Module Code & Module Title

CS4001NI Advanced Database Systems Development

Assessment Weightage & Type

25% Individual Coursework / 50% Group Coursework

Year and Semester

2018-19 Autumn / 2018-19 Spring

Student Name: Ayush Amatya

London Met ID: 18029908

College ID: NP01CP4A180129

Assignment Due Date: 19 April 2019

Assignment Submission Date: 19 April 2019

Word Count (Where Required):

Contents

Introduction:

This project report was done to add a class to the project of coursework 1 to make a graphical user interface (GUI) for a system that stores details of developers. This coursework was given to us in 20th week and was required to be submitted by 24th week.

This coursework was all about writing program in java. Java is one of the most popular programming languages. Many users choose java for developing different program as it is very versatile and compatible in nature. Java can be used for a large number of things including software development, mobile applications, and large systems development. Writing, compiling and debugging a program is easy in java. It helps to create modular programs and reusable code.

The java program developed in this coursework consists of main user interface GUI where the user performs all the activities. Here, the class RigoTechnology acts like a bridge to all the other classes. Staying in RigoTechnology class, user can use and interact with all the function of all the other classes.

We can perform all the activities of developer, seniorDevloper and juniorDeveloper staying in RigoTechnology class including adding a platform, hiring developer, terminating contract of developer and many other.

Class Diagram

RigoTechnology	
+ list: ArrayList<Developer>	+ myFrame: JFrame
+ lblAdvanceSalary: JLabel	+ lblSeniorDeveloper: JLabel
+ lblStaffRoomNumber: JLabel	+ lblPlatformS: JLabel
+ lblSelectPlatformS: JLabel	+ lblInterviewerNameS: JLabel
+ lblEndLine: JLabel	+ lblWorkingHoursS: JLabel
+ lblJuniorDeveloper: JLabel	+ lblDotLineSen: JLabel
+ lblPlatformJr: JLabel	+ lblSalaryS: JLabel
+ lblInterviewerNameJ: JLabel	+ lblContractPeriod: JLabel
+ lblSalaryJ: JLabel	+ lblDeveloperNameS: JLabel
+ lblWorkingHoursJ: JLabel	+ lblJoiningDate: JLabel
+ lblAppointedBy: JLabel	+ lblDotLineJun: JLabel
+ lblTerminationDate1: JLabel	+ lblDeveloperNameJ: JLabel
+ lblSelectPlatformJ: JLabel	+ lblAppointedDate: JLabel
+ lblEvaluationPeriod: JLabel	+ lblSpecialization: JLabel
+ lblTerminationDate2: JLabel	+ txtStaffRoomNumber: JTextField
+ txtPlatfomS: JTextField	+ txtPlatformJ: JTextField
+ txtInterviewerNameS: JTextField	+ txtInterviewerNameJ: JTextField
+ txtWorkingHoursS: JTextField	+ txtSalaryJ: JTextField
+ txtSalaryS: JTextField	+ txtWorkingHoursJ: JTextField
+ txtContractPeriod: JTextField	+ txtAppointedBy: JTextField
+ txtDeveloperNameS: JTextField	+ txtTerminationDate1: JTextField
+ txtJoiningDate: JTextField	+ txtTerminatinDate2: JTextField
+ txtAdvanceSalary: JTextField	+ txtDeveloperNameJ: JTextField
+ txtEvaluationPeriod: JTextField	+ txtAppointedDate: JTextField
+ btnAddSen: JButton	+ txtSpecialization: JTextField
+ btnAppoint: JButton	+ btnHire: JButton
+ btnDisplayAll: JButton	+ btnTerminate: JButton
+ btnClear: JButton	+ btnAddJun: JButton
+cmbSelectPlatformJ: JComboBox	+ cmbSelectPlatformS: JComboBox
+ gui: void	+ addJun: void
+ actionPerformed: void	+ hireSen: void
+ addSen: void	+ terminateSen: void
+ clear: void	+ appointJun: void
+ main(String [ ]args): static void	+ displayAll: void

Pseudocode

Following are the pseudocode for all the method used in RigoTechnology class:

1. gui():

```
Create a public void method: gui()
Do:
   Create object of JFrame: myFrame;
   Declare size of myFrame;

   Create object of JLabel: lblSeniorDeveloper (“Senior Developer”);
   Set bound for lblSeniorDeveloper;
   Add lblSeniorDeveloper to myFrame;

   Create object of JLabel: lblPlatformS ("Platform:");
   Set bound for lblPlatformS;
   Add lblPlatformS to myFrame;

   Create object of JTextField: txtPlatformS;
   Set bound for txtPlatformS;
   Add txtPlatformS to myFrame;

   Create object of JLabel: lblInterviewerNameS ("Interviewer Name:");
   Set bound for lblInterviewerNameS;
   Add lblInterviewerNameS to myFrame;

   Create object of JTextField: txtInterviewerNameS;
   Set bounds for txtInterviewerNameS;
   Add txtInterviewerNameS to myFrame;

   Create object of JLabel: lblWorkingHoursS ("Working Hours:");
   Set bound for lblWorkingHoursS;
   Add lblWorkingHoursS to myFrame;
```

```
Create object of JTextField: txtWorkingHoursS;
Set bounds for txtWorkingHoursS;
Add txtWorkingHoursS to myFrame;

Create object of JLabel:  lblSalaryS (“Salary:");
Set bound for lblSalaryS;
Add lblSalaryS to myFrame;

Create object of JTextField: txtSalaryS;
Set bounds for txtSalaryS;
Add txtSalaryS to myFrame;

Create object of JLabel: lblContractPeriod ("Contract Period:");
Set bound for lblContractPeriod;
Add lblContractPeriod to myFrame;

Create object of JTextField: txtContractPeriod;
Set bounds for txtContractPeriod;
Add txtContractPeriod to myFrame;

Create object of JButton: btnAddSen ("Add");
Set bounds for btnAddSen;
Add btnAddSen to myFrame;

Create object of JLabel: lblDotLineSen ("- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -");
Set bound for lblDotLineSen;
Add lblDotLineSen to myFrame;

Create object of JLabel: lblDeveloperNameS ("Developer Name:");
Set bound for lblDeveloperNameS;
Add lblDeveloperNameS to myFrame;
```

```
Create object of JTextField: txtDeveloperNameS;
Set bounds for txtDeveloperNameS;
Add txtDeveloperNameS to myFrame;

Create object of JLabel: lblJoiningDate ("Joining Date:");
Set bound for lblJoiningDate;
Add lblJoiningDate to myFrame;

Create object of JTextField: txtJoiningDate;
Set bounds for txtJoiningDate;
Add txtJoiningDate to myFrame;

Create object of JLabel: lblStaffRoomNumber ("Staff Room Number:");
Set bound for lblStaffRoomNumber;
Add lblStaffRoomNumber to myFrame;

Create object of JTextField: txtStaffRoomNumber;
Set bounds for txtStaffRoomNumber;
Add txtStaffRoomNumber to myFrame;

Create object of JLabel: lblAdvanceSalary ("Advance Salary:");
Set bound for lblAdvanceSalary;
Add lblAdvanceSalary to myFrame;

Create object of JTextField: txtAdvanceSalary;
Set bounds for txtAdvanceSalary;
Add txtAdvanceSalary to myFrame;

Create object of JLabel: lblSelectPlatformsS ("Select Platform:");
Set bound for lblSelectPlatformsS;
Add lblSelectPlatformsS to myFrame;

Create a list: selectPlatformS[] = {"Select platform", "No platform registered"};
Create object of JComboBox: cmbSelectPlatformS(selectPlatformS);
```

```
Set bound for cmbSelectPlatformS;
Add cmbSelectPlatformS to myFrame;

Create object of JButton: btnHire ("Hire");
Set bounds for btnHire;
Add btnHire to myFrame;

Create object of JButton: btnTerminate ("Terminate");
Set bounds for btnTerminate;
Add btnTerminate to myFrame;

Create object of JLabel: lblEndLine
("=======================================================
================================");
Set bound for lblEndLine;
Add lblEndLine to myFrame;

Create object of JLabel: lblJuniorDeveloper ("JUNIOR DEVELOPER");
Set bound for lblJuniorDeveloper;
Add lblJuniorDeveloper to myFrame;

Create object of JLabel: lblPlatformJ ("Platform:");
Set bound for lblPlatformJ;
Add lblPlatformJ to myFrame;

Create object of JTextField: txtPlatformJ;
Set bounds for txtPlatformJ;
Add txtPlatformJ to myFrame;

Create object of JLabel: lblInterviewerNameJ ("Interviewer Name:");
Set bound for lblInterviewerNameJ;
Add lblInterviewerNameJ to myFrame;

Create object of JTextField: txtInterviewerNameJ;
```

```
Set bounds for txtInterviewerNameJ;
Add txtInterviewerNameJ to myFrame;

Create object of JLabel: lblAppointedBy ("Appointed By:");
Set bound for lblAppointedBy;
Add lblAppointedBy to myFrame;

Create object of JTextField: txtAppointedBy;
Set bounds for txtAppointedBy;
Add txtAppointedBy to myFrame;

Create object of JLabel: lblWorkingHoursJ ("Working Hours:");
Set bound for lblWorkingHoursJ;
Add lblWorkingHoursJ to myFrame;

Create object of JTextField: txtWorkingHoursJ;
Set bounds for txtWorkingHoursJ;
Add txtWorkingHoursJ to myFrame;

Create object of JLabel: lblSalaryJ ("Salary:");
Set bound for lblSalaryJ;
Add lblSalaryJ to myFrame;

Create object of JTextField: txtSalaryJ;
Set bounds for txtSalaryJ;
Add txtSalaryJ to myFrame;

Create object of JLabel: lblTerminationDate1 ("Termination Date:");
Set bound for lblTerminationDate1;
Add lblTerminationDate1 to myFrame;

Create object of JTextField: txtTerminationDate1;
Set bounds for txtTerminationDate1;
Add txtTerminationDate1 to myFrame;
```

```
Create object of JButton: btnAddJun ("Add");
Set bounds for btnAddJun;
Add btnAddJun to myFrame;

Create object of JLabel: lblDotLineJun ("- - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -");
Set bound for lblDotLineJun;
Add lblDotLineJun myFrame;

Create object of JLabel: lblDeveloperNameJ ("Developer Name:");
Set bound for lblDeveloperNameJ;
Add lblDeveloperNameJ to myFrame;

Create object of JTextField: txtDeveloperNameJ;
Set bounds for txtDeveloperNameJ;
Add txtDeveloperNameJ to myFrame;

Create object of JLabel: lblAppointedDate ("Appointed Date:");
Set bound for lblAppointedDate;
Add lblAppointedDate to myFrame;

Create object of JTextField: txtAppointedDate;
Set bounds for txtAppointedDate;
Add txtAppointedDate to myFrame;

Create object of JLabel: lblEvaluationPeriod ("Evaluation Period:");
Set bound for lblEvaluationPeriod;
Add lblEvaluationPeriod to myFrame;

Create object of JTextField: txtEvaluationPeriod;
Set bounds for txtEvaluationPeriod;
Add txtEvaluationPeriod to myFrame;
```

```
Create object of JLabel: lblSpecialization ("Specialization:");
Set bound for lblSpecialization;
Add lblSpecialization to myFrame;

Create object of JTextField: txtSpecialization;
Set bounds for txtSpecialization;
Add txtSpecialization to myFrame;

Create object of JLabel: lblSelectPlatformJ ("Select Platform:");
Set bound for lblSelectPlatformJ;
Add lblSelectPlatformJ to myFrame;

Create a list: selectPlatformJ[] = {"Select platform", "No platform registered"};
Create object of JComboBox: cmbSelectPlatformJ (selectPlatformJ);
Set bound for cmbSelectPlatformJ;
Add cmbSelectPlatformJ to myFrame;

Create object of JButton: btnAppoint ("Appoint");
Set bounds for btnAppoint;
Add btnApppoint to myFrame;

Create object of JButton: btnDisplayAll ("Display All");
Set bounds for btnDisplayAll;
Add btnDisplay to myFrame;

Create object of JButton: btnClear ("Clear");
Set bounds for btnClear;
Add btnClear to myFrame;

Create object of JLabel: lblTerminationDate2 ("Termination Date:");
Set bound for lblTerminationDate2;
Add lblTerminationDate2 to myFrame;

Create object of JTextField: txtTerminationDate2;
```

```
        Set bounds for txtTerminationDate2;
        Add txtTerminationDate2 to myFrame;

        Call btnClear.addActionListener(this);
        Call btnAddSen.addActionListener(this);
        Call btnAddJun.addActionListener(this);
        Call btnHire.addActionListener(this);
        Call btnTerminate.addActionListener(this);
        Call btnAppoint.addActionListener(this);
        Call btnDisplayAll.addActionListener(this);
    End do
```

