

## **Project 2 by**

**-Manideep Paturi([mp672@njit.edu](mailto:mp672@njit.edu))**

- **As part of Coursework for Java Programming**
- **Under Professor Leon Vaks**
- **Submitted on 05/07/2017**

Project 2:

**a : Calculate the hypotenuse of a right triangle**

**b: Validate user entries**

**c: Manage customer data (GUI)**

**d: Tortoise and the Hare Race.**

**The contents for each project are as follows**

## **Contents**

1. Description
2. Functionality
3. Working Images
4. API Listing
5. Error Codes and Limitations
6. Assumptions/Design Decisions
7. Source code

- **a : Calculate the hypotenuse of a right triangle**

### **Descripton:**

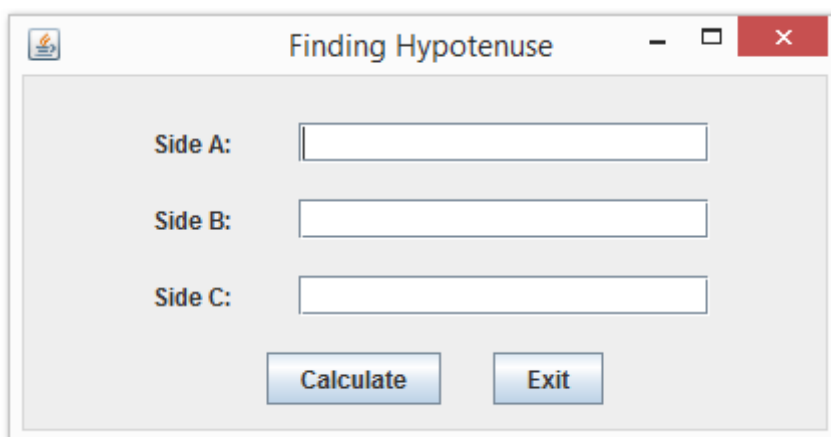
- This application lets the user enter the lengths of the two shortest sides of a right triangle. When the user clicks the Calculate button, the application calculates and displays the length of the third side.

### **Functionality:**

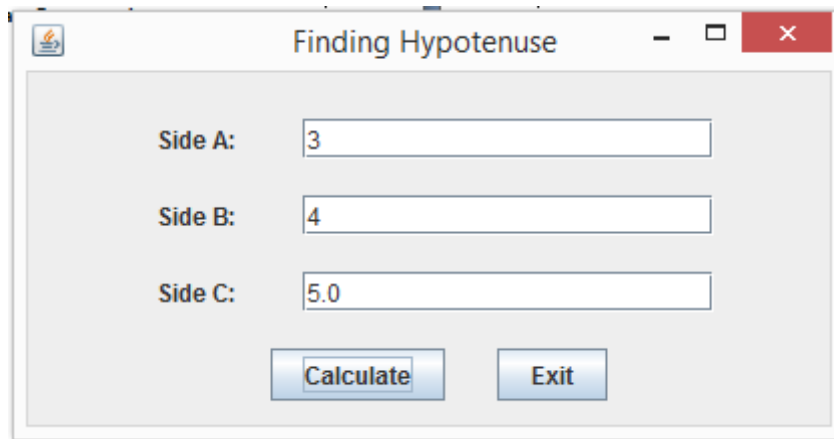
- Used the Pythagorean Theorem to calculate the length of the third side. The Pythagorean Theorem states that the square of the hypotenuse of a right-triangle is equal to the sum of the squares of the opposite sides:
- Validated the user input so that the user must enter a double value for side A and B of the triangle.

### **Working images:**

#### **1.Application is Running**

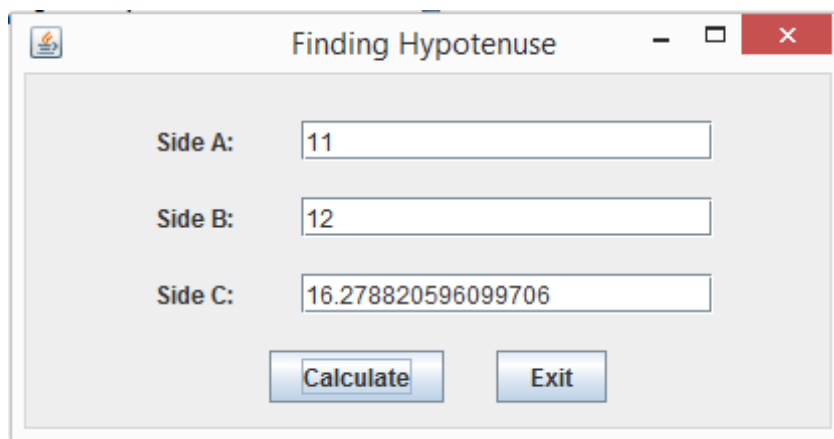


## 2.Entering Values as 3 and 4 gives result 5



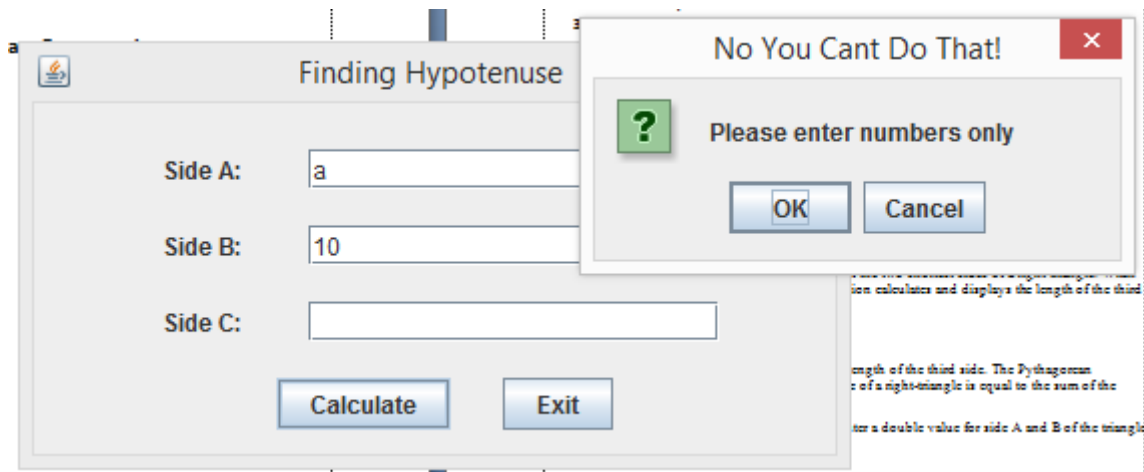
The screenshot shows a window titled "Finding Hypotenuse". It contains three input fields: "Side A:" with the value "3", "Side B:" with the value "4", and "Side C:" with the value "5.0". Below the input fields are two buttons: "Calculate" and "Exit".

## 3.Another Set of Values,gives a double value



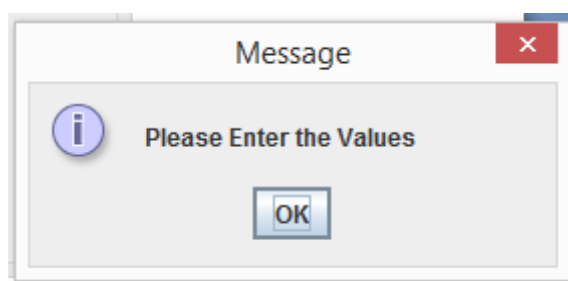
The screenshot shows the same "Finding Hypotenuse" window. The input fields now contain "Side A:" with "11", "Side B:" with "12", and "Side C:" with the long decimal value "16.278820596099706". The "Calculate" and "Exit" buttons are still present.

## 4.When a string value is entered,it validates



The screenshot shows the "Finding Hypotenuse" window with "Side A:" containing the string "a" and "Side B:" containing "10". A validation error dialog box is overlaid on top of the main window. The dialog box is titled "No You Cant Do That!" and contains a green question mark icon, the text "Please enter numbers only", and "OK" and "Cancel" buttons.

## 5.When no values are entered.



The screenshot shows a "Message" dialog box. It has a title bar with a close button. Inside, there is an information icon (i) followed by the text "Please Enter the Values". At the bottom, there is an "OK" button.

## **4.API Listing**

**These are the API used in the project  
Java Swing-**

### **Action**

**The Action interface provides a useful extension to the ActionListener interface in cases where the same functionality may be accessed by several controls.**

### **ComboBoxEditor**

**The editor component used for JComboBox components.**

### **JApplet**

**An extended version of java.applet.Applet that adds support for the JFC/Swing component architecture.**

### **JButton**

**An implementation of a "push" button.**

### **JCheckBox**

**An implementation of a check box -- an item that can be selected or deselected, and which displays its state to the user.**

### **JCheckBoxMenuItem**

**A menu item that can be selected or deselected.**

### **JColorChooser**

**JColorChooser provides a pane of controls designed to allow a user to manipulate and select a color.**

### **JComboBox<E>**

**A component that combines a button or editable field and a drop-down list.**

## **JComponent**

The base class for all Swing components except top-level containers. JDesktopPane

A container used to create a multiple-document interface or a virtual desktop.

## **JDialog**

The main class for creating a dialog window. JTable

The JTable is used to display and edit regular two-dimensional tables of cells. JTable.DropLocation

A subclass of TransferHandler.DropLocation representing a drop location for a JTable.

## **JEditorPane**

A text component to edit various kinds of content.

## **Error Codes and Limitations:**

There are no errors and limitations as far I ran, In this particular project

## **Assumptions/Design Decisions:**

Design and Assumptions are exactly in accordance to the requirement given. There was no need of any assumptions

## **Source Code:**

The source code is given at the end of this pdf

**b: Validate user entries**

## Descripton:

- This application accepts user entries and validates them according to the specifications below.
- If the data is valid, this app displays the data in a dialog box. Then, when the user clicks OK in the dialog box, the application clears the text fields, so the user can make another entry.
- If the data is not valid, this app displays a dialog box with an appropriate error message and attempts to move the focus to the text field with the invalid data.

