LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

MODULE CODE & MODULE TTILE

CS4001NI - PROGRAMMING

50% Individual Coursework

2019-20 Autumn

Student Name: NIMESH POUDEL

GROUP:C2

London Met ID: 19031195

College ID: NP01CP4A190166

Assignment Due Date: 2020-06-05

Assignment Submission Date: 2020-06-05

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

.

Table of Contents

Table of table

Table of figure

1.Introduction

1.1 Java

(lifewire.com, n.d.)

Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere! (Java.com, 2020)

Java is a broadly useful programming language that is class-based, object-situated, and intended to possess as not many execution conditions as could be expected under the circumstances. Java application designers compose once, run anyplace (WORA), implying that assembled Java code can run on all stages that help Java without the requirement for recompilation.

1.2 BlueJ

BlueJ is an Integrated Development Environment (IDE) for the Java programming language. This software application helps to provide a more precise interface for creating projects and coding in Java. BlueJ was primarily built to assist with user education on object-oriented programming. The interface supports visual views of classes and coded objects. The idea is that by ordering and organizing visual representations of Java code, these kinds of tools can make programming languages like Java easier to use. (Techopedia, 2020)

1.3 Object oriented Programing language

Object Oriented programming is a way of programming which is associated with the concepts which is related with the ideas of Object, Class, Polymorphism, Inheritance, Encapsulation, Abstraction as they are real world entities the main of Object Oriented Programming is to used these concept in programming This concept is also Knows as Oop's . In this Coursework we had use this concept as java is Object Oriented Programming Language. Each of these concepts has their own rules. Security is one of the important topic in software developing so the main implements' of oop's concept is to provide security and code reusability. Developer can use same code for different method.

1.4 Graphical User Interface (GUI)

The graphical User Interface is a type of interface that permits or kind of computer programme that enables user to connect with electronic gadgets through graphical symbols and sound marker, for example, essential documentation, rather than content based user interfaces, composed order names or content route. GUIs were acquainted in response with the apparent soak expectation to absorb information of order line interfaces which expect orders to be composed on a PC console. The activities in a GUI are normally performed through direct control of the graphical components like mouse. GUIs are utilized in numerous handheld cell phones, for example, MP3 players, convenient media players, gaming gadgets, cell phones and littler family, office and mechanical controls

Graphical User Interfaces (GUIs) are systems for permitting user to enter information within the most affordable and clear way conceivable. GUI that is intended to permit a user to settle a text style type, style and size. Not only that it also enables to click and drag and drop items. There are various controls, for example, Buttons, Text Fields, Label, combo box, decision things and check box things too. GUI also permits the user to pick the choices effectively, while it likewise permits the software engineer or developer to painstakingly control the way that the client can enter the information, keeping the client from entering invalid choices.

Java gives two components to creating UI applications in Java - AWT and Swing. AWT (Abstract Windowing Toolkit) is the package. In this project both packages are used to

make this project reliable and user interface good or user friendly. The Abstract Window Toolkit (AWT) is Java's unique stage subordinate windowing, designs, and User interface gadget toolbox, going before Swing. The AWT is a piece of the Java Foundation Classes (JFC) the standard API for giving a GUI for a Java program . Java Swing is a lightweight Graphical User Interface (GUI) toolbox that incorporates a rich arrangement of gadgets. It incorporates bundle that hat features a fashionable to make GUI segments for your Java applications, and It is stage independent. The Swing library is made on the Java Abstract Widget Toolkit (AWT), a more established, stage subordinate GUI toolbox. which can utilize the Java GUI parts like button , textbox, and so forth from the library and don't need to make the segments without any preparation.

1.5 ArrayList

Arraylist class executes List interface and it depends on an Array information structure. It is generally utilized due to the usefulness and adaptability it offers. The majority of the designers pick Arraylist over Array as it's an awesome option of customary java exhibits. ArrayList is a resizable-cluster execution of the List interface. It executes all discretionary rundown tasks, and allows all components, including null and duplicate.

In this project Array List is used in order to pass and keep all the attributes like vacancy number, designation, job type working hour, salary, wager per hour , staff name, qualification, appointed by and joining date.

1.6 Introduction of project.

This project was given as a second coursework for the modules CS4001NI programming. In this coursework students should make Graphical User Interface program by the helps of java using the concept of object oriented Programming. The main aims of this project is to check the programming ability of the students by creating the Vacancy management software with GUI . Where user can manage task like Add vacancy for Full time as well as part Time , appoint the Staff in particular vacancy for both types of Job types and user can terminate the staff of part time in needed and user also can display the current information about vacancy and staff too.

There are four program Ing Nepal, Staff hire , Full time staff hire and part time staff hire where staff hire is the class and Part time staff hire and Full time staff hire are sub class of staff hire. Ing Nepal is the GUI Class which is inherit in staff hire, part time staff hire, full time staff hire.Staff hire cotains Vacancy no, degisanation and job type they are called in sub classes i.e. full time and part time by using super keyword. In Ing Nepal all of the Attributes like vacancy number, designation, job type working hour, salary, wager per hour , staff name, qualification, appointed by and joining date are called by the helps of Associate and Arraylist used in this project when user provides value of each attributes in text box of graphical frame. Array list helps to store value and provide the value of each attributes when it is called. There are JButton, Jpanel,Jframe, Jtextfield, imageicon,J label are used inside the Ing Nepal in order to make Graphical User Interface program. If else, is used to run the program in different different condition. Different Boolean Flag are also created inside this class to run the program efficient way and to reduce data redundancy. Try Catch is used to for handling different error according to requirement of the project when error is occur it will pop up with error message. There are many method which converts String value to integer. At First there are six button Vacancy Add for full time, Part Time similarly appoint full time and part time. , terminate and display. When user click on Add vacancy for full time new frame will be open and user can put the values in each text fields. Those values of the text field i.e. vacancy number, designation, job type,

salary and working hour per day will create a new object of type FullTime StaffHire which is added to array list of StaffHire class. Similarly when user click on Add vacancy for Part Time StaffHire new frame will be open and user can put the values in each textfields . Thoose values of text fields i.e. vacancy number,designation, job type, working hours per day, wages per hour and shift will create a new object of type PartTime StaffHire which is added to an array list of StaffHire class. More over when user click on Appoint Full time staff new frame will open and ask the used to provide the vacancy number to appoint the staff it means in which vacancy number user want to hire the staff if vacancy number is in arraylist the new frame will open in order to appoint staff other wise it will say vacancy no is not found. when user click on Appoint Part time staff new frame will open and ask the used to provide the vacancy number to appoint the staff it means in which vacancy number user want to hire the staff if vacancy number is in arraylist the new frame will open in order to appoint staff other wise it will say vacancy no is not found. When user click in terminate button new frame will open and ask user to input the vacancy number same process if vacancy number user enter matches in vacancy number in arraylist the user will terminate otherwise it will say error. This terminate is inly for part time staffhire. When user click Display button ,the information relating to the appropriate class will displayed. There are two button more clear and cancel if user click on cancel button then frame will dispose and if user click on clear button then the value in text field will erase. All of this event is handle by using Action listener.

To sum up, This program was developed for the company which helps in hiring staff. This program show all the details information of vacancy post, designation, vacancy number and job type as well. And in sub classes it provide the information about working hour, qualification ,wages and appointment too. By the helps of this program company can keeps record of the staff and it also can terminated if any this happened and again new staff can be hire. Vacancy repetition error is solved so that user can used this program with easy and reliable ways

2.Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. (Visual-paradigm, 2020)

The Class Diagram of ING Nepal is

Class ING Nepal	
- frame	JFrame
-frame1	JFrame
-frame2	JFrame
-framein	JFrame
-Frameck	JFrame
-framedis	JFrame
-frametm	JFrame
-cointainfu	JPanel
-cointain	JPanel
-cointainfuap	JPanel
-InformationStaff	JPanel
-Terminatepanel	JPanel
-Inputvano	JTextField
-Inputde	JTextField
-Inputjb	JTextField
-Inputsa	JTextField
-Inputwh	JTextField
-Inputwg	JTextField
-viworkHour	JTextField
-vistaff	JTextField
-vishift	JTextField

-viappol	JTextField
-videsign	JTextField
-vijobtype	JTextField
-vivac	JTextField
-inputvcba	JTextField
-inputsh	JTextField
-vijoindate	JTextField
-viquali	JTextField
-viwageper	JTextField
-visalary	JTextField
-inputtem	JTextField
-realVacancy	Int
-realSalary	Int
-realWorkHour	Int
-realWage	Int
-choice	String
-realDesignation	String
-realJob	String
-realShift	String
-copyCheck	Boolean
-toAddinside	Boolean
-terbo	Boolean
-Fullva	JButton
-Fullap	JButton
-Partva	JButton
-Partap	JButton
-Display	JButton
-Terminate	JButton
-Hire	JButton
-Cancel	JButton
-submit1	JButton

-submit2 -infoClear -infoSave -ckcl -ckbtn -btnterminate	JButton JButton JButton JButton JButton JButton
+GuiBox(); +Fulltimeva(); +Fulltimeapoint(); +InsideHiring(); +termGui(); +displaymethod(); +vacancyforward(); +vacancyFrom(); +designationFrom(); +workHourFrom(); +salaryFrom(); +shiftFrom(); +wagePHourFrom(); +staffNameFrom(); +appointeByFrom(); +joingDateFrom(); +qualificationFrom(); +inputtemret(); +hiringPartTimeStaff(String staffName,Joiningdate,qualification,appointBy); +hiringFullTimeStaff(String staffName,Joiningdate,qualification,appointBy); +backendterminat(); +repetationStaff();	Void Void Void Void Void Void Void Int String String int String Int String String String String Int Void void Void Void

+dataStore();	Void
+dataCheck();	Void
+FullTimeData();	String
+partTimeData();	String

3. Relation Diagram

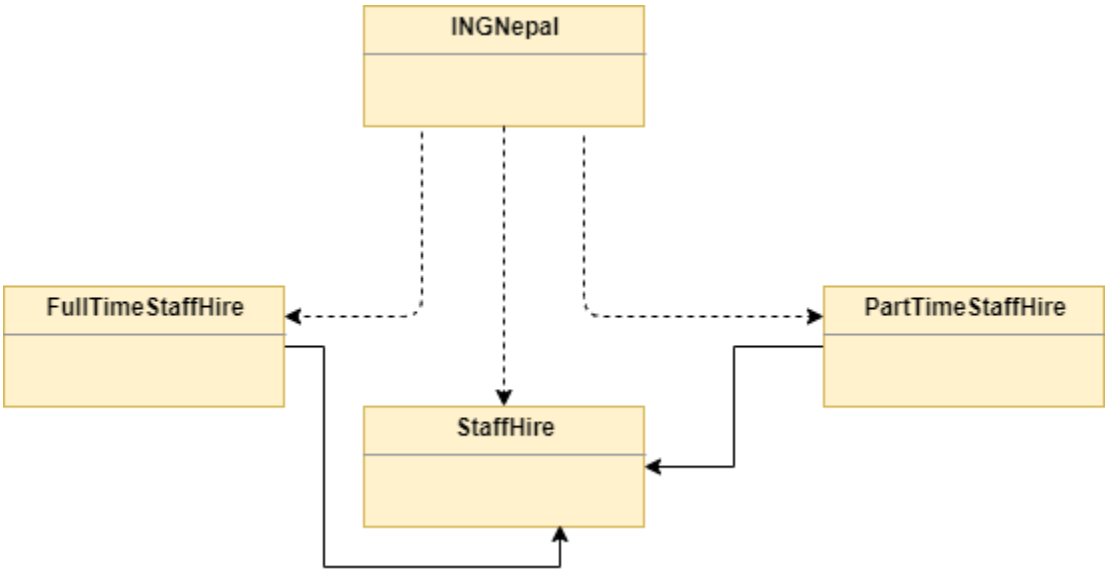


Figure 1 Relation of all the classes

Relationships in UML are used to represent a connection between various things. It is also called a link that describes how two or more things can relate to each other during the execution of a system. A relationship is a connection amongst things such as structural, behavioral, or grouping things in the unified modeling language. (guru99, 2020)

In this diagram StaffHire is the super class for FullTimeStaffHire and PartTimeStaffHire. FullTimeStaffHire and PartTimeStaffHire are Connected to the StaffHire class by using extends keyword i.e. inheritance and INGNepal is the class for GUI , where other three class are associative Arraylist of StaffHire is create in INGNepal class.

