222     posCombo.setModel(new javax.swing.DefaultComboBoxModel<>(new
    String[] { "N/A", "Forward", "Guard" }));
223     getContentPane().add(posCombo, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(320, 200, 100, -1));
224
225     UpdateInfo.setFont(new java.awt.Font("Times New Roman", 1,
    14)); // NOI18N
226     UpdateInfo.setText("Update Info");
227     UpdateInfo.addActionListener(new java.awt.event.ActionListener() {
228         public void actionPerformed(java.awt.event.ActionEvent evt) {
229             UpdateInfoActionPerformed(evt);
230         }
231     });
232     getContentPane().add(UpdateInfo, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(40, 350, 110, -1));
233
234     Cancel.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
235     Cancel.setText("Cancel");
236     Cancel.addActionListener(new java.awt.event.ActionListener() {
237         public void actionPerformed(java.awt.event.ActionEvent evt) {
238             CancelActionPerformed(evt);
239         }
240     });
241     getContentPane().add(Cancel, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(160, 350, 110, -1));
242
243     jLabel1.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/basketball/1.jpg"))); //
NOI18N
244     getContentPane().add(jLabel1, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 700, 400));
245
246     setSize(new java.awt.Dimension(716, 439));
```

## updateInfo.java

```
247     setLocationRelativeTo(null);
248 }// </editor-fold>//GEN-END:initComponents
249 private void updateScreen(){
250     name1.setText(temp.fName);
251     name2.setText(temp.lName);
252     posit.setText(temp.position);
253     numb.setText(""+temp.jerseyNumber);
254     pt.setText(""+temp.points);
255     ast.setText(""+temp.assist);
256     rb.setText(""+temp.rebounds);
257     gms.setText(""+temp.totalGames);
258 }
259 private void CancelActionPerformed(java.awt.event.ActionEvent evt) {//
GEN-FIRST:event_CancelActionPerformed
260     // TODO add your handling code here:
261     this.setVisible(false);
262     ViewPlayers pg = new ViewPlayers(ppl);
263     pg.setVisible(true);
264 }//GEN-LAST:event_CancelActionPerformed
265
266 private void UpdateInfoActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_UpdateInfoActionPerformed
267     // TODO add your handling code here:
268     String name1 = newName1.getText();
269     String name2 = newName2.getText();
270     String position = (String) posCombo.getSelectedItem();
271     int number = 0;
272     int points = 0;
273     int assists = 0;
274     int rebounds = 0;
275     int gmsPlayed = 0;
276     if(!nu.getText().equals("")){
277         if(!pts.getText().equals("")){
278             if(!newAst.getText().equals("")){
279                 if(!newReb.getText().equals("")){
280                     if(!newGms.getText().equals("")){
281                         number = Integer.parseInt(nu.getText());
282                         points = Integer.parseInt(pts.getText());
283                         assists = Integer.parseInt(newAst.getText());
284                         rebounds = Integer.parseInt(newReb.getText());
285                         gmsPlayed =
Integer.parseInt(newGms.getText());
```

# updateInfo.java

```

286         }
287     }
288 }
289 }
290 }
291     boolean updatePlayer = true;
292     if(name1.equals("") || name2.equals("") || position.equals("N/A") ||
    nu.getText().equals("") || pts.getText().equals("") ||
    newAst.getText().equals("") || newReb.getText().equals("") ||
    newGms.getText().equals("")){
293         JOptionPane.showMessageDialog(null, "Please input all
    fields");
294         updatePlayer = false;
295     }
296     if(updatePlayer == true){
297         for(int i = 0; i < ppl.length; i++){
298             if(ppl[i] != null){
299                 if(ppl[i].fName.equals(temp.fName) &&
    ppl[i].lName.equals(temp.lName)){
300                     ppl[i] = new Player(name1, name2, gmsPlayed,
    number, rebounds, assists, position, points);
301                     JOptionPane.showMessageDialog(null, "The Player's
    Stats have been updated.");
302                     //update the file and go back to screen
303                     try{
304                         FileOutputStream fos= new
    FileOutputStream("Players");
305                         ObjectOutputStream oos= new
    ObjectOutputStream(fos);
306                         oos.writeObject(ppl); //this is a way to store
    the arraylist in a file
307                         oos.close();
308                         fos.close();
309                     }catch(IOException ioe){
310                         ioe.printStackTrace();
311                     }
312
313                     this.setVisible(false);
314                     ViewPlayers plt = new ViewPlayers(ppl);
315                     plt.setVisible(true);
316                 }
317             }

```

## updateInfo.java