2. actionPerformed(ActionEvent e):

```
Create a public void method: actionPerformed(AvtionEvent e)
Do
        If (e.getSource() = btnClear) Do
                Call clear();
        End Do

        If (e.getSource() = btnAddSen) Do
                Call addSen();
        End Do

        If (e.getSource() = btnAddJun) Do
                Call addJun();
        End Do


        If (e.getSource() = btnHire) Do
                Call hireSen();
        End Do
        If (e.getSource() = btnTerminate) Do
            Call terminateSen();
        End Do

        If (e.getSource() = btnAppoint) Do
            Call appointJun();
        End Do

        If (e.getSource() = btnDisplayAll) Do
            Call displayAll();
        End Do
    End Do
```

3. addSen():

```
Create a public void method: addSen()
Do
  Try Do
    String platformS = txtPlatformS.getText();
    String interviewerNameS = txtInterviewerNameS.getText();
    String workingHoursS = txtWorkingHoursS.getText();
    String salaryS = txtSalaryS.getText();
    String contractPeriod = txtContractPeriod.getText();

       If (platformS.equals("") or interviewerNameS.equals("") or
       workingHoursS.equals("") or salaryS.equals("") or
       contractPeriod.equals("")) Do
         Print: "Please fill up all the required information");
       End DO
     Else Do
       int intWorkingHoursS = Integer.parseInt(workingHoursS);
       int intSalaryS = Integer.parseInt(salaryS);
```

```
            int intContractPeriod = Integer.parseInt(contractPeriod);

             SeniorDeveloper senDev = new SeniorDeveloper(platformS,
             interviewerNameS, intWorkingHoursS, intSalaryS, intContractPeriod);

             Add SenDev to list;

            If (cmbSelectPlatformS.getItemAt(1).equals("No platform registered")) Do
               Remove “No platform registered from cmbSelectPlatformS;
            End Do

            Add ("(id: " + i + ") " + platformS) to cmbSelectPlatformS;
            i=i+1;

            Print:  "The platform is sucessfully registered to SeniorDeveloper";
         End Do
   End Do
   Catch (Exception exp) Do
            Print: "Working hours, Salary and contract period must be a numeric
            value.";
   End Do
```

4. hireSen():

```
Create a public void method: hireSen()
Do
   Try Do
         String developerNameS = txtDeveloperNameS.getText();
         String staffRoomNumber = txtStaffRoomNumber.getText();
         String selectPlatformS = cmbSelectPlatformS.getSelectedItem().toString();
         String joiningDate = txtJoiningDate.getText();
         String advanceSalary = txtAdvanceSalary.getText();
```

```
      If (developerNameS.equals("") or staffRoomNumber.equals("") or
      selectPlatformS.equals("Select platform") or joiningDate.equals("") or
      advanceSalary.equals("")) Do
         Print: "Please fill up all the required information");
      End Do
      Else if (selectPlatformS.equals ("No platform registered")) Do
          Print: "There are no any platform registered");
      End Do
      Else Do
         float floatAdvanceSalary = Float.parseFloat(advanceSalary);

         int indexS = Character.getNumericValue(selectPlatformS.charAt(5));
         SeniorDeveloper senVar = (SeniorDeveloper)list.get(indexS);

         If (senVar.getAppointed() = true) Do
            Print: “The developer is already hired in this platform: "
         End Do
         Else Do
              Call senVar.hire(developerNameS, joiningDate, floatAdvanceSalary,
              staffRoomNumber);

              Print: "The developer: is successfully hired in platform: ";
         End Do
      End Do
   End Do
   Catch (Exception exp) Do
      Print: "The value of Advance salary should be float";
   End Do
```

5. TerminateSen():

Create a public void method: terminateSen()

```
Do
   String selectPlatformS = cmbSelectPlatformS.getSelectedItem().toString();
```

```
    If (selectPlatformS.equals("Select platform") or selectPlatformS.equals("No
    platform registered")) Do
        Print: "Please select the platform that you want to terminate";
    End Do
    Else Do
        int index = Character.getNumericValue(selectPlatformS.charAt(5));
        SeniorDeveloper senVar = (SeniorDeveloper)list.get(index);

        If (senVar.getTerminated() = true) Do
            Print: "The contract of developer of platform is already terminated";
        End Do
        Else Do
            Print: "The contract of developer of platform is sucessfully terminated";
            Call senVar.termination();
        End Do
    End Do
End Do
```

6. addJun():

```
Create a public void method: addJun()
Do
    Try Do
        String platformJ = txtPlatformJ.getText();
        String interviewerNameJ = txtInterviewerNameJ.getText();
        String workingHoursJ = txtWorkingHoursJ.getText();
        String salaryJ = txtSalaryJ.getText();
        String terminationDate1 = txtTerminationDate1.getText();
        String appointedBy = txtAppointedBy.getText();

        If (platformJ.equals("") or| interviewerNameJ.equals("") or
        workingHoursJ.equals("") or salaryJ.equals("") or
        terminationDate1.equals("") or appointedBy.equals("")) Do
            Print: "Please fill up all the required information");
        End Do

        Else Do
            int intWorkingHoursJ = Integer.parseInt(workingHoursJ);
            int intSalaryJ = Integer.parseInt(salaryJ);

            JuniorDeveloper junDev = new JuniorDeveloper(platformJ,
            interviewerNameJ, intWorkingHoursJ, intSalaryJ, appointedBy,
            terminationDate1);

            Add junDev to list;

            If (cmbSelectPlatformJ.getItemAt(1).equals("No platform registered")) Do
                Remove “No platform registered from cmbSelectPlatformJ;
            End Do
            Add ("(id: " + i + ") " + platformJ) to cmbSelectPlatformJ;
            i=i+1;

            Print: "The platform is sucessfully registered to Junior Developer";
         End Do
      End Do
      Catch (Exception exp) Do
          Print: "Working hours and Salary must be a numeric value.");
      End Do
   End Do
```

7. appointJun():

```
Create a public void method: appointJun()
Do
      String developerNameJ = txtDeveloperNameJ.getText();
      String selectPlatformJ = cmbSelectPlatformJ.getSelectedItem().toString();
      String specialization = txtSpecialization.getText();
      String evaluationPeriod = txtEvaluationPeriod.getText();
      String appointedDate = txtAppointedDate.getText();
      String terminationDate2 = txtTerminationDate2.getText();

      If (developerNameJ.equals("") or selectPlatformJ.equals("Select platform") or
      specialization.equals("") or evaluationPeriod.equals("") or
      appointedDate.equals("") or terminationDate2.equals("")) Do
          Print: "Please fill all the required information";
      End Do
      Else if (selectPlatformJ.equals("No platform registered")) Do
           Print: "There are no any platform registered";
      End Do
      Else Do
         int indexJ = Character.getNumericValue(selectPlatformJ.charAt(5));
         JuniorDeveloper junVar = (JuniorDeveloper)list.get(indexJ);
         If (junVar.getJoined() = true) Do
            Print: "The developer is already appointed in this platform";
         End Do
         Else Do
             Call junVar.appoint(developerNameJ, terminationDate2, appointedDate,
             specialization, evaluationPeriod);
             Print: "The developer is sucessfully appointed”;
         End Do
      End Do
   End Do
```

8. displayAll():

```
Create a public void method: displayAll()
Do
      For (Developer obj: list) Do
         If (obj is instance of SeniorDeveloper) Do
            Print: "Senior Developer:";
            Print: "id: " + Integer.toString(list.indexOf(obj));
            Call obj.display();
         End Do
         Else Do
            Print: "Junior Developer:";
            Print: "id: " + Integer.toString(list.indexOf(obj));
            Call obj.display();
         End Do
         Print blank line;
      End Do
   End Do
```

9. clear():

```
Do
   Set text of txtPlatformS as "";
   Set text of txtInterviewerNameS as "";
   Set text of txtWorkingHoursS as "";
   Set text of txtSalaryS as "";
   Set text of txtContractPeriod as "";
   Set text of txtDeveloperNameS as "";
   Set text of txtJoiningDate as "";
   Set item of cmbSelectPlatformS as "Select platform";
   Set text of txtAdvanceSalary as "";
   Set text of txtStaffRoomNumber as "";
   Set text of txtPlatformJ as "";
   Set text of txtInterviewerNameJ as "";
   Set text of txtSalaryJ as "";
   Set text of txtWorkingHoursJ as "";
   Set text of txtAppointedBy as "";
   Set text of txtTerminationDate1 as "";
   Set text of txtTerminationDate2 as "";
   Set text of txtDeveloperNameJ as "";
   Set text of txtAppointedDate as "";
   Set text of txtSpecialization as "";
   Set item of cmbSelectPlatformJ as "Select platform";
   Set text of txtEvaluationPeriod as "";
End DO
```

10. main(String [ ]args):

Create a public static void main(String []args) method:

```
Do
   Create an object of RigoTechnology() and call the method gui();
End Do
```

Method description:

To complete this program, many methods were used. The method used in this coursework are given below with its short description:

1. gui():

This is the method which determine all the deigns, layouts and contains of the frame created in this class. All the text box, label, buttons required in the frame are declared in this method. Then their bounds are declared and finally they are added to the frame. It accepts no parameters.

2. actionPerformed(ActionEvent e):

This is the method that determines the functions of all the buttons present in the frame. It decides the activities that is to be performed when the specific button is clicked by the user. It accepts (ActionEvent e) as the parameter. The source of e can be used to know which button is clicked.

