basic education
Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

JUNE EXAMINATION 2015

INFORMATION TECHNOLOGY P1

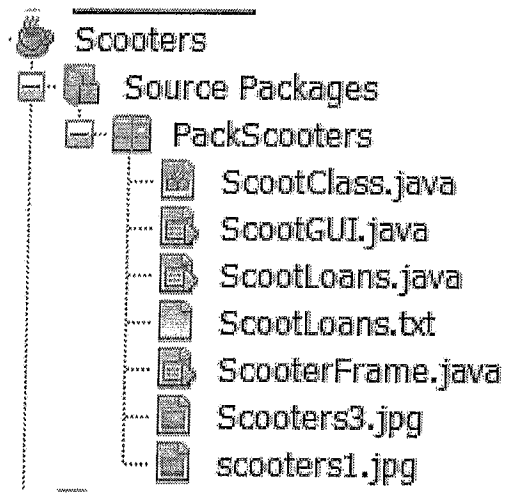
GRADE 12

EXAMINER: M PADAYACHEE **MODERATOR: S NAIDOO**

DURATION: 3HRS **Marks: 150** **DATE: 23 – 06 – 2015**

INSTRUCTIONS TO LEARNERS:

1. Ensure that this paper contains 9 pages and 3 questions.
2. Answer all questions.
3. Save your work at every 10 minute interval.
4. Ensure that you have the following files in the Exam folder.

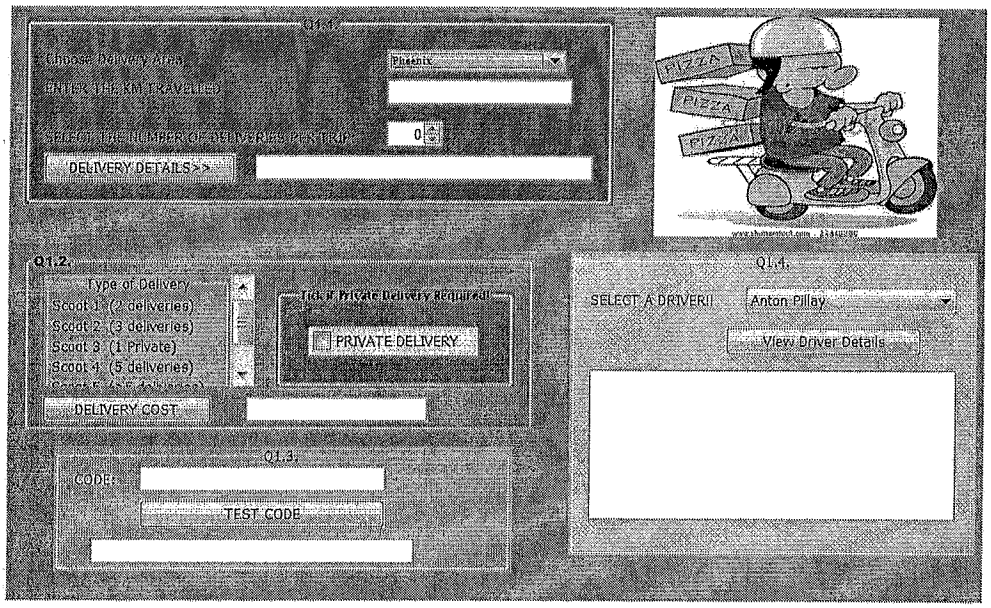


SCENARIO:

Sceeters is a nationwide Pizza franchise that is rated number one in pizza deliveries. You are requested to assist in developing software to help the local branch at the Phoenix plaza to manage their transport and delivery system.

QUESTION ONE (*General programming*)

Open the project called **Scooters**, then open the **ScooterFrame** found in **PackScooters**. Compile and execute. The frame has no code for the given GUI components. You are required to examine the example of the GUI given below and write the code for the 5 sections(1.1. to 1.5.):



1.1. Code the **DELIVERY DETAILS** button as follows:

- Place a line of text in the textfield alongside the button as shown in the example input and output below:

Input

Output

[5]

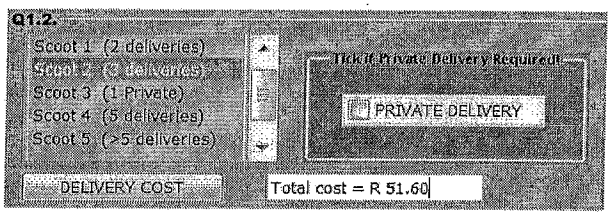
1.2. You are required to calculate the Total Charge of a delivery. A driver is given a scooter according to the category of deliveries. Since there are 5 scooters, you are required to write a code to randomly generate a type from 1 to 5 of the list provided. Type 3 is private delivery. The total charge is what the driver makes after his delivery per trip. Use the table below to determine the delivery charge according to the list type generated. The driver will receive an additional R8 per home delivered to. Ignore the number of deliveries in question 1.1.