```
318         }
319     }
320     }//GEN-LAST:event_UpdateInfoActionPerformed
321
322     private void newName1KeyTyped(java.awt.event.KeyEvent evt) {//GEN-FIRST:event_newName1KeyTyped
323         // TODO add your handling code here:
324         char c=evt.getKeyChar();
325         if(!(Character.isAlphabetic(c) || (c==KeyEvent.VK_BACK_SPACE)||
326             c==KeyEvent.VK_DELETE ))
327             evt.consume();
328     }//GEN-LAST:event_newName1KeyTyped
329
330     private void newName2KeyTyped(java.awt.event.KeyEvent evt) {//GEN-FIRST:event_newName2KeyTyped
331         // TODO add your handling code here:
332         char c=evt.getKeyChar();
333         if(!(Character.isAlphabetic(c) || (c==KeyEvent.VK_BACK_SPACE)||
334             c==KeyEvent.VK_DELETE ))
335             evt.consume();
336     }//GEN-LAST:event_newName2KeyTyped
337
338     private void nuKeyTyped(java.awt.event.KeyEvent evt) {//GEN-FIRST:event_nuKeyTyped
339         // TODO add your handling code here:
340         char c = evt.getKeyChar();
341         if(!(Character.isDigit(c) || c == KeyEvent.VK_BACK_SPACE) || c ==
342             KeyEvent.VK_DELETE){
343             evt.consume();
344         }
345     }//GEN-LAST:event_nuKeyTyped
346
347     private void ptsKeyTyped(java.awt.event.KeyEvent evt) {//GEN-FIRST:event_ptsKeyTyped
348         // TODO add your handling code here:
349         char c = evt.getKeyChar();
350         if(!(Character.isDigit(c) || c == KeyEvent.VK_BACK_SPACE) || c ==
351             KeyEvent.VK_DELETE){
352             evt.consume();
353         }
354     }//GEN-LAST:event_ptsKeyTyped
355
```

## updateInfo.java

```
352     private void newAstKeyTyped(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_newAstKeyTyped
353         // TODO add your handling code here:
354         char c = evt.getKeyChar();
355         if(!(Character.isDigit(c) || c == KeyEvent.VK_BACK_SPACE) || c ==
KeyEvent.VK_DELETE){
356             evt.consume();
357         }
358     }//GEN-LAST:event_newAstKeyTyped
359
360     private void newRebKeyTyped(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_newRebKeyTyped
361         // TODO add your handling code here:
362         char c = evt.getKeyChar();
363         if(!(Character.isDigit(c) || c == KeyEvent.VK_BACK_SPACE) || c ==
KeyEvent.VK_DELETE){
364             evt.consume();
365         }
366     }//GEN-LAST:event_newRebKeyTyped
367
368     private void newGmsKeyTyped(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_newGmsKeyTyped
369         // TODO add your handling code here:
370         char c = evt.getKeyChar();
371         if(!(Character.isDigit(c) || c == KeyEvent.VK_BACK_SPACE) || c ==
KeyEvent.VK_DELETE){
372             evt.consume();
373         }
374     }//GEN-LAST:event_newGmsKeyTyped
375
376     private void name1ActionPerformed(java.awt.event.ActionEvent evt) {//
GEN-FIRST:event_name1ActionPerformed
377         // TODO add your handling code here:
378     }//GEN-LAST:event_name1ActionPerformed
379
380     /**
381     * @param args the command line arguments
382     */
383     public static void main(String args[]) {
384         /* Set the Nimbus look and feel */
385         //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
```

## updateInfo.java