## Functionality:

- The Name field is required.
- The Age field is required and must be a valid integer value.
- The Sales field is required and must be a valid double value.
- A class is created and named SwingValidator to perform the validation. This class contains these methods:

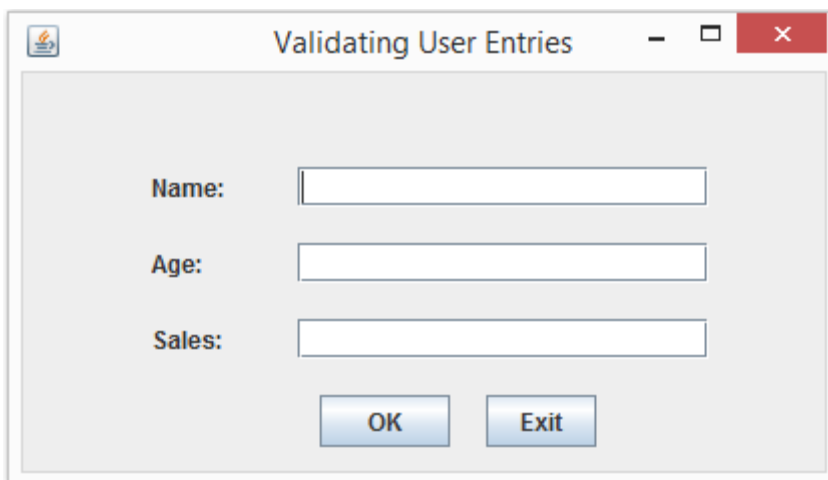
```
boolean isEmpty(JTextField field, String fieldName)
```

```
boolean isInteger(JTextField field, String fieldName)
```

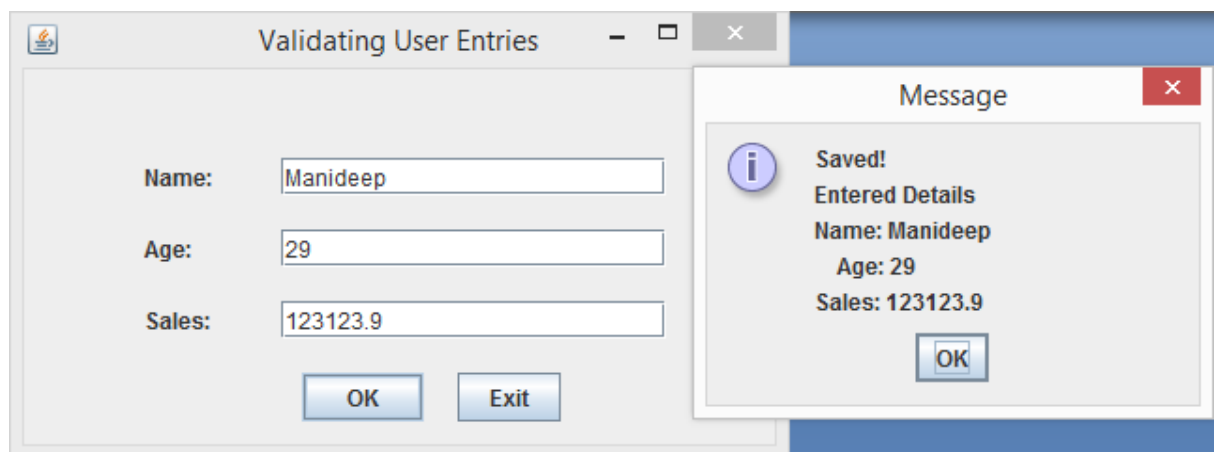
```
boolean isDouble(JTextField field, String fieldName)
```

## Working images:

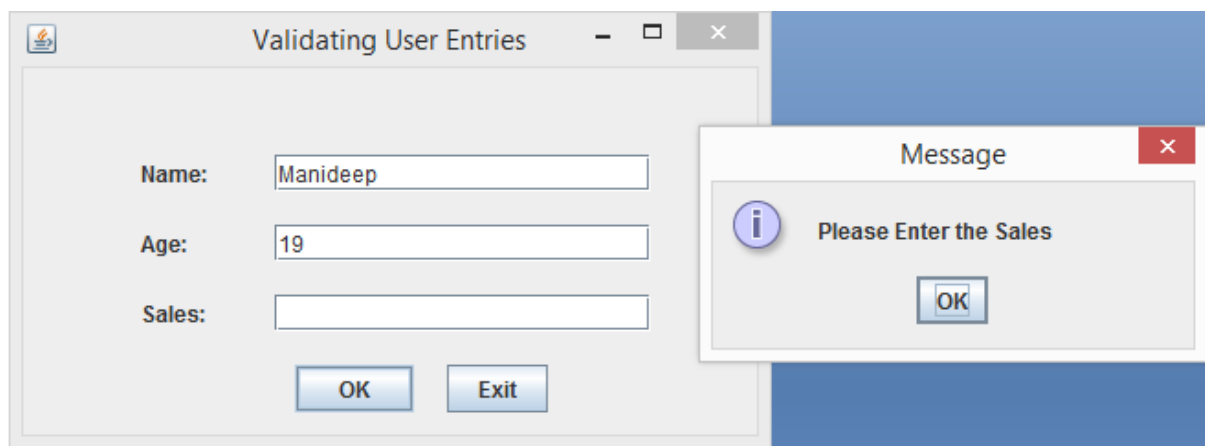
### 1.Application is Running



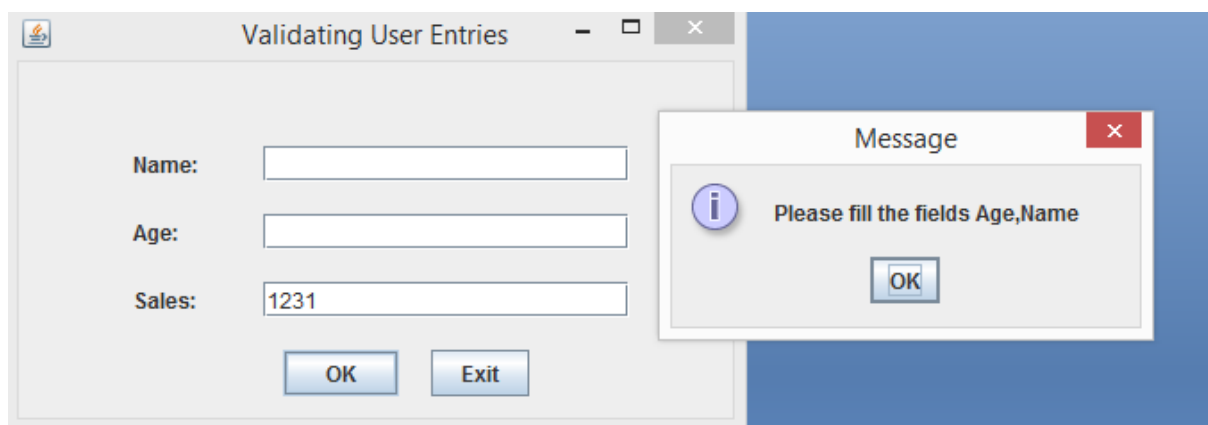
### 2.User Details are successfully entered



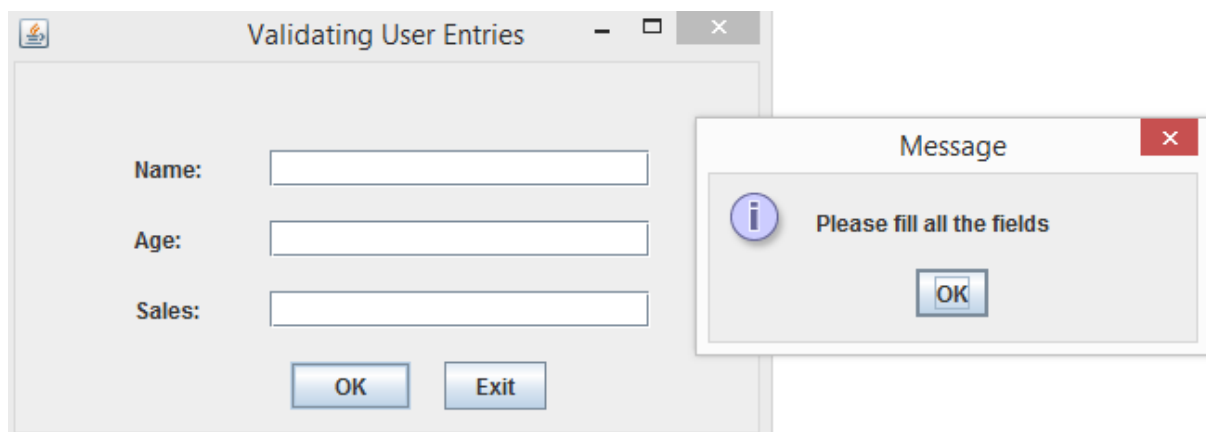
### 3. When a field is not entered



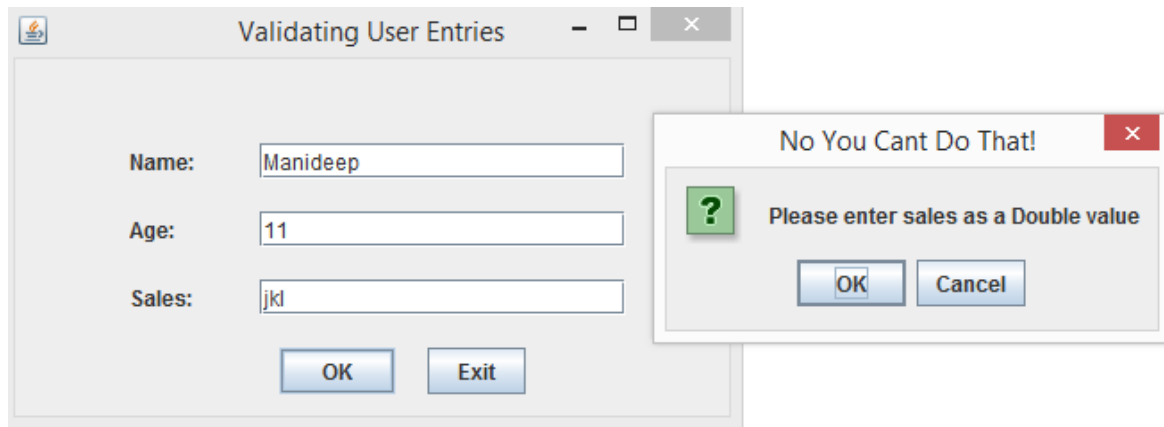
### 4. When more than one fields is not entered



### 5. When no field is entered



### 6. When a wrong data type is entered



**7. There are some parts which cannot be shown in screenshots**

- Fields are cleared after a data is entered successfully, or a mismatch occurs in data type, this can only be seen when the application is running.