By using the type selected from the list and the tariff table, display the Total charge.:

TYPE	Number of Deliveries	Charge per KM
Scoot 1	2 deliveries	R0,95/KM
Scoot 2	3 deliveries	R1.15/KM
Scoot 3	1Private	R9 / KM
Scoot 4	4 deliveries	R4.50/KM
Scoot 5	>4 Deliveries	R7/KM

(Total charge = Charge per KM *number of KM + R8*number of homes)

Sample Input and Output: number of km from text field is 24km.



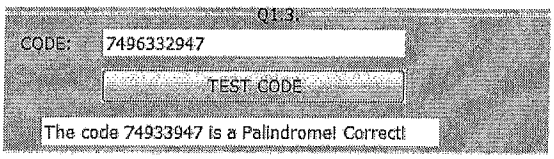
[10]

1.3. Every driver is given a generated code which is validated every time they enter it. The driver is allowed to enter code in order to view his deliveries and his amount earned. After the driver enters his code, program the TEST CODE button to validate the code entry.

Below is a flow diagram that demonstrates the validation rule:

- ✓ Enter a 10 digit code.
- ✓ Remove the 4th and the 8th digits from the code.
- ✓ The rest of the remaining digits of the code must form a Palindrome number.

Sample input and Output:



[15]

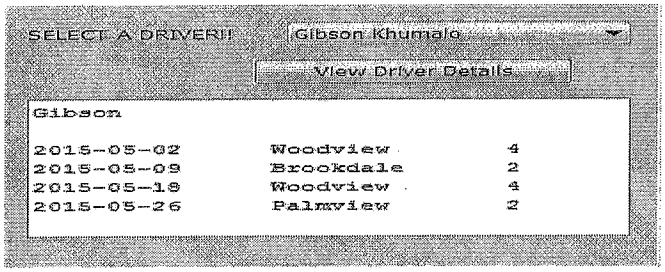
1.4. An array of deliveries is provided, for May 2014, in the variable section of the code. Your task is to transfer all the array details into a textfile. The name of the textfile should be "May2014" followed by the name of the driver in the textfield selected. Example : "May2014Gibson".

You are required to use the driver selected from the textfield and display all the details from the text file created as follows:

The format of the text file is: (Date#Name of driver#Area of delivery#number of homes delivered to)

```
2015-05-02#Gibson#Woodview#4
2015-05-05#Florence#Southgate#3
2015-05-09#Gibson#Brookdale#2
2015-05-11#Devan#Broardlands#3
2015-05-12#Anton#Phoenix#1
2015-05-12#Stacey#Palmview#15
2015-05-18#Gibson#Woodview#4
2015-05-22#Florence#Broadlands#6
2015-05-25#Devan#Southgate#3
2015-05-26#Gibson#Palmview#2
```

Name of driver, followed by all the different deliveries made by the driver selected.
Sample output:



[15]

TOTAL QUESTION ONE = 45

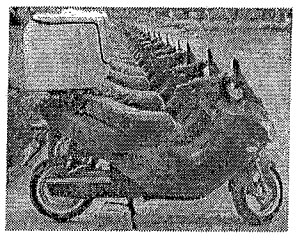
QUESTION TWO

The bikes at Scooters are classified into three different classes according to the ditances they are equipped to cover:

Classification	Distance
Class1	<10km
Class2	10km<=dist<15km
Class3	15km<=dist<20
Class4	>=20km

You have been supplied with a java class called **ScootClass** and a framed class called **ScootGUI** in the PackScooters package. Open the ScootGUI and ScootClass, insert your name and grade/div in the first line of each code then compile and execute them.