4.Pseudo Code

4.1 **Method Name** vacancyFrom()

```
        FUNCTION vacancyFrom(){
 DO
  RETURN Integer.parseInt(inputvano.getText())
END DO
}
```

4.2 **Method Name** designationFrom ()

```
FUNCTION  designationFrom(){
DO
    RETURN inputde.getText()
END DO
 }
```

4.3 **Method Name** workHourFrom ()

```
FUNCTION workHourFrom (){
DO
    RETURN Integer.parseInt(inputwh.getText());
END DO
}
```

4.4 **Method Name** salaryFrom ()

```
FUNCTION salaryFrom (){
DO
    RETURN Integer.parseInt(inputsa.getText());
END DO
}
```

4.5 **Method Name** shiftFrom ()

```
FUNCTION  shiftFrom (){
DO
    RETURN inputsh.getText();
END DO
 }
```

4.6 **Method Name** wagePHourFrom ()

```
FUNCTION wagePHourFrom (){
DO
    RETURN Integer.parseInt(inputwg.getText());
END DO
}
```

4.7 **Method Name** staffNameFrom ()

```
FUNCTION  staffNameFrom (){
DO
 RETURN vistaff.getText();
END DO
 }
```

4.8 **Method Name** appointeByFrom ()

```
FUNCTION  appointeByFrom (){
DO
    RETURN viappol.getText();
END DO
 }
```

4.9 **Method Name** joingDateFrom ()

```
FUNCTION  joingDateFrom (){
DO
    RETURN vijoindate.getText();
END DO
 }
```

4.10 **Method Name** qualificationFrom ()

```
FUNCTION  qualificationFrom (){
DO
    RETURN viquali.getText();
END DO
 }
```

4.11 **Method Name** inputtemret ()

```
FUNCTION inputtemret (){
DO
    RETURN Integer.parseInt(inputtem.getText());
```

END DO

}

4.12 **Assign** variables

DECLARE staffVacacyhire as ArrayList <StaffHire>

FUNCTION main(String[] args)

DECLARE obj1 as an object to call non-static methods.

CALL guiBox();

SET UIManager.put("OptionPane.background", Color.*darkGray*);

SET UIManager.put("Panel.background", Color.*darkGray*);

SET UIManager.put("Button.background", Color.*lightGray*)

SET UIManager.put("OptionPane.messageForeground",Color.*lightGray*);

END FUNCTION

4.13 **Method Name** guiBox()

FUNCTION guiBox(){

DO

INITIALIZE JFrame as frame

SET Preferred Size(500,380); in frame

INITIALIZE JPanel as cointain

SET Layout null in cointain

SET Background Color as DARK_GRAY in cointain

ADD ImageIcon as img,img1,img2,img3

INITIALIZE JLabel as titJlabel in cointain

SET Bounds (150,0,250,40) in a titJlabel

SET Fonts SansSeri and size 38 in titJlabel

SET Foreground Color as White as in titJLabel

INITIALIZE JButton Fullva

SET Bounds (100,50,250,40) in a Fullva

SET Background Color as GRAY in a Fullva

ADD img in a Fullva

INITIALIZE JButton Fullap

SET Bounds (100,150,250,40) in a Fullap

SET Background Color as LIGHT_GRAY in a Fullap

ADD img1 in a Fullap

INITIALIZE JButton Partva

SET Bounds (100,100,250,40) in a Partva

SET Background Color as GRAY in a Partva

ADD img in a Partva

INITIALIZE JButton Partap

SET Bounds (100,200,250,40 in a Partap

SET Background Color as LIGHT_GRAY in a Partap

ADD img1 in a Partap

INITIALIZE JButton display

SET Bounds (100,245,250,40) in a display

SET Background Color as GRAY in a display

ADD img2 in a display

INITIALIZE JButton terminate

SET Bounds (100,290,250,40 in a terminate

SET Background Color as LIGHT_GRAY terminate

ADD img3 in terminate

ADD titJLabel in a cointain

ADD Fullva in a cointain

ADD Fullap in a cointain

ADD Partap in a cointain

ADD Partva in a cointain

ADD display in a cointain

ADD terminate in a cointain

ADD Action Listner to Fullva

ADD Action Listner to Fullap

ADD Action Listner to Partva

ADD Action Listner to Partap

ADD Action Listner to display

ADD Action Listner to terminate

ADD cointain in frame

DECLARE frame as pack

SET DefaultCloseOperation(JFrame.EXIT_ON_CLOSE) in frame

SET Visible true in frame

SET LocationRelativeTo null in frame

END DO

}

4.14 **Method Name** Fulltimeva ()

FUNCTION Fulltimeva (){

DO

INITIALIZE JFrame as frame1

SET Preferred Size(500,380); in frame1

SET title as Add Vacancy in frame1

INITIALIZE JPanel as cointainfu

SET Layout null in cointainfu

SET Background Color as DARK_GRAY in cointainfu

ADD ImageIcon as img,img1

INITIALIZE JLabel as titJlabel in cointainfu
SET Bounds (150,0,250,40) in a titJlabel
SET Fonts SansSeri and size 38 in titJlabel
SET Foreground Color as White as in titJLabel

INITIALIZE JLabel as lbl1in cointainfu
SET Bounds (20, 60, 150, 20) in a lbl1
SET Fonts SansSeri and size 15 in lbl1
SET Foreground Color as LIGHT_GRAY as in lbl1

INITIALIZE JTextField as inputvano in cointainfu
SET Bounds (260, 60, 180, 25) in a inputvano
SET Fonts Calibri and size 15 in inputvano

INITIALIZE JLabel as lbl2 in cointainfu
SET Bounds (20, 95, 150, 20) in a lbl2
SET Fonts SansSeri and size 15 in lbl2
SET Foreground Color as LIGHT_GRAY as in lbl2

INITIALIZE JTextField as inputdein cointainfu
SET Bounds (260, 95, 180, 25) in a inputde
SET Fonts Calibri and size 15 in inputde

INITIALIZE JLabel as lbl3 in cointainfu
SET Bounds (20, 130, 150, 20) in a lbl3
SET Fonts SansSeri and size 15 in lbl3
SET Foreground Color as LIGHT_GRAY as in lbl3

INITIALIZE JTextField as inputjb inn cointainfu
SET Bounds (260, 130, 180, 25) in a inputjb
SET Fonts Calibri and size 15 in inputjb

SET Editable as false in inputjb

SET Text as choice in injubjb

INITIALIZE JLabel as lbl4 in cointainfu

SET Bounds (20, 165, 150, 20) in a lbl4

SET Fonts SansSeri and size 15 in lbl4

SET Foreground Color as LIGHT_GRAY as in lbl4

INITIALIZE JTextField as inputsa in cointainfu

SET Bounds (260, 165, 180, 25) in a inputsa

SET Fonts Calibri and size 15 in inputsa

INITIALIZE JLabel as lbl5 in cointainfu

SET Bounds (20, 200, 150, 20) in a lbl5

SET Fonts SansSeri and size 15 in lbl5

SET Foreground Color as LIGHT_GRAY as in lb5

INITIALIZE JTextField as inputwh in cointainfu

SET Bounds (260, 200, 180, 25) in a inputwh

SET Fonts Calibri and size 15 in inputwh

INITIALIZE JLabel as lbl6 in cointainfu

SET Bounds (20, 165, 150, 20) in a lbl6

SET Fonts SansSeri and size 15 in lbl6

SET Foreground Color as LIGHT_GRAY as in lb6

INITIALIZE JTextField as inputwg in cointainfu

SET Bounds (260, 165, 180, 25) in a inputwg

SET Fonts Calibri and size 15 in inputwg

INITIALIZE JLabel as lbl7 in cointainfu

SET Bounds (20, 235, 150, 20) in a lbl7

SET Fonts SansSeri and size 15 in lbl7

SET Foreground Color as LIGHT_GRAY as in lb7

INITIALIZE JTextField as inputsh in cointainfu

SET Bounds (260, 235, 180, 25) in a inputsh

SET Fonts Calibri and size 15 in inputsh

INITIALIZE JButton as hire in cointainfu

SET Bounds (80, 280, 120, 40) in a hire

ADD img in hire

ADD Action Listner in hire

SET Background Color as LIGHT_GRAY

INITIALIZE JButton as cancelin cointainfu

SET Bounds (230, 280, 120, 40) in a cancel

ADD Action Listner in cancel

ADD img1 in cancel

SET Background Color as LIGHT_GRAY

ADD titJLabel in a cointainfu

ADD lbl1 in a cointainfu

ADD lbl2 in a cointainfu

ADD lbl3 in a cointainfu

ADD lbl5 in a cointainfu

ADD inputvano in a cointainfu

ADD inputde in a cointainfu

ADD inputjb in a cointainfu

ADD inputwh in a cointainfu

ADD hire in a cointainfu

```
        ADD cancel in a cointainfu

        ADD lbl6 in a cointainfu
        ADD lbl7 in a cointainfu

        IF(choice==”full time”){
        DO
        ADD lbl4 in a cointainfu
        ADD inputsa in a cointainfu
        END DO
        END IF
        }

        IF (choice==”part time”){
        DO
        ADD lbl6 in a cointainfu
        ADD lbl7 in a cointainfu
        ADD inputwg in a cointainfu
        ADD inputsh in a cointainfu
        END DO
        END IF
        }
        ADD cointainfu in frame1
        DECLARE frame1 as pack
        SET DefaultCloseOperation(JFrame.EXIT_ON_CLOSE) in frame1
        SET Visible true in frame1
        END DO
}
```

.