### **Error Codes and Limitations:**

There are no errors and limitations as far I ran, In this particular project

### **Assumptions/Design Decisions:**

Design and Assumptions are exactly in accordance to the requirement given. There was no need of any assumptions

### **API listing:**

It is almost similar to first part. The full api list used will be shown at the end of the project.

### **Source Code:**

The source code is given at the end of this pdf

## **c: Manage customer data (GUI)**

### **Description**

- This application begins by displaying a table of customer data.
- If the user clicks the Add button, the application allows the user to add customer data to the table (and the underlying database).



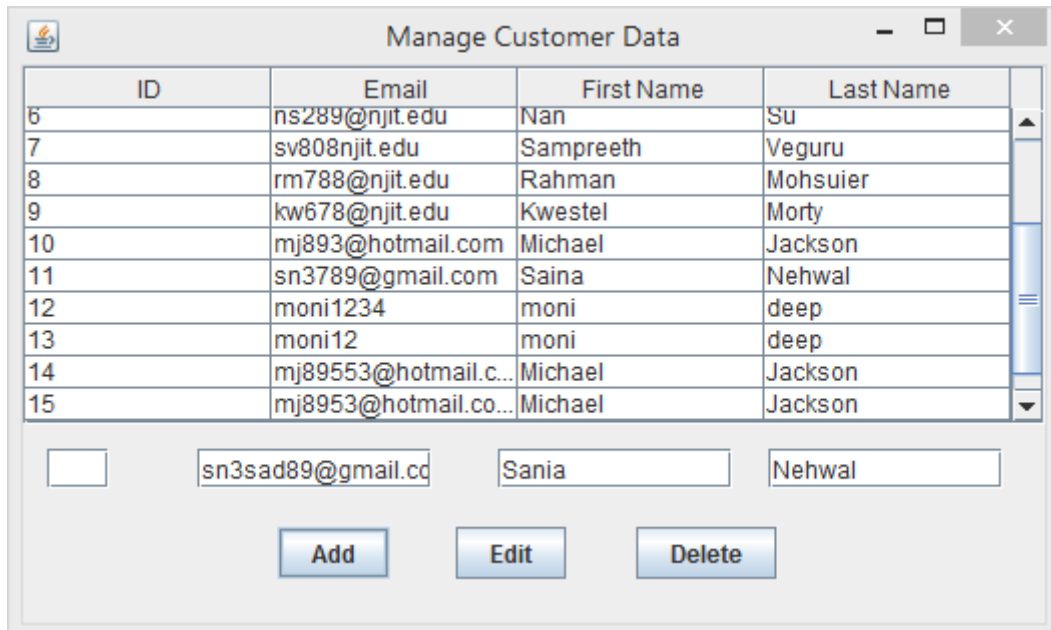
- If the user selects a customer row and clicks the Edit button, the application allows the user to update the data for the selected customer row in the table (and the database).
- If the user selects a customer row and clicks the Delete button, the application deletes the selected customer row from the table (and the database).

### **Functionality**

- A table in the NJIT database to store the necessary data is created. To do that, I used an SQL script
- A class named Customer that stores data for the user's id, email address, first name, and last name is created
- A class named CustomerDB that contains the methods necessary to get an array list of Customer objects, to get a Customer object for the customer with the specified id, and to add, update, or delete the specified customer is created . This frame99 displays a table of customer data as well as the Add, Edit, and Delete buttons. This class uses the Customer and CustomerDB classes to work with the customer data.
- The main method , a CustomerForm class that allows the user to add or edit customer data is created.

## Working Images

### 1. Showing the loaded table when the application started

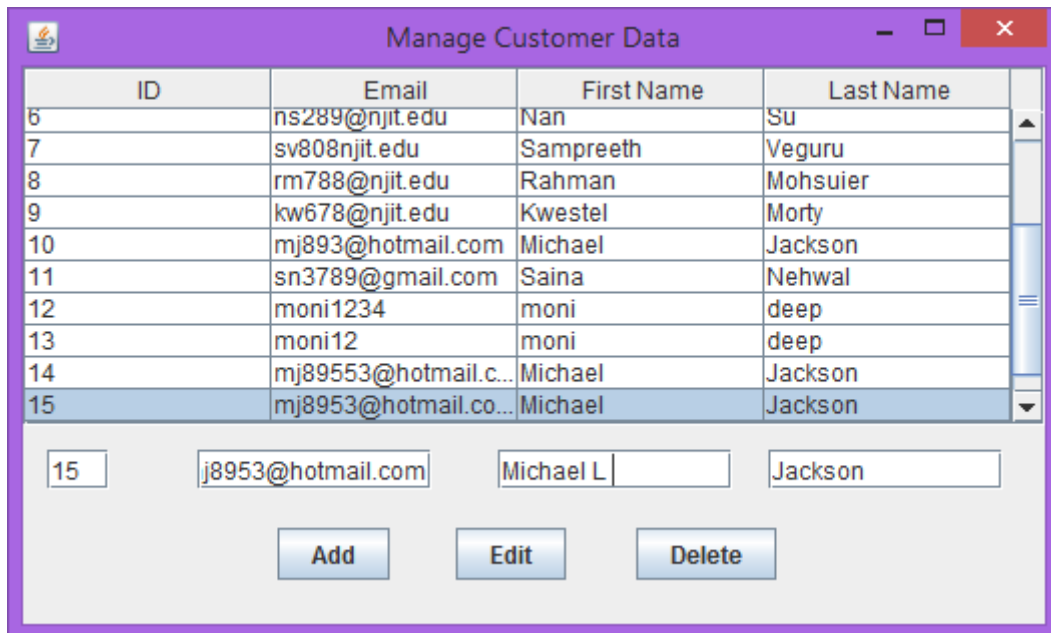
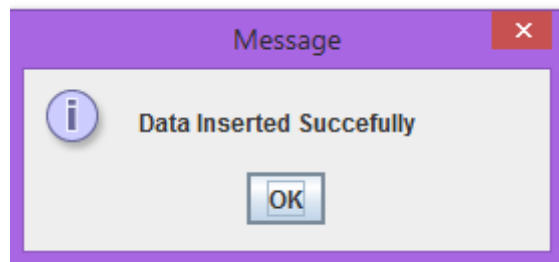


The screenshot shows a window titled "Manage Customer Data" with a table containing 15 rows of customer data. The table has four columns: ID, Email, First Name, and Last Name. Below the table, there are four input fields for adding new data, followed by "Add", "Edit", and "Delete" buttons.

ID	Email	First Name	Last Name
6	ns289@njit.edu	Nan	Su
7	sv808njit.edu	Sampreeth	Veguru
8	rm788@njit.edu	Rahman	Mohsuier
9	kw678@njit.edu	Kwestel	Morty
10	mj893@hotmail.com	Michael	Jackson
11	sn3789@gmail.com	Saina	Nehwal
12	moni1234	moni	deep
13	moni12	moni	deep
14	mj89553@hotmail.c...	Michael	Jackson
15	mj8953@hotmail.co...	Michael	Jackson

Below the table, there are four input fields for adding new data: ID (empty), Email (sn3sad89@gmail.co), First Name (Sania), and Last Name (Nehwal). Below these fields are three buttons: "Add", "Edit", and "Delete".

### 2. Screenshots related to adding a data



The screenshot shows the "Manage Customer Data" window with the table updated. The row with ID 15 is highlighted. Below the table, the input fields are filled with the new data: ID (15), Email (j8953@hotmail.com), First Name (Michael L), and Last Name (Jackson). The "Add", "Edit", and "Delete" buttons are still present.

ID	Email	First Name	Last Name
6	ns289@njit.edu	Nan	Su
7	sv808njit.edu	Sampreeth	Veguru
8	rm788@njit.edu	Rahman	Mohsuier
9	kw678@njit.edu	Kwestel	Morty
10	mj893@hotmail.com	Michael	Jackson
11	sn3789@gmail.com	Saina	Nehwal
12	moni1234	moni	deep
13	moni12	moni	deep
14	mj89553@hotmail.c...	Michael	Jackson
15	mj8953@hotmail.co...	Michael	Jackson

Below the table, the input fields are filled with the new data: ID (15), Email (j8953@hotmail.com), First Name (Michael L), and Last Name (Jackson). Below these fields are three buttons: "Add", "Edit", and "Delete".

### 3. Showing data when updated


Manage Customer Data

ID	Email	First Name	Last Name
6	ns289@njit.edu	Nan	Su
7	sv808njit.edu	Sampreeth	Veguru
8	rm788@njit.edu	Rahman	Mohsuier
9	kw678@njit.edu	Kwestel	Morty
10	mj893@hotmail.com	Michael	Jackson
11	sn3789@gmail.com	Saina	Nehwal
12	moni1234	moni	deep
13	moni12	moni	deep
14	mj89553@hotmail.c...	Michael	Jackson
15	mj8953@hotmail.co...	Michael L	Jackson

15    j8953@hotmail.com    Michael L    Jackson

Add Edit Delete

Message

 Data Updated Succesfully

OK

4.A row is selected and data is deleted

See that ID 15 is removed after deletion


Manage Customer Data

ID	Email	First Name	Last Name
6	ns289@njit.edu	Nan	Su
7	sv808njit.edu	Sampreeth	Veguru
8	rm788@njit.edu	Rahman	Mohsuier
9	kw678@njit.edu	Kwestel	Morty
10	mj893@hotmail.com	Michael	Jackson
11	sn3789@gmail.com	Saina	Nehwal
12	moni1234	moni	deep
13	moni12	moni	deep
14	mj89553@hotmail.c...	Michael	Jackson
16	sn3sad89@gmail.c...	Sania	Nehwal

15    j8953@hotmail.com    Michael L    Jackson

Add Edit Delete

Message

 Data Deleted Succesfully

OK

## **API LISTING:**

These are the API used in the project

### **Java Swing-**

#### **Action**

The Action interface provides a useful extension to the ActionListener interface in cases where the same functionality may be accessed by several controls.