```
386      /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
387      * For details see http://download.oracle.com/javase/tutorial/
uiswing/lookandfeel/plaf.html
388      */
389      try {
390          for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
391              if ("Nimbus".equals(info.getName())) {
392
javax.swing.UIManager.setLookAndFeel(info.getClassName());
393                  break;
394              }
395          }
396      } catch (ClassNotFoundException ex) {
397
java.util.logging.Logger.getLogger(updateInfo.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
398      } catch (InstantiationException ex) {
399
java.util.logging.Logger.getLogger(updateInfo.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
400      } catch (IllegalAccessException ex) {
401
java.util.logging.Logger.getLogger(updateInfo.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
402      } catch (javax.swing.UnsupportedLookAndFeelException ex) {
403
java.util.logging.Logger.getLogger(updateInfo.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
404      }
405      //</editor-fold>
406
407      /* Create and display the form */
408      java.awt.EventQueue.invokeLater(new Runnable() {
409          public void run() {
410              new updateInfo().setVisible(true);
411          }
412      });
413  }
414
415  // Variables declaration - do not modify//GEN-BEGIN:variables
```

## updateInfo.java

```
416     private javax.swing.JLabel Assists;
417     private javax.swing.JLabel BBallTeam;
418     private javax.swing.JButton Cancel;
419     private javax.swing.JLabel FName;
420     private javax.swing.JLabel GRU;
421     private javax.swing.JLabel GmsPlayed;
422     private javax.swing.JLabel LName;
423     private javax.swing.JLabel Number;
424     private javax.swing.JLabel Points;
425     private javax.swing.JLabel Position;
426     private javax.swing.JLabel Rebounds;
427     private javax.swing.JButton UpdateInfo;
428     private javax.swing.JTextField ast;
429     private javax.swing.JTextField gms;
430     private javax.swing.JLabel jLabel1;
431     private javax.swing.JLabel jLabel12;
432     private javax.swing.JLabel jLabel13;
433     private javax.swing.JTextField name1;
434     private javax.swing.JTextField name2;
435     private javax.swing.JTextField newAst;
436     private javax.swing.JTextField newGms;
437     private javax.swing.JTextField newName1;
438     private javax.swing.JTextField newName2;
439     private javax.swing.JTextField newReb;
440     private javax.swing.JTextField nu;
441     private javax.swing.JTextField numb;
442     private javax.swing.JComboBox<String> posCombo;
443     private javax.swing.JTextField posit;
444     private javax.swing.JTextField pt;
445     private javax.swing.JTextField pts;
446     private javax.swing.JTextField rb;
447     // End of variables declaration//GEN-END:variables
448 }
449
```



## ViewPlayers.java

```
2  * To change this license header, choose License Headers in Project
   Properties.
11 import java.util.*;
15 public class ViewPlayers extends javax.swing.JFrame {
16
17     /**
18      * Creates new form ViewPlayers
19      */
20     Player[] playerList = new Player[30];
21     Player [] seeing; //the list that is printed on the screen
22
23     ///public ViewPlayers(){
24         // initComponents();
25     ///}
26
27     public ViewPlayers(Player[] fat) {
28         playerList = fat;
29         seeing = fat;
30         initComponents();
31         seeList();
32     }
33
34     /**
35      * This method is called from within the constructor to initialize the
   form.
36      * WARNING: Do NOT modify this code. The content of this method is
   always
37      * regenerated by the Form Editor.
38      */
39     @SuppressWarnings("unchecked")
40     // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
   BEGIN: initComponents
41     private void initComponents() {
42
43         jScrollPane1 = new javax.swing.JScrollPane();
44         jList1 = new javax.swing.JList<>();
45         jScrollPane2 = new javax.swing.JScrollPane();
46         show = new javax.swing.JList<>();
47         GRU = new javax.swing.JLabel();
48         BBallTeam = new javax.swing.JLabel();
49         Sort = new javax.swing.JButton();
50         sortCombo = new javax.swing.JComboBox<>();
```

## ViewPlayers.java