4.15 **Method Name** Fulltimeapoint ()

FUNCTION Fulltimeapoint (){

DO

INITIALIZE JFrame as frame2

SET Preferred Size(500,380); in frame2

SET title as Appoint Staff in frame2

INITIALIZE JPanel as cointainfuap

SET Layout null in cointainfuap

SET Background Color as DARK_GRAY in cointainfuap

ADD ImageIcon as img1

INITIALIZE JLabel as titJlabel in cointainfuap

SET Bounds (150,0,250,40) in a titJlabel

SET Fonts SansSeri and size 38 in titJlabel

SET Foreground Color as White as in titJLabel

INITIALIZE JLabel as lbl1in cointainfuap

SET Bounds (140, 110, 300, 20) in a lbl1

SET Fonts SansSeri and size 15 in lbl1

SET Foreground Color as LIGHT_GRAY as in lbl1

INITIALIZE JTextField as inputvcba in cointainfuap

SET Bounds (150, 140, 180, 30) in a inputvcba

SET Fonts Calibri and size 15 in inputvcba

ADD titJLabel in a cointainfuap

ADD lbl1 in a cointainfuap

ADD inputvcba in a cointainfuap

ADD cointainfuap in a frame2

```
IF(choice==”full time”){
DO
INITIALIZE JButton as submit1 in cointainfuap
SET Bounds (180, 180, 120, 40) in a submit1
ADD Action Listner in submit1
ADD img1 in submit1
SET Background Color as LIGHT_GRAY
ADD submit1 in a cointainfuap
END DO
END IF
}

IF (choice==”part time”){
DO
INITIALIZE JButton as submit2 in cointainfuap
SET Bounds (180, 180, 120, 40) in a submit2
ADD Action Listner in submit2
ADD img1 in submit2
SET Background Color as LIGHT_GRAY
ADD submit2 in a cointainfuap
END DO
END IF
}
ADD cointainfuap in frame2
DECLARE frame2as pack
SET DefaultCloseOperation(JFrame. DISPOSE_ON_CLOSE) in frame2
SET Visible true in frame2
END DO
}
```

4.16 **Method Name** InsideHiring ()

FUNCTION InsideHiring (){

DO

INITIALIZE JFrame as framein

SET Preferred Size(700,380); in framein

SET title as Appoint Staff in framein

INITIALIZE JPanel as InformationStaff

SET Layout null in InformationStaff

SET Background Color as DARK_GRAY in InformationStaff

ADD ImageIcon as img1,img

INITIALIZE JLabel as titJlabel in InformationStaff

SET Bounds (250,0,250,40) in a titJlabel

SET Fonts SansSeri and size 38 in titJlabel

SET Foreground Color as White as in titJLabel

INITIALIZE JLabel as lblvacancy in InformationStaff

INITIALIZE JLabel as lbljobtype in InformationStaff

INITIALIZE JLabel as lblDesigna in InformationStaff

INITIALIZE JLabel as lblstaffName in InformationStaff

INITIALIZE JLabel as lblquali in InformationStaff

INITIALIZE JLabel as lblworkHour in InformationStaff

INITIALIZE JLabel as lbljoinDate in InformationStaff

INITIALIZE JLabel as lblappol in InformationStaff

INITIALIZE JLabel as lblshift in InformationStaff

INITIALIZE JLabel as lblwageper in InformationStaff

INITIALIZE JLabel as lblsalary in InformationStaff

DECLARE String salar = Integer.toString(realSalary)

DECLARE String hour = Integer.toString(realWorkHour)

```
DECLARE  String wage = Integer.toString(realWage)
DECLARE  String vac = Integer.toString(realVacancy)

INITIALIZE JTextField as vivac in InformationStaff
INITIALIZE JTextField as vijobtype in InformationStaff
INITIALIZE JTextField as videsign in InformationStaff
INITIALIZE JTextField as vistaff in InformationStaff
INITIALIZE JTextField as viquali in InformationStaff
INITIALIZE JTextField as viworkHour in InformationStaff
INITIALIZE JTextField as vijoindate in InformationStaff
INITIALIZE JTextField as viappol in InformationStaff
INITIALIZE JTextField as vishift in InformationStaff
INITIALIZE JTextField as viwageper in InformationStaff
INITIALIZE JTextField as visalary in InformationStaff

SET Editable as false in visalary
SET  Text as salar in visalary
SET Editable as false in viwageper
SET  Text as wage in viwageper
SET Editable as false in viworkHour
SET  Text as hour in viworkHour
SET Editable as false in vivac
SET  Text as vac in vivac
SET Editable as false in vijobtype
SET  Text as realJob in vijobtype
SET Editable as false in videsign
SET  Text as realDesignation in videsign
SET Editable as false in vishift
SET  Text as realShift in vishift

SET Bounds (20, 60, 150, 20) in a lblvacancy
```

SET Fonts SansSeri and size 15 in lblvacancy

SET Foreground Color as LIGHT_GRAY as in lblvacancy

SET Bounds (200, 60, 150, 20) in a lbljobtype

SET Fonts SansSeri and size 15 in lbljobtype

SET Foreground Color as LIGHT_GRAY as in lbljobtype

SET Bounds (360, 60, 150, 20)) in a lblDesigna

SET Fonts SansSeri and size 15 in lblDesigna

SET Foreground Color as LIGHT_GRAY as in lblDesigna

SET Bounds (520, 130, 150, 20) in a lblappol

SET Fonts SansSeri and size 15 in lblappol

SET Foreground Color as LIGHT_GRAY as in lblappol

SET Bounds (20, 130, 150, 20) in a lblstaffName

SET Fonts SansSeri and size 15 in lblstaffName

SET Foreground Color as LIGHT_GRAY as in lblstaffName

SET Bounds (200, 130, 150, 20)in a lblquali

SET Fonts SansSeri and size 15 in lblquali

SET Foreground Color as LIGHT_GRAY as in lblquali

SET Bounds (520, 60, 150, 20) in a lblworkHour

SET Fonts SansSeri and size 15 in lblworkHour

SET Foreground Color as LIGHT_GRAY as in lblworkHour

SET Bounds (360, 130, 150, 20) in a lbljoinDate

SET Fonts SansSeri and size 15 in lbljoinDate

SET Foreground Color as LIGHT_GRAY as in lbljoinDate

```
SET Bounds (20, 200, 150, 20)in a lblshift
SET Fonts SansSeri and size 15 in lblshift
SET Foreground Color as LIGHT_GRAY as in lblshift
SET Bounds (200, 200, 150, 20) in a lblwageper
SET Fonts SansSeri and size 15 in lblwageper
SET Foreground Color as LIGHT_GRAY as in lblwageper

SET Bounds (20, 200, 150, 20) in a lblsalary
SET Fonts SansSeri and size 15 in lblsalary
SET Foreground Color as LIGHT_GRAY as in lblsalary

SET Bounds (150, 140, 180, 30) in a inputvcba
SET Fonts Calibri and size 15 in inputvcba

SET Bounds (150, 140, 180, 30) in a inputvcba
SET Fonts Calibri and size 15 in inputvcba

SET Bounds (150, 140, 180, 30) in a inputvcba
SET Fonts Calibri and size 15 in inputvcba

SET Bounds (150, 140, 180, 30) in a inputvcba
SET Fonts Calibri and size 15 in inputvcba

SET Bounds (150, 140, 180, 30) in a inputvcba
SET Fonts Calibri and size 15 in inputvcba

SET Bounds (150, 140, 180, 30) in a inputvcba
SET Fonts Calibri and size 15 in inputvcba

SET Bounds (150, 140, 180, 30) in a inputvcba
SET Fonts Calibri and size 15 in inputvcba
```

SET Bounds (150, 140, 180, 30) in a inputvcba
SET Fonts Calibri and size 15 in inputvcba

SET Bounds (150, 140, 180, 30) in a inputvcba
SET Fonts Calibri and size 15 in inputvcba

SET Bounds (150, 140, 180, 30) in a inputvcba
SET Fonts Calibri and size 15 in inputvcba

SET Bounds (150, 140, 180, 30) in a inputvcba
SET Fonts Calibri and size 15 in inputvcba

INITIALIZE JButton as infoSave in InformationStaff
SET Bounds (420, 290, 120, 40)in a infoSave
ADD Action Listner in infoSave
ADD img1 in infoSave
SET Background Color as LIGHT_GRAY

INITIALIZE JButton as infoClearin InformationStaff
SET Bounds (420, 290, 120, 40)in a infoClear
ADD Action Listner in infoClear
ADD img in infoSave
SET Background Color as LIGHT_GRAY

ADD titJLabel in a InformationStaff
ADD lblvacancy in a InformationStaff
ADD lbljobtype in a InformationStaff
ADD lblDesigna in a InformationStaff

```
ADD lblappol in a InformationStaff
ADD lblstaffName in a InformationStaff
ADD lblquali in a InformationStaff
ADD lblworkHour in a InformationStaff
ADD lbljoinDate in a InformationStaff

ADD vivac in a InformationStaff
ADD vijobtype in a InformationStaff
ADD videsign in a InformationStaff
ADD viappol in a InformationStaff
ADD vistaff in a InformationStaff
ADD viquali in a InformationStaff
ADD viworkHour in a InformationStaff
ADD vijoindate in a InformationStaff

IF(choice==”full time”){
DO
ADD lblsalary in a InformationStaff
ADD visalary in a InformationStaff
SET Editable as false in visalary
SET  Text as realShift in visalary

END DO
END IF
}

IF (choice==”part time”){
DO
ADD lblshift in a InformationStaff
ADD vishift in a InformationStaff
ADD lblwageper in a InformationStaff
```

ADD viwageper in a InformationStaff

END DO

END IF

}

ADD InformationStaff in framein

DECLARE framein as pack

SET DefaultCloseOperation(JFrame. DISPOSE_ON_CLOSE) in framein

SET Visible true in framein

END DO

}

4.17 **Method Name** termGui ()

FUNCTION termGui (){

DO

INITIALIZE JFrame as frametm

SET Preferred Size(500,380); in frametm

SET title as Terminate Staff in frametm

INITIALIZE JPanel as terminatepanel

SET Layout null in terminatepanel

SET Background Color as DARK_GRAY in terminatepanel

ADD ImageIcon as img1

INITIALIZE JLabel as titJlabel in terminatepanel

SET Bounds (150,0,250,40) in a titJlabel

SET Fonts SansSeri and size 38 in titJlabel

SET Foreground Color as White as in titJLabel

INITIALIZE JLabel as lbltem in terminatepanel
SET Bounds (140, 110, 300, 20) in a lbltem
SET Fonts SansSeri and size 15 in lbltem
SET Foreground Color as LIGHT_GRAY as in lbltem

INITIALIZE JTextField as inputtem in terminatepanel
SET Bounds (150, 140, 180, 30) in a inputtem
SET Fonts Calibri and size 15 in inputtem