2.1. The incomplete object class (ScootClass) contains attributes of scooters used for deliveries.

Identifier Names	Description
	
deliveryNumber	Number given to a specific delivery(1,2,3...)
scooterNumber	Number eg.(S1,S2,S3...)
fuelUsed	Amount of fuel used during delivery
meterReading1	Odometer reading at start
meterReading2	Odeometer reading at end of delivery

2.1.1. Complete the code in the incomplete class for the declaration of the attributes that are missing. (3)

2.1.2. Write a code for a constructor to accept the delivery number, scooterNumber, start odometer and end odometer readings as parameters. Pass these values to the relevant global attributes of the class. (4)

2.1.3. Write the relevant mutator and accessor methods for the fuelUsed identifier. (4)

2.1.4. Write a method called **distanceTravelled** to calculate and return the total distance covered by a delivery based on the two odometer readings. (5)

2.2. A text file called **ScooterDetails.txt** is supplied with an unknown number of lines of information on past recorded deliveries. The information found on each line of the text file has the following format:

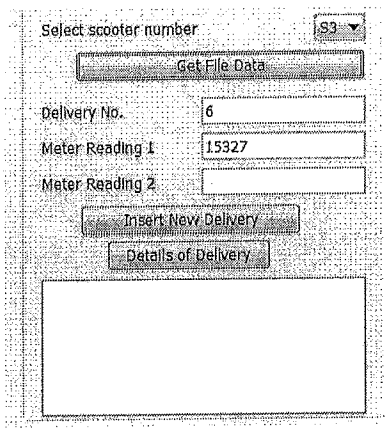
<deliveryNumber>#<scooterNumber>#<meterReading2>

Sample data in textfile

```
1#S1#11110
2#S2#8020
3#S3#15219
4#S4#7300
5#S1#11190
6#S315327
...
...
```

2.2.1. Complete code for the button **Get File Data** as follows:

- ✓ Select the scooter number from the combo box
- ✓ Use the given text file **ScooterDetails** to determine the following:
- ➢ The last delivery number of the selected scooter number in the text file.
- ➢ The start odometer reading(Meter reading1) is the last reading in the text file for the selected scooter. (Eg. If S4 is selected, 7300 must be used as meter reading1 for the new delivery).
- ✓ Display the delivery number and start odometer reading in the boxes provided.
  Example of content of relevant values displayed for S3:



(15)

2.2.2. To program the **Insert New Delivery** button, Instantiate the new **ScootClass** object by inserting the new selection of the Scooter from the combo box.
Meter reading 1 will be the last meter reading for that scooter from the text file.
Enter the odometer reading after the delivery (Meter Reading2). This reading must be greater than the previous odometer reading(Reading 1) for that delivery.

Obtain the distance travelled by calling the **distanceTravelled** method and calculate the estimated litres of fuel used for the delivery. One litre of fuel is used to every five kilometres travelled. Use the fuelUsed mutator to set the attribute to the calculated value. (12)

2.2.3. Program the Details of Delivery button to display the object in the text area provided in the GUI. The object is identified by the scooter number as follows.

Sample Output:

```
Delivery number: 4
Scooter number: S4
Odometer reading:
    (Before):7300
    (After): 7392
Fuel Used: 18.4Litres
```

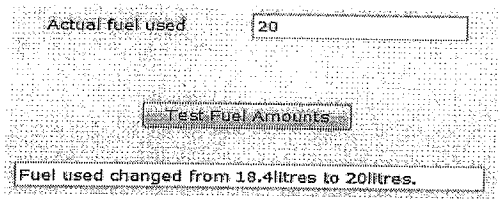
(5)

2.2.4. Each driver has to refill the tank at the end of a delivery. The difference in the routes means that the refill amount may differ from the fuel used calculated.
Write code to do the following:

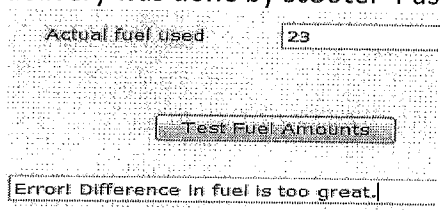
- Enter the amount of fuel used to refill the tank in the text box provided.
- Obtain the calculated amount of fuel used from the relevant attribute in the object class.
- Calculate the difference between the fuel used and the calculated fuel used as stored in the attribute of the object class.
    - If the difference is less than 15%, the fuel used attribute must be updated by setting the value to the new value as entered in the text box.