3. addSen():

This is a sub method declared in this class. This method contains all the code that is to be executed if user clicks ADD button for seniorDeveloper. This method is latter called in actionPerformed method if the source of action event is btnAddSen.

4. hireSen():

This is a sub method declared in this class. This method contains all the code that is to be executed if user clicks HIRE button for seniorDeveloper. This method is latter called in actionPerformed method if the source of action event is btnHireSen.

5. terminateSen():

This is a sub method declared in this class. This method contains all the code that is to be executed if user clicks TERMINATE button for seniorDeveloper. This method is latter called in actionPerformed method if the source of action event is btnterminateSen.

6. addJun():

This is a sub method declared in this class. This method contains all the code that is to be executed if user clicks ADD button for juniorDeveloper. This method is latter called in actionPerformed method if the source of action event is btnAddJun.

7. appointJun():

This is a sub method declared in this class. This method contains all the code that is to be executed if user clicks APPOINT button for juniorDeveloper. This method is latter called in actionPerformed method if the source of action event is btnAppoint.

8. displayAll():

This is a sub method declared in this class. This method contains all the code that is to be executed if user clicks DISPLAY ALL button. This method is latter called in actionPerformed method if the source of action event is btnDisplayAll. It displays all the platforms and developer working of both senior and junior developer.

9. clear():

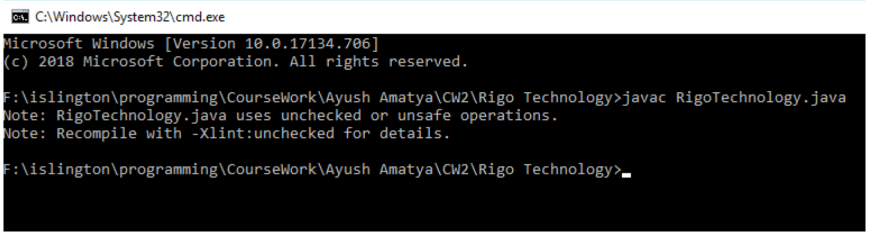
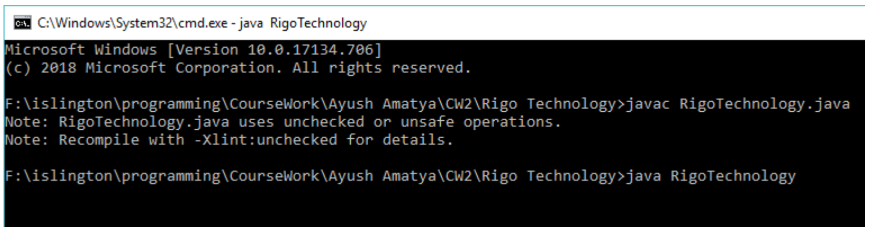
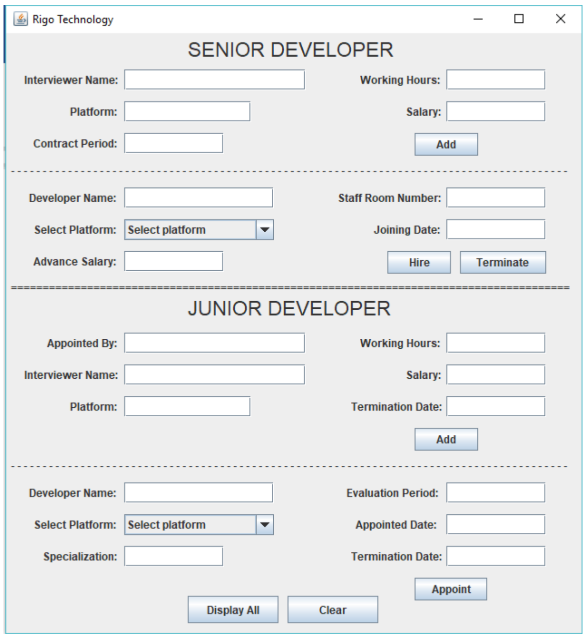
This is a sub method declared in this class. This method contains all the code that is to be executed if user clicks CLEAR button. This method is latter called in actionPerformed method if the source of action event is btnClear. This clears all the text boxes of the frame if there are any text written there by the user.

10. Main(String [ ] args):

This is the main method of this class. This is the only method that is directly called by the user in real time. Gui() method is called in this method. When user runs this method then the frame is opened in the window.

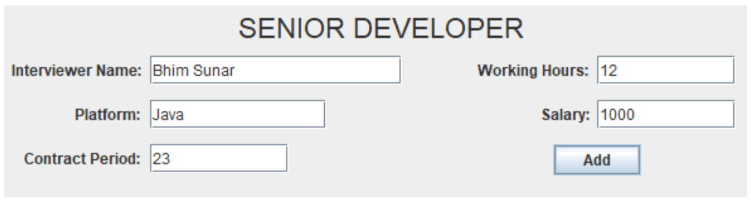
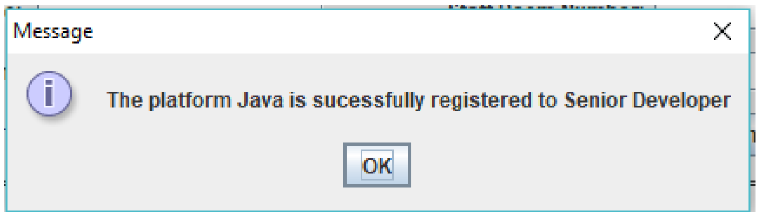
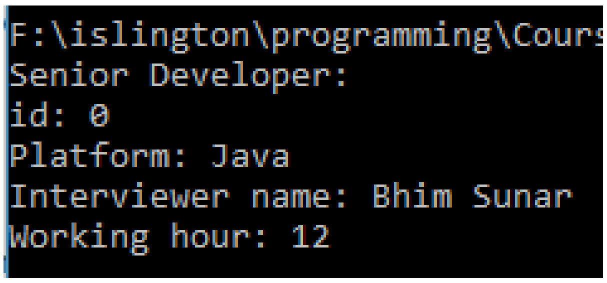
Testing:

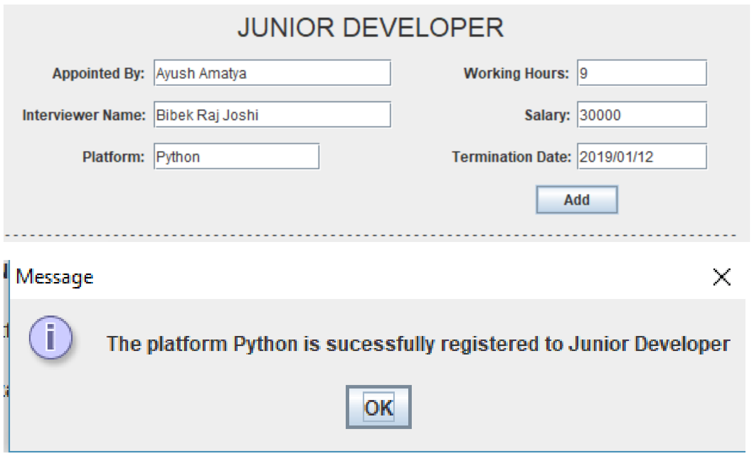
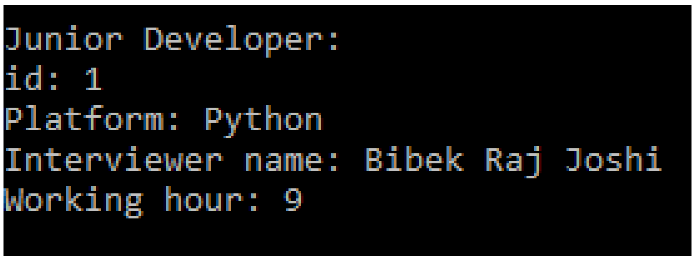
Test case number:	Test 1: Test that the program can be compiled and run using the command prompt.

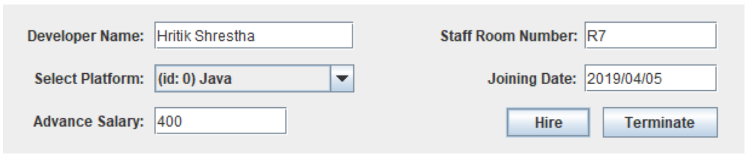
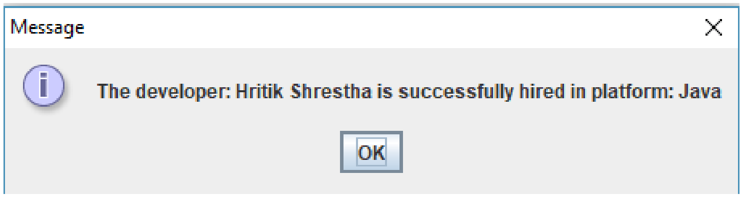
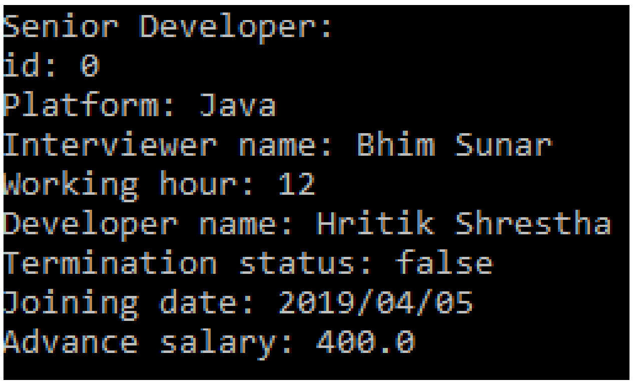
Objective/Steps:	1) Compile the program using command prompt
	
	C:\Windows\System32\cmd.exe
	`Microsoft Windows [Version 10.0.17134.706]`
	`(c) 2018 Microsoft Corporation. All rights reserved.`
	
	`F:\islington\programming\CourseWork\Ayush Amatya\CW2\Rigo Technology>javac RigoTechnology.java`
	`Note: RigoTechnology.java uses unchecked or unsafe operations.`
	`Note: Recompile with -Xlint:unchecked for details.`
	
	`F:\islington\programming\CourseWork\Ayush Amatya\CW2\Rigo Technology>`
	
	2) Run the program using command prompt
	
	C:\Windows\System32\cmd.exe - java RigoTechnology
	`Microsoft Windows [Version 10.0.17134.706]`
	`(c) 2018 Microsoft Corporation. All rights reserved.`
	
	`F:\islington\programming\CourseWork\Ayush Amatya\CW2\Rigo Technology>javac RigoTechnology.java`
	`Note: RigoTechnology.java uses unchecked or unsafe operations.`
	`Note: Recompile with -Xlint:unchecked for details.`
	
	`F:\islington\programming\CourseWork\Ayush Amatya\CW2\Rigo Technology>java RigoTechnology`
	