INITIALIZE JButton as btnterminate in terminatepanel
SET Bounds (180, 180, 120, 40) in a btnterminate
ADD Action Listner in btnterminate
ADD img1 in btnterminate
SET Background Color as LIGHT_GRAY
ADD titJLabel in a terminatepanel
ADD lbl1 in a terminatepanel
ADD inputtem in a terminatepanel
ADD btnterminate in a terminatepanel

ADD terminatepanel in frametm
DECLARE frametm as pack
SET DefaultCloseOperation(JFrame. DISPOSE_ON_CLOSE) in frametm
SET Visible true in frametm
END DO
}

4.18 **Method Name** displaymethod ()
FUNCTION displaymethod (){

DO

INITIALIZE JFrame as framedis

SET Preferred Size(600,580); in framedis

SET title as display in framedise

INITIALIZE JPanel as panelDisplayJPanel

SET Layout null in panelDisplayJPanel

INITIALIZE JLabel as titJlabel in Scroll

SET Fonts SansSeri and size 18 in titJlabel

SET Foreground Color as White as in titJLabel

SET Opaque true in a titJlabel

SET Backgrounf Color as DARK_GRAY as in titJLabel

CALL fullTimeData()+partTimeData() in titJLabel

INITIALIZE JScrollPane as scroll in panelDisplayJPanel

SET ViewreportView as titJlabel in scroll

SET HorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS) in scroll

SET VerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS) in scroll

SET Bounds(5, 50, 575, 470) in scroll

INITIALIZE JLabel as DisplayLabel in panelDisplayJPanel

SET Bounds (220, 0, 250, 40) In DisplayLabel

SET Fonts SansSeri and size 38 in DisplayLabel

SET Foreground Color as White as in DisplayLabel

ADD DisplayLabel in panelDisplayJPanel

ADD scroll in panelDisplayJPanel

ADD panelDisplayJPanel in framedis

```
DECLARE framedisas pack
SET DefaultCloseOperation(JFrame. DISPOSE_ON_CLOSE) in framedis
SET Visible true in framedis
END DO
}
```

4.19 **Method Name** fullTimeData ()

```
FUNCTION fullTimeData (){
DO
ASSIGN String  Com=”full time”
FOR ( StaffHire temp : ingArrayList){
IF(niii instanceof FullTimeStaffHire){
DECLARE fulhi = FullTimeStaffHire of nii
 SET com = com +"Vacancy no. "+temp.getVacancyNo();
SET com = com +"Staff Name : "+ objdisplay.getstaffName()
      SET     com = com +"Qualification : "+ objdisplay.getqualification()
          SET com = com +"Job Type : "+ temp.getJobType()
         SET com = com +"Designation : " + temp.getDesignation(;

         SET com = com +"Salary : "+ objdisplay.getsalary()
          SET com = com +"Work Hour : "+ objdisplay.getworkHour()

          SET com = com +"Join Date : "+ objdisplay.getjoinDate()

          SET com = com +"Appointed By : "+ objdisplay.getappointedBy()
        SET    com = com +" “
END IF
        }
END FOR
     }
```

```
        RETURN com;
    END DO
     }
```

4.20 **Method Name** partTimeData ()

```
    FUNCTION partTimeData (){
    DO
    ASSIGN String  Com=" PART TIME"
    FOR( StaffHire temp : ingArrayList){
            IF(temp instanceof PartTimeStaffHire){
    DECLARE  objdisplay = PartTimeStaffHire of temp
            SET com = com +" Vacancy no. "+temp.getVacancyNo()

            SET com = com +"Staff Name : "+objdisplay.getstaffName()
            SET com = com +"Qualification : "+ objdisplay.getqualification()
            SET com = com +"Job Type : "+temp.getJobType()
            SET com = com +"Designation : " + temp.getDesignation()

            SET com = com +"Work Hour : "+objdisplay.getworkHour()
            SET com = com +"Wage per Hour : "+ objdisplay.getwagePerHour()
            SET com = com +"Shift : "+ objdisplay.getshifts()

            SET com = com +"Join Date : "+objdisplay.getjoinDate()

            SET com = com +"Appointed By : "+objdisplay.getappointedBy()
            SET com = com +""
    END IF
            }
    END FOR
```

```
        }
        RETURN com+"</html>";
    END DO
    }
```

4.21 **Method Name** vacancyforward ()

```
    FUNCTION vacancyforward (){
    DO
    DECLARE boolean Cck s false
    IF(choice==”full time”){
    DO
    IF (inputvano= “ “ OR inputde = “ “ OR inputde= “ “ OR inputwh = “ “{
    DO
    JOptionPane.showMessageDialog(frame1, "Please fill out all the text box", "main",
    JOptionPane.INFORMATION_MESSAGE);
            SET  Cck=true
    END DO
    END IF
    }
    END IF
    END DO
    }
    IF(choice==( "Part Time"){
    DO
    IF (inputvano= “ “ OR inputde = “ “ OR inputde= “ “ inputsh = “ “ inputwg= “ “ OR
    inputwh = “ “{
    DO
    JOptionPane.showMessageDialog(frame1, "Please fill out all the text BOX",
    "main", JOptionPane.INFORMATION_MESSAGE);
            SET  Cck=true
```

```
END DO
END IF
}
END IF
END DO
}
IF(Cck=false){
DO
TRY{
IF (choice.equals("Full Time")) {
DO
SET objfu= new FullTimeStaffHire(vacancyFrom(), designationFrom(), choice,
salaryFrom(), workHourFrom());
                    ingArrayList.add(objfu);
END DO
END IF
 }
IF (choice.equals("Part Time")) {
DO
SET objpa= new PartTimeStaffHire(vacancyFrom(), designationFrom(), choice,
workHourFrom(), wagePHourFrom(), shiftFrom());
                    ingArrayList.add(objpa);
END DO
END IF
}END TRY
CATCH(NumberFormatException UR){
DO
JOptionPane.showMessageDialog(frame1,exeError!",
JOptionPane.WARNING_MESSAGE);
END DO
```

```
END CATCH
                }
END IF
END DO
            }
END IF
END DO

        }
```

4.22 **Method Name** backendterminat ()

```
FUNCTION backendterminat (){
DO
DECLARE boolean  temboolean as false
TRY{
DO

 FOR (staffHire termi as ingArray){

 IF (termi instanceof PartTimeStaffHire) {

 SET objTem = (PartTimeStaffHire)
IF (VacancyNo() == inputtemret()){
IF (joined() == true) {
SET  flagInTerminate as true;
CALL  objTem.terminate }
END IF
IF{
SET  JOptionPane.showMessageDialog(frame, "No Staff has been appointed in order to terminate", "Error!", JOptionPane.INFORMATION_MESSAGE);
```

```
SET terbo as false;
                }
            }
        }
    }
    IF (flagInTerminate==false AND terbo==true){
SET JOptionPane.showMessageDialog(frame,"No Valid record found for termination","Error!",JOptionPane.WARNING_MESSAGE);
    }

    }
END TRY
CATCH ( Exception err) {
    SETJOptionPane.showMessageDialog(frametm,err ,"Error!",JOptionPane.WARNING_MESSAGE); END CATCH
END IF
                }
    END DO
        }
```

4.23 **Method Name** repetationStaff ()

```
FUNCTION repetationStaff (){
DO
DECLARE boolean  nimcheck as false
TRY {
FOR (StaffHire temp: ingArrayList) {
IF (VacancyNo() == vacancyFrom(){
 SET JOptionPane.showMessageDialog(frame1, "This vacancy no. has already been used", "Error!", JOptionPane.INFORMATION_MESSAGE);
  SET nimcheck as true
```

```
BREAK
END IF
}
END FOR
}
    IF(nimcheck) {
   END IF
   } IF{
  CALL vacancyforward()
      SET frame1.dispose()
     SET  frame.dispose()
    CALL   GuiBox()
END IF
}
END IF
END TRY
} CATCH (Exception e) {
IF (inputvano="" OR  inputde ="" OR  inputwg="" OR  inputsh="" OR
CALL vacancyforward();
END IF}
ELSE{
 SET
JOptionPane.showMessageDialog(frame1,e,"Error!",JOptionPane.ERROR_MESSAGE)
}END ELSE
} END CATCH
}END IF
END DO
```

4.24 **Method Name** dataStore ()

```
FUNCTION dataStore (){
DO
IF(choice==("Full Time")){
FOR(StaffHire ckob:ingArrayList){
SET fulob= (FullTimeStaffHire) ckob;
  IF(VacancyNo()==realVacancy) {
 IF(getjoined()==false){
SET
fullhire(staffNameFrom(),joingDateFrom(),qualificationFrom(),appointeByFrom());
  SET    JOptionPane.showMessageDialog(framein,"Staff has been hired!");
END IF
}
END IF
}
END FOR
}
END IF

IF(choice==(" Part Time")){
FOR(StaffHire ckob:ingArrayList){
SET paob= (PartTimeStaffHire) ckob;
  IF(VacancyNo()==realVacancy) {
 IF(getjoined()==false){
SET
part (staffNameFrom(),joingDateFrom(),qualificationFrom(),appointeByFrom());
  SET  JOptionPane.showMessageDialog(framein,"Staff has been hired!");
END IF
}
```

```
END IF
}
END FOR
}
END IF
END DO
```

4.25 **Method Name** dataCheck ()

```
FUNCTION dataCheck (){
DO
FOR (StaffHire datac : staffVacancy) {
        IF (VacancyNo() == Integer.parseInt(inputvcba.getText())) {
INITIALIZE JFrame as Check

SET Preferred Size(500,380); in Check
SET title as conform, in Check

INITIALIZE JPanel as panelck

SET Layout null in panelck
SET Background Color as DARK_GRAY in panelck
ADD ImageIcon as img1

INITIALIZE JLabel as titJlabel in panelck
SET Bounds (150,0,250,40) in a panelck
SET Fonts SansSeri and size 38 in panelck
SET Foreground Color as White as in panelck

INITIALIZE JLabel as lbl1in panelck
SET Bounds (140, 110, 300, 20) in a lbl1
```

SET Fonts SansSeri and size 15 in lbl1

SET Foreground Color as LIGHT_GRAY as in lbl1

INITIALIZE JButton as ckbtn in panelck

SET Bounds (80, 280, 120, 40) in a ckbtn

ADD img in ckbtn

ADD Action Listner in ckbtn

SET Background Color as LIGHT_GRAY

INITIALIZE JButton as ckcl panelck

SET Bounds (230, 280, 120, 40) in a ckcl

ADD Action Listner in ckcl

ADD img1 in ckcl

SET Background Color as LIGHT_GRAY

ADD titJLabel in a panelck

ADD lbl1 in a panelck

ADD ckbtn in a panelck

ADD ckbtn in a Check

ADD panelck in Check

DECLARE Check as pack

SET DefaultCloseOperation(JFrame. DISPOSE_ON_CLOSE) in Check

SET Visible true in Check

END DO

4.26 **Method Name** actionPerformed ()