```
51     jScrollPane3 = new javax.swing.JScrollPane();
52     display = new javax.swing.JTextArea();
53     Cancel = new javax.swing.JButton();
54     EditPlayer = new javax.swing.JButton();
55     ShowStats = new javax.swing.JButton();
56     removeButton = new javax.swing.JButton();
57     jLabel1 = new javax.swing.JLabel();
58
59     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
60     setMinimumSize(new java.awt.Dimension(700, 400));
61     getContentPane().setLayout(new
org.netbeans.lib.awtextra.AbsoluteLayout());
62
63     jList1.setModel(new javax.swing.AbstractListModel<String>() {
64         String[] strings = { "Item 1", "Item 2", "Item 3", "Item 4",
"Item 5" };
65         public int getSize() { return strings.length; }
66         public String getElementAt(int i) { return strings[i]; }
67     });
68     jScrollPane1.setViewportView(jList1);
69
70     getContentPane().add(jScrollPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(714, 93, -1, -1));
71
72     show.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
NOI18N
73     jScrollPane2.setViewportView(show);
74
75     getContentPane().add(jScrollPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 130, 370, 210));
76
77     GRU.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N
78     GRU.setForeground(new java.awt.Color(255, 255, 255));
79     GRU.setText("Grand Ridge University ");
80     getContentPane().add(GRU, new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 20, -1, -1));
81
82     BBallTeam.setFont(new java.awt.Font("Times New Roman", 1, 36)); //
NOI18N
83     BBallTeam.setForeground(new java.awt.Color(255, 255, 255));
84     BBallTeam.setText("Basketball Team");
```

## ViewPlayers.java

```

85         getContentPane().add(BBallTeam, new
org.netbeans.lib.awtextra.AbsoluteConstraints(380, 70, -1, -1));
86
87         Sort.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
88         Sort.setText("Sort");
89         Sort.addActionListener(new java.awt.event.ActionListener() {
90             public void actionPerformed(java.awt.event.ActionEvent evt) {
91                 SortActionPerformed(evt);
92             }
93         });
94         getContentPane().add(Sort, new
org.netbeans.lib.awtextra.AbsoluteConstraints(280, 90, 80, 30));
95
96         sortCombo.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
97         sortCombo.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "View All", "Jersey Number", "Last Name", "Guards",
"Forwards" }));
98         getContentPane().add(sortCombo, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 90, 190, 30));
99
100        display.setEditable(false);
101        display.setColumns(20);
102        display.setFont(new java.awt.Font("Times New Roman", 0, 18)); //
NOI18N
103        display.setRows(5);
104        jScrollPane3.setViewportView(display);
105
106        getContentPane().add(jScrollPane3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(420, 130, 250, 210));
107
108        Cancel.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
109        Cancel.setText("Cancel");
110        Cancel.addActionListener(new java.awt.event.ActionListener() {
111            public void actionPerformed(java.awt.event.ActionEvent evt) {
112                CancelActionPerformed(evt);
113            }
114        });
115        getContentPane().add(Cancel, new
org.netbeans.lib.awtextra.AbsoluteConstraints(560, 350, 100, 30));

```

## ViewPlayers.java