	 Rigo Technology
	SENIOR DEVELOPER
	Interviewer Name:
	Working Hours:
	Platform:
	Salary:
	Contract Period:
	Add
	Developer Name:
	Staff Room Number:
	Select Platform:
	Select platform
	Joining Date:
	Advance Salary:
	Hire
	Terminate
	JUNIOR DEVELOPER
	Appointed By:
	Working Hours:
	Interviewer Name:
	Salary:
	Platform:
	Termination Date:
	Add
	Developer Name:
	Evaluation Period:
	Select Platform:
	Select platform
	Appointed Date:
	Specialization:
	Termination Date:
	Appoint
	Display All
	Clear

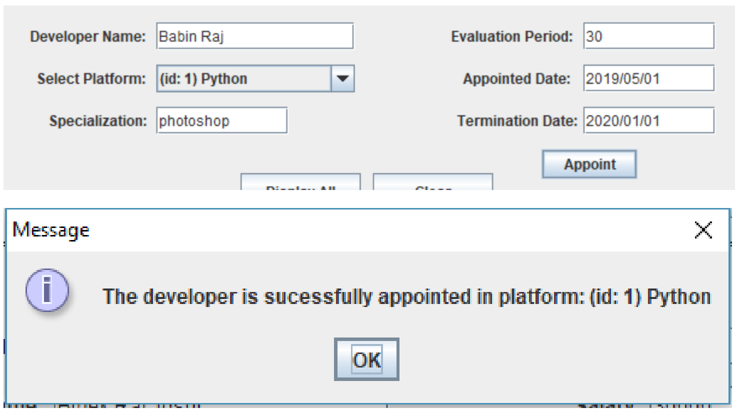
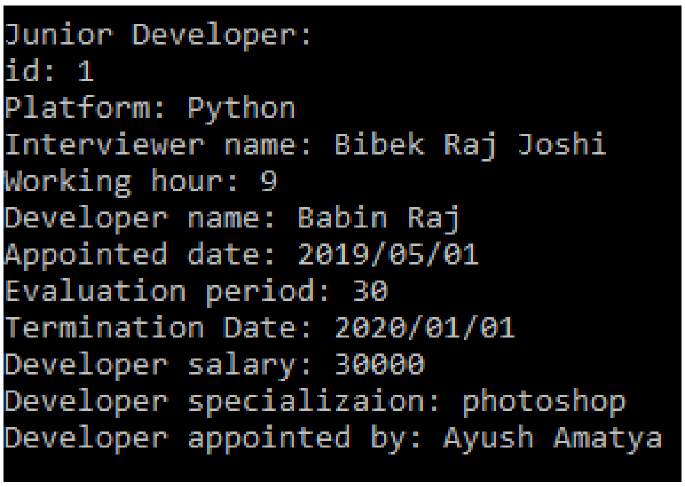
Expected result:	The program should be successfully complied using command prompt without any errors. The frame should be displayed when the program is run.

Actual result:	The program was successfully complied using command prompt without any errors. The frame was displayed when the program is run.
Conclusion:	The program passed Test 1.

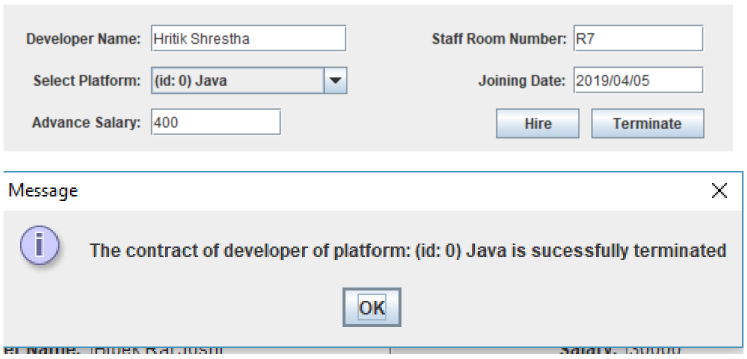
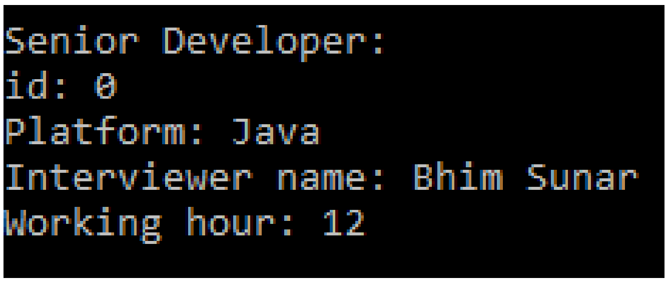
Test case number:	Test 2: Evidence should be shown of adding platform for senior developer
Objective/Steps:	1. add platform for senior developer  SENIOR DEVELOPER Interviewer Name: Bhim Sunar Working Hours: 12 Platform: Java Salary: 1000 Contract Period: 23  Message The platform Java is sucessfully registered to Senior Developer 2. click displayAll button and check output  F:\islington\programming\Cours Senior Developer: id: 0 Platform: Java Interviewer name: Bhim Sunar Working hour: 12
Expected result:	Success message should appear. Information about platform should be displayed when display all button is clicked
Actual result:	Success message was appeared. Information about platform was displayed when display all button is clicked
Conclusion:	The program passed Test 2.

Test case number:	Test 3: Evidence should be shown of adding platform for junior developer
Objective/Steps:	1. add platform for junior developer  JUNIOR DEVELOPER Appointed By: Ayush Amatya Working Hours: 9 Interviewer Name: Bibek Raj Joshi Salary: 30000 Platform: Python Termination Date: 2019/01/12 Add Message The platform Python is sucessfully registered to Junior Developer OK 2. click displayAll button and check output  Junior Developer: id: 1 Platform: Python Interviewer name: Bibek Raj Joshi Working hour: 9
Expected result:	Success message should appear. Information about platform should be displayed when display all button is clicked
Actual result:	Success message was appeared. Information about platform was displayed when display all button is clicked
Conclusion:	The program passed Test 3.

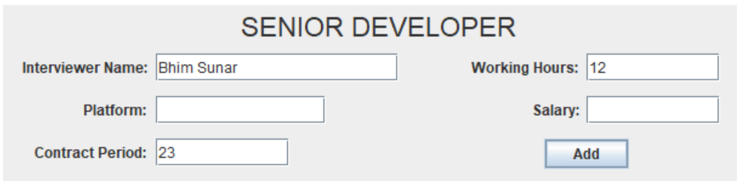
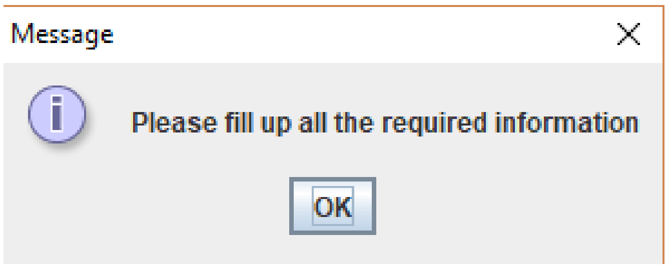
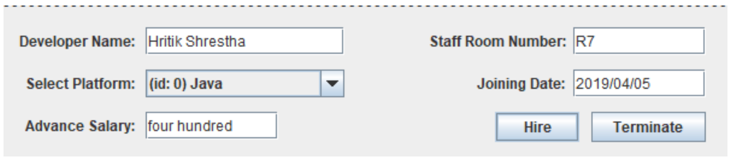
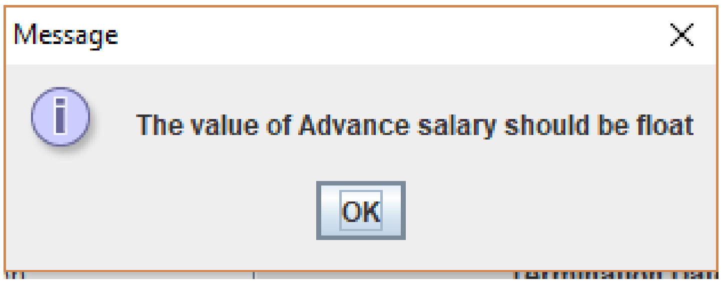
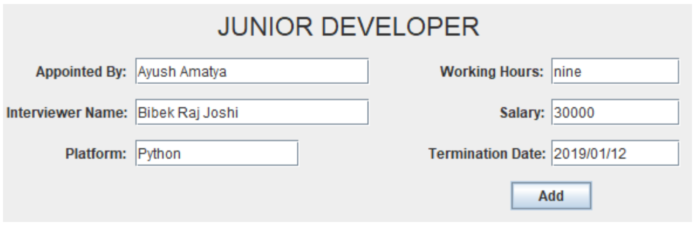
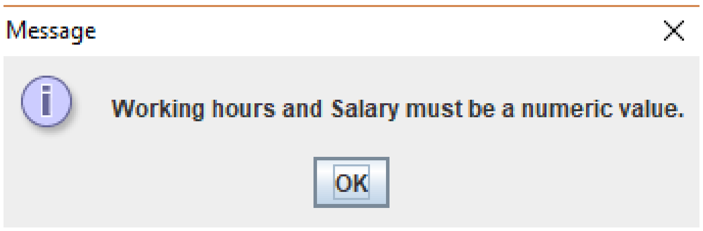
Test case number:	Test 4: Evidence should be shown of appointing senior developer
Objective/Steps:	1. hire developer for senior developer  Developer Name: Hritik Shrestha; Staff Room Number: R7; Select Platform: (id: 0) Java; Joining Date: 2019/04/05; Advance Salary: 400  The developer: Hritik Shrestha is successfully hired in platform: Java 2. click displayAll button and check output  Senior Developer: id: 0 Platform: Java Interviewer name: Bhim Sunar Working hour: 12 Developer name: Hritik Shrestha Termination status: false Joining date: 2019/04/05 Advance salary: 400.0
Expected result:	Success message should appear. Information about platform with developer details should be displayed when display all button is clicked
Actual result:	Success message was appeared. Information about platform with developer details was displayed when display all button is clicked
Conclusion:	The program passed Test 4.

Test case number:	Test 5: Evidence should be shown of appointing junior Developer
Objective/Steps:	1. appoint developer for junior developer  Developer Name: Babin Raj; Select Platform: (id: 1) Python; Specialization: photoshop; Evaluation Period: 30; Appointed Date: 2019/05/01; Termination Date: 2020/01/01; Appoint. Message: The developer is sucessfully appointed in platform: (id: 1) Python. OK 2. click displayAll button and check output  Junior Developer: id: 1 Platform: Python Interviewer name: Bibek Raj Joshi Working hour: 9 Developer name: Babin Raj Appointed date: 2019/05/01 Evaluation period: 30 Termination Date: 2020/01/01 Developer salary: 30000 Developer specializaion: photoshop Developer appointed by: Ayush Amatya
Expected result:	Success message should appear. Information about platform with developer details should be displayed when display all button is clicked
Actual result:	Success message was appeared. Information about platform with developer details was displayed when display all button is clicked
Conclusion:	The program passed Test 5.