FUNCTION actionPerformed (){

DO

CALL ActionEvent e

DECLEARE throws ConcurrentModificationException

```
IF (e.getSource()==Fullva){
  SET choice = "Full Time"
  CALL Fulltimeva()
  END IF
        }
IF (e.getSource()==Fullap){
SET  choice = "Full Time";
CALL Fulltimeapoint();
END IF
        }
IF (e.getSource()==Partva){
  SET choice = " Part Time"
  CALL Fulltimeva()
  END IF
        }

 (e.getSource()==Partap){
SET  choice = "Part Time";
CALL  Fulltimeapoint();
      END IF
        }

        IF (e.getSource()==ckcl){
            SET  frameck.dispose()
    END IF
        }
```

```
IF (e.getSource()==terminate){
                CALL termGui();
                     END IF
          }
IF (e.getSource()==btnterminate){
        IF (inputtem.getText().equals("")){
          SET JOptionPane.showMessageDialog(frametm,"Please fill out the Text
Field","Info",JOptionPane.INFORMATION_MESSAGE);
END IF
        }ELSE {
          CALL   backendterminat();
        }
END ELSE
     }
IF (e.getSource()==display){
  CALL      displaymethod();
END IF

     }
IF (e.getSource()==hire){
                CALL   repetationStaff();
END IF

     }
IF (e.getSource()==cancel){
        SET  inputvano.setText("");
    SET     inputsh.setText("");
     SET     inputwh.setText("");
     SET     inputde.setText("");
    SET     inputwg.setText("");
    SET     inputsa.setText("");
```

```
END IF

    }
        IF (e.getSource()==infoClear){
   SET      viappol.setText("");
   SET      viquali.setText("");
  SET       vistaff.setText("");
 SET        vijoindate.setText("");
END IF

    }

        IF (e.getSource()==infoSave){
                IF  (vistaff.getText().equals("")OR  viquali.getText().equals("")  OR
viappol.getText().equals("")OR vijoindate.getText().equals("")){
       SET    JOptionPane.showMessageDialog(framein,"Please fill out all the text
fields.","ERROR",JOptionPane.ERROR_MESSAGE);
END IF
       }ELSE {
          SET framein.dispose();
          CALL dataStore();
END ELSE
          }END IF
                }
IF (e.getSource()==submit1){
       TRY {
       SET    boolean falgINbtn = false
          FOR (StaffHire temp : ingArray)
             IF(temp.getVacancyNo() == Integer.parseInt(inputvcba.getText())AND
temp.getJobType().equals("Full Time")) {
              SET   falgINbtn = true;
```

```
                SET     realVacancy = temp.getVacancyNo();
                SET     realDesignation = temp.getDesignation();
                SET     realJob = temp.getJobType();
                SET     FullTimeStaffHire objBtnfull = (FullTimeStaffHire) temp;
                SET    realSalary = objBtnfull.getsalary();
               SET     realWorkHour = objBtnfull.getworkHour();
                CALL  dataCheck();
END IF
              }
END FOR
          }
          IF (falgINbtn == false) {

            SET   JOptionPane.showMessageDialog(frame2, "No vacancy found",
"Message", JOptionPane.ERROR_MESSAGE);
          }
END IF
END TRY
        }CATCH(Exception ee){
          IF(inputvcba.getText().equals("")){
          SET   JOptionPane.showMessageDialog(framein,"Please fill out the Text
Field","Info",JOptionPane.ERROR_MESSAGE);
          }
          ELSE{
         SET     JOptionPane.showMessageDialog(framein,exe);
END ELSE
          }
END CATCH
        }
END IF
     }
```

```
IF (e.getSource()==submit2){
        TRY {
            boolean falgINbtn = false;
            FOR(StaffHire temp: ingArray) {
                IF (temp.getVacancyNo() == Integer.parseInt(inputvcba.getText()) ) {
                  IF (temp.getJobType()=="Part Time") {
                  SET       falgINbtn = true;
                     SET    realVacancy = temp.getVacancyNo();
                   SET      realDesignation = temp.getDesignation();
                  SET       realJob = temp.getJobType();
                  SET       PartTimeStaffHire objBtnpart = (PartTimeStaffHire) temp;
                  SET       realWorkHour = objBtnpart.getworkHour();
                 SET        realWage = objBtnpart.getwagePerHour();
                 SET        realShift = objBtnpart.getshifts();
                CALL      dataCheck();
                   }
END IF
                }
             IF ( falgINbtn == false){
              SET   JOptionPane.showMessageDialog(frame2,"No   vacancy   found"
,"Message",JOptionPane.ERROR_MESSAGE);
            }
        END TRY
 }END CATCH (Exception exe){
            IF (inputvcba.getText().equals("")){
           SET     JOptionPane.showMessageDialog(framein,"Please fill out the Text
Field","Info",JOptionPane.ERROR_MESSAGE);
            }
            ELSE{
            SET         JOptionPane.showMessageDialog(framein,exe);
END ELSE
```

```
            }
END CATCH
        }
END IF
    }
IF (e.getSource()==ckbtn){
      SET   frame2.dispose();
        IF(choice=="Full Time") {
            FOR (StaffHire temp : ingArray) {
                IF(temp instanceof FullTimeStaffHire) {  // This checks whether object
is an instance of FullTimeStaffHire subclass.
                    SET      obful = (FullTimeStaffHire) temp;
                    IF(obful.getVacancyNo()==realVacancy) {
             SET  (obful.getjoined() == false) {
                     CALL       InsideHiring();
                     SET       frameck.dispose();
.
END IF
                    } ELSE{
                    SET          JOptionPane.showMessageDialog(frame,  "Staff  has
already        been        appointed        for        this        vacancy",        "Info",
JOptionPane.INFORMATION_MESSAGE);
                      SET  frameck.dispose();
                    }
END ELSE
                    }
                }
            }
        }
        IF (choice=="Part Time") {
            FOR (StaffHire temp : ingArray) {
```

```
                IF (temp instanceof PartTimeStaffHire) { // This checks whether object
is an instance of PartTimeStaffHire subclass.
                    PartTimeStaffHire obpar = (PartTimeStaffHire) temp;
                    IF (obpar.getVacancyNo()==realVacancy) {
                    IF (obpar.getjoined() == false) {
                        CALL InsideHiring();
                       SET  frameck.dispose();
                    } ELSE {
SET  JOptionPane.showMessageDialog(frame, "Staff has already been appointed
for this vacancy", "Info", JOptionPane.INFORMATION_MESSAGE);
                    }
.END ELSE
                    }
END IF
                }
END IF
            }
END IF
        }
END FOR
    }
END DO
   //
 }
```

5.Method Descprition

5.1 Method Description vacancyFrom():

This method uses to convert the vacancy number of String value provide by user to interger and return it.

5.2 Method Description designationFrom():

This method uses return designation.

5.3 Method Description workHourFrom():

This method uses to convert the String value of work hour provide by user to interger and return it.

5.4 Method Description salaryFrom():

This method uses to convert the String value of salary provide by user to interger and return it.

5.5 Method Description shiftFrom():

This method uses return Shift.

5.6 Method Description wagePHourFrom():

This method uses to convert the String value of wage per hour provide by user to interger and return it.

5.7 Method Description staffNameFrom():

This method uses return Staff Name.

5.8 Method Description appointedFrom() :

This method uses return Appoint By.

5.9 Method Description joiningDateFrom():

This method uses return Joining Date.

5.10 Method Description qualificationFrom():

This method uses return Qualification of staff.

5.11 Method Description inputtermret():

This method uses to convert the String value of vacancy no for terminate provide by user to integer and return it.

5.12 AssignVariables :

This is the main method of program in this method object is creates i.e. ob1 in order to call non static method and guiBox(); method is called here and this Function will run first in the program. And in this method the design of Joption pane is also done.

5.13 Method DescriptionguiBox():

This method is about the home section of the program . this method contains six button action listener is added in button when user click for add vacancy, hire staff, display the details and terminate. This method will display at first because is called in main method

5.14 Method Description fullTimeva():

This method is about the GUI for Add vacancy for both Part Time and Full Time. This methods contains Text Fields and j label for Adding vacancy and two button are there hire and cancel . each button has action listener for adding vacancy and clearing form. If choice is full time then salay will be show in form and if choice is part time then wage per hour and shift will be show in the form

5.15 Method Description fullTimeApoint():

This methods runs when user click on hire full time staff or part time staff. when user add vacancy for any job type and want to hire staff then this methods helps to hire the staff. When user input the vacancy number in textbox then it will search for that particular vacancy number if it is found then it will open hiring form. It like a mediator between vacancy number put by user during hiring and vacancy number put during add vacancy. In this method it cointain Jlabel, JButton, JTextFields all of them are managed by using set keywords

5.16 Method Description insideHiring():

This methods is contains forms for hiring the staff when the vacancy number is found and vacant. In this method there are several JLabe,JTextFields. User can input staffName, qualificlation, appoint by and joining date. And other text fields are non

editiable and the value are called for them from Fulltimeva methods. In this methods some of integers value are convert in to string in order to get values in the text field. The method is runs when user add vacancy and want to appoint staff.

5.17 Method Description termiGui():

This method is for terminate the staff in this method it contains one text field, one button and 2 JLabel. The method is only for GUI of terminate when user click on button the backend will run . backendTerminate method.

5.18 Method Description displayMethod():

In this method it display the details of vacancy and staff for those vacancy. In this method two methods are called data of full time staff hire and part time staff hire is called. When user click in display button in Fultimeva then this methods will run in the program. This method contains scroll pane in order to scroll the display to look more vacany and staff details.

5.19 Method Description fullTimeData():

This method is created in order to display full time staff and vacancy data. In this method when user input the data in each text field and add vacancy and hire staff those value will be save in Arraylist and those saved value in array list for full time staff hire is called in this method in order to display them. In this method there is use of html because to make display result more efficient and clear

5.20 Method Description partTimeData():

This method is created in order to display part time staff and vacancy data. In this method when user input the data in each text field and add vacancy and hire staff those value will be save in Arraylist and those saved value in array list for part time staff hire is called in this method in order to display them. In this method there is use of html because to make display result more efficient and clear

5.21 Method Description vacancyForward() :

The main purpose of this method is to check whether the text box in blank or not it blank it will show please fill out all the text area other wise this method allow the user to save

the input vacancy . repetition vacancy number is also checked in repetition staff method. Repetitions staff method invokes this method

5.22 Method Description backendTerminate():

This method is for the terminate the part time staff. In this method array list value is called. And the object of PArtTimeStaffHire is created when user input the vacancy number in text field if that vacancy number is match in array list for partime staff and also It will check where the vacancy of that vacancy number is hire or not. If it hired then this method terminate the staff for that particular vacancy otherwise it till show no appointment for vacancy. And if vacancy number is not match with the vacancy number in array list for part time then it say vacancy no not found. For this operation different Boolean flags as well as if condition is used. Try catch is used here in order to handle expectation

5.23 Method Description repetationStaff():

This method is for checking whether the vacancy no is already used or not. In this method when user input the vacancy number in textbox then from button this method is called and this method checks for available vacancy if it is already used the dialog box appear and say vacancy number is already used other wise this from this method those value will pass to vacancyforward method as it is called in this method. Try catch is used here in order to handle expectation.