      Display an appropriate message to indicate whether the fuel used attribute has been updated or not.

      E.g. of the output if the delivery was done by scooter 4 as shown above in sample output.



If the difference is greater than or equal to 15%, display an error message should be displayed and the value of the fuelUsed attribute must not be updated.
E.g. of output if delivery was done by scooter 4 as shown above.



(12)

[60]

QUESTION THREE

Scooters offers loans to their employees. They are in the process of designing an app to handle all loan accounts.
Study the **ScootLoans** app and help them complete the code to assist in managing the loan system.
Below is a copy of the text file that contains the loans taken by employees of the company.

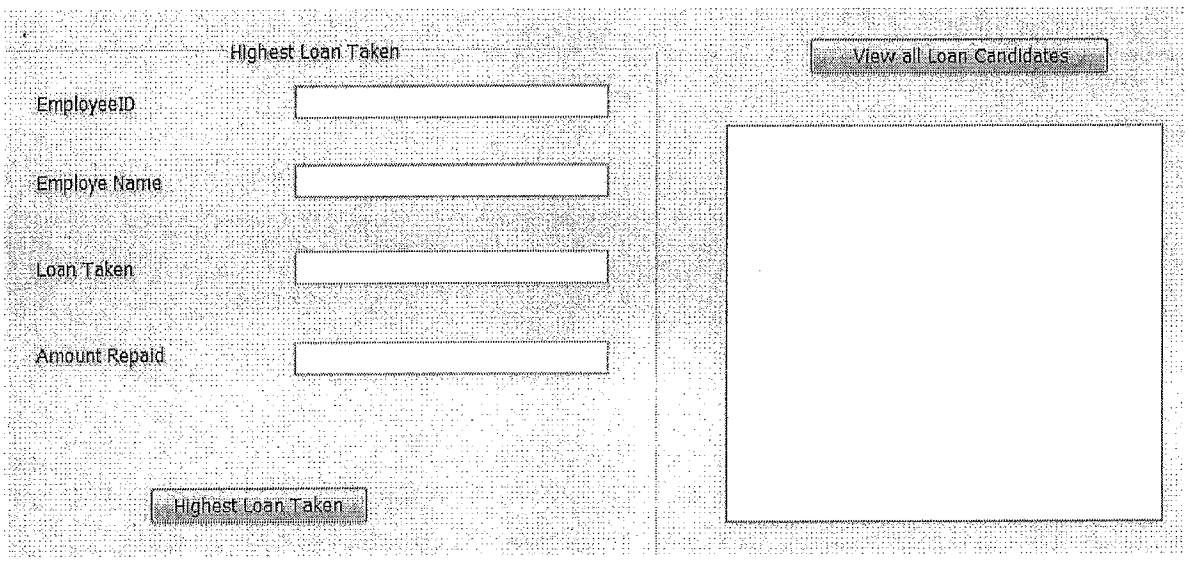
```
1021,Jake_Everret,R5000,R450
1099,Candice_Miller,R3000,R220
1123,Amy_Richter,R9000,R4350
1211,Mary_Malhouse,R500,R0
1001,Stephanie_Kule,R300,R300
1231,Jane_Kench,R4200,R4000
1372,Carol_Jacobs,R3000,R3000
1000,Trudy_Trotter,R250,R100
1036,Alan_Tobias,R600,R600
1222,Richard_Thomas,R5600,R4200
```

3.1. Code the action event **View All Loan** Candidates to access the text file called "**ScootLoans.text**" provided, and populate the following arrays:

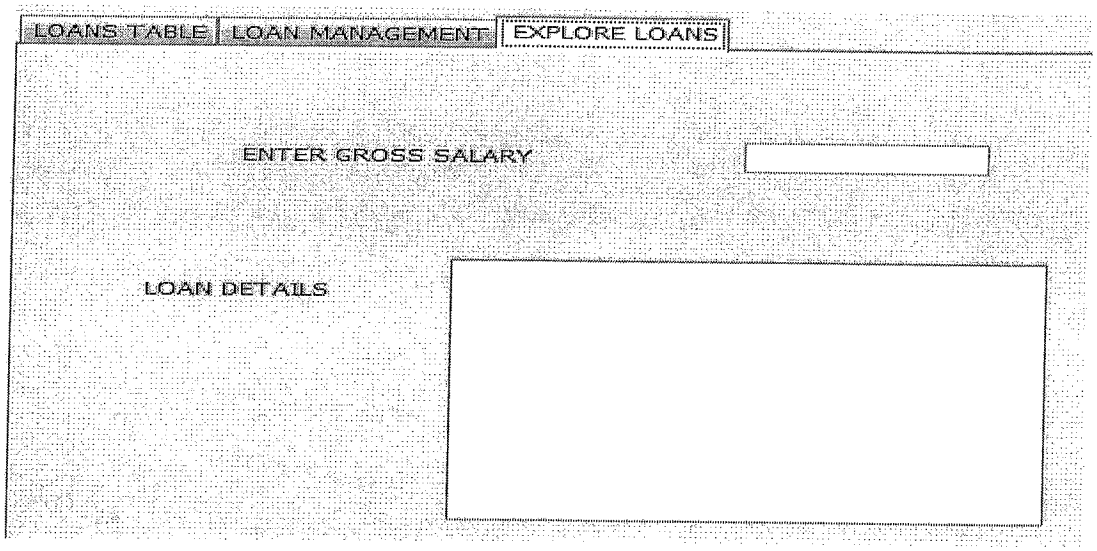
- A one dimensional array that contains the LoanID's.
- A one dimensional array that contains the names of loan candidates.
- A two dimensional array that will store the loan amount taken, the amount repaid and the amount outstanding.

The contents of the arrays above must be placed in the table in the Loans Table tab under the suitable columns. See sample table below: (15)

3.2. Program the **Highest Loan Taken** button to display all details in the **Highest Loan Taken** panel. (7)



3.2. Program the action listeners(buttons) in the LOAN MANAGEMENT tab as follows:



3.2.1. The **VIEW PAID UP LOANS** button should list all names with a zero balance. (6)

3.2.2. The **VIEW OUTSTANDING AMOUNT** must be calculated and displayed with all the clients details. (8)

3.3. The **EXPLORE LOANS** tab allows the prospective client to view the feasibility of loan amounts with respect to their earnings.

3.3.1. If the company does micro lending (small loans), set the slider to the respective text field in accordance with a lending range of (R1000 to R10000). Then allow the client to enter their details in the respective text fields as per request. (3)

3.3.2. Program the **PROCESS REQUEST** button to determine if a person qualifies for a loan. Use the criteria below to determine if the client qualifies.

- The loan amount should not be greater than the 30% of the gross salary.

Display the gross, the qualifying amount and whether or not the applicant qualifies.

(6)