#### **ComboBoxEditor**

The editor component used for JComboBox components.

#### **JApplet**

An extended version of java.applet.Applet that adds support for the JFC/Swing component architecture.

#### **JButton**

An implementation of a "push" button.

#### **JCheckBox**

An implementation of a check box -- an item that can be selected or deselected, and which displays its state to the user.

#### **JCheckBoxMenuItem**

A menu item that can be selected or deselected.

#### **JColorChooser**

JColorChooser provides a pane of controls designed to allow a user to manipulate and select a color.

#### **JComboBox<E>**

A component that combines a button or editable field and a drop-down list.

## **JComponent**

The base class for all Swing components except top-level containers.

## **JDesktopPane**

A container used to create a multiple-document interface or a virtual desktop.

## **JDialog**

The main class for creating a dialog window.

## **JTable**

The JTable is used to display and edit regular two-dimensional tables of cells.

## **JTable.DropLocation**

A subclass of TransferHandler.DropLocation representing a drop location for a JTable.

## **JEditorPane**

A text component to edit various kinds of content.

## **Java.Sql**

## **DriverManager**

The basic service for managing a set of JDBC drivers.

**NOTE:** The DataSource interface, new in the JDBC 2.0 API, provides another way to connect to a data source.

## **SQLException**

**The subclass of SQLException thrown when the SQLState class value is '22', or under vendor-specified conditions.**

### **SQLException**

**An exception that provides information on a database access error or other errors.**

### **SQLInput**

**An input stream that contains a stream of values representing an instance of an SQL structured type or an SQL distinct type.**

### **SQLOutput**

**The output stream for writing the attributes of a user-defined type back to the database.**

### **SQLXML**

**The mapping in the JavaTM programming language for the SQL XML type.**

### **Statement**

**The object used for executing a static SQL statement and returning the results it produces.**

### **Java.Utility**

### **ArrayList**

**Resizable-array implementation of the List interface. Implements all optional list operations, and permits all elements, including null. In addition to implementing the List interface, this class provides methods to manipulate**

the size of the array that is used internally to store the list. (This class is roughly equivalent to Vector, except that it is unsynchronized.)

**public ArrayList(int initialCapacity)**

Constructs an empty list with the specified initial capacity.

**Parameters:**

**initialCapacity** - the initial capacity of the list

**Throws:**

**IllegalArgumentException** - if the specified initial capacity is negative

## **Error Codes and Limitations:**

Due to the time and other constraints there may be certain expected and unexpected errors showing up in the project .Some of them are

- There might be connection errors with the database due to the timer
- When a non expected field like integer value is typed by the user, the app doesn't do anything
- Internally it may be throwing error but in the UI it will be silent
- The auto increment is set once in database, and since the database cannot be maintained like formal projects the increment value stays where it is left.
- For example if there are 15 entries and if a user deletes one value the new ID value will continue from 16 instead of 15.

- If there was a little more time a new column showing total of transactions and to export the details into excel format – these two functionalities may be added
- Other than the above errors the app runs fine, and never crashes at least in my test period .There may be other unforeseen errors which will be dealt upon in near future.

### **Assumptions/Design Decisions:**

- Design and Assumptions are exactly in accordance to the requirement given. There was no need of any assumptions
- The only thing different from actual is I used only 1 table in database instead of proposed 2 different tables.
- Considering the memory aspects I took this decision, since I felt the second table was unnecessary

### **Source Code:**

The source code is given at the end of this pdf

**d: Tortoise and the Hare Race.**

### **Description:**

- This application simulates a race between two or more runners. The runners differ in their speed and how often they need to rest. One of the runners, named “Tortoise,” is slow but never rests. The other runner, named “Hare,” is ten times as fast but rests 90% of the time.
- There is a random element to the runners’ performance, so the outcome of the race is different each time the application is run.
- The race is run over a course of 1000 meters. Each time one of the runners moves, the application displays the runner’s new position on the course. The first runner to reach 1000 wins the race.

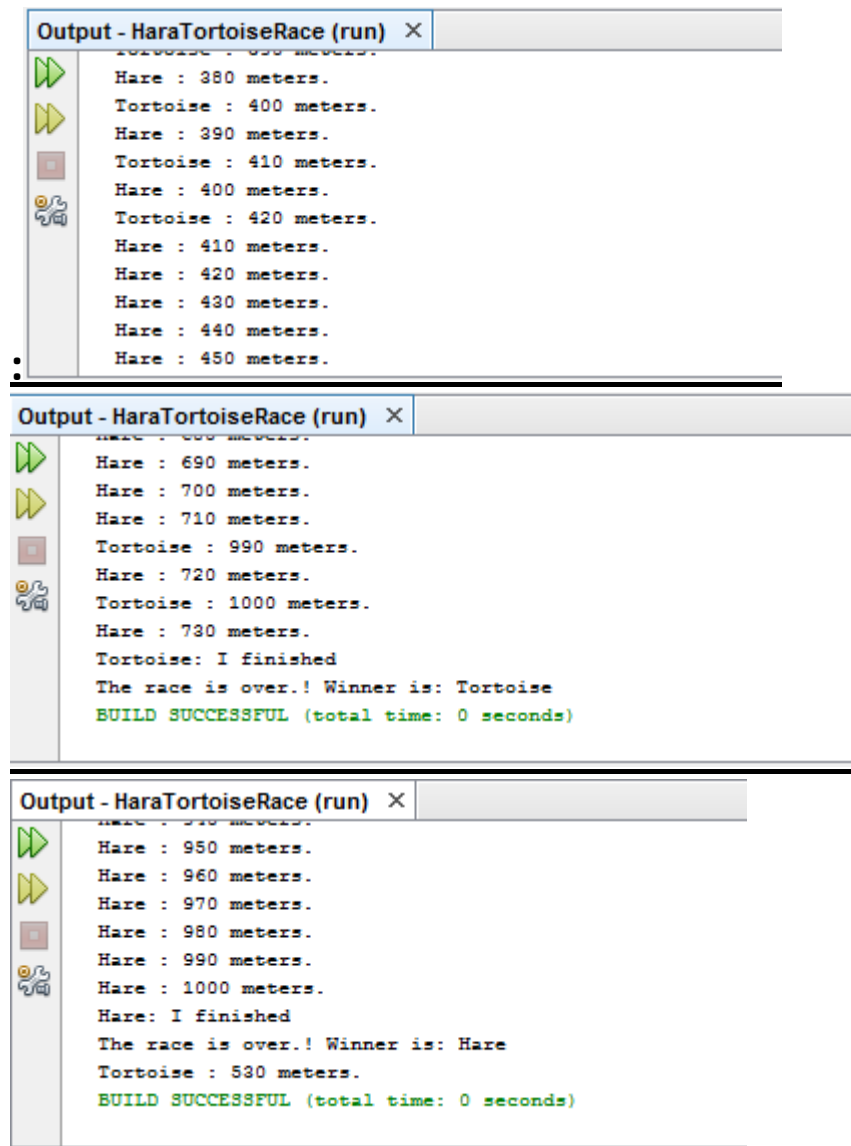


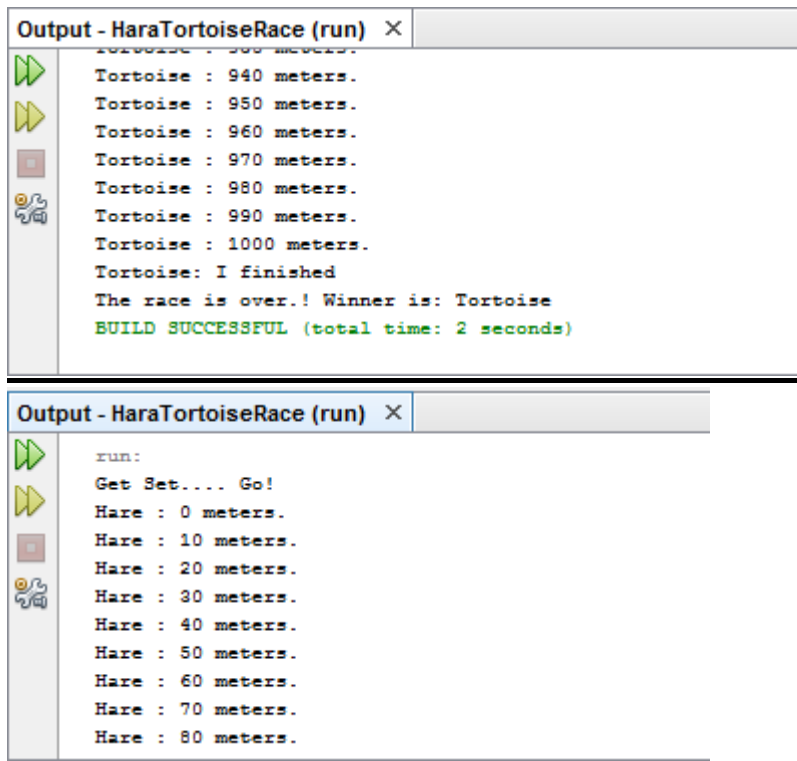
- When one of the runners finishes the race, the application declares that runner to be the winner and the other runner concedes.

## Functionality:

- Each runner is implemented as a separate thread using a class named ThreadRunner. The ThreadRunner class an int value that indicates the runners speed—that is, how many meters the runner travels in each move.
- The racer method of the ThreadRunner class consists of a loop that repeats until the runner has reached 1000 meters. Each time through the loop, the thread decides whether it should run or rest based on a random number . If this random number indicates that the runner should run, the class. The run method sleeps for 100 milliseconds on each repetition of the loop.
- The main method of the application’s main class shows created two runner threads and starts them. One of the threads named as “Tortoise.” It runs only 10 meters each move, but plods along without ever resting. The other thread should be named “Hare.” It should run 10 meters each move, but should rest 90% of the time.