```
116
117     EditPlayer.setFont(new java.awt.Font("Times New Roman", 1,
118 14)); // NOI18N
119     EditPlayer.setText("Edit Player");
120     EditPlayer.addActionListener(new java.awt.event.ActionListener() {
121         public void actionPerformed(java.awt.event.ActionEvent evt) {
122             EditPlayerActionPerformed(evt);
123         }
124     });
125     getContentPane().add(EditPlayer, new
126 org.netbeans.lib.awtextra.AbsoluteConstraints(440, 350, -1, 30));
127
128     ShowStats.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
129 NOI18N
130     ShowStats.setText("Show Stats");
131     ShowStats.addActionListener(new java.awt.event.ActionListener() {
132         public void actionPerformed(java.awt.event.ActionEvent evt) {
133             ShowStatsActionPerformed(evt);
134         }
135     });
136     getContentPane().add(ShowStats, new
137 org.netbeans.lib.awtextra.AbsoluteConstraints(210, 350, 140, 30));
138
139     removeButton.setFont(new java.awt.Font("Times New Roman", 1,
140 14)); // NOI18N
141     removeButton.setText("Remove");
142     removeButton.addActionListener(new java.awt.event.ActionListener()
143 {
144     public void actionPerformed(java.awt.event.ActionEvent evt) {
145         removeButtonActionPerformed(evt);
146     }
147 });
148     getContentPane().add(removeButton, new
149 org.netbeans.lib.awtextra.AbsoluteConstraints(60, 350, 140, 30));
150
151     jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
152 NOI18N
153     jLabel1.setIcon(new
154 javax.swing.ImageIcon(getClass().getResource("/basketball/1.jpg"))); //
155 NOI18N
156     getContentPane().add(jLabel1, new
157 org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 700, 400));
```

## ViewPlayers.java

```
147
148     setSize(new java.awt.Dimension(716, 439));
149     setLocationRelativeTo(null);
150 }// </editor-fold>//GEN-END:initComponents
151
152 private void seeList(){
153     DefaultListModel a = new DefaultListModel();
154
155     for(int i = 0; i< seeing.length; i++){
156         if(seeing[i] != null){
157             a.addElement(seeing[i].fName + " " + seeing[i].lName);
158         }
159     }
160     show.setModel(a);
161 }
162 private void CancelActionPerformed(java.awt.event.ActionEvent evt) {//
GEN-FIRST:event_CancelActionPerformed
163     StartFrame stf = new StartFrame();
164     this.setVisible(false);
165     stf.setVisible(true);
166 }//GEN-LAST:event_CancelActionPerformed
167
168 private void EditPlayerActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_EditPlayerActionPerformed
169     // TODO add your handling code here:
170     Player pl = null;
171     String full = "";
172     if(show.getSelectedValue() != null){
173         full = show.getSelectedValue();
174         Scanner kboard = new Scanner(full);
175         String f = kboard.next();
176         String l = kboard.next();
177         for(int i = 0; i< playerList.length;i++ ){
178             if(playerList[i]!= null){
179                 if(playerList[i].fName.equals(f) &&
playerList[i].lName.equals(l)){
180                     pl = playerList[i];
181                 }
182             }
183         }
184     }
185     if(show.getSelectedValue() != null && pl != null){
```

## ViewPlayers.java