Test case number:	Test 6: Evidence should be shown of terminating developer contract of senior developer

Objective/Steps:	1. terminate developer for junior developer
	
	Developer Name: Hritik Shrestha
	Staff Room Number: R7
	Select Platform: (id: 0) Java
	Joining Date: 2019/04/05
	Advance Salary: 400
	Message
	The contract of developer of platform: (id: 0) Java is sucessfully terminated
	2. click displayAll button and check output
	
	`Senior Developer:`
	`id: 0`
	`Platform: Java`
	`Interviewer name: Bhim Sunar`
	`Working hour: 12`

Expected result:	Success message should appear. Information about platform without any developer details should be displayed when display all button is clicked
Actual result:	Success message was appeared. Information about platform without any developer details was displayed when display all button is clicked
Conclusion:	The program passed Test 6.

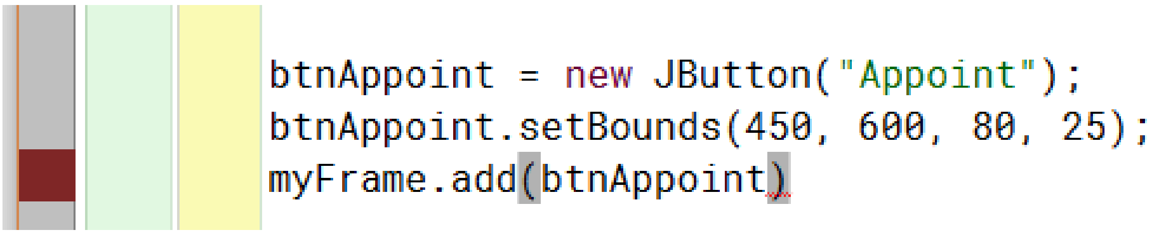
Test case number:	Test 7: Test that appropriate dialog boxes appear when unsuitable values are entered.
Objective/Steps:	1. try to add platform with some empty fields.  SENIOR DEVELOPER Interviewer Name: Bhim Sunar Working Hours: 12 Platform: Salary: Contract Period: 23  Message Please fill up all the required information 2. try to hire developer with the value of advance salary as string  Developer Name: Hritik Shrestha Staff Room Number: R7 Select Platform: (id: 0) Java Joining Date: 2019/04/05 Advance Salary: four hundred  Message The value of Advance salary should be float 3. try to add platform with the value of working hour as string  JUNIOR DEVELOPER Appointed By: Ayush Amatya Working Hours: nine Interviewer Name: Bibek Raj Joshi Salary: 30000 Platform: Python Termination Date: 2019/01/12  Message Working hours and Salary must be a numeric value.

Expected result:	Dialog box with suitable message should be displayed.
Actual result:	Suitable dialog box with suitable message was displayed.
Conclusion:	The program passed Test 7.

Error Detection and Error Correction:

Some of the errors that occurred during writing my code are as follows:

Error 1:

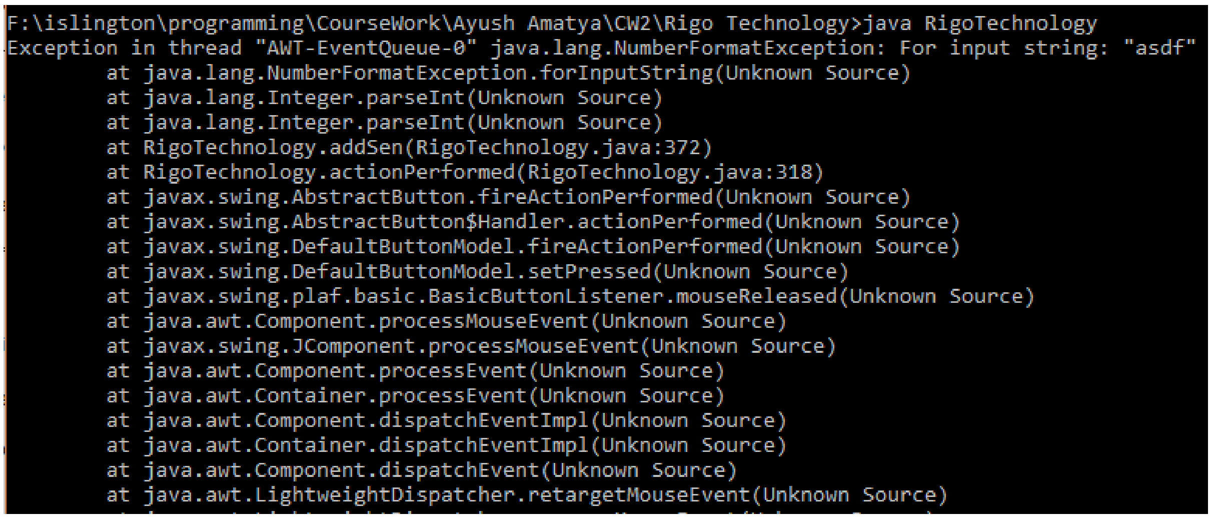


```
btnAppoint = new JButton("Appoint");
btnAppoint.setBounds(450, 600, 80, 25);
myFrame.add(btnAppoint)
```

This was a syntax error where I forgot to write ";" after returning the value.

It was easy to correct as the compiler detected the error and gave the solution. Hence, I corrected this error by adding ";" at the end of the syntax.

Error 2:

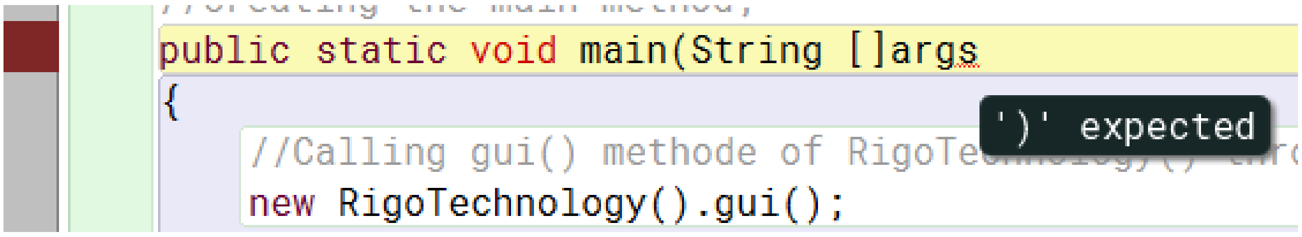


```
F:\islington\programming\CourseWork\Ayush Amatya\CW2\Rigo Technology>java RigoTechnology
Exception in thread "AWT-EventQueue-0" java.lang.NumberFormatException: For input string: "asdf"
        at java.lang.NumberFormatException.forInputString(Unknown Source)
        at java.lang.Integer.parseInt(Unknown Source)
        at java.lang.Integer.parseInt(Unknown Source)
        at RigoTechnology.addSen(RigoTechnology.java:372)
        at RigoTechnology.actionPerformed(RigoTechnology.java:318)
        at javax.swing.AbstractButton.fireActionPerformed(Unknown Source)
        at javax.swing.AbstractButton$Handler.actionPerformed(Unknown Source)
        at javax.swing.DefaultButtonModel.fireActionPerformed(Unknown Source)
        at javax.swing.DefaultButtonModel.setPressed(Unknown Source)
        at javax.swing.plaf.basic.BasicButtonListener.mouseReleased(Unknown Source)
        at java.awt.Component.processMouseEvent(Unknown Source)
        at javax.swing.JComponent.processMouseEvent(Unknown Source)
        at java.awt.Component.processEvent(Unknown Source)
        at java.awt.Container.processEvent(Unknown Source)
        at java.awt.Component.dispatchEventImpl(Unknown Source)
        at java.awt.Container.dispatchEventImpl(Unknown Source)
        at java.awt.Component.dispatchEvent(Unknown Source)
        at java.awt.LightweightDispatcher.retargetMouseEvent(Unknown Source)
```

This was a runtime error that occurred while I tried to enter string value where integer value was expected when hiring a developer for senior developer.

It was little difficult to solve this error. Since it was a run time error, the compiler was unable to detect the exact location of the error. After my hard effort, I found that I forgot to use try catch in hireSen() method. Hence, I corrected this error by using try catch in hireSen() method.

Error 3:



```
public static void main(String []args
{
    //Calling gui() methode of RigoTe
    new RigoTechnology().gui();
```

This was a syntax error where I forgot to close small bracket.

It was easy to correct as the compiler detected the error and gave the solution. Hence, I corrected this error by closing the bracket.

Conclusion:

All the tasks assigned in the coursework was finally completed through much trial and errors. Conclusion: Finally, after complition of this project I have learn many new things. The tasks assigned weren't that easy. Alot of reasearch and study was done about syntax, types of errors, frame, increase font size, use of JComboBox and many other topic related to java. After the completion of this project I learn many different things about java and its programming styles. I also learned about class diagram and pseudocode. Their was no bugs and errors in final check and the submission was done in time.

This coursework helps to improve ourself and get better in programming. It increase our skills and capacity of doing project in vast way. Learning this project in detial would be alot helpful in pursue of development of career as a good programmer. It provied us a detail idea about how we can iteract with the user as a programmer. This coursework teaches us the combine use of all the different classes and method of different classes. It gave us detailed knowledge about how to use the methods of other classes in another class. It made our vision about objects more clear and vast. We are now able to interact with the user from different means like buttons, text box, combo box, etc.

It requires nights of hardwork and must be determinant for better improvement. Even it was tough it was fun doing it. This project was sucessfully completed, but their are more thing we must learn about programming in very vast ways. It feels great if our codes would be correct and program would run without bug and error. In process of completing this coursework I faced lots of problems while designing the frame and read the inputs of the user. I sloved this problem by the help of my friends, teacher and ofcourse internet. I googled lots of thing related to the functions of some inbuit java class and method.