## Working Images





The image shows two screenshots of a Java IDE's output window. The top screenshot shows the output of a race where the Tortoise wins. The bottom screenshot shows the output of a race where the Hare wins.

```
Output - HaraTortoiseRace (run) X
Tortoise : 940 meters.
Tortoise : 950 meters.
Tortoise : 960 meters.
Tortoise : 970 meters.
Tortoise : 980 meters.
Tortoise : 990 meters.
Tortoise : 1000 meters.
Tortoise: I finished
The race is over.! Winner is: Tortoise
BUILD SUCCESSFUL (total time: 2 seconds)

Output - HaraTortoiseRace (run) X
run:
Get Set.... Go!
Hare : 0 meters.
Hare : 10 meters.
Hare : 20 meters.
Hare : 30 meters.
Hare : 40 meters.
Hare : 50 meters.
Hare : 60 meters.
Hare : 70 meters.
Hare : 80 meters.
```

In some scenarios Hare is the winner ,and in some tortoise wins.

### **Assumptions/Design Decisions:**

- I gave the steps to Hare as 10 which is same as Tortoise, but due to Hares sleeping mechanism the winner differs
- Couldn't implement all the enhancements, although just by adding few more thread objects , race can be increased to 9 participants

- **Limitations:**

Certain limitations are mentioned in the inline comments in source code

### **Source code :**

**Is provided at the end**

***a : Source code for Calculate the hypotenuse of a right triangle***

```
package righttriangles1;
```

```
import javax.swing.JOptionPane;
```

```
/*
```

```
 * To change this license header, choose  
   License Headers in Project Properties.
```

```
 * To change this template file, choose Tools |  
   Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
/**
```

```
 *
```

```
 * @author Manideep
```

```
*/
```

```
public class HypotenuseFinder extends
```

```
javax.swing.JFrame {

    /**
     * Creates new form HypotenuseFinder
     */
    public HypotenuseFinder() {
        initComponents();
    }

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

```
private void initComponents() {  
  
    jTextField_SideA = new  
    javax.swing.JTextField();  
  
    jTextField_SideB = new  
    javax.swing.JTextField();  
  
    jTextField_SideC = new  
    javax.swing.JTextField();  
  
    jButton_Calculate = new  
    javax.swing.JButton();  
  
    jButton_Exit = new javax.swing.JButton();  
    jLabel1 = new javax.swing.JLabel();  
    jLabel2 = new javax.swing.JLabel();  
    jLabel3 = new javax.swing.JLabel();  
  
    setDefaultCloseOperation(javax.swing.Windo  
wConstants.EXIT_ON_CLOSE);  
    setTitle("Finding Hypotenuse ");
```

```
jButton_Calculate.setText("Calculate");

jButton_Calculate.addMouseListener(new
java.awt.event.MouseAdapter() {

    public void
mouseClicked(java.awt.event.MouseEvent
evt) {

        jButton_CalculateMouseClicked(evt);

    }

});

jButton_Calculate.addActionListener(new
java.awt.event.ActionListener() {

    public void
actionPerformed(java.awt.event.ActionEvent
evt) {

jButton_CalculateActionPerformed(evt);

    }

});

jButton_Exit.setText("Exit");

jButton_Exit.addMouseListener(new
```

```
java.awt.event.MouseAdapter() {  
    public void  
    mouseClicked(java.awt.event.MouseEvent  
    evt) {  
        JButton_ExitMouseClicked(evt);  
    }  
});
```

```
jLabel1.setText("Side B:");
```

```
jLabel2.setText("Side A:");
```

```
jLabel3.setText("Side C:");
```

```
javax.swing.GroupLayout layout = new  
javax.swing.GroupLayout(getContentPane());  
getContentPane().setLayout(layout);  
layout.setHorizontalGroup
```

```
layout.createParallelGroup(javax.swing.Grou
```

pLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

    .addGap(121, 121, 121)

    .addComponent(jButton\_Calculate)

    .addGap(26, 26, 26)

    .addComponent(jButton\_Exit)

.addContainerGap(javax.swing.GroupLayout.  
DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignme  
nt.TRAILING, layout.createSequentialGroup())

    .addContainerGap(64,  
Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.s  
wing.GroupLayout.Alignment.LEADING)

    .addComponent(jLabel1,  
javax.swing.GroupLayout.PREFERRED\_SIZE,  
44,  
javax.swing.GroupLayout.PREFERRED\_SIZE)



```
        .addComponent(jLabel3,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
44,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
        .addComponent(jLabel2,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
44,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addGap(29, 29, 29)
```

```
.addGroup(layout.createParallelGroup(javax.s  
wing.GroupLayout.Alignment.LEADING)
```

```
        .addComponent(jTextField_SideC,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
205,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jTextField_SideB,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
205,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jTextField_SideA,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
205,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
    .addGap(58, 58, 58))
```

```
);
```

```
layout.setVerticalGroup(
```

```
layout.createParallelGroup(javax.swing.Grou  
pLayout.Alignment.LEADING)
```

```
.addGroup(layout.createSequentialGroup())
```

```
    .addContainerGap(23,  
Short.MAX_VALUE)
```

```
.addGroup(layout.createParallelGroup(javax.s  
wing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jTextField_SideA,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(jLabel2))
```

```
.addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.s  
wing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jTextField_SideB,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(jLabel1))
```

```
    .addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.s  
wing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jTextField_SideC,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(jLabel3))
```

```
    .addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.s  
wing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jButton_Calculate)
```

```
    .addComponent(jButton_Exit))
```

```
        .addContainerGap()

    );

    pack();

} // </editor-fold> // GEN-END: initComponents

private void
jButton_CalculateActionPerformed(java.awt.
event.ActionEvent evt) { // GEN-
FIRST:event_jButton_CalculateActionPerform
ed

    // TODO add your handling code here:

} // GEN-
LAST:event_jButton_CalculateActionPerform
ed

private void
jButton_CalculateMouseClicked(java.awt.eve
nt.MouseEvent evt) { // GEN-
FIRST:event_jButton_CalculateMouseClicked

    // TODO add your handling code here:
```

```
try{

if(jTextField_SideA.getText().equals("") || jTextField_SideB.getText().equals("")){

    JOptionPane.showMessageDialog(null,
    "Please Enter the Values");

}

else{

    double a=
    Double.parseDouble(jTextField_SideA.getText());

    double b=
    Double.parseDouble(jTextField_SideB.getText());

    //pythagoreon formula

    double t=Math.sqrt((a*a)+(b*b));

jTextField_SideC.setText(Double.toString(t));
}

catch (NumberFormatException e){

    JOptionPane.showConfirmDialog(null,
```

```
"Please enter numbers only", "No You Cant  
Do That!", JOptionPane.CANCEL_OPTION);
```

```
}
```

```
}//GEN-
```

```
LAST:event_jButton_CalculateMouseClicked
```

```
private void
```

```
jButton_ExitMouseClicked(java.awt.event.Mo  
useEvent evt) {//GEN-
```

```
FIRST:event_jButton_ExitMouseClicked
```

```
    // TODO add your handling code here:
```

```
    System.exit(0);
```

```
}//GEN-
```

```
LAST:event_jButton_ExitMouseClicked
```

```
/**
```

```
 * @param args the command line  
arguments
```

```
 */
```

```
public static void main(String args[]) {
```

```
/* Set the Nimbus look and feel */

//<editor-fold defaultstate="collapsed"
desc=" Look and feel setting code (optional)
">

/* If Nimbus (introduced in Java SE 6) is not
available, stay with the default look and feel.

* For details see
http://download.oracle.com/javase/tutorial/
uiswing/lookandfeel/plaf.html

*/

try {

    for
(javax.swing.UIManager.LookAndFeelInfo
info :
javax.swing.UIManager.getInstalledLookAndF
eels()) {

        if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.
getClassName());

            break;

        }

    }

}
```

```
}
```

```
} catch (ClassNotFoundException ex) {
```

```
java.util.logging.Logger.getLogger(HypotenuseFinder.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
} catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(HypotenuseFinder.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
} catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(HypotenuseFinder.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
} catch
```

```
(javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(HypotenuseFinder.class.getName()).log(java.util.logging.
```



```
Level.SEVERE, null, ex);
```

```
}
```

```
//</editor-fold>
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

```
java.awt.EventQueue.invokeLater(new  
Runnable() {
```

```
    public void run() {
```

```
        new
```

```
HypotenuseFinder().setVisible(true);
```

```
    }
```

```
});
```

```
}
```

```
// Variables declaration - do not  
modify//GEN-BEGIN:variables
```

```
public javax.swing.JButton jButton_Calculate;
```

```
private javax.swing.JButton jButton_Exit;
```

```
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JTextField
jTextField_SideA;
private javax.swing.JTextField
jTextField_SideB;
private javax.swing.JTextField
jTextField_SideC;

// End of variables declaration//GEN-
END:variables

}

/*
 * To change this license header, choose
 * License Headers in Project Properties.
 * To change this template file, choose Tools |
 * Templates
 * and open the template in the editor.
 */
package righttriangles1;
```

```
/**
 *
 * @author Manideep
 */
public class RightTriangles1 {

    /**
     * @param args the command line
     arguments
     */
    public static void main(String[] args) {
        java.awt.EventQueue.invokeLater(new
        Runnable() {
            public void run() {
                new
                HypotenuseFinder().setVisible(true);
            }
        });
    }
}
```

```
    // TODO code application logic here  
}
```

```
}
```