TOTAL : 150

GREENBERRY [illegible] SCHOOL
[illegible] SCIENCES
[illegible] PILLAY

```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5 package PackScooters;
6
7 import java.io.FileNotFoundException;
8 import java.io.FileReader;
9 import java.util.Scanner;
10
11 /**
12  *
13  * @author comp13
14  */
15 public class ScootLoans extends javax.swing.JFrame {
16
17     /**
18      * Creates new form ScootLoans
19      */
20     String s[];
21      String  loanID [] = new String[100];
22       String empName [] = new String [100];
23       String amounts[][] = new String [100][100];
24       int counter = 0;
25     public ScootLoans() {
26         initComponents();
27     }
28
29     /**
30      * This method is called from within the constructor to initialize the form.
31      * WARNING: Do NOT modify this code. The content of this method is always
32      * regenerated by the Form Editor.
33      */
34     @SuppressWarnings("unchecked")
35     // <editor-fold defaultstate="collapsed" desc="Generated Code">
36     private void initComponents() {
37
38         jTabbedPane1 = new javax.swing.JTabbedPane();
39         jPanel1 = new javax.swing.JPanel();
40         jButton1 = new javax.swing.JButton();
41         jPanel3 = new javax.swing.JPanel();
42         jFormattedTextField3 = new javax.swing.JFormattedTextField();
43         jTextField1 = new javax.swing.JTextField();
44         jFormattedTextField1 = new javax.swing.JFormattedTextField();
45         jFormattedTextField2 = new javax.swing.JFormattedTextField();
46         jLabel1 = new javax.swing.JLabel();
47         jLabel2 = new javax.swing.JLabel();
48         jLabel3 = new javax.swing.JLabel();
49         jLabel4 = new javax.swing.JLabel();
50         jButton4 = new javax.swing.JButton();
51         jScrollPane3 = new javax.swing.JScrollPane();
52         jTextArea2 = new javax.swing.JTextArea();
53         jPanel2 = new javax.swing.JPanel();
54         jScrollPane2 = new javax.swing.JScrollPane();
55         jTextArea1 = new javax.swing.JTextArea();
56         jButton2 = new javax.swing.JButton();
57         jButton3 = new javax.swing.JButton();
58         jPanel4 = new javax.swing.JPanel();
59         jLabel5 = new javax.swing.JLabel();
60         jTextField2 = new javax.swing.JTextField();
61         jScrollPane4 = new javax.swing.JScrollPane();
```

```
62      jTextArea3 = new javax.swing.JTextArea();
63      jLabel6 = new javax.swing.JLabel();
64
65      setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
66
67      jButton1.setText("View all Loan Candidates");
68      jButton1.addActionListener(new java.awt.event.ActionListener() {
69          public void actionPerformed(java.awt.event.ActionEvent evt) {
70              jButton1ActionPerformed(evt);
71          }
72      });
73
74      jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Highest Loan Taken",
javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.DEFAULT_POSITION));
75
76      jLabel1.setText("EmployeeID");
77
78      jLabel2.setText("Employe Name");
79
80      jLabel3.setText("Loan Taken");
81
82      jLabel4.setText("Amount Repaid");
83
84      jButton4.setText("Highest Loan Taken");
85
86      javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
87      jPanel3.setLayout(jPanel3Layout);
88      jPanel3Layout.setHorizontalGroup(
89          jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
90          .addGroup(jPanel3Layout.createSequentialGroup()
91              .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
92                  .addGroup(jPanel3Layout.createSequentialGroup()
93                      .addGap(38, 38, 38)
94                      .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
95                          .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)
96                          .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)
97                          .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)
98                          .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))
99                      .addGap(60, 60, 60)
100                     .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
101                         .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 209,
javax.swing.GroupLayout.PREFERRED_SIZE)
102                         .addComponent(jFormattedTextField3, javax.swing.GroupLayout.PREFERRED_SIZE, 209,
javax.swing.GroupLayout.PREFERRED_SIZE)
103                         .addComponent(jFormattedTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 209,
javax.swing.GroupLayout.PREFERRED_SIZE)
104                         .addComponent(jFormattedTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 209,
javax.swing.GroupLayout.PREFERRED_SIZE)))
105                 .addGroup(jPanel3Layout.createSequentialGroup()
106                     .addGap(115, 115, 115)
107                     .addComponent(jButton4)))
108             .addContainerGap(29, Short.MAX_VALUE))
109     );
110     jPanel3Layout.setVerticalGroup(
111         jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
112         .addGroup(jPanel3Layout.createSequentialGroup()
113             .addContainerGap()
```

```
114             .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
115                 .addComponent(jFormattedTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
116                 .addComponent(jLabel1))
117             .addGap(30, 30, 30)
118             .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
119                 .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
120                 .addComponent(jLabel2))
121             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
122             .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
123                 .addComponent(jFormattedTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
124                 .addComponent(jLabel3))
125             .addGap(38, 38, 38)
