

# Advance Java FA1

---

SEPTEMBER 2

---

COMPANY NAME

Authored by: Albert Michael Ludick



Logo  
Name



---

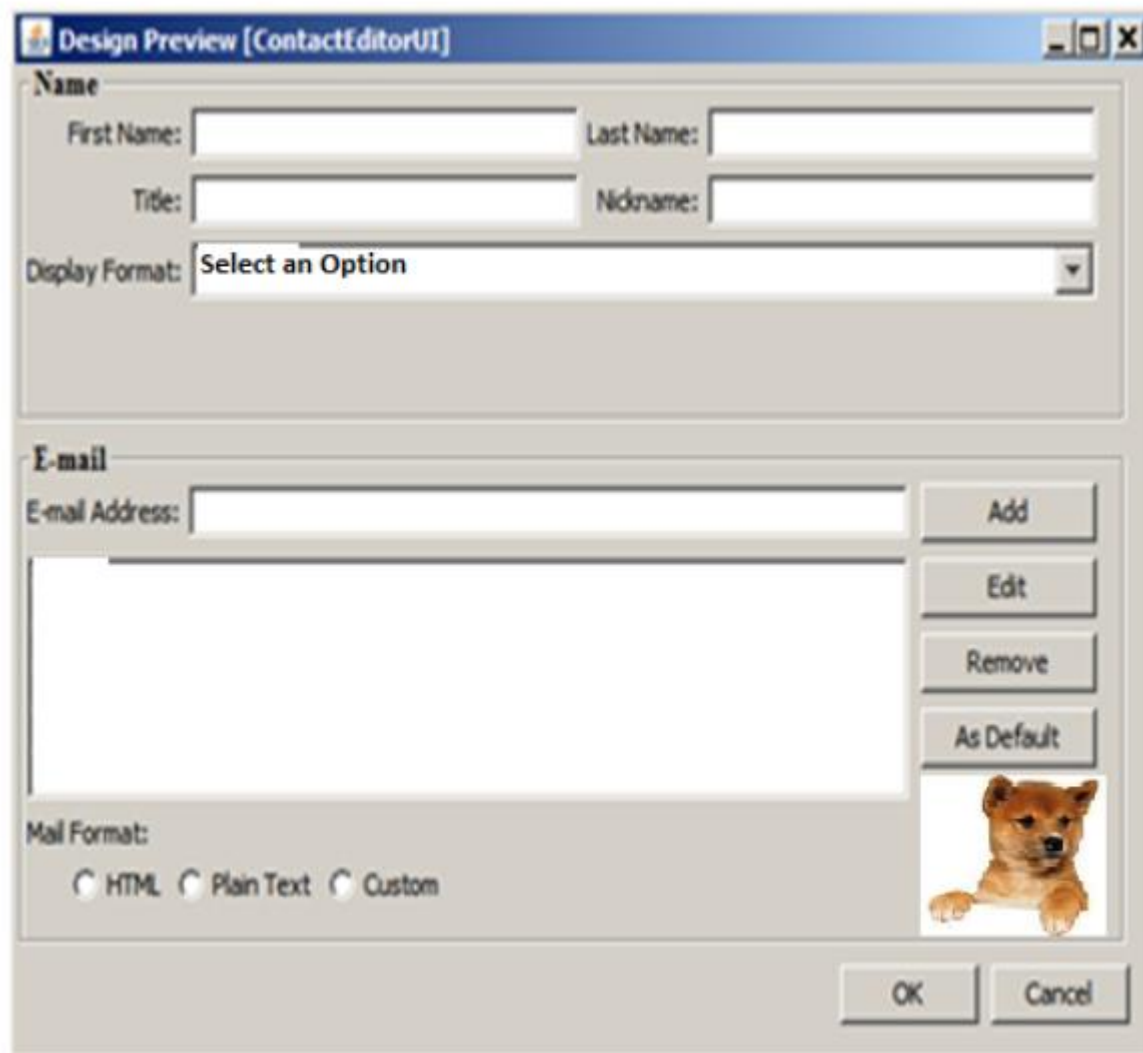
## Table of Contents

Intorduction .....	3
Question 2 .....	16
Question 2 Output .....	19
Question 2 Source Code.....	21
Conclution .....	25
Refences .....	26
Appendix .....	27

# Intorduction

## Question 1

Design the below interface using Java NetBeans ensure that the design looks exactly shown below. You can use any picture you want on the interface. To add title borders to the JPanels:



The image shows a 'Design Preview' window for a Java Swing application titled 'ContactEditorUI'. The interface is divided into two main sections: 'Name' and 'E-mail'.