### ***b: Source code for Validate user entries***

```
/*
```

```
* To change this license header, choose  
  License Headers in Project Properties.
```

```
* To change this template file, choose Tools |  
  Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package validateuserentries1;
```

```
import javax.swing.JOptionPane;
```

```
import javax.swing.JTextField;
```

```
/**
```

```
*
```

```
* @author Manideep
*/
//class design as per specification
//to ease the maintaining
public class SwingValidator {

    boolean isEmpty(JTextField field, String
fieldName){

        return !fieldName.equals("");
    }

    boolean isInteger(JTextField field, String
fieldName){

        try{

            int b= Integer.parseInt(fieldName);}

        catch (NumberFormatException e){

            JOptionPane.showConfirmDialog(null,
            "Please enter Age as integer value", "No You
            Cant Do That! ",
            JOptionPane.CANCEL_OPTION);

            return false;
        }
    }
}
```

```
    }  
    return true;}  
  
    boolean isDouble(JTextField field, String  
    fieldName){  
        try{  
            double  
t=Double.parseDouble(fieldName);}  
            catch (NumberFormatException e){  
                JOptionPane.showConfirmDialog(null,  
                "Please enter sales as a Double value", "No  
                You Cant Do That! ",  
                JOptionPane.CANCEL_OPTION);  
                return false;  
            }  
        return true;}  
  
    }
```

```
package validateuserentries1;
```

```
import javax.swing.JOptionPane;
```

```
/*
```

```
 * To change this license header, choose  
   License Headers in Project Properties.
```

```
 * To change this template file, choose Tools |  
   Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
/**
```

```
 *
```

```
 * @author Manideep
```

```
*/
```

```
public class ValidateUserEntries extends
```

```
javax.swing.JFrame {

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



```
jLabel2 = new javax.swing.JLabel();  
jLabel1 = new javax.swing.JLabel();  
jLabel3 = new javax.swing.JLabel();  
  
jTextField_Sales = new  
javax.swing.JTextField();  
  
jTextField_Age = new  
javax.swing.JTextField();  
  
jTextField_Name = new  
javax.swing.JTextField();  
  
jButton_OK = new javax.swing.JButton();  
jButton_Exit = new javax.swing.JButton();  
  
  
setDefaultCloseOperation(javax.swing.Window  
Constants.EXIT_ON_CLOSE);  
  
setTitle("Validating User Entries");  
  
  
jLabel2.setText("Name:");
```

```
jLabel1.setText("Age:");
```

```
jLabel3.setText("Sales:");
```

```
jButton_OK.setText("OK");
```

```
jButton_OK.addMouseListener(new  
java.awt.event.MouseAdapter() {  
    public void  
mouseClicked(java.awt.event.MouseEvent  
evt) {
```

```
        jButton_OKMouseClicked(evt);
```

```
    }
```

```
});
```

```
jButton_OK.addActionListener(new  
java.awt.event.ActionListener() {
```

```
    public void  
actionPerformed(java.awt.event.ActionEvent  
evt) {
```

```
        jButton_OKActionPerformed(evt);
```

```
    }
```

```
});
```

```
jButton_Exit.setText("Exit");
```

```
jButton_Exit.addMouseListener(new  
java.awt.event.MouseAdapter() {
```

```
    public void  
mouseClicked(java.awt.event.MouseEvent  
evt) {
```

```
        jButton_ExitMouseClicked(evt);
```

```
    }
```

```
});
```

```
jButton_Exit.addActionListener(new  
java.awt.event.ActionListener() {
```

```
    public void  
actionPerformed(java.awt.event.ActionEvent  
evt) {
```

```
        jButton_ExitActionPerformed(evt);
```

```
    }
```

```
});
```

```
    javax.swing.GroupLayout layout = new  
    javax.swing.GroupLayout(getContentPane());  
    getContentPane().setLayout(layout);  
    layout.setHorizontalGroup(
```

```
    layout.createParallelGroup(javax.swing.Grou  
    pLayout.Alignment.LEADING)
```

```
    .addGroup(layout.createSequentialGroup()
```

```
        .addGap(148, 148, 148)
```

```
        .addComponent(jButton_OK,  
        javax.swing.GroupLayout.PREFERRED_SIZE,  
        65,
```

```
        javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addGap(18, 18, 18)
```

```
        .addComponent(jButton_Exit)
```

```
    .addContainerGap(javax.swing.GroupLayout.  
    DEFAULT_SIZE, Short.MAX_VALUE))
```

```
    .addGroup(javax.swing.GroupLayout.Alignme  
    nt.TRAILING, layout.createSequentialGroup()
```

```
        .addContainerGap(64,  
Short.MAX_VALUE)
```

```
    .addGroup(layout.createParallelGroup(javax.s  
wing.GroupLayout.Alignment.LEADING)
```

```
        .addComponent(jLabel1,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
44,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jLabel3,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
44,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jLabel2,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
44,  
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
    .addGap(29, 29, 29)
```

```
    .addGroup(layout.createParallelGroup(javax.s  
wing.GroupLayout.Alignment.LEADING)
```

```
        .addComponent(jTextField_Sales,  
javax.swing.GroupLayout.PREFERRED_SIZE,
```

```

205,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jTextField_Age,
javax.swing.GroupLayout.PREFERRED_SIZE,
205,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jTextField_Name,
javax.swing.GroupLayout.PREFERRED_SIZE,
205,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(58, 58, 58))
);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.Grou
pLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup())
        .addContainerGap(47,
Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.s

```

```
wing.GroupLayout.Alignment.BASELINE)
```

```
        .addComponent(jTextField_Name,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jLabel2))
```

```
        .addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.s  
wing.GroupLayout.Alignment.BASELINE)
```

```
        .addComponent(jTextField_Age,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jLabel1))
```

```
        .addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.s  
wing.GroupLayout.Alignment.BASELINE)
```

```
        .addComponent(jTextField_Sales,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel3))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.s
wing.GroupLayout.Alignment.BASELINE)
        .addComponent(jButton_OK)
        .addComponent(jButton_Exit))
        .addContainerGap())
);

pack();

} // </editor-fold> // GEN-END: initComponents

private void
jButton_OKMouseClicked(java.awt.event.Mo
useEvent evt) { // GEN-
FIRST:event_jButton_OKMouseClicked
    // TODO add your handling code here:
```



```
SwingValidator();
```

```
}//GEN-
```

```
LAST:event_jButton_OKMouseClicked
```

```
public void SwingValidator(){
```

```
    try{
```

```
        if(new
```

```
SwingValidator().isEmpty(jTextField_Name,
```

```
jTextField_Name.getText())==false || new
```

```
SwingValidator().isEmpty(jTextField_Age,
```

```
jTextField_Age.getText())==false || new
```

```
SwingValidator().isEmpty(jTextField_Sales
```

```
, jTextField_Sales.getText())==false){
```

```
        if( new
```

```
SwingValidator().isEmpty(jTextField_Name,
```

```
jTextField_Name.getText())==false&&!new
```

```
SwingValidator().isEmpty(jTextField_Age,
```

```
jTextField_Age.getText())==false&&!new
```

```
SwingValidator().isEmpty(jTextField_Sales
```

```

, jTextField_Sales.getText())==false){

    JOptionPane.showMessageDialog(null,
    "Please Enter the Name");}

    if( !new
    SwingValidator().isEmpty(jTextField_Nam
    e, jTextField_Name.getText())==false&&new
    SwingValidator().isEmpty(jTextField_Age,
    jTextField_Age.getText())==false&&!new
    SwingValidator().isEmpty(jTextField_Sales
    , jTextField_Sales.getText())==false){

        JOptionPane.showMessageDialog(null,
        "Please Enter the Age");}

        if( !new
        SwingValidator().isEmpty(jTextField_Nam
        e, jTextField_Name.getText())==false&&!new
        SwingValidator().isEmpty(jTextField_Age,
        jTextField_Age.getText())==false&&new
        SwingValidator().isEmpty(jTextField_Sales
        , jTextField_Sales.getText())==false){

            JOptionPane.showMessageDialog(null,
            "Please Enter the Sales");}

            if( !new
            SwingValidator().isEmpty(jTextField_Nam

```

```
e, jTextField_Name.getText())==false&&new  
SwingValidator().isEmpty(jTextField_Age,  
jTextField_Age.getText())==false&&new  
SwingValidator().isEmpty(jTextField_Sales  
, jTextField_Sales.getText())==false){
```

```
    JOptionPane.showMessageDialog(null,  
"Please fill the fields Age,Sales");}
```

```
    if( new  
SwingValidator().isEmpty(jTextField_Nam  
e, jTextField_Name.getText())==false&&new  
SwingValidator().isEmpty(jTextField_Age,  
jTextField_Age.getText())==false&&!new  
SwingValidator().isEmpty(jTextField_Sales  
, jTextField_Sales.getText())==false){
```

```
        JOptionPane.showMessageDialog(null,  
"Please fill the fields Age,Name");}
```

```
    if( !new  
SwingValidator().isEmpty(jTextField_Nam  
e, jTextField_Name.getText())==false&&!new  
SwingValidator().isEmpty(jTextField_Age,  
jTextField_Age.getText())==false&&!new  
SwingValidator().isEmpty(jTextField_Sales  
, jTextField_Sales.getText())==false){
```

```
        JOptionPane.showMessageDialog(null,
```

```

    "Please fill the fields Sales,Name");}

        if( new
SwingValidator().isEmpty(jTextField_Name
e, jTextField_Name.getText())==false&&new
SwingValidator().isEmpty(jTextField_Age,
jTextField_Age.getText())==false&&new
SwingValidator().isEmpty(jTextField_Sales
, jTextField_Sales.getText())==false){

            JOptionPane.showMessageDialog(null,
"Please fill all the fields");}

        }

        else{

            String a= (jTextField_Name.getText());

            if(new
SwingValidator().isInteger(jTextField_Age,
jTextField_Age.getText())==true&&new
SwingValidator().isDouble(jTextField_Sales,
jTextField_Sales.getText())==true){

                int b=
Integer.parseInt(jTextField_Age.getText());

                double

```

```
t=Double.parseDouble(jTextField_Sales.getText());
```

```
    JOptionPane.showMessageDialog(null,  
    "Saved!\nEntered Details\nName: "+a+"\n  
Age: "+b+"\nSales: "+t);
```

```
        jTextField_Age.setText("");
```

```
        jTextField_Name.setText("");
```

```
        jTextField_Sales.setText("");
```

```
    }
```

```
}}
```

```
catch (NumberFormatException e){
```

```
    JOptionPane.showConfirmDialog(null,  
    "Please enter Details in correct format", "No  
You Cant Do That! ",  
    JOptionPane.CANCEL_OPTION);
```