5.24 Method Description dataStrore():

This method work when user want to hire the staff for particular vacancy . initially when vacancy number is match for hiring staff then then new frame in open with the form for hiring staff and when the user input all the data for hiring staff and want to save in array list then this method will work. So the main work of this method is to save the data of staff for particular vacancy. Try catch is used here in order to handle expectation

5.25 Method Description dataCheck():

This method is for the Check where the staff for particular vacancy number is already hire or not. This methods work like a mediator. When user input the vacancy number for hiring staff this method works and if it already hire then this methods show staff is already

hire and if staff is not hired it will allow user to hire the staff . Try catch is used here in order to handle expectation

5.26Method Description actionPerformed():

This method is override method . it is an action listener method in this method action preformed by each component are managed like wen one button is pressed the one method is called . similarly try catch is also used in this method inside action event in order to handle error. And many check are also done in this method like null value pass etc./

6.TESTING

After the completed this project . the testing of project is done according to the requirement of the project . below three testing are done and the output is also tabulated with proper screenshot

6.1 Test1 Run the program using Command Prompt

Objective	Run the program using Command Prompt for learning aid
Action	At first command prompt is opened and the directory of the java file is opened and then StaffHire.java file is compiled, FullTimeStaffHire.java is also compiled ,PartTimeStaffHire.java is also compiled and lastly INGNepal.java is compiled. And INGNepal.java file was run.

Expected Result	During Compile time class file should be created inside the program files and program should open.
Actual Result	Class File was created and program runs
Conclusion	The test was successfully performed and screenshot was given below

Table 1 Test1

OUTPUT

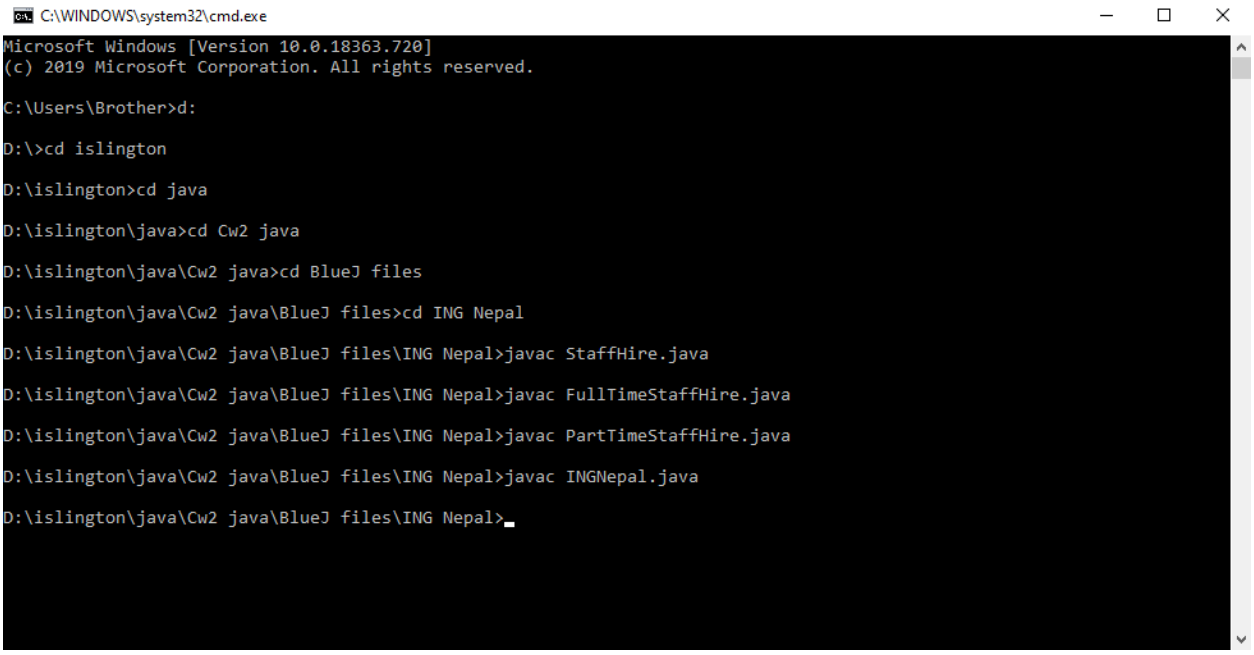
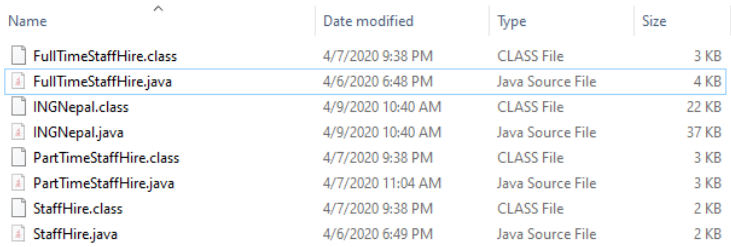


Figure 2 Command Prompt Learning aid – compile



Name	Date modified	Type	Size
FullTimeStaffHire.class	4/7/2020 9:38 PM	CLASS File	3 KB
FullTimeStaffHire.java	4/6/2020 6:48 PM	Java Source File	4 KB
INGNepal.class	4/9/2020 10:40 AM	CLASS File	22 KB
INGNepal.java	4/9/2020 10:40 AM	Java Source File	37 KB
PartTimeStaffHire.class	4/7/2020 9:38 PM	CLASS File	3 KB
PartTimeStaffHire.java	4/7/2020 11:04 AM	Java Source File	3 KB
StaffHire.class	4/7/2020 9:38 PM	CLASS File	2 KB
StaffHire.java	4/6/2020 6:49 PM	Java Source File	2 KB

Figure 3 Command Prompt Learning aid- class file proof

Figure 4 Command prompt learning aid - program running

6.2 TEST 2 Testing of Add vacancy for Full Time Staff and Part Time Staff, Appoint Full Time Staff and Part Time Staff, Terminate Part Time Staff

Objective	To of Add vacancy for Full Time Staff and Part Time Staff, Appoint f Full Time Staff and Part Time Staff, Terminate Part Time Staff.
Action	At first program was run and mouse cursor was clicked over the Add Full time Vacancy and the all the Test Box was Filled Vacancy number=4 Designation=manager Salary=1000 Working Hour =2

	and vacancy was added for Full time. Similarly mouse cursor was clicked over the Add Part Time Vacancy and the all the Test Box was Filled Vacancy number=2 Designation=CEO Working Hour =8 wagesPerHour=1000 shifts= Day and vacancy was added for Part time. And mouse cursor is clicked in Appoint Full time Staff and Vacancy number was entered which was kept during Adding Vacancy for Full Time then hire button is clicked and in confirm box confirm button is pressed and Staff name= Nimesh Poudel Qualification =+2 Appointed by =Pramod Joining date=2020-02-02 was entered. And Save button is pressed. And mouse cursor is clicked in Appoint part time Staff and Vacancy number was entered which was kept during Adding Vacancy for part Time then hire button is

	clicked and in confirm box confirm button is pressed and Staff name = Ram Qualification=MScIT Appointed by= Zyan Joining date=2018-02-09 was entered. And Save button is pressed. Display button was Pressed and mouse cursor was clicked over terminate and vacancy number=2 entered terminate staff button was clicked. Again display button is clicked
Expected Result	While Adding vacancy for full time and part time it should be added and while appointing staff staff should be appoint and while clicking display after appoint all the data should be displayed and while terminate staff . Staff should be removed
Actual Result	Vacancy for both part time and full time is added and staff is also hired for both types of job it is showing when display button is pressed and . when staff is terminate for part time staff is removed
Conclusion	The test was successfully performed and screenshot was given below