Appendix:

The codes to complete this coursework was written in blueJ. The codes written in this project are as follows:

Class: RigoTechnology

```
/**
 * Write a description of class RigoTechnology here.
 *
 * @author Ayush Amatya
 * @version 1.1
 */
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
//Creating a class RigoTechnology that implements an interface ActionListener;
public class RigoTechnology implements ActionListener
{
    //Creating array list named "list";
    ArrayList<Developer> list = new ArrayList<Developer>();

    //Declearing 1st value of integer i as 0. This is later used as id in combo box;
    int i = 0;

    //Declaring frame;
    JFrame myFrame;

    //Declaring all the JLabel that will be used in the frame;
    JLabel lblSeniorDeveloper, lblPlatformS, lblInterviewerNameS,
    lblWorkingHoursS, lblDotLineSen, lblSalaryS, lblContractPeriod,
lblDeveloperNameS,
    lblJoiningDate, lblAdvanceSalary, lblStaffRoomNumber, lblSelectPlatformS,
    lblEndLine, lblJuniorDeveloper, lblPlatformJ, lblInterviewerNameJ, lblSalaryJ,
    lblWorkingHoursJ, lblAppointedBy, lblTerminationDate1, lblDotLineJun,
    lblDeveloperNameJ, lblAppointedDate, lblSpecialization, lblSelectPlatformJ,
    lblEvaluationPeriod, lblTerminationDate2;

    //Declaring all the JTextBox that will be used in the frame;
    JTextField txtPlatformS, txtInterviewerNameS, txtWorkingHoursS, txtSalaryS,
    txtContractPeriod, txtDeveloperNameS, txtJoiningDate, txtAdvanceSalary,
    txtStaffRoomNumber, txtPlatformJ, txtInterviewerNameJ, txtSalaryJ,
    txtWorkingHoursJ, txtAppointedBy, txtTerminationDate1, txtTerminationDate2,
txtDeveloperNameJ,
    txtAppointedDate, txtSpecialization, txtEvaluationPeriod;

    //Declearing all the JButton that will be used in the frame;
    JButton btnAddSen, btnHire, btnTerminate, btnAddJun, btnAppoint, btnDisplayAll,
btnClear;

    //Declaring all the JComboBox that will be used in the frame;
    JComboBox cmbSelectPlatformS, cmbSelectPlatformJ;

    //Creating a methode named "gui" where all the code that displays required
information in frame is written. This methode will be called in main methode.
    public void gui(){
        //Creating frame "myFrame";
        myFrame = new JFrame("Rigo Technology");
        myFrame.setVisible(true);
        //Declearing size of the frame;
        myFrame.setSize(650,700);
        myFrame.setLayout(null);
```

```java
        //Creating JLabel "lblSeniorDeveloper" which displays "Senior Developer" in our frame;
        lblSeniorDeveloper = new JLabel ("SENIOR DEVELOPER");
        //Declearing font of the text;
        lblSeniorDeveloper.setFont(new Font("",Font.PLAIN, 22));
        //Declearing location and size of lblSeniorDeveloper. (x-axis, y-axis, width, height);
        lblSeniorDeveloper.setBounds(200, 5, 300, 25);
        //adding lblSeniorDeveloper to the frame;
        myFrame.add(lblSeniorDeveloper);

        lblPlatformS = new JLabel ("Platform:");
        lblPlatformS.setBounds (70, 75, 60, 23);
        myFrame.add(lblPlatformS);

        //Creating JTextField "txtPlatformS" to enter the value of platform;
        txtPlatformS = new JTextField ();
        //Declearing location and size of txtPlatformS. (x-axis, y-axis, width, height);
        txtPlatformS.setBounds(130, 75, 140, 23);
        //adding txtPlatformS to the frame;
        myFrame.add(txtPlatformS);

        lblInterviewerNameS = new JLabel("Interviewer Name:");
        lblInterviewerNameS.setBounds(20, 40, 110, 23);
        myFrame.add(lblInterviewerNameS);

        txtInterviewerNameS = new JTextField();
        txtInterviewerNameS.setBounds(130, 40, 200, 23);
        myFrame.add(txtInterviewerNameS);

        lblWorkingHoursS = new JLabel("Working Hours:");
```

```
    lblWorkingHoursS.setBounds(390, 40, 90, 23);
    myFrame.add(lblWorkingHoursS);

    txtWorkingHoursS = new JTextField();
    txtWorkingHoursS.setBounds(485, 40, 110, 23);
    myFrame.add(txtWorkingHoursS);

    lblSalaryS = new JLabel("Salary:");
    lblSalaryS.setBounds(440, 75, 60, 23);
    myFrame.add(lblSalaryS);

    txtSalaryS = new JTextField();
    txtSalaryS.setBounds(485, 75, 110, 23);
    myFrame.add(txtSalaryS);

    lblContractPeriod = new JLabel("Contract Period:");
    lblContractPeriod.setBounds(30, 110, 100, 23);
    myFrame.add(lblContractPeriod);

    txtContractPeriod = new JTextField();
    txtContractPeriod.setBounds(130, 110, 110, 23);
    myFrame.add(txtContractPeriod);

    btnAddSen = new JButton("Add");
    btnAddSen.setBounds(450, 110, 70, 25);
    myFrame.add(btnAddSen);

    lblDotLineSen = new JLabel("- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -");
    lblDotLineSen.setBounds(5, 140, 650, 20);
    myFrame.add(lblDotLineSen);
```

```
lblDeveloperNameS = new JLabel("Developer Name:");
lblDeveloperNameS.setBounds(25, 170, 100, 23);
myFrame.add(lblDeveloperNameS);

txtDeveloperNameS = new JTextField();
txtDeveloperNameS.setBounds(130, 170, 165, 23);
myFrame.add(txtDeveloperNameS);

lblJoiningDate = new JLabel("Joining Date:");
lblJoiningDate.setBounds(405, 205, 90, 23);
myFrame.add(lblJoiningDate);

txtJoiningDate = new JTextField();
txtJoiningDate.setBounds(485, 205, 110, 23);
myFrame.add(txtJoiningDate);

lblStaffRoomNumber = new JLabel("Staff Room Number:");
lblStaffRoomNumber.setBounds(365, 170, 120, 23);
myFrame.add(lblStaffRoomNumber);

txtStaffRoomNumber = new JTextField();
txtStaffRoomNumber.setBounds(485, 170, 110, 23);
myFrame.add(txtStaffRoomNumber);

lblAdvanceSalary = new JLabel("Advance Salary:");
lblAdvanceSalary.setBounds(30, 240, 100, 23);
myFrame.add(lblAdvanceSalary);

txtAdvanceSalary = new JTextField();
txtAdvanceSalary.setBounds(130, 240, 110, 23);
```

```
        myFrame.add(txtAdvanceSalary);


        lblSelectPlatformS = new JLabel("Select Platform:");
        lblSelectPlatformS.setBounds(30, 205, 100, 23);
        myFrame.add(lblSelectPlatformS);


        //Ceating JComboBox "cmbSelectPlatformS" that allows the user to select the
required platform;
        //Creating list "selectPlatformS[]" that stored the options of platform that is
displayed in the combo box;
        String selectPlatformS[] = {"Select platform", "No platform registered"};
        cmbSelectPlatformS = new JComboBox(selectPlatformS);
        //Declearing the location and size of cmbSelectPlatformS. (x-axis, y-axis, width,
height);
        cmbSelectPlatformS.setBounds(130, 205, 165, 23);
        //Adding cmbSeletPlatform to the frame;
        myFrame.add(cmbSelectPlatformS);


        //Creating JButton "btnHire";
        btnHire = new JButton("Hire");
        btnHire.setBounds(420, 240, 70, 25);
        myFrame.add(btnHire);


        btnTerminate = new JButton("Terminate");
        btnTerminate.setBounds(500, 240, 95, 25);
        myFrame.add(btnTerminate);


        lblEndLine = new
JLabel("===========================================================
==============================");
        lblEndLine.setBounds(5, 270, 650, 20);
        myFrame.add(lblEndLine);
```

```
lblJuniorDeveloper = new JLabel ("JUNIOR DEVELOPER");
lblJuniorDeveloper.setFont(new Font("",Font.PLAIN, 22));
lblJuniorDeveloper.setBounds(200, 290, 300, 25);
myFrame.add(lblJuniorDeveloper);

lblPlatformJ = new JLabel ("Platform:");
lblPlatformJ.setBounds (70, 400, 60, 23);
myFrame.add(lblPlatformJ);

txtPlatformJ = new JTextField ();
txtPlatformJ.setBounds(130, 400, 140, 23);
myFrame.add(txtPlatformJ);

lblInterviewerNameJ = new JLabel("Interviewer Name:");
lblInterviewerNameJ.setBounds(20, 365, 110, 23);
myFrame.add(lblInterviewerNameJ);

txtInterviewerNameJ = new JTextField();
txtInterviewerNameJ.setBounds(130, 365, 200, 23);
myFrame.add(txtInterviewerNameJ);

lblAppointedBy = new JLabel("Appointed By:");
lblAppointedBy.setBounds(45, 330, 110, 23);
myFrame.add(lblAppointedBy);

txtAppointedBy = new JTextField();
txtAppointedBy.setBounds(130, 330, 200, 23);
myFrame.add(txtAppointedBy);

lblWorkingHoursJ = new JLabel("Working Hours:");
```

```
        lblWorkingHoursJ.setBounds(390, 330, 90, 23);
        myFrame.add(lblWorkingHoursJ);

        txtWorkingHoursJ = new JTextField();
        txtWorkingHoursJ.setBounds(485, 330, 110, 23);
        myFrame.add(txtWorkingHoursJ);

        lblSalaryJ = new JLabel("Salary:");
        lblSalaryJ.setBounds(440, 365, 60, 23);
        myFrame.add(lblSalaryJ);

        txtSalaryJ = new JTextField();
        txtSalaryJ.setBounds(485, 365, 110, 23);
        myFrame.add(txtSalaryJ);

        lblTerminationDate1 = new JLabel("Termination Date:");
        lblTerminationDate1.setBounds(380, 400, 100, 23);
        myFrame.add(lblTerminationDate1);

        txtTerminationDate1= new JTextField();
        txtTerminationDate1.setBounds(485, 400, 110, 23);
        myFrame.add(txtTerminationDate1);

        btnAddJun = new JButton("Add");
        btnAddJun.setBounds(450, 435, 70, 25);
        myFrame.add(btnAddJun);

        lblDotLineJun = new JLabel("- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -");
        lblDotLineJun.setBounds(5, 465, 650, 20);
        myFrame.add(lblDotLineJun);
```

```
lblDeveloperNameJ = new JLabel("Developer Name:");
lblDeveloperNameJ.setBounds(25, 495, 100, 23);
myFrame.add(lblDeveloperNameJ);

txtDeveloperNameJ = new JTextField();
txtDeveloperNameJ.setBounds(130, 495, 165, 23);
myFrame.add(txtDeveloperNameJ);

lblAppointedDate = new JLabel("Appointed Date:");
lblAppointedDate.setBounds(385, 530, 90, 23);
myFrame.add(lblAppointedDate);

txtAppointedDate = new JTextField();
txtAppointedDate.setBounds(485, 530, 110, 23);
myFrame.add(txtAppointedDate);

lblEvaluationPeriod = new JLabel("Evaluation Period:");
lblEvaluationPeriod.setBounds(375, 495, 120, 23);
myFrame.add(lblEvaluationPeriod);

txtEvaluationPeriod = new JTextField();
txtEvaluationPeriod.setBounds(485, 495, 110, 23);
myFrame.add(txtEvaluationPeriod);

lblSpecialization = new JLabel("Specialization:");
lblSpecialization.setBounds(40, 565, 85, 23);
myFrame.add(lblSpecialization);

txtSpecialization = new JTextField();
txtSpecialization.setBounds(130, 565, 110, 23);
```

```
myFrame.add(txtSpecialization);

lblSelectPlatformJ = new JLabel("Select Platform:");
lblSelectPlatformJ.setBounds(30, 530, 100, 23);
myFrame.add(lblSelectPlatformJ);

String selectPlatformJ[] = {"Select platform", "No platform registered"};
cmbSelectPlatformJ = new JComboBox(selectPlatformJ);
cmbSelectPlatformJ.setBounds(130, 530, 165, 23);
myFrame.add(cmbSelectPlatformJ);

btnAppoint = new JButton("Appoint");
btnAppoint.setBounds(450, 600, 80, 25);
myFrame.add(btnAppoint);

btnDisplayAll = new JButton("Display All");
btnDisplayAll.setBounds(200, 620, 100, 30);
myFrame.add(btnDisplayAll);

btnClear = new JButton("Clear");
btnClear.setBounds(310, 620, 100, 30);
myFrame.add(btnClear);

lblTerminationDate2 = new JLabel("Termination Date:");
lblTerminationDate2.setBounds(380, 565, 100, 23);
myFrame.add(lblTerminationDate2);

txtTerminationDate2= new JTextField();
txtTerminationDate2.setBounds(485, 565, 110, 23);
myFrame.add(txtTerminationDate2);
```

```
        //adding ActionListener to all the button used in this frame. This decides the
functions of the button;
        btnClear.addActionListener(this);
        btnAddSen.addActionListener(this);
        btnAddJun.addActionListener(this);
        btnHire.addActionListener(this);
        btnTerminate.addActionListener(this);
        btnAppoint.addActionListener(this);
        btnDisplayAll.addActionListener(this);

        myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    //Creating a methode actionPerformed that is inside ActionListener;
    public void actionPerformed(ActionEvent e)
    {
        //This code runs if user click btnClear;
        if (e.getSource() == btnClear){
            //Calling method clear() of same class(this class);
            clear();
        }

        //This code runs if user click btnAddSen;
        if(e.getSource() == btnAddSen){
            //Calling method addSen() of same class(this class);
            addSen();
        }

        //This code runs if user click btnAddJun;
        if(e.getSource() == btnAddJun){
            //Calling method addJun() of same class(this class);
            addJun();
        }

        //This code runs if user click btnHire;
        if(e.getSource() == btnHire){
            //Calling method hire() of same class(this class);
            hireSen();
        }

        //This code runs if user click btnTerminate;
        if(e.getSource() == btnTerminate){
            //Calling method terminateSen() of same class(this class);
            terminateSen();
        }

        //This code runs if user click btnAppoint;
        if(e.getSource() == btnAppoint){