**Name Section:**

- First Name:
- Last Name:
- Title:
- Nickname:
- Display Format:

**E-mail Section:**

- E-mail Address:
- 
- 
- 
- 
- Mail Format:
  - ☐ HTML
  - ☐ Plain Text
  - ☐ Custom

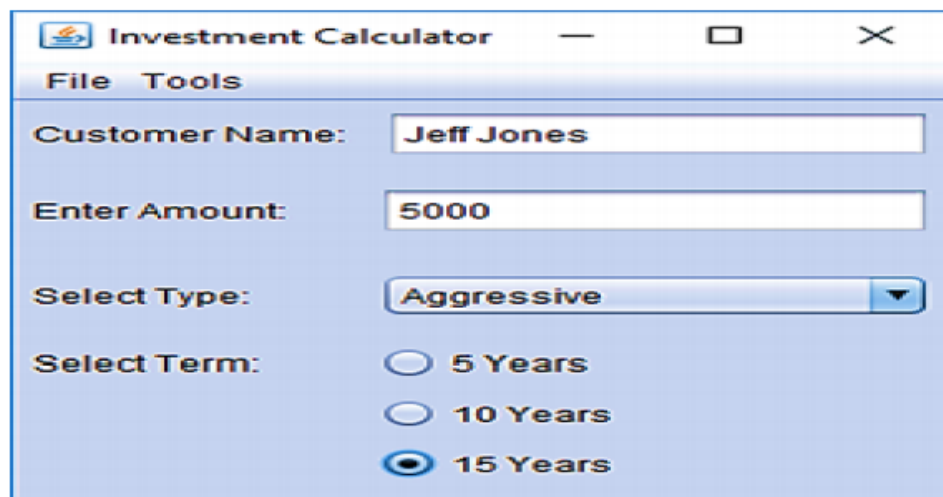
At the bottom right of the 'E-mail' section, there is a small image of a brown dog. At the very bottom of the window, there are 'OK' and 'Cancel' buttons.

## Question 2

[80]

Develop a Java GUI application that will produce an investment report based on various criteria, such as investment amount, investment type and term. On the form create two text fields, one to capture the customer name and another to capture the amount to invest. Also create a combo box for the user to select the investment type which will be moderate or aggressive. Finally add three radio buttons for the user to select the term which is five years, 10 years or 15 years.

You are also required to create a menu system which will allow the user to exit the application under the file menu and calculate and clear under the tools menu. Marks will be allocated to the presentation and effectiveness of the layout.



The screenshot shows a Java Swing window titled "Investment Calculator". It has a menu bar with "File" and "Tools". The main area contains four input controls: a text field for "Customer Name" with the value "Jeff Jones", a text field for "Enter Amount" with the value "5000", a dropdown menu for "Select Type" with "Aggressive" selected, and three radio buttons for "Select Term": "5 Years", "10 Years", and "15 Years". The "15 Years" radio button is selected.

The application must calculate the compound interest of the amount entered at the required investment type. Moderate investments receive 10% interest, while the aggressive receives 15% interest per year for the selected term. A sample output based on the selection above would be:

## Question 1

Properties

background	[240,240,240]	...
border	(No Border)	...
foreground	[0,0,0]	...