126             .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
127                 .addComponent(jFormattedTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
128                 .addComponent(jLabel4))
129             .addGap(73, 73, 73)
130             .addComponent(jButton4)
131             .addGap(58, 58, 58))
132     );
133
134     jTextArea2.setColumns(20);
135     jTextArea2.setRows(5);
136     jScrollPane3.setViewportView(jTextArea2);
137
138     javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
139     jPanel1.setLayout(jPanel1Layout);
140     jPanel1Layout.setHorizontalGroup(
141         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
142         .addGroup(jPanel1Layout.createSequentialGroup()
143             .addGap(22, 22, 22)
144             .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
145             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
146             .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
147                 .addComponent(jScrollPane3, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 292, javax.swing.GroupLayout.PREFERRED_SIZE)
148                 .addComponent(jButton1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 201, javax.swing.GroupLayout.PREFERRED_SIZE))
149             .addGap(29, 29, 29))
150     );
151     jPanel1Layout.setVerticalGroup(
152         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
153         .addGroup(jPanel1Layout.createSequentialGroup()
154             .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
155                 .addGroup(jPanel1Layout.createSequentialGroup()
156                     .addGap(33, 33, 33)
157                     .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
158                 .addGroup(jPanel1Layout.createSequentialGroup()
159                     .addGap(22, 22, 22)
160                     .addComponent(jButton1)
161                     .addGap(44, 44, 44)
162                     .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE, 262, Short.MAX_VALUE)
163                     .addGap(65, 65, 65)))
164             .addContainerGap())
```

```
165     );
166
167     jTabbedPane1.addTab("LOANS TABLE", jPanel1);
168
169     jTextArea1.setColumns(20);
170     jTextArea1.setRows(5);
171     jScrollPane2.setViewportView(jTextArea1);
172
173     jButton2.setText("VIEW PAID UP LOANS");
174
175     jButton3.setText("VIEW OUTSTANDING AMOUNTS");
176
177     javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
178     jPanel2.setLayout(jPanel2Layout);
179     jPanel2Layout.setHorizontalGroup(
180         jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
181         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()
182             .addGap(68, 68, 68)
183             .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
184                 .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 228,
javax.swing.GroupLayout.PREFERRED_SIZE)
185                 .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 234,
javax.swing.GroupLayout.PREFERRED_SIZE))
186             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
187             .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 420,
javax.swing.GroupLayout.PREFERRED_SIZE)
188             .addGap(43, 43, 43))
189     );
190     jPanel2Layout.setVerticalGroup(
191         jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
192         .addGroup(jPanel2Layout.createSequentialGroup()
193             .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
194                 .addGroup(jPanel2Layout.createSequentialGroup()
195                     .addGap(41, 41, 41)
196                     .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 369,
javax.swing.GroupLayout.PREFERRED_SIZE))
197                 .addGroup(jPanel2Layout.createSequentialGroup()
198                     .addGap(64, 64, 64)
199                     .addComponent(jButton2)
200                     .addGap(45, 45, 45)
201                     .addComponent(jButton3)))
202             .addContainerGap(21, Short.MAX_VALUE))
203     );
204
205     jTabbedPane1.addTab("LOAN MANAGEMENT", jPanel2);
206
207     jLabel5.setText("ENTER GROSS SALARY");
208
209     jTextArea3.setColumns(20);
210     jTextArea3.setRows(5);
211     jScrollPane4.setViewportView(jTextArea3);
212
213     jLabel6.setText("LOAN DETAILS");
214
215     javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
216     jPanel4.setLayout(jPanel4Layout);
217     jPanel4Layout.setHorizontalGroup(
218         jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
219         .addGroup(jPanel4Layout.createSequentialGroup()
220             .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
221                 .addGroup(jPanel4Layout.createSequentialGroup()
222                     .addGap(102, 102, 102)
223                     .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 158,
javax.swing.GroupLayout.PREFERRED_SIZE)
224                     .addGap(72, 72, 72)
225                     .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 112,
javax.swing.GroupLayout.PREFERRED_SIZE))
226                 .addGroup(jPanel4Layout.createSequentialGroup()
227                     .addGap(59, 59, 59)
228                     .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE)
229                     .addGap(29, 29, 29)
230                     .addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE, 275,
javax.swing.GroupLayout.PREFERRED_SIZE)))
231             .addContainerGap(249, Short.MAX_VALUE))
232         );
233         jPanel4Layout.setVerticalGroup(
234             jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
235             .addGroup(jPanel4Layout.createSequentialGroup()
236                 .addGap(65, 65, 65)
237                 .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
238                     .addComponent(jLabel5)
239                     .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