```
        jTextField_Age.setText("");
```

```
        jTextField_Sales.setText("");
```

```
    }  
}  
  
private void  
jButton_OKActionPerformed(java.awt.event.  
ActionEvent evt) { //GEN-  
FIRST:event_jButton_OKActionPerformed  
    // TODO add your handling code here:  
} //GEN-  
LAST:event_jButton_OKActionPerformed
```

```
private void  
jButton_ExitMouseClicked(java.awt.event.Mo  
useEvent evt) { //GEN-  
FIRST:event_jButton_ExitMouseClicked  
    // TODO add your handling code here:  
    System.exit(0);  
} //GEN-  
LAST:event_jButton_ExitMouseClicked
```

```
private void  
jButton_ExitActionPerformed(java.awt.event.
```

```
ActionEvent evt) {  
    //GEN-FIRST:event_jButton_ExitActionPerformed  
    // TODO add your handling code here:  
    }  
    //GEN-LAST:event_jButton_ExitActionPerformed
```

```
/**  
 * @param args the command line  
 arguments  
 */  
public static void main(String args[]) {  
    /* Set the Nimbus look and feel */  
    //<editor-fold defaultstate="collapsed"  
    desc=" Look and feel setting code (optional)  
    ">  
    /* If Nimbus (introduced in Java SE 6) is not  
    available, stay with the default look and feel.  
    * For details see  
    http://download.oracle.com/javase/tutorial/  
    uiswing/lookandfeel/plaf.html  
    */
```

```

        try {
            for
(javax.swing.UIManager.LookAndFeelInfo
info :
javax.swing.UIManager.getInstalledLookAndF
eels()) {
                if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.
getClassName());

                    break;
                }
            }

        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(ValidateUs
erEntries.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ValidateUs
erEntries.class.getName()).log(java.util.loggin

```



```
g.Level.SEVERE, null, ex);
```

```
    } catch (IllegalAccessException ex) {
```

```
        java.util.logging.Logger.getLogger(ValidateUs  
erEntries.class.getName()).log(java.util.loggin  
g.Level.SEVERE, null, ex);
```

```
    } catch
```

```
(javax.swing.UnsupportedLookAndFeelExcept  
ion ex) {
```

```
        java.util.logging.Logger.getLogger(ValidateUs  
erEntries.class.getName()).log(java.util.loggin  
g.Level.SEVERE, null, ex);
```

```
    }
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

```
    java.awt.EventQueue.invokeLater(new  
Runnable() {
```

```
        public void run() {
```

```
            new
```

```
ValidateUserEntries().setVisible(true);  
    }  
});  
}
```

```
// Variables declaration - do not  
modify//GEN-BEGIN:variables  
public javax.swing.JButton jButton_Exit;  
private javax.swing.JButton jButton_OK;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
public javax.swing.JTextField jTextField_Age;  
public javax.swing.JTextField  
jTextField_Name;  
public javax.swing.JTextField  
jTextField_Sales;  
// End of variables declaration//GEN-  
END:variables  
}
```

```
/*  
    * To change this license header, choose  
      License Headers in Project Properties.  
    * To change this template file, choose Tools |  
      Templates  
    * and open the template in the editor.  
    */
```

```
package validateuserentries1;
```

```
/**  
    *  
    * @author Manideep  
    */  
public class ValidateUserEntries1 {
```

```
    /**  
        * @param args the command line  
        arguments  
    */
```

```
public static void main(String[] args) {  
    // TODO code application logic here  
  
    java.awt.EventQueue.invokeLater(new  
Runnable() {  
        public void run() {  
            new  
ValidateUserEntries().setVisible(true);  
        }  
    });  
}  
  
}
```

***c: Source code for Manage customer data  
(GUI)***

/\*

- \* To change this license header, choose License Headers in Project Properties.
- \* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package managecustomerdata.specifications;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableModel;

/\*\*

\*

\* @author Manideep

\*/

public class CustomerDB extends  
    javax.swing.JFrame {

```
/*class named CustomerDB that contains the  
methods necessary to  
  
get an array list of Customer objects, to get a  
Customer object  
  
for the customer with the specified id, and to  
add, update, or  
  
delete the specified customer.  
*/
```

```
/**  
 * Creates new form ManageCustDataForm  
 */
```

```
public CustomerDB() {  
    initComponents();  
    Show_Users_In_JTable();  
}  
  
public ArrayList<Customer> getUsersList(){  
    ArrayList<Customer> userList=new  
    ArrayList<Customer>();
```

```
Connection connection = getConnection();
//Deriving all data
String query ="SELECT * FROM `Customer`
";
Statement st;
ResultSet rs;

try {
    st=connection.createStatement();
    rs=st.executeQuery(query);
    Customer user;

    //while loop is used to get 'ALL'
rows.Untill next() is empty this runs
    while(rs.next())
    {
        user= new
Customer(rs.getInt("ID"),rs.getString("EmailId
"),rs.getString("FirstName"),rs.getString("Last
Name"));
        usersList.add(user);
    }
}
```

```

        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    return userList;
}

public Connection getConnection()
{

    Connection con;

    try{

        //Establishing connection on NJIT
        database with my login details

        con=
        DriverManager.getConnection("jdbc:mysql://
        sql2.njit.edu/mp672", "mp672",
        "Jbmjysrhjr89");

        return con;

    }

```



```
        catch(Exception e){
            e.printStackTrace();
            return null;
        }

    }

    public void Show_Users_In_JTable(){

        ArrayList<Customer> list= getUsersList();

        DefaultTableModel model
        =(DefaultTableModel)jTable_Display_Users.g
        etModel();

        Object[] row=new Object[4];
        for(int i=0;i<list.size();i++){

            row[0]=list.get(i).getID();
            row[1]=list.get(i).getEmailid();
            row[2]=list.get(i).getFirstname();
            row[3]=list.get(i).getLastname();
```

```
        model.addRow(row);
    }
}

public void executeSQLQuery(String query,
String message)
{
    Connection con = getConnection();
    Statement st;
    try{
        st = con.createStatement();
        if((st.executeUpdate(query)) == 1)
        {
            // refreshing table data

            DefaultTableModel model =
(DefaultTableModel)jTable_Display_Users.get
```

```
Model();

        model.setRowCount(0);
        Show_Users_In_JTable();

        JOptionPane.showMessageDialog(null,
        "Data "+message+" Succesfully");
    }else{
        JOptionPane.showMessageDialog(null,
        "Data Not "+message);
    }
}catch(Exception ex){
    ex.printStackTrace();
}
}
```

```
/**
```

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed"

desc="Generated Code">//GEN-

BEGIN:initComponents

private void initComponents() {

jScrollPane1 = new  
javax.swing.JScrollPane();

jTable\_Display\_Users = new  
javax.swing.JTable();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jTextField\_EmailID = new  
javax.swing.JTextField();

jTextField\_FirstName = new  
javax.swing.JTextField();

jTextField\_LastName = new  
javax.swing.JTextField();

```
    jTextField_ID = new  
    javax.swing.JTextField();
```

```
    setDefaultCloseOperation(javax.swing.Windo  
wConstants.EXIT_ON_CLOSE);
```

```
    setTitle("Manage Customer Data");
```

```
    jTable_Display_Users.setModel(new  
    javax.swing.table.DefaultTableModel(  
        new Object [][] {
```

```
            new Object [] {
```

```
                },
```

```
                new String [] {
```

```
                    "ID", "Email", "First Name", "Last  
Name"
```

```
                }  
            });
```

```
    jTable_Display_Users.addMouseListener(new  
    java.awt.event.MouseAdapter() {
```

```
        public void  
mouseClicked(java.awt.event.MouseEvent  
evt) {
```

```
    jTable_Display_UsersMouseClicked(evt);  
        }  
    });
```

```
jScrollPane1.setViewportViewView(jTable_Display  
_Users);
```

```
        jButton1.setText("Add");  
        jButton1.addMouseListener(new  
java.awt.event.MouseAdapter() {  
            public void  
mouseClicked(java.awt.event.MouseEvent  
evt) {  
                jButton1MouseClicked(evt);  
            }  
        });
```

```
jButton2.setText("Edit");

jButton2.addMouseListener(new
java.awt.event.MouseAdapter() {

    public void
mouseClicked(java.awt.event.MouseEvent
evt) {

        jButton2MouseClicked(evt);

    }

});
```

```
jButton3.setText("Delete");

jButton3.addMouseListener(new
java.awt.event.MouseAdapter() {

    public void
mouseClicked(java.awt.event.MouseEvent
evt) {

        jButton3MouseClicked(evt);

    }

});
```

```
    javax.swing.GroupLayout layout = new  
    javax.swing.GroupLayout(getContentPane());  
    getContentPane().setLayout(layout);  
    layout.setHorizontalGroup(
```

```
    layout.createParallelGroup(javax.swing.Grou  
pLayout.Alignment.LEADING)
```

```
        .addComponent(jScrollPane1,  
        javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT_SIZE,  
        510, Short.MAX_VALUE)
```

```
        .addGroup(javax.swing.GroupLayout.Alignme  
nt.TRAILING, layout.createSequentialGroup()
```

```
        .addContainerGap(javax.swing.GroupLayout.  
DEFAULT_SIZE, Short.MAX_VALUE)
```

```
            .addComponent(jButton1)
```

```
            .addGap(33, 33, 33)
```

```
            .addComponent(jButton2)
```

```
            .addGap(35, 35, 35)
```



```
.addComponent(jButton3)
.addGap(134, 134, 134))

.addGroup(layout.createSequentialGroup()
.addContainerGap()
.addComponent(jTextField_ID,
javax.swing.GroupLayout.PREFERRED_SIZE,
31,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.Co
mponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addComponent(jTextField_EmailID,
javax.swing.GroupLayout.PREFERRED_SIZE,
117,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(33, 33, 33)

.addComponent(jTextField_FirstName,
javax.swing.GroupLayout.PREFERRED_SIZE,
117,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18)
    .addComponent(jTextField_LastName,
javax.swing.GroupLayout.PREFERRED_SIZE,
117,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(21, 21, 21))
);
```

```
layout.setVerticalGroup(
```

```
layout.createParallelGroup(javax.swing.Grou
pLayout.Alignment.LEADING)
```

```
.addGroup(layout.createSequentialGroup())
```

```
    .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
179,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.Co
mponentPlacement.UNRELATED)
```

```
.addGroup(layout.createParallelGroup(javax.s  
wing.GroupLayout.Alignment.BASELINE)
```

```
        .addComponent(jTextField_EmailID,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addComponent(jTextField_FirstName,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addComponent(jTextField_LastName,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jTextField_ID,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
        .addGap(19, 19, 19)
```

```
.addGroup(layout.createParallelGroup(javax.s
```

```

wing.GroupLayout.Alignment.BASELINE)
        .addComponent(jButton1)
        .addComponent(jButton2)
        .addComponent(jButton3))
    .addGap(0, 22, Short.MAX_VALUE))
);

pack();
} // </editor-fold> // GEN-END: initComponents

private void
jButton1MouseClicked(java.awt.event.MouseEvent evt) { // GEN-
FIRST:event_jButton1MouseClicked
    if(jTextField_ID.getText().equals("")){
        {String query = "INSERT INTO `Customer`
VALUES
(NULL,'" + jTextField_EmailID.getText() + "','" + jTextField_FirstName.getText() + "','" + jTextField_LastName.getText() + "')";
        executeSQLQuery(query, "Inserted");}}