Click, or select and press CTRL-SPACE to open custom editor, right click for me

Other Properties

jPanel1 [JPanel] - border

Set **jPanel1's border** property using: Border customizer

Available Borders

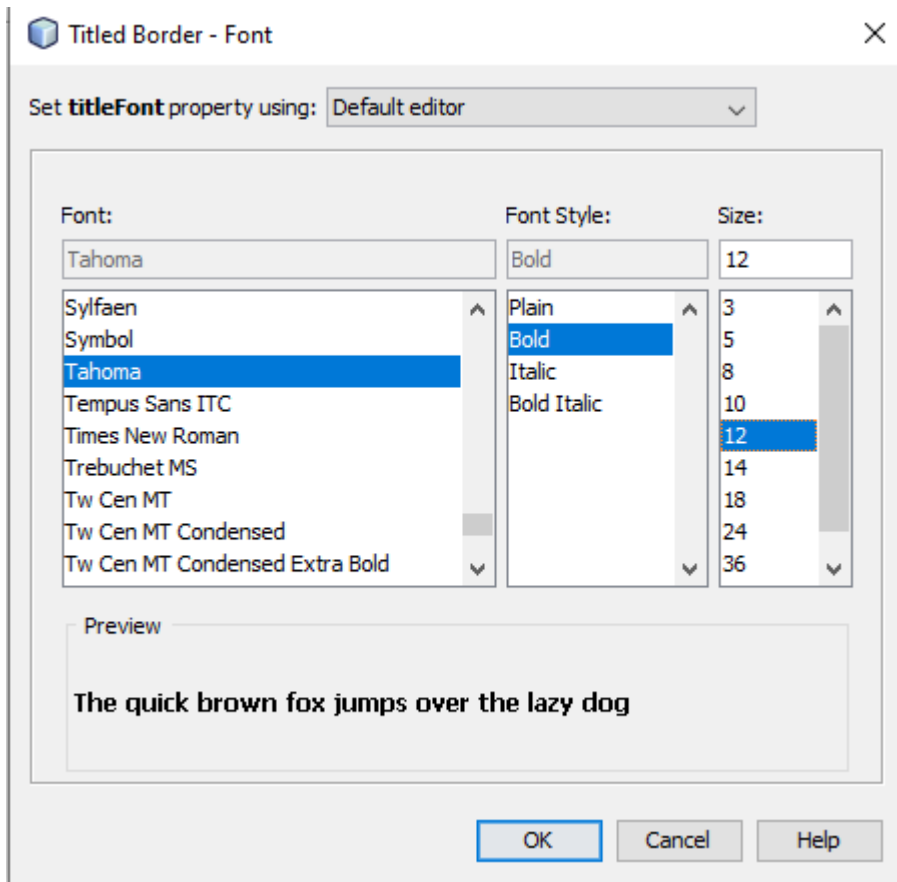
- (No Border)
- Bevel Border
- Compound Border
- Empty Border
- Etched Border
- Line Border
- Matte Border
- Soft Bevel Border
- Titled Border**