Table 2 TEST 2

OUTPUT

Figure 5 Front page of program

Figure 6 Clicked over Add vacancy for full time

Figure 7 Before Adding Vacancy for full time

Figure 8 After filling form to add vacancy for full time

Figure 9 Vacancy is Added for Full time

Figure 10 Clicked over Add Vacancy for Part Time

Figure 11 Before Adding Vacancy for part time

Figure 12 After filling form for Add vacancy part time

Figure 13 Vacancy is added for part time

Figure 14 Clicked over Appoint Full Time Staff

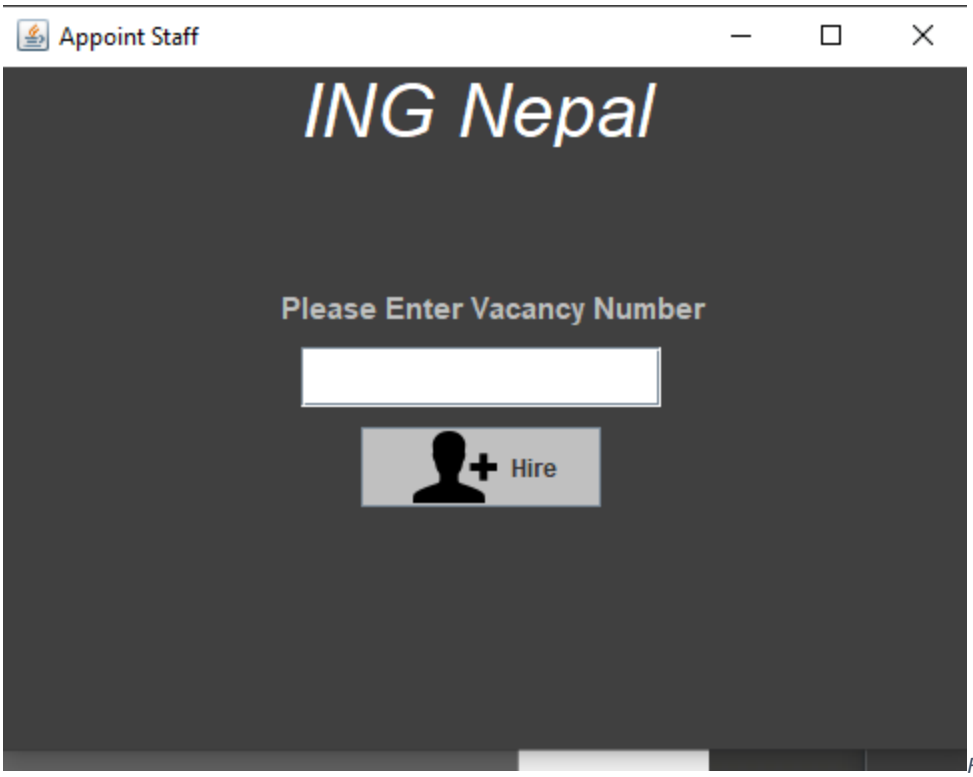


Figure 15 Before entering vacancy number for appoint full time

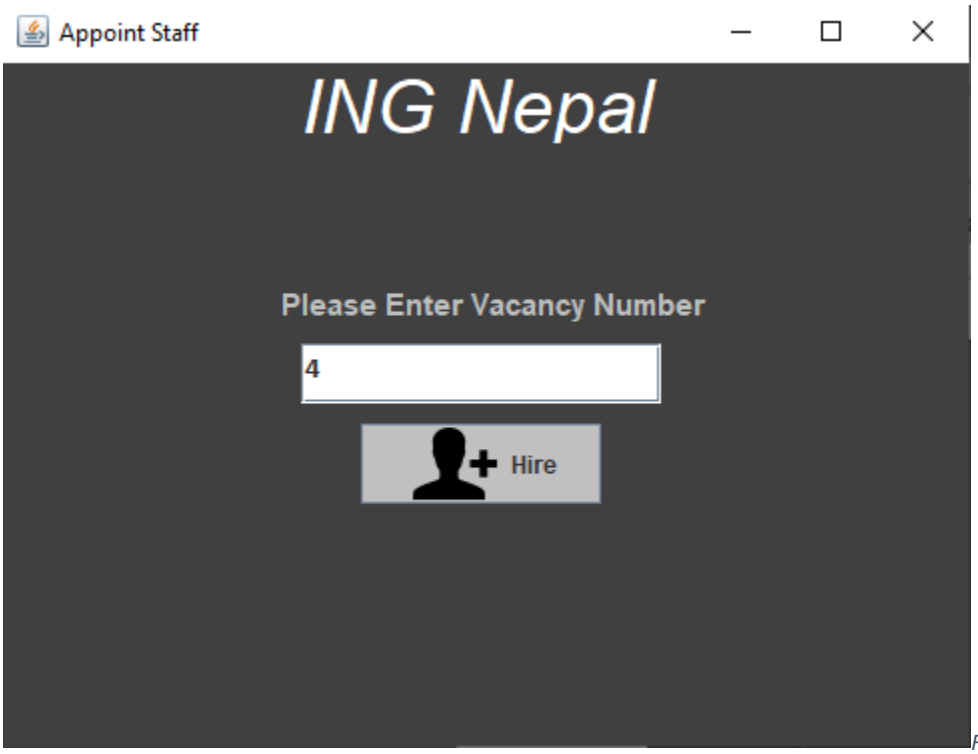


Figure 16 entering vacancy number for hiring staff in full time

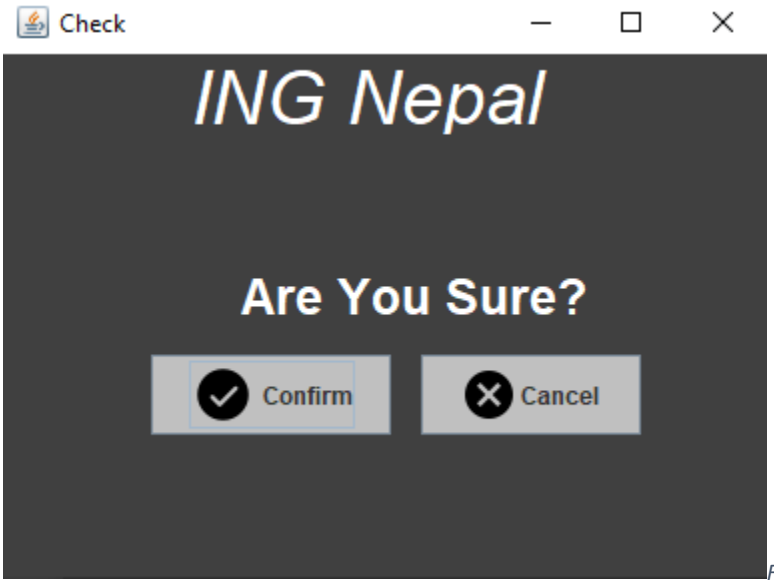


Figure 17 confirm box in full time

Figure 18 Before appointing staff in full time

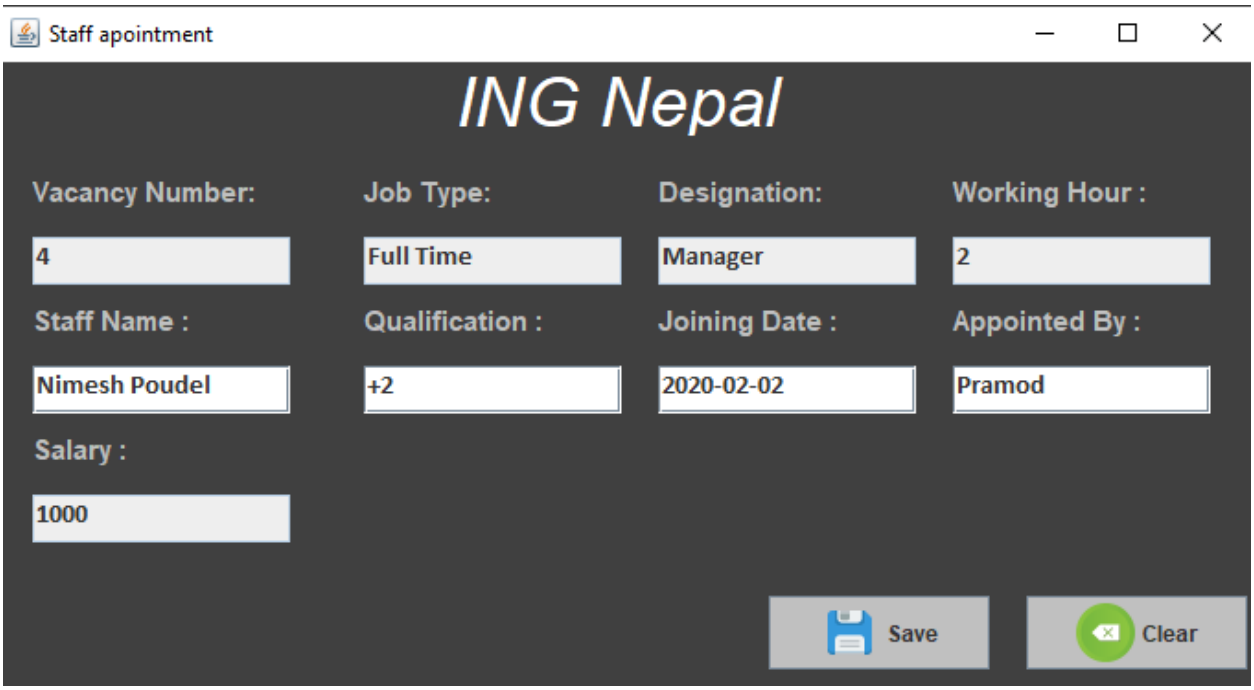