```

186         this.setVisible(false);
187         updateInfo ui = new updateInfo(playerList, pl);
188         ui.setVisible(true);
189     }
190 }//GEN-LAST:event_EditPlayerActionPerformed
191
192 private void ShowStatsActionPerformed(java.awt.event.ActionEvent evt)
193 { //GEN-FIRST:event_ShowStatsActionPerformed
194     // TODO add your handling code here:
195
196     String se = "";
197     String f = "";
198     String l = "";
199     if(show.getSelectedValue() != null){
200         se = show.getSelectedValue();
201
202         Scanner kboard = new Scanner(se);
203         f = kboard.next();
204         l = kboard.next();
205     }
206     DecimalFormat df = new DecimalFormat("#.##"); //for decimal format
207     when rounding
208     if(!se.equals("")){
209         for(int i = 0; i < seeing.length; i++){
210             if(seeing[i] != null){
211                 if(seeing[i].fName.equals(f) &&
212                     seeing[i].lName.equals(l)){
213                     display.setText("Name: "+seeing[i].fName + " " +
214                         seeing[i].lName+"\nPosition: " + seeing[i].position+"\nJersey Number: " +
215                         seeing[i].jerseyNumber+"\nTotal Points: " + seeing[i].points+"    PPG:
216                         "+df.format((double)seeing[i].points/seeing[i].totalGames)+"\nAssists: " +
217                         seeing[i].assist+"    APG: "+df.format((double)seeing[i].assist/
218                         seeing[i].totalGames)+"\nRebounds: "+seeing[i].rebounds+"    RPG:
219                         "+df.format((double)seeing[i].rebounds/seeing[i].totalGames)+"\nGames
220                         Played: "+ seeing[i].totalGames);
221                 }
222             }
223         }
224     }
225 }//GEN-LAST:event_ShowStatsActionPerformed
226
227 private void removeButtonActionPerformed(java.awt.event.ActionEvent

```

## ViewPlayers.java

```
    evt) { //GEN-FIRST:event_removeButtonActionPerformed
218        // TODO add your handling code here:
219        String f = "";
220        String l = "";
221        if(show.getSelectedValue() != null){
222            Scanner kboard = new Scanner(show.getSelectedValue());
223            f = kboard.next();
224            l = kboard.next();
225
226            for(int i = 0; i < playerList.length; i++){
227                if(playerList[i] != null){
228                    if(playerList[i].fName.equals(f) &&
playerList[i].lName.equals(l)){
229                        playerList[i] = null;
230                        seeing = playerList;
231                        seeList();
232                        //update the file
233                        try{
234                            FileOutputStream fos = new
FileOutputStream("Players");
235                            ObjectOutputStream oos = new
ObjectOutputStream(fos);
236                            oos.writeObject(playerList); //this is a way
to store the arraylist in a file
237                            oos.close();
238                            fos.close();
239                        }catch(IOException ioe){
240                            ioe.printStackTrace();
241                        }
242                    }
243                }
244            }
245        }
246    } //GEN-LAST:event_removeButtonActionPerformed
247
248    private void SortActionPerformed(java.awt.event.ActionEvent evt) { //
GEN-FIRST:event_SortActionPerformed
249        // TODO add your handling code here:
250        String select = (String) sortCombo.getSelectedItem();
251        if(select != null){
252            if(select.equalsIgnoreCase("Jersey Number")){
253                seeing = readFile();
```

## ViewPlayers.java

```

254         for(int i = 0; i < seeing.length-1; i++){
255             int index = i; //the first index is the smallest
                number
256             for(int j = i+1; j< seeing.length; j++){
257                 try{//fof all the null pointers, there's
                    exceptions
258                     if(seeing[j].jerseyNumber <
                        seeing[index].jerseyNumber)
259                         index = j; //the neew index is where the
                            lowest number is
260                     }catch(NullPointerException n){}
261                 }//loop through the entire list to find the lowest
                    number then make the swap
262                 Player high =seeing[i];
263                 seeing [i] = seeing[index];
264                 seeing[index] = high;
265             }
266             seeList();
267         }else if(select.equalsIgnoreCase("View All")){
268             seeing = readFile();
269             seeList();
270         }else if(select.equalsIgnoreCase("Last Name")){
271             seeing = readFile();
272             for(int i = 0; i < seeing.length-1; i++){
273                 int index = i; //the first index is the smallest
                    number
274                 for(int j = i+1; j< seeing.length; j++){
275                     try{//fof all the null pointers, there's
                        exceptions
276                         if(seeing[j].lName.toLowerCase().compareTo(seeing[index].lName.toLowerCase()
                            ()) < 0)
277                             index = j; //the neew index is where the
                                lowest number is
278                         }catch(NullPointerException n){}
279                     }//loop through the entire list to find the lowest
                        number then make the swap
281                     Player high =seeing[i];
282                     seeing [i] = seeing[index];
283                     seeing[index] = high;
284                 }

```



## ViewPlayers.java