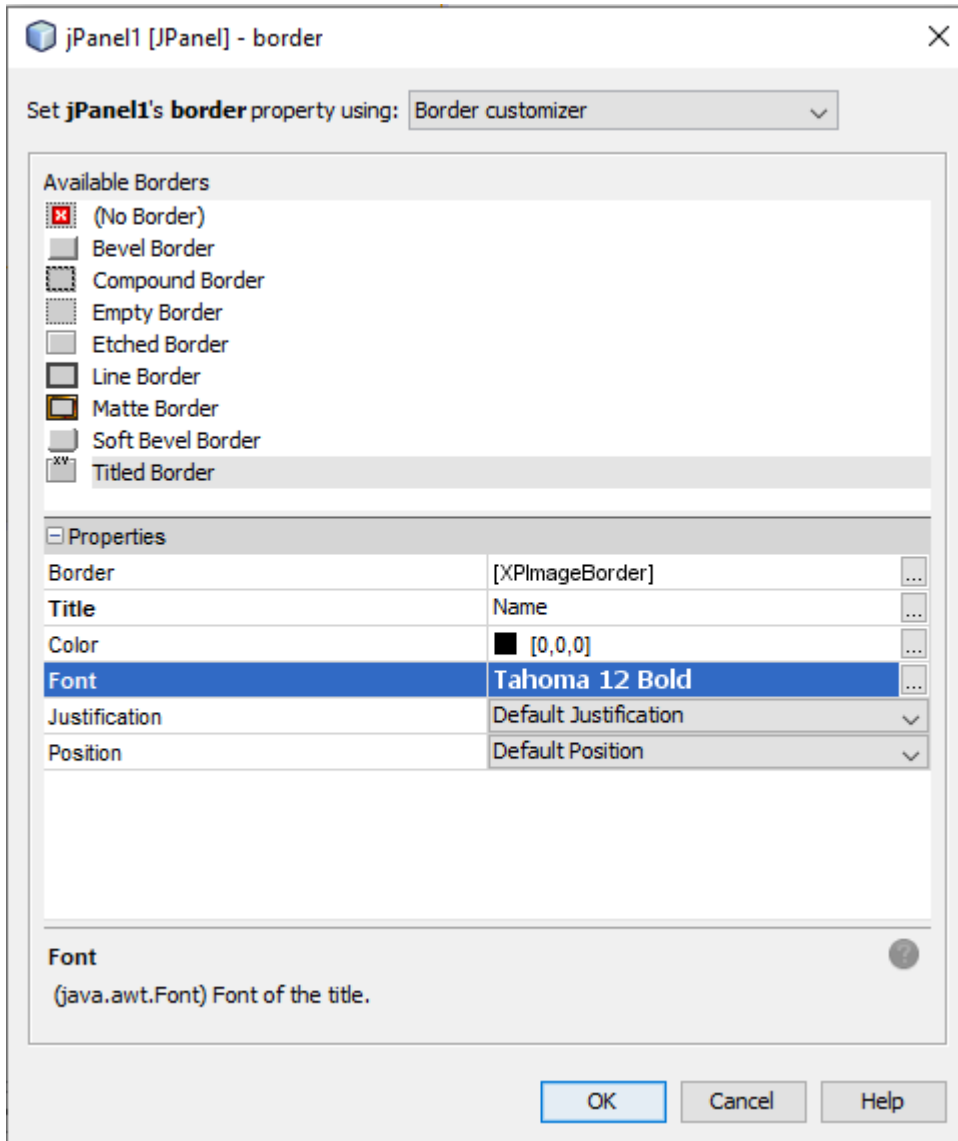
Properties

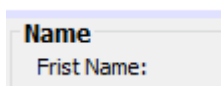
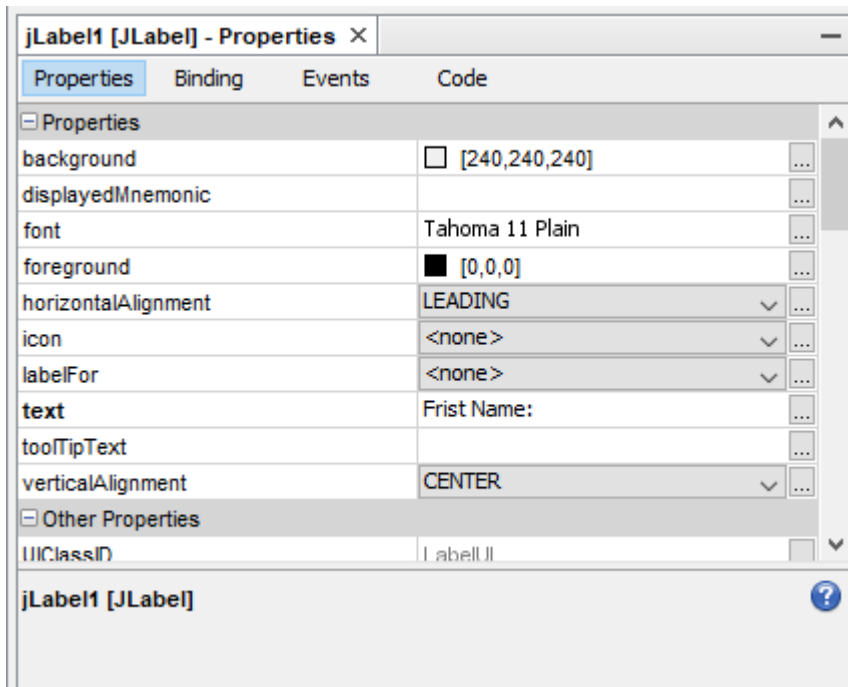
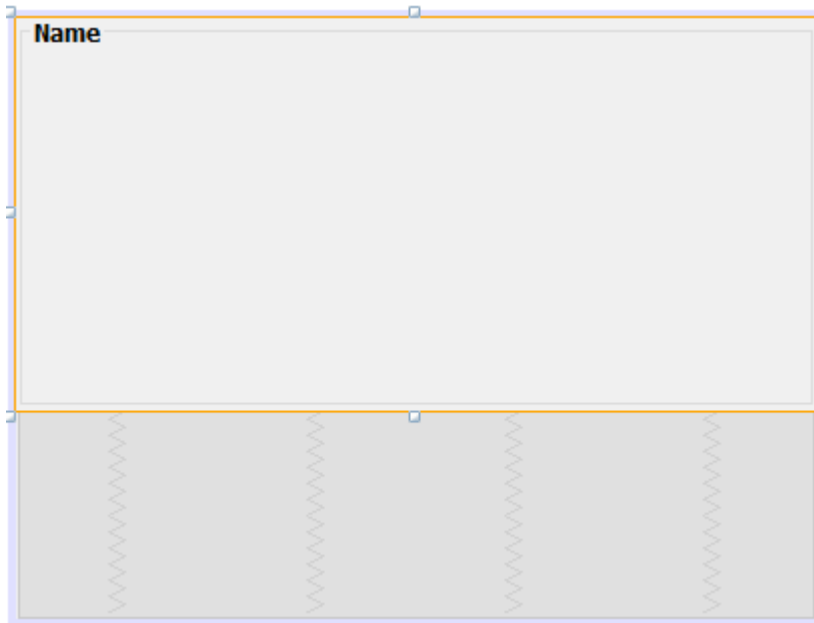
Border	[XpImageBorder]	...
Title		...
Color	[0,0,0]	...
Font	Tahoma 11 Plain	...
Justification	Default Justification	...
Position	Default Position	...

**Titled Border**  
javax.swing.border.TitledBorder  
A border with a String title in a specified position and justification.