240                 .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
241                     .addGroup(jPanel4Layout.createSequentialGroup()
242                         .addGap(65, 65, 65)
243                         .addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE, 187,
javax.swing.GroupLayout.PREFERRED_SIZE))
244                     .addGroup(jPanel4Layout.createSequentialGroup()
245                         .addGap(79, 79, 79)
246                         .addComponent(jLabel6)))
247                 .addContainerGap(92, Short.MAX_VALUE))
248         );
249
250         jTabbedPane1.addTab("EXPLORE LOANS", jPanel4);
251
252         javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
253         getContentPane().setLayout(layout);
254         layout.setHorizontalGroup(
255             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
256             .addGroup(layout.createSequentialGroup()
257                 .addContainerGap()
258                 .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 727,
javax.swing.GroupLayout.PREFERRED_SIZE)
259                 .addContainerGap(28, Short.MAX_VALUE))
260         );
261         layout.setVerticalGroup(
262             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
263             .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
264                 .addContainerGap()
265                 .addComponent(jTabbedPane1)
266                 .addContainerGap())
267         );
268
269         pack();
270     }// </editor-fold>
271
272     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
273         Scanner contents = null;
274
```

```
275
276     String st;
277
278 try {
279       contents = new Scanner(new FileReader("ScootLoans.txt"));
280       while (contents.hasNext()) {
281           st = contents.next();
282           s = st.split("\\,");
283         //  jTextArea2.append(st+"\n");
284
285
286
287          loanID[counter] = s[0];
288          empName [counter] = s[1];
289           amounts[counter][0] = s[2];
290           amounts[counter][1] = s[3];
291
292
293
294           counter ++;
295       }
296          for (int i = 0; i < counter; i++) {
297             jTextArea2.append(loanID[i] + "  "+empName[i]);
298         //    JTable.setValueAt(loanID[i], i,0);
299          //jTable1.setValueAt(empName[i], i,1);
300
301                  jTextArea2.append(" "+amounts[i][0]+"  "+amounts[i][1]);
302                // jTable1.setValueAt(amounts[i][0],i,2);
303              //   jTable1.setValueAt(amounts[i][1],i,3);
304
305
306             jTextArea2.append("\n");
307          }
308          for (int i = 0; i < counter; i++) {
309           //  jTable1.setValueAt(amounts[i][0],i,2);
310            // jTable1.setValueAt(amounts[i][1], i,3);
311
312
313     }
314       }
315   catch (FileNotFoundException e) {
316      System.out.println("ERROR"+e.getMessage());
317   }
318    }
319
320    /**
321     * @param args the command line arguments
322     */
323    public static void main(String args[]) {
324        /* Set the Nimbus look and feel */
325        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
326        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
327         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
328         */
329        try {
330            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
331                if ("Nimbus".equals(info.getName())) {
332                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
333                    break;
334                }
335            }
```

```
336         } catch (ClassNotFoundException ex) {
337             java.util.logging.Logger.getLogger(ScootLoans.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
338         } catch (InstantiationException ex) {
339             java.util.logging.Logger.getLogger(ScootLoans.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
340         } catch (IllegalAccessException ex) {
341             java.util.logging.Logger.getLogger(ScootLoans.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
342         } catch (javax.swing.UnsupportedLookAndFeelException ex) {
343             java.util.logging.Logger.getLogger(ScootLoans.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
344         }
345         //</editor-fold>
346
347         /* Create and display the form */
348         java.awt.EventQueue.invokeLater(new Runnable() {
349             public void run() {
350                 new ScootLoans().setVisible(true);
351             }
352         });
353     }
354     // Variables declaration - do not modify
355     private javax.swing.JButton jButton1;
356     private javax.swing.JButton jButton2;
357     private javax.swing.JButton jButton3;
358     private javax.swing.JButton jButton4;
359     private javax.swing.JFormattedTextField jFormattedTextField1;
360     private javax.swing.JFormattedTextField jFormattedTextField2;
361     private javax.swing.JFormattedTextField jFormattedTextField3;
362     private javax.swing.JLabel jLabel1;
363     private javax.swing.JLabel jLabel2;
364     private javax.swing.JLabel jLabel3;
365     private javax.swing.JLabel jLabel4;
366     private javax.swing.JLabel jLabel5;
367     private javax.swing.JLabel jLabel6;
368     private javax.swing.JPanel jPanel1;
369     private javax.swing.JPanel jPanel2;
370     private javax.swing.JPanel jPanel3;
371     private javax.swing.JPanel jPanel4;
372     private javax.swing.JScrollPane jScrollPane2;
373     private javax.swing.JScrollPane jScrollPane3;
374     private javax.swing.JScrollPane jScrollPane4;
375     private javax.swing.JTabbedPane jTabbedPane1;
376     private javax.swing.JTextArea jTextArea1;
377     private javax.swing.JTextArea jTextArea2;
378     private javax.swing.JTextArea jTextArea3;
379     private javax.swing.JTextField jTextField1;
380     private javax.swing.JTextField jTextField2;
381     // End of variables declaration
382 }
```