```

285         seeList();
286     }else if(select.equalsIgnoreCase("Guards")){
287         seeing = readFile();
288         for(int i = 0; i< seeing.length; i++){
289             if(seeing[i]!= null){
290                 if(!seeing[i].position.equalsIgnoreCase("Guard")){
291                     seeing[i] = null;
292                 }
293             }
294         }
295         seeList();
296     }else if(select.equalsIgnoreCase("Forwards")){
297         seeing = readFile();
298         for(int i = 0; i< seeing.length; i++){
299             if(seeing[i]!= null){
300                 if(!
seeing[i].position.equalsIgnoreCase("Forward")){
301                     seeing[i] = null;
302                 }
303             }
304         }
305         seeList();
306     }
307 }
308 }//GEN-LAST:event_SortActionPerformed
309 private Player [] readFile(){
310     Player [] temp = null;
311     try
312     {
313         FileInputStream fis = new FileInputStream("Players");
314         ObjectInputStream ois = new ObjectInputStream(fis);
315         temp = (Player[]) ois.readObject();
316         ois.close();
317         fis.close();
318     }catch(IOException ioe){
319         ioe.printStackTrace();
320         //return;
321     }catch(ClassNotFoundException c){
322         System.out.println("Class not found");
323         c.printStackTrace();
324         //return;
325     }

```

## ViewPlayers.java

```
326         return temp;
327     }
328     private Player[] clear(Player[] temp){
329
330         for(int i = 0; i< temp.length; i++){
331             temp[i] = null;
332         }
333         return temp;
334     }
335
336     /**
337      * @param args the command line arguments
338      */
339     public static void main(String args[]) {
340         /* Set the Nimbus look and feel */
341         //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
342         /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
343          * For details see http://download.oracle.com/javase/tutorial/
uiswing/lookandfeel/plaf.html
344          */
345         try {
346             for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
347                 if ("Nimbus".equals(info.getName())) {
348
349                     javax.swing.UIManager.setLookAndFeel(info.getClassName());
350                     break;
351                 }
352             } catch (ClassNotFoundException ex) {
353
354                 java.util.logging.Logger.getLogger(ViewPlayers.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
355             } catch (InstantiationException ex) {
356
357                 java.util.logging.Logger.getLogger(ViewPlayers.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
358             } catch (IllegalAccessException ex) {
359
360                 java.util.logging.Logger.getLogger(ViewPlayers.class.getName()).log(java.u
```

## ViewPlayers.java

```
        til.logging.Level.SEVERE, null, ex);
358    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
359
360        java.util.logging.Logger.getLogger(ViewPlayers.class.getName()).log(java.u
        til.logging.Level.SEVERE, null, ex);
360    }
361    //</editor-fold>
362
363    /* Create and display the form */
364    java.awt.EventQueue.invokeLater(new Runnable() {
365        public void run() {
366            //new ViewPlayers().setVisible(true);
367        }
368    });
369 }
370
371 // Variables declaration - do not modify//GEN-BEGIN:variables
372 private javax.swing.JLabel BBallTeam;
373 private javax.swing.JButton Cancel;
374 private javax.swing.JButton EditPlayer;
375 private javax.swing.JLabel GRU;
376 private javax.swing.JButton ShowStats;
377 private javax.swing.JButton Sort;
378 private javax.swing.JTextArea display;
379 private javax.swing.JLabel jLabel1;
380 private javax.swing.JList<String> jList1;
381 private javax.swing.JScrollPane jScrollPane1;
382 private javax.swing.JScrollPane jScrollPane2;
383 private javax.swing.JScrollPane jScrollPane3;
384 private javax.swing.JButton removeButton;
385 private javax.swing.JList<String> show;
386 private javax.swing.JComboBox<String> sortCombo;
387 // End of variables declaration//GEN-END:variables
388 }
389
```