            //Calling method appointJun() of same class(this class);
            appointJun();
        }

        //This code runs if user click btnDisplayAll;
        if(e.getSource() == btnDisplayAll){
            //Calling method displayAll() of same class(this class);
            displayAll();
        }
    }

    //Declearing methode addSen() which decleares functions of btnAddSen;
    public void addSen()
    {
```

```
    //declearing a try except so that there are no RunTime errors if there occur
some errors;

    //try{

        //Storing the values entered by user in the text box of frame into string
variables;

        String platformS = txtPlatformS.getText();

        String interviewerNameS = txtInterviewerNameS.getText();

        String workingHoursS = txtWorkingHoursS.getText();

        String salaryS = txtSalaryS.getText();

        String contractPeriod = txtContractPeriod.getText();


        //Checking if any textboxes are empty or not;

        if(platformS.equals("") || interviewerNameS.equals("") ||
workingHoursS.equals("") || salaryS.equals("") ||

        contractPeriod.equals("")){

            //Displaying message in a dialogbox using JOptionPane;

            JOptionPane.showMessageDialog(myFrame, "Please fill up all the
required information");

        }

        else{

            //convertig values of working hours, slary and contract period into integer;

            int intWorkingHoursS = Integer.parseInt(workingHoursS);

            int intSalaryS = Integer.parseInt(salaryS);

            int intContractPeriod = Integer.parseInt(contractPeriod);


            //Creating the object(senDev) of SeniorDeveloper with all the required
arguments;

            SeniorDeveloper senDev = new SeniorDeveloper(platformS,
interviewerNameS, intWorkingHoursS, intSalaryS, intContractPeriod);


            //Adding the object(senDev) of SeniorDeveloper in the arraylist(list);

            list.add(senDev);
            if(cmbSelectPlatformS.getItemAt(1).equals("No platform registered")){
                cmbSelectPlatformS.removeItemAt(1);
            }
            //Adding the name of the platform available in SeniorDeveloper in combo
box(cmbSelectPlatformS);
            cmbSelectPlatformS.addItem("(id: " + i + ") " + platformS);
            i=i+1;

            //Displaying sucess message in the dialogbox;
            JOptionPane.showMessageDialog(myFrame, "The platform "+ platformS +
" is sucessfully registered to Senior Developer");
        }
      //}
      //This contains the code that is to be runned if the above code produces any
error;
      //catch(Exception exp){
          JOptionPane.showMessageDialog(myFrame, "Working hours, Salary and
contract period must be a numeric value.");
      //}
  }

  //Declearing methode hireSen() which decleares functions of btnHire;
  public void hireSen()
  {
      try{
          String developerNameS = txtDeveloperNameS.getText();
          String staffRoomNumber = txtStaffRoomNumber.getText();
          //Storing the value selected by user in combo box in new String
variable(selectPlatformS);
          String selectPlatformS = cmbSelectPlatformS.getSelectedItem().toString();
          String joiningDate = txtJoiningDate.getText();
          String advanceSalary = txtAdvanceSalary.getText();
        if(developerNameS.equals("") || staffRoomNumber.equals("") ||
selectPlatformS.equals("Select platform") ||

        joiningDate.equals("") || advanceSalary.equals("")){

            JOptionPane.showMessageDialog(myFrame, "Please fill up all the
required information");

        }

        else if(selectPlatformS.equals("No platform registered")){

            JOptionPane.showMessageDialog(myFrame, "There are no any platform
registered");

        }

        else{

            float floatAdvanceSalary = Float.parseFloat(advanceSalary);


            //Storing the index of the platform selected by user in new integer
variable(indexS) from selectPlatformS;

            int indexS = Character.getNumericValue(selectPlatformS.charAt(5));


            //Casting the object of Developer into object of SeniorDeveloper;

            //Here list.get(indexS) is the object of Developer class stored in
index(indexS) of arraylist(list) and senVar stores that object as the object of
SeniorDeveloper;

            SeniorDeveloper senVar = (SeniorDeveloper)list.get(indexS);


            if (senVar.getAppointed() == true){

                JOptionPane.showMessageDialog(myFrame, "The developer is already
hired in this platform: " + selectPlatformS);

            }

            else{

                //caling hire methode of SeniorDeveloper using SeniorDeveloper
object(senVar);

                senVar.hire(developerNameS, joiningDate, floatAdvanceSalary,
staffRoomNumber);
```

```
                JOptionPane.showMessageDialog(myFrame, "The developer: "+
developerNameS +" is successfully hired in platform: " + senVar.platform);
            }
          }


        }
        catch(Exception exp){
            JOptionPane.showMessageDialog(myFrame, "The value of Advance salary
should be float");
        }
    }


    //Declearing methode terminateSen() which decleares functions of btnTerminate;
    public void terminateSen()
    {
        String selectPlatformS = cmbSelectPlatformS.getSelectedItem().toString();
        if (selectPlatformS.equals("Select platform") || selectPlatformS.equals("No
platform registered")){
            JOptionPane.showMessageDialog(myFrame, "Please select the platform that
you want to terminate");
        }
        else{
            //Storing the index of the platform selected by user in new integer
variable(index) from selectPlatformS;
            int index = Character.getNumericValue(selectPlatformS.charAt(5));
            //Casting the object of Developer into object of SeniorDeveloper;
            //Here list.get(indexS) is the object of Developer class stored in index(indexS)
of arraylist(list) and senVar stores that object as the object of SeniorDeveloper;
            SeniorDeveloper senVar = (SeniorDeveloper)list.get(index);


            if(senVar.getTerminated() == true){
                JOptionPane.showMessageDialog(myFrame, "The contract of developer
of platform: " + selectPlatformS + " is already terminated");
```

```
            }
            else{
                JOptionPane.showMessageDialog(myFrame, "The contract of developer
of platform: " + selectPlatformS + " is sucessfully terminated");
                //Calling termination() method of SeniorDeveloper through object od
SeniorDeveloper(senVar);
                senVar.termination();
            }
        }
    }

    //Declearing methode addJun() which decleares functions of btnAddJun;
    public void addJun()
    {
        //declearing a try except so that there are no RunTime errors if there occur
some errors;
        try{
            //Storing the values entered by user in the text box of frame into string
variables;
            String platformJ = txtPlatformJ.getText();
            String interviewerNameJ = txtInterviewerNameJ.getText();
            String workingHoursJ = txtWorkingHoursJ.getText();
            String salaryJ = txtSalaryJ.getText();
            String terminationDate1 = txtTerminationDate1.getText();
            String appointedBy = txtAppointedBy.getText();

            //Checking if any textboxes are empty or not;
            if(platformJ.equals("") || interviewerNameJ.equals("") ||
workingHoursJ.equals("") || salaryJ.equals("") ||
            terminationDate1.equals("") || appointedBy.equals("")){
                JOptionPane.showMessageDialog(myFrame, "Please fill up all the
required information");
            }
            else{
                //convertig values of working hours and salary into integer;
                int intWorkingHoursJ = Integer.parseInt(workingHoursJ);
                int intSalaryJ = Integer.parseInt(salaryJ);

                //Creating the object(junDev) of JuniorDeveloper with all the required
arguments;
                JuniorDeveloper junDev = new JuniorDeveloper(platformJ,
interviewerNameJ, intWorkingHoursJ, intSalaryJ, appointedBy, terminationDate1);

                //Adding the object(junDev) of JuniorDeveloper in the arraylist(list);
                list.add(junDev);

                if(cmbSelectPlatformJ.getItemAt(1).equals("No platform registered")){
                    cmbSelectPlatformJ.removeItemAt(1);
                }
                //Adding the name of the platform available in JuniorDeveloper in combo
box(cmbSelectPlatformJ);
                cmbSelectPlatformJ.addItem("(id: " + i + ") " + platformJ);
                i=i+1;

                JOptionPane.showMessageDialog(myFrame, "The platform "+ platformJ +
" is sucessfully registered to Junior Developer");
            }
        }
        //This contains the code that is to be runned if the above code produces any
error;
        catch(Exception exp){
            JOptionPane.showMessageDialog(myFrame, "Working hours and Salary
must be a numeric value.");
        }
    }
```

```
   //Declearing methode appointJun() which decleares functions of btnAppoint;
   public void appointJun()
   {
       String developerNameJ = txtDeveloperNameJ.getText();
       String selectPlatformJ = cmbSelectPlatformJ.getSelectedItem().toString();
       String specialization = txtSpecialization.getText();
       String evaluationPeriod = txtEvaluationPeriod.getText();
       String appointedDate = txtAppointedDate.getText();
       String terminationDate2 = txtTerminationDate2.getText();

       if(developerNameJ.equals("") || selectPlatformJ.equals("Select platform") ||
specialization.equals("") ||
       evaluationPeriod.equals("") || appointedDate.equals("") ||
terminationDate2.equals("")){
           JOptionPane.showMessageDialog(myFrame, "Please fill all the required
information");
       }
       else if (selectPlatformJ.equals("No platform registered")){
           JOptionPane.showMessageDialog(myFrame, "There are no any platform
registered");
       }
       else{
           int indexJ = Character.getNumericValue(selectPlatformJ.charAt(5));
           JuniorDeveloper junVar = (JuniorDeveloper)list.get(indexJ);

           if(junVar.getJoined() == true){
               JOptionPane.showMessageDialog(myFrame, "The developer is already
appointed in this platform");
           }
           else{
               junVar.appoint(developerNameJ, terminationDate2, appointedDate,
specialization, evaluationPeriod);
```

```
                JOptionPane.showMessageDialog(myFrame, "The developer is
sucessfully appointed in platform: "+ selectPlatformJ);
            }
        }
    }

    //Declearing methode displayAll() which decleares functions of btnDisplayAll;
    public void displayAll()
    {
        //loopig all the items of arraylist(list) using for loop;
        for (Developer obj: list){
            //checking of the object is instance of SeniorDeveloper or not;
            if(obj instanceof SeniorDeveloper){
                System.out.println("Senior Developer:");
                System.out.println("id: " + Integer.toString(list.indexOf(obj)));
                obj.display();
            }
            else{
                System.out.println("Junior Developer:");
                System.out.println("id: " + Integer.toString(list.indexOf(obj)));
                obj.display();
            }
            System.out.println("");
        }
    }

    //Declearing methode clear() which decleares functions of btnClear;
    public void clear()
    {
        txtPlatformS.setText("");
        txtInterviewerNameS.setText("");
        txtWorkingHoursS.setText("");
        txtSalaryS.setText("");
        txtContractPeriod.setText("");
        txtDeveloperNameS.setText("");
        txtJoiningDate.setText("");
        cmbSelectPlatformS.setSelectedItem("Select platform");
        txtAdvanceSalary.setText("");
        txtStaffRoomNumber.setText("");
        txtPlatformJ.setText("");
        txtInterviewerNameJ.setText("");
        txtSalaryJ.setText("");
        txtWorkingHoursJ.setText("");
        txtAppointedBy.setText("");
        txtTerminationDate1.setText("");
        txtTerminationDate2.setText("");
        txtDeveloperNameJ.setText("");
        txtAppointedDate.setText("");
        txtSpecialization.setText("");
        cmbSelectPlatformJ.setSelectedItem("Select platform");
        txtEvaluationPeriod.setText("");
    }
    //Creating the main method;
    public static void main(String []args)
    {
        //Calling gui() methode of RigoTechnology() through the object of
RigoTechnology();
        new RigoTechnology().gui();
    }
}
```