```

```
//If a specific ID is mentioned then this part  
is executed
```

```
else{
```

```
    {String query = "INSERT INTO `Customer`  
VALUES  
('"+jTextField_ID.getText()+"','"+jTextField_E  
mailID.getText()+"','"+jTextField_FirstName.g  
etText()+"','"+jTextField_LastName.getText()+"  
")";
```

```
    executeSQLQuery(query, "Inserted");}
```

```
}
```

```
// TODO add your handling code here:
```

```
}//GEN-LAST:event_jButton1MouseClicked
```

```
private void  
jTable_Display_UsersMouseClicked(java.awt.  
event.MouseEvent evt) {//GEN-  
FIRST:event_jTable_Display_UsersMouseClick  
ed
```

```
// TODO add your handling code here:

    int i =
jTable_Display_Users.getSelectedRow();

    TableModel model =
jTable_Display_Users.getModel();

jTextField_ID.setText(model.getValueAt(i,0).t
oString());

jTextField_EmailID.setText(model.getValueAt
(i,1).toString());

jTextField_FirstName.setText(model.getValue
At(i,2).toString());
```

```
textField_LastName.setText(model.getValue  
At(i,3).toString());
```

```
}//GEN-
```

```
LAST:event_jTable_Display_UsersMouseClick  
ed
```

```
private void
```

```
jButton2MouseClicked(java.awt.event.Mouse  
Event evt) {//GEN-
```

```
FIRST:event_jButton2MouseClicked
```

```
String query = "UPDATE `Customer` SET  
`EmailId`='"+textField_EmailID.getText().toSt  
ring()+"',`FirstName`='"+textField_FirstName  
.getText()+"',`LastName`='"+textField_LastN  
ame.getText()+"' WHERE `ID` =  
"+textField_ID.getText();
```

```
executeSQLQuery(query, "Updated");  
// TODO add your handling code here:
```

```
}//GEN-LAST:event_jButton2MouseClicked
```

```

private void
jButton3MouseClicked(java.awt.event.Mouse
Event evt) { //GEN-
FIRST:event_jButton3MouseClicked

String query = "DELETE FROM `Customer`
WHERE ID = "+jTextField_ID.getText();

        executeSQLQuery(query, "Deleted");    //
TODO add your handling code here:

} //GEN-LAST:event_jButton3MouseClicked

/**
 * @param args the command line
arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed"
desc=" Look and feel setting code (optional)
">

    /* If Nimbus (introduced in Java SE 6) is not

```



available, stay with the default look and feel.

\* For details see

<http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html>

\*/

try {

for

(javax.swing.UIManager.LookAndFeelInfo

info :

javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.  
getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(CustomerD

```
B.class.getName()).log(java.util.logging.Level.  
SEVERE, null, ex);
```

```
    } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(CustomerD  
B.class.getName()).log(java.util.logging.Level.  
SEVERE, null, ex);
```

```
    } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(CustomerD  
B.class.getName()).log(java.util.logging.Level.  
SEVERE, null, ex);
```

```
    } catch
```

```
(javax.swing.UnsupportedLookAndFeelExcept  
ion ex) {
```

```
java.util.logging.Logger.getLogger(CustomerD  
B.class.getName()).log(java.util.logging.Level.  
SEVERE, null, ex);
```

```
}
```

```
//</editor-fold>
```

```
//</editor-fold>
```

```
//</editor-fold>
```

```
//</editor-fold>
```

```
//</editor-fold>
```

```
//</editor-fold>
```

```
//</editor-fold>
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

```
java.awt.EventQueue.invokeLater(new  
Runnable() {  
    public void run() {  
        new CustomerDB().setVisible(true);  
    }  
});  
}
```

```
// Variables declaration - do not  
modify//GEN-BEGIN:variables
```

```
private javax.swing.JButton jButton1;
```

```
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable
jTable_Display_Users;
private javax.swing.JTextField
jTextField_EmailID;
private javax.swing.JTextField
jTextField_FirstName;
private javax.swing.JTextField jTextField_ID;
private javax.swing.JTextField
jTextField_LastName;
// End of variables declaration//GEN-
END:variables
}

package managecustomerdata.specifications;

/*
 * To change this license header, choose
 * License Headers in Project Properties.
```

- \* To change this template file, choose Tools | Templates

- \* and open the template in the editor.

- \*/

/\*\*

\*

\* @author Manideep

\*/

//Java class designed for getter methods used  
in Designed Frame

```
public class Customer {  
    private int ID;  
    private String emailid;  
    private String firstname;  
    private String lastname;
```

//a class named Customer that stores data  
for the user's id, email address, first name,  
and last name

```
public Customer( int ID,String emailid,String  
firstname,String lastname){
```

```
    this.ID=ID;
```

```
    this.emailid=emailid;
```

```
    this.firstname=firstname;
```

```
    this.lastname=lastname;
```

```
}
```

```
/**
```

```
 * @return the ID
```

```
 */
```

```
public int getID() {  
    return ID;  
}
```

```
/**  
 * @return the emailid  
 */  
public String getEmailid() {  
    return emailid;  
}
```

```
/**  
 * @return the firstname  
 */  
public String getFirstname() {  
    return firstname;  
}
```

```
/**
```

```
* @return the lastname
*/
public String getLastName() {
    return lastname;
}
```

```
/**
 * @return the PayeeOrSource
 */
```

```
}
```

```
/*
```

- \* To change this license header, choose License Headers in Project Properties.
- \* To change this template file, choose Tools | Templates



\* and open the template in the editor.

\*/

package managecustomerdata.specifications;

/\*\*

\*

\* @author Manideep

\*/

//a CustomerForm class that allows the user to  
add or edit customer data.

public class CustomerForm {

/\*\*

\* @param args the command line  
arguments

\*/

public static void main(String[] args) {

```
// TODO code application logic here

    java.awt.EventQueue.invokeLater(new
Runnable() {

        public void run() {

            new CustomerDB().setVisible(true);

        }

    });

}
```

***d: Source code for Tortoise and the Hare Race.***

```
import java.util.Random;
```

```
public class Racer implements Runnable {
```

```
public static String h;
```

```
public void race() {
```

```
    //Random numbber generated to tell the  
    Hare when to sleep
```

```
    Random rand = new Random();
```

```
    //Here a random integer is generated form  
    1 to 1111
```

```
    //This is to satisfy the specification of 90%  
    sleep periof of Hare
```

```
    //from 1-1111 there is a 90% probability  
    that a random nubmer is generated from 1-  
    1000
```

```
    //Hence Satisfying the requirement
```

```
    final int HARES_Lazy = rand.nextInt(1111) +
```

1;

```
    for (int distance = 0; distance <= 1000;  
distance+=10) {
```

```
System.out.println(Thread.currentThread().g  
etName() + " : " + distance + " meters.");
```

```
    //Checking if hare needs to sleep
```

```
    if (distance == HARES_Lazy &&  
Thread.currentThread().getName().equals("H  
are")) {
```

```
        //The Hare decides to take a nap  
        midway through since he is so far ahead
```

```
        try {
```

```
System.out.println(Thread.currentThread().g  
etName() + " is sleeeeeping ZZZZZZZZZZ");
```

```
        Thread.sleep(100);
```

```
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

    }

    //Check if race is finished
    boolean isRaceWon =
this.isRaceWon(distance);

    if (isRaceWon)
        break;

    }

}

private boolean isRaceWon(int totalDistance)
```

```
{
```

```
    boolean isRaceWon = false;
```

```
    if ((Racer.h == null) && (totalDistance ==  
1000)) {
```

```
        String winnerName =  
Thread.currentThread().getName();
```

```
        Racer.h = winnerName;
```

```
        System.out.println(Racer.h+": I  
finished");
```

```
        System.out.println("The race is over.!  
Winner is: " + Racer.h);
```

```
        isRaceWon = true;
```

```
    } else if (Racer.h == null) {
```

```
isRaceWon = false;
```

```
} else if (Racer.h != null) {
```

```
isRaceWon = true;
```

```
}
```

```
//one small limitation of my project is that  
the loop executes one last time
```

```
//after the race is over.This can be  
overcomed in future design.
```

```
return isRaceWon;
```

```
}
```

```
@Override
```

```
public void run() {
```

```
    this.race();  
  
}  
  
}
```

```
public class ThreadRunner {  
  
    public static void main(String[] args) {  
        //for(int i=0;i<3;i++){  
            Racer racer = new Racer();  
            //two thread objects are created  
            //this can be extended to 9 threads as  
            asked in enhancements  
            Thread tortoise = new Thread(racer,  
            "Tortoise");  
  
            Thread hare = new Thread(racer, "Hare");
```



```
//Starting the race
```

```
System.out.println("Get Set.... Go!");
```

```
hare.start();
```

```
tortoise.start();
```

```
// }
```

```
}
```

```
}
```