OK Cancel Help









**Name**

Frist Name:  Last Name:

Title:  NickName:

Display Format:

jPanel2 [JPanel] - border

Set jPanel2's **border** property using:

**Available Borders**

- ☒ (No Border)
- ☐ Bevel Border
- ☐ Compound Border
- ☐ Empty Border
- ☐ Etched Border
- ☐ Line Border
- ☒ Matte Border
- ☐ Soft Bevel Border
- ☐ Titled Border


**Properties**

Border	[XpImageBorder]	...
<b>Title</b>	E-mail	...
Color	<input type="color" value="#000000"/> [0,0,0]	...
<b>Font</b>	<b>Tahoma 12 Bold</b>	...
Justification	Default Justification	▼
Position	Default Position	▼

**Title**

(java.lang.String) Title text

OK Reset to Default Cancel Help

 Design Preview [ContactEditorUI] — □ ×

**Name**  
Frist Name:  Last Name:   
Title:  NickName:   
Display Format: Select An Option ▼

**E-mail**  
E-mail Address:  Add  

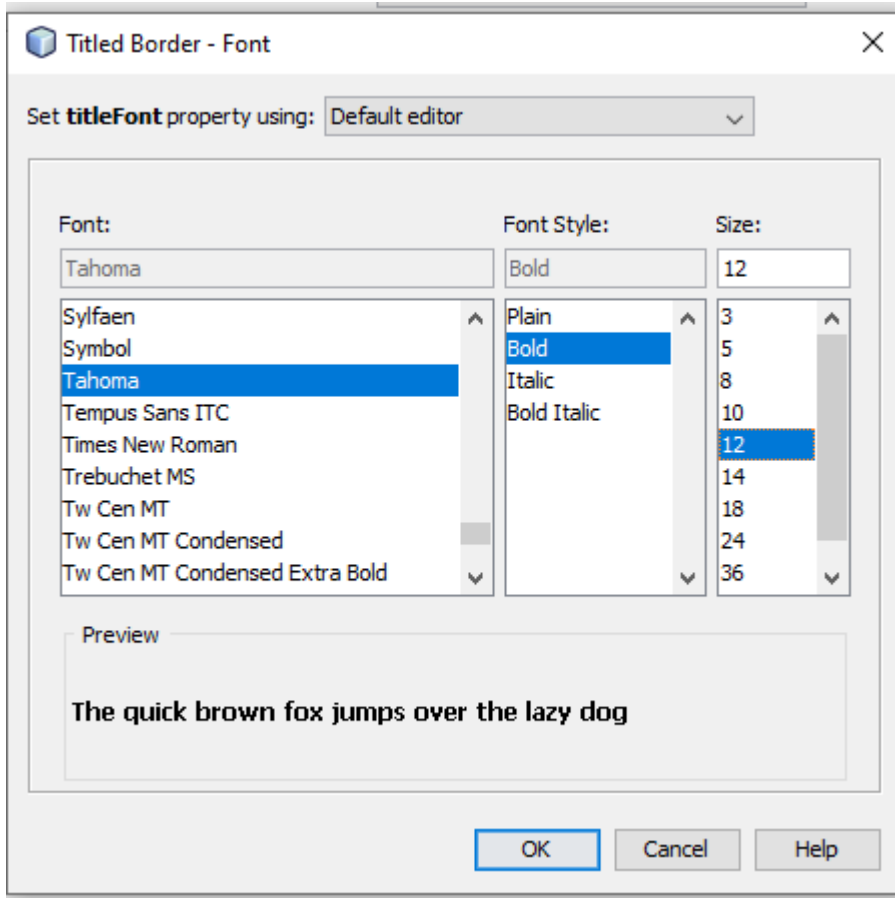
Edit

Remove

As Default

  
Mail Formats:  
☐ HTML ☐ Plain Text ☐ Custom

OK Cancel



Name

Frist Name:

Last Name:

Title:

NickName:

Display Format:

Select An Option

E-mail

E-mail Address:

Add

Edit

Remove

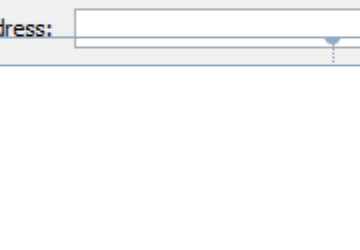
As Default

Mail Formats:

☐HTML

☐Plain Text


☐Custom



OK

Cancel



 Design Preview [ContactEditorUI] — □ ×

**Name**  
Frist Name:  Last Name:   
Title:  NickName:   
Display Format: Select An Option ▼

**E-mail**  
E-mail Address:   


Add

Edit

Remove

As Default


Mail Formats:  
☐ HTML ☐ Plain Text ☐ Custom



OK

Cancel

14

 Design Preview [ContactEditorUI] — □ ×

**Name**  
Frist Name:  Last Name:   
Title:  NickName:   
Display Format: Select An Option ▼


**E-mail**  
E-mail Address:   

Add

Edit

Remove

As Default



Mail Formats:  
☐ HTML ☐ Plain Text ☐ Custom

OK

Cancel

15

# Question 2

## GUI Design

Edit

Customer Name:

Enter Amount:

Select Type: Aggressive

Select term:

☒ 5 years

☐ 10 years

☐ 15 years

Design Preview [InvestmentCalculator] — □ ×

File Edit

Customer Name:

Enter Amount:

Select Type: Aggressive

Select term:

☒ 5 years

☐ 10 years

☐ 15 years



Design Preview [InvestmentCalculator]

File Edit

Customer Name:

Enter Amount:

Select Type: 

Aggressive ☐

Aggressive ☒

Moderate ☐

Select term: ☒ 5 years  
☐ 10 years  
☐ 15 years

Design Preview [InvestmentCalculator]

File Edit

Customer Name:

Enter Amount:

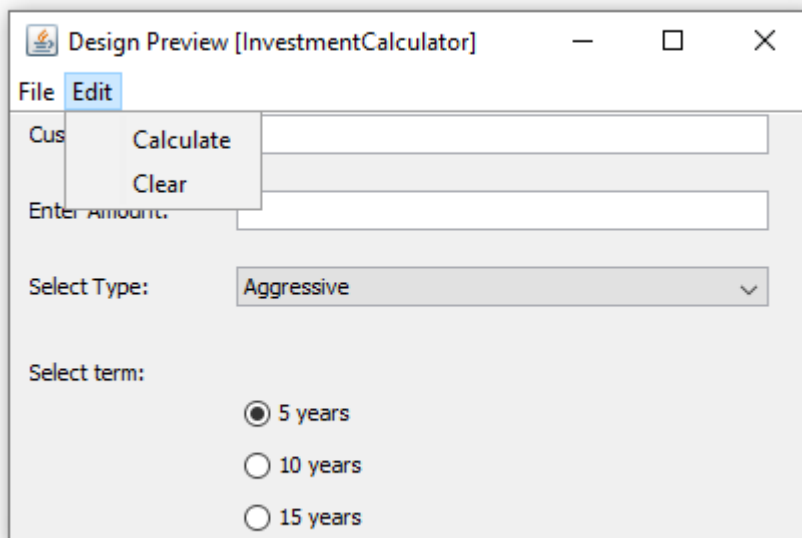
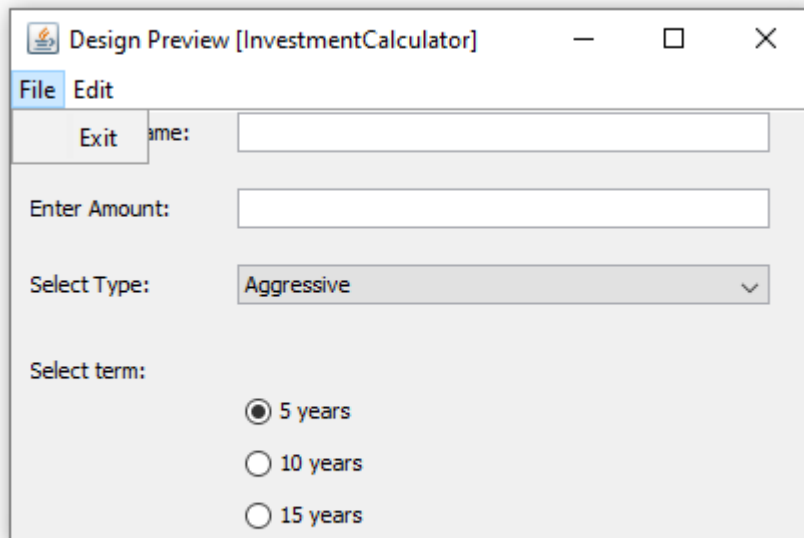
Select Type: 