Class: Developer

```
/** Creating Developer class **/
public class Developer
{
   /* Creating instance variables */
   String platform;
   String interviewerName;
   String developerName;
   int workingHours;

   /* Creating a constructor of Developer */
   public Developer(String platform, String interviewerName, int workingHours)
   {
      /* Assigning values entered by user as parameter to its corresponding variables
*/
      this.platform= platform;
      this.interviewerName= interviewerName;
      this.workingHours= workingHours;
      this.developerName="";
   }

   /* Crearing a accesor method from where user can access value of platform */
   public String getPlatform()
   {
      return platform;
   }

   /* Crearing a accesor method from where user can access value of interviewer
name*/
   public String getInterviewerName()
   {
      return interviewerName;
   }

   /* Crearing a accesor method from where user can access value of working
hours*/
   public int getWorkingHours()
   {
      return workingHours;
   }

   /* Crearing a accesor method from where user can access value of developer
name */
   public String getDeveloperName()
   {
      return developerName;
```

```
    }

    /* Creating a method from where user can change developer name*/
    public void setDeveloperName(String developerName)
    {
        this.developerName= developerName;
    }

    /* Creating a method that displays all the details to the user. */
    public void display()
    {
        System.out.println("Platform: "+platform);
        System.out.println("Interviewer name: "+ interviewerName);
        System.out.println("Working hour: "+ workingHours);
        if (!developerName.equals("")){
            System.out.println("Developer name: "+developerName);
        }
    }
}
```

Class: SeniorDeveloper

```
/** Creating SeniorDeveloper class **/
public class SeniorDeveloper extends Developer
{
    /* Creating instance variables */
    private int salary;
    private String joiningDate;
    private String staffRoomNumber;
    private int contractPeriod;
    private float advanceSalary;
    private boolean appointed;
    private boolean terminated;

    /* Creating a constructor of SeniorDeveloper */
    public SeniorDeveloper(String platform, String interviewerName, int workingHours,
int salary, int contractPeriod)
    {
        /* Calling constructor of super class i.e. Developer class*/
        super(platform, interviewerName, workingHours);
        /* Assigning values entered by user as parameter to its corresponding
variables*/
        this.salary= salary;
        this.contractPeriod= contractPeriod;
        joiningDate="";
        staffRoomNumber="";
        advanceSalary=0.0f;
        appointed= false;
        terminated= false;
    }

    /* Crearing a accesor method from where user can access value of advance
salary */
    public float getAdvanceSalary()
    {
        return advanceSalary;
    }

    /* Crearing a accesor method from where user can access value of joining date*/
    public String getJoiningDate()
    {
        return joiningDate;
    }

    /* Crearing a accesor method from where user can access value of staff room
number */
    public String getStaffRoomNumber()
    {
        return staffRoomNumber;
    }

    /* Crearing a accesor method from where user can access value of salary*/
    public int getSalary()
    {
        return salary;
    }

    /* Crearing a accesor method from where user can access value of contact
period*/
    public int getContractPeriod()
    {
        return contractPeriod;
    }

    /* Crearing a accesor method from where user can access value of appointed */
    public boolean getAppointed()
    {
        return appointed;
    }

    /* Crearing a accesor method from where user can access value of terminated */
    public boolean getTerminated()
    {
        return terminated;
    }

    /* Creating a method from where user can change salary*/
    public void setSalary(int salary)
    {
        this.salary = salary;
    }

    /* Creating a method from where user can change contract period*/
    public void setContractPeriod(int contractPeriod)
    {
        this.contractPeriod = contractPeriod;
    }

    /* Creating a method to hire new developer */
    public void hire(String developerName, String joiningDate, float advanceSalary,
String staffRoomNumber)
    {
        if (appointed == true){
            /* This code runs if appointed is true */
            System.out.println("The developer is already appointed");
            System.out.println("Developer name: "+ developerName);
            System.out.println ("Room number: "+ staffRoomNumber);
        }
        else{
            /* This code runs if appointed is false */
            /* Calling the method to set the developer name from parent class i.e.
Developer class with the developer name as parameter*/
            setDeveloperName(developerName);
            this.joiningDate = joiningDate;
            this.advanceSalary = advanceSalary;
            this.staffRoomNumber= staffRoomNumber;
            appointed= true;
            terminated= false;
        }
    }

    /* Creating a method that terminates the developer contract */
    public void termination()
    {
        if (terminated == true){
            /* This code runs if termination is true */
            System.out.println("The developer contract is already terminated.");
        }
        else{
            /* This code runs if termination is false. */
            /* Calling the method to set the developer name from parent class i.e.
Developer class with the empty string as parameter*/
            setDeveloperName("");
            this.joiningDate="";
            this.advanceSalary = 0.0f;
            this.staffRoomNumber = “”;
            appointed= false;
            terminated= true;
        }
    }

    /* Creating a method to display platform, interviewer name and salary to the user
*/
    public void print()
    {
        System.out.println("Platform: "+ getPlatform());
        System.out.println("Interviewer name: "+ getInterviewerName());
        System.out.println("Developer salary: "+ getSalary());
    }

    /* Creating a method that displays all the details to the user. */
    public void display()
    {
        /* Calling display method of super class i.e. Developer class*/
        super.display();
        if (appointed==true){
            /* This code runs if appointed is true */
            System.out.println("Termination status: "+ terminated);
            System.out.println("Joining date: "+ joiningDate);
            System.out.println("Advance salary: "+ advanceSalary);
        }
    }
}
```

Class: JuniorDeveloper

```
/** Creating SeniorDeveloper class **/
public class JuniorDeveloper extends Developer
{
    /* Creating instance variables */
    private int salary;
    private String appointedDate;
    private String evaluationPeriod;
    private String terminationDate;
    private String specialization;
    private String appointedBy;
```

```
   private boolean joined;

   /* Creating a constructor of JuniorDeveloper */
   public JuniorDeveloper(String platform, String interviewerName, int workingHours,
int salary, String appointedBy, String terminationDate)
   {
      /* Calling constructor of super class i.e. Developer class*/
      super(platform, interviewerName, workingHours);
      /* Assigning values entered by user as parameter to its corresponding
variables*/
      this.platform= platform;
      this.interviewerName= interviewerName;
      this.workingHours= workingHours;
      this.salary= salary;
      this.appointedBy= appointedBy;
      this.terminationDate= terminationDate;
      appointedDate="";
      evaluationPeriod="";
      specialization="";
      joined= false;
   }

   /* Crearing a accesor method from where user can access value of salary*/
   public int getSalary()
   {
      return salary;
   }

   /* Crearing a accesor method from where user can access value of appointed
date*/
   public String getAppointedDate()
   {
      return appointedDate;
   }

   /* Crearing a accesor method from where user can access value of evaluation
period*/
   public String getEvaluationPeriod()
   {
      return evaluationPeriod;
   }

   /* Crearing a accesor method from where user can access value of termination
date*/
   public String getTerminationDate()
   {
      return terminationDate;
```

```
    }

    /* Crearing a accesor method from where user can access value of appointed by*/
    public String getAppointedBy()
    {
        return appointedBy;
    }

    /* Crearing a accesor method from where user can access value of specialization
*/
    public String getSpecialization()
    {
        return specialization;
    }

    /* Crearing a accesor method from where user can access value of joined */
    public boolean getJoined()
    {
        return joined;
    }

    /* Creating a method from where user can change salary*/
    public void setSalary(int salary)
    {
        if (joined==false){
            /* This code runs if joined is false*/
            this.salary=salary;
        }
        else{
            /* This code runs if joined is true*/
            System.out.println("The developer has already joined. Therefore it is not
possible to change the salary");
        }
    }

    /* Creating a method to appoint new developer */
    public void appoint(String developerName, String appointedDate, String
terminationDate, String specialization, String evaluationPeriod)
    {
        if (joined==false){
            /* This code runs if joined is false*/
            setDeveloperName(developerName);
            joined= true;
            this.appointedDate= appointedDate;
            this.terminationDate= terminationDate;
            this.specialization= specialization;
            this.evaluationPeriod = evaluationPeriod;
        }
        else{
            /* This code runs if joined is true*/
            System.out.println("The developer is already appointed");
            System.out.println("Appointed Date: "+ appointedDate);
        }
    }

    /* Creating a method that displays all the details to the user. */
    public void display()
    {
        /* Calling display method of super class i.e. Developer class*/
        super.display();
        if (joined== true){
            /* This code runs if joined is true */
            System.out.println("Appointed date: "+ appointedDate);
            System.out.println("Evaluation period: "+ evaluationPeriod);
            System.out.println("Termination Date: "+ terminationDate);
            System.out.println("Developer salary: "+ salary);
            System.out.println("Developer specializaion: "+ specialization);
            System.out.println("Developer appointed by: "+ appointedBy);
        }
    }
}
```