Aggressive ☐

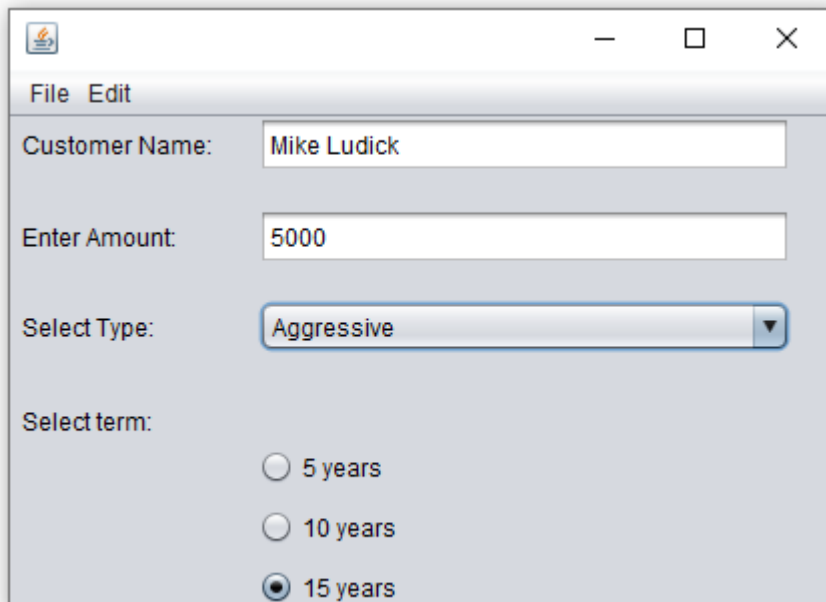
Aggressive ☒

Moderate ☐

Select term: ☐ 5 years  
☐ 10 years  
☒ 15 years



## Question 2 Output



A screenshot of a software application window. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. Below the title bar is a menu bar with 'File' and 'Edit' options. The main area contains four input fields: 'Customer Name' with the text 'Mike Ludick', 'Enter Amount' with the value '5000', 'Select Type' with a dropdown menu showing 'Aggressive', and 'Select term' with three radio button options: '5 years', '10 years', and '15 years'. The '15 years' option is selected.

File Edit

Customer Name: Mike Ludick

Enter Amount: 5000

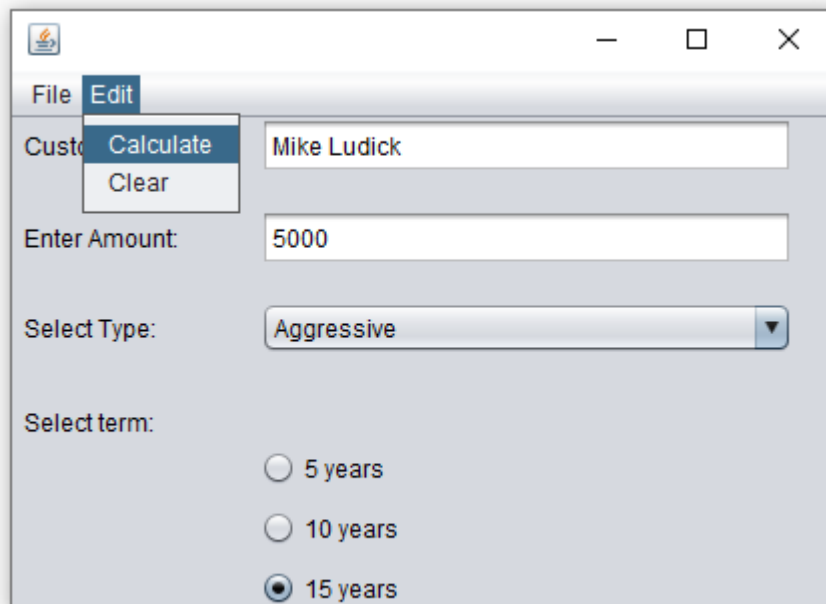
Select Type: Aggressive

Select term:

☐ 5 years

☐ 10 years

☒ 15 years



A screenshot of the same software application window, but with the 'Edit' menu open. The menu shows two options: 'Calculate' and 'Clear'. The 'Calculate' option is highlighted. The input fields remain the same as in the previous screenshot: 'Customer Name' (Mike Ludick), 'Enter Amount' (5000), 'Select Type' (Aggressive), and 'Select term' (15 years selected).

File Edit

Calculate

Clear

Customer Name: Mike Ludick

Enter Amount: 5000

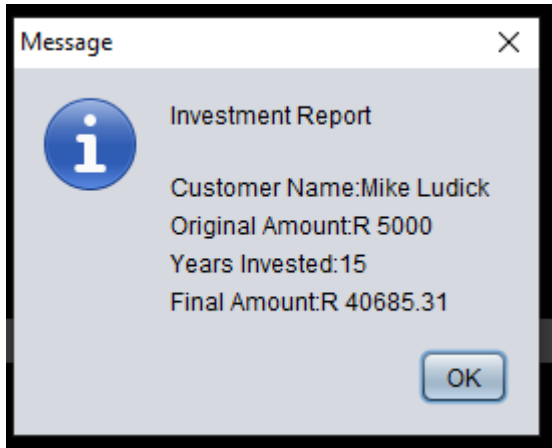
Select Type: Aggressive

Select term:

☐ 5 years

☐ 10 years

☒ 15 years



## Question 2 Source Code

Generated Code

```
/*
 * this code here cleans all and reset all the controls values
 * to the default values
 */
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    txtName.setText("");
    txtAmount.setText("");
    rb5.setSelected(true);
    rb10.setSelected(false);
    rb15.setSelected(false);
    cmbType.setSelectedIndex(0);
}
```

```
/*
 * this controls exits out of the application
 */
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
```

```
/*
 * this calls the Calculation method for the compound financial
 * problem presented in the app
 */
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        FinancialCalculator();
    }
    catch(Exception ex){
        JOptionPane.showMessageDialog(null, ex, "ErrorMessage"
            , JOptionPane.ERROR_MESSAGE);
    }
}
```

```
/*
these methods is just checking if other radio button is
selected that it should unselect them
*/
private void rb5ActionPerformed(java.awt.event.ActionEvent evt) {
    if(rb10.isSelected()){
        rb10.setSelected(false);
    }
    if(rb15.isSelected()){
        rb15.setSelected(false);
    }
}
```

```
/*
these methods is just checking if other radio button is
selected that it should unselect them
*/
private void rb15ActionPerformed(java.awt.event.ActionEvent evt) {
    if(rb10.isSelected()){
        rb10.setSelected(false);
    }
    if(rb5.isSelected()){
        rb5.setSelected(false);
    }
}
```

```
/*
these methods is just checking if other radio button is
selected that it should unselect them
*/
private void rb10ActionPerformed(java.awt.event.ActionEvent evt) {
    if(rb15.isSelected()){
        rb15.setSelected(false);
    }
    if(rb5.isSelected()){
        rb5.setSelected(false);
    }
}
```

```

/**
 * this will work out the compound mathematical expression of
 *  $A=p(1+r)^t$  modified according to the App needs and requirements
 * becuase the original is  $A=P(1=r/n)^{nt}$ 
 * @throws Exception
 */
private void FinancialCalculator() throws Exception{
    double finalAmount;
    double initialBalance;
    double interrestRate=0;
    int years;
    if(rb5.isSelected()){
        years=5;
    }
    else if(rb10.isSelected()){
        years=10;
    }
    else if(rb15.isSelected()){
        years=15;
    }
    else{
        throw new Exception("No Selected Years value! _code Line 191");
    }
    initialBalance = Double.parseDouble(txtAmount.getText());
    if(cmbType.getSelectedIndex()==0){
        interrestRate =0.15;
    }
    else if(cmbType.getSelectedIndex()==1){
        interrestRate =0.10;
    }
}

```

```

252         int years;
253         if(rb5.isSelected()){
254             years=5;
255         }
256         else if(rb10.isSelected()){
257             years=10;
258         }
259         else if(rb15.isSelected()){
260             years=15;
261         }
262         else{
263             throw new Exception("No Selected Years value! _code Line 191");
264         }
265         initialBalance = Double.parseDouble(txtAmount.getText());
266         if(cmbType.getSelectedIndex()==0){
267             interestRate =0.15;
268         }
269         else if(cmbType.getSelectedIndex()==1){
270             interestRate =0.10;
271         }
272         finalAmount= initialBalance*(Math.pow(1+interestRate, years));
273         PrintToScreen(finalAmount,years);
274     }
275     /**
276     * this method will print the data to screen in a showmessage daialog
277     * in a nicely formatted way
278     * @param finalAmount
279     * @param numberOfYears
280     */

```

```

/**
 * this method will print the data to screen in a showmessage daialog
 * in a nicely formatted way
 * @param finalAmount
 * @param numberOfYears
 */
private void PrintToScreen(double finalAmount,int numberOfYears){
    String message = "Investment Report"+"\\n"+"\\n"+"Customer Name:"
        +txtName.getText()+"\\n"+"Original Amount:R "
        +txtAmount.getText()+"\\n"+"Years Invested:"
        +numberOfYears+"\\n"+"Final Amount:R "
        +String.format("%.2f", finalAmount);
    JOptionPane.showMessageDialog(null,message);
}

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

```



---

# Conclusion

The math app works! And

The first question and my frame is twins!

---

# References

Websites: <https://www.geeksforgeeks.org/>

# Appendix



## Completed Declaration of Authenticity

I Albert Michael Ludick hereby  
(FULL NAME)

declare that the contents of this assignment Advance Java FA 1 is entirely my own work except for the following documents: (List the documents and page numbers of work in this portfolio that were generated in a group)

Activity	Date
Make both frames look the same	2020/09/01
And code for both if needed	2020/09/01

Signature: Albert Ludick Date: 2020/09/01