Figure 19 Filling form for appointing staff in full time

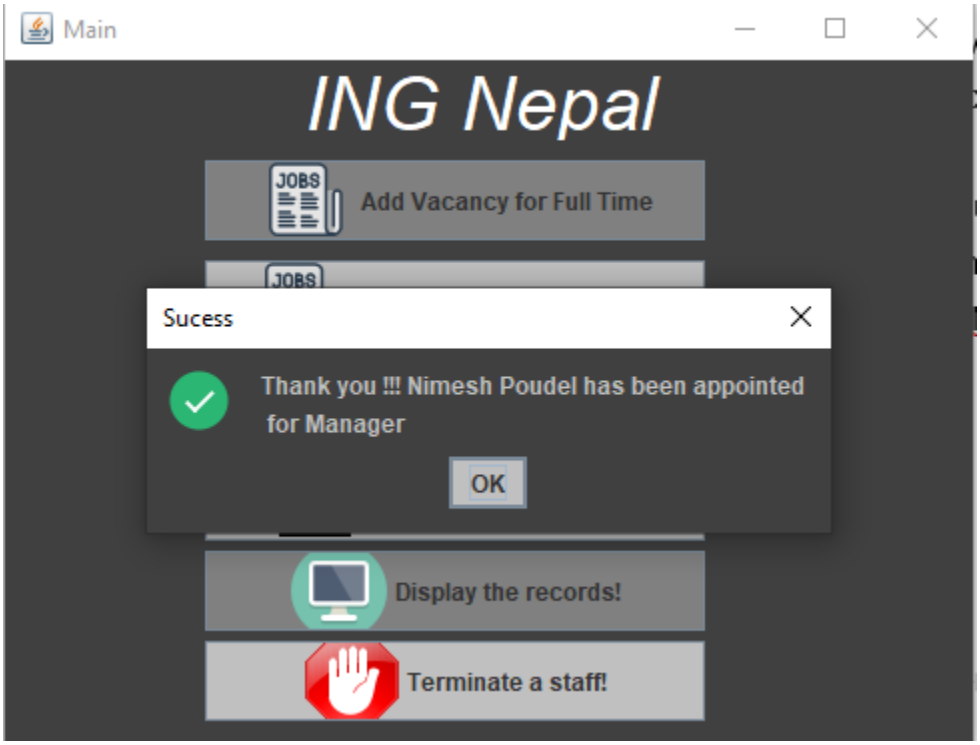


Figure 20 Staff is appointed for full time

Figure 21 clicked over appoint part time staff

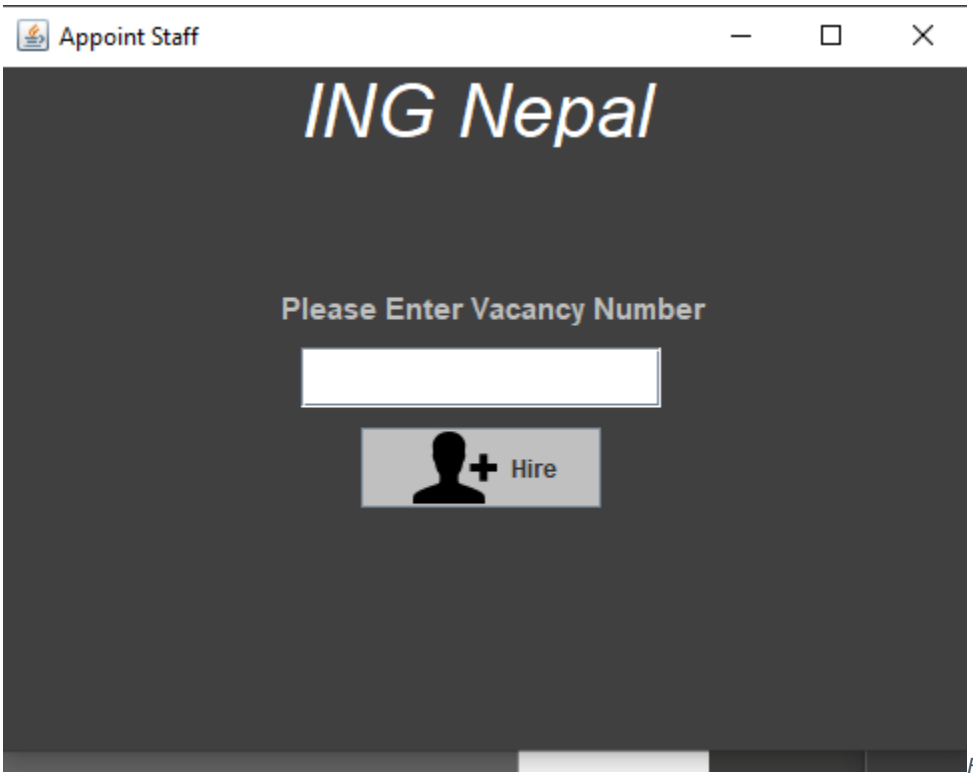


Figure 22 before entering vacancy number for appointing staff

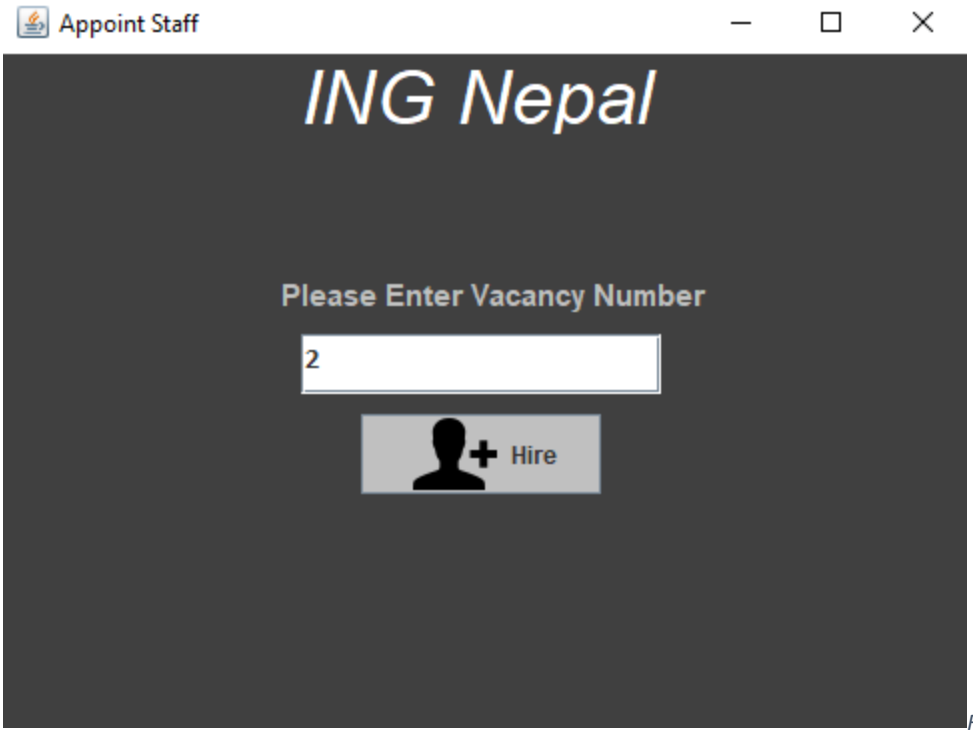


Figure 23 ENtering vacancy number for appointing part time staff

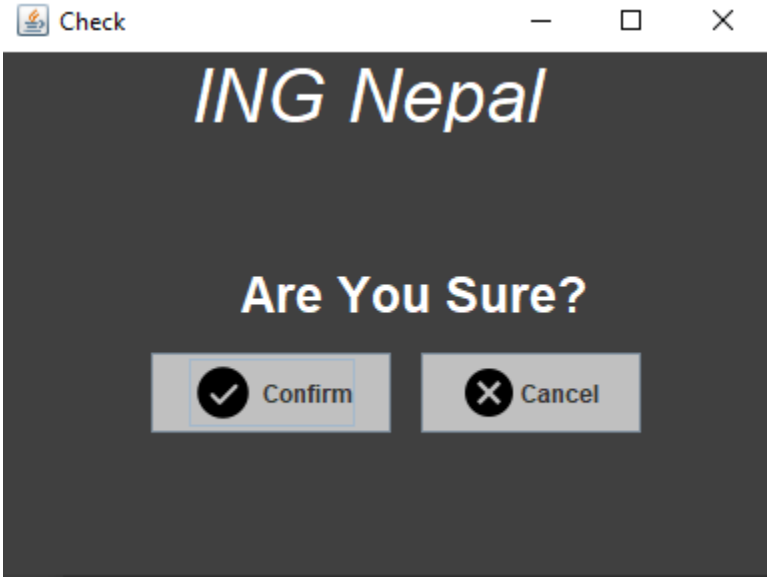


Figure 24 Confirm box in part time staff

Figure 25 Before appointing staff for part time

Figure 26 Filling form for appointing staff for part time

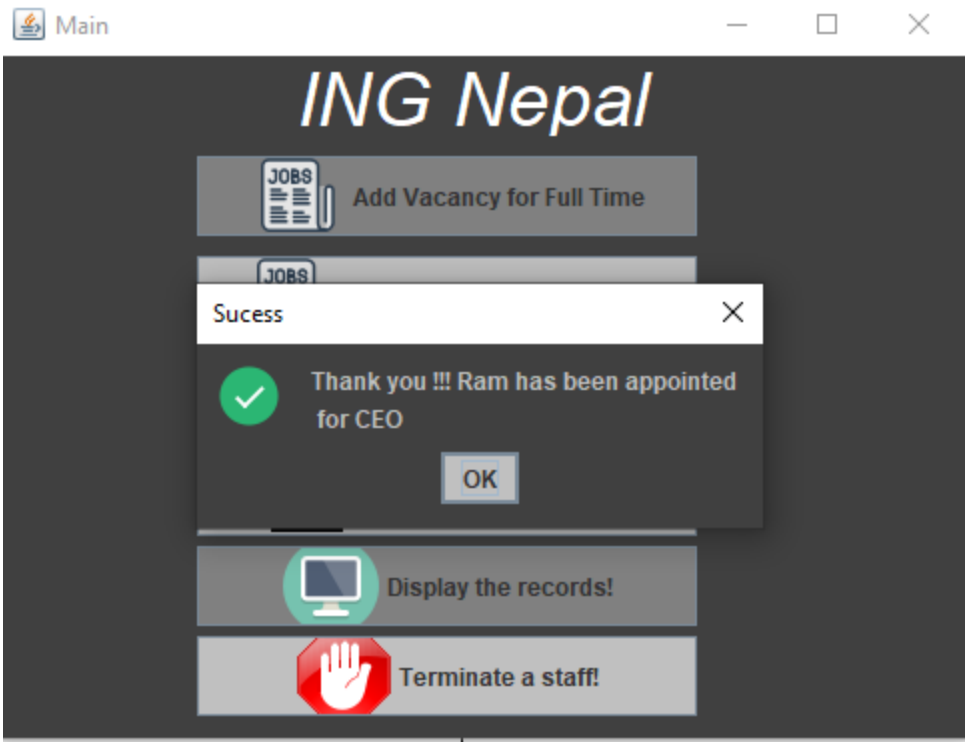


Figure 27 Staff is appointed for part time

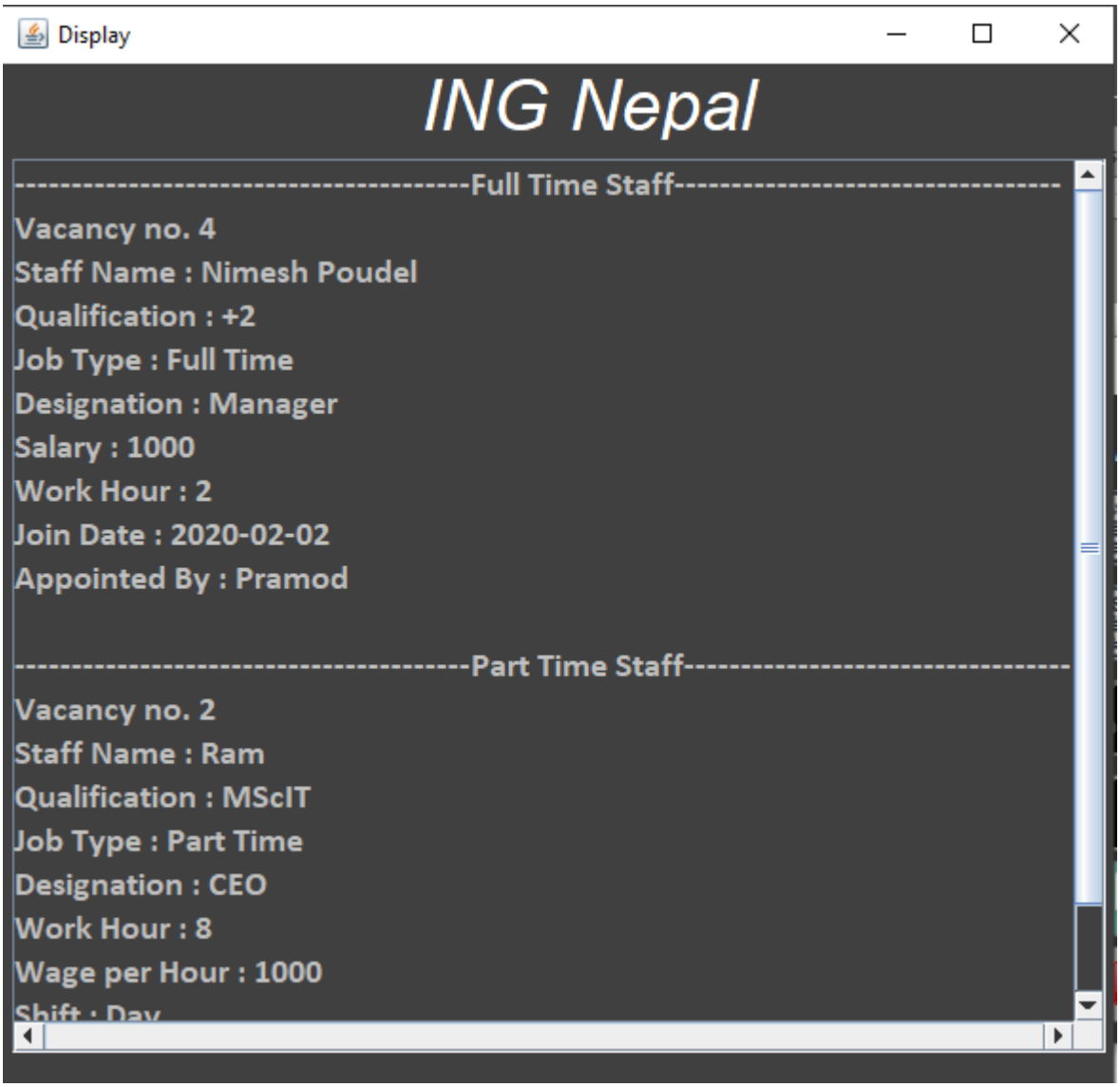


Figure 28 DIsplay staff details

Figure 29 display staff details

Figure 30 clicked over terminate staff

Figure 31 Vacancy number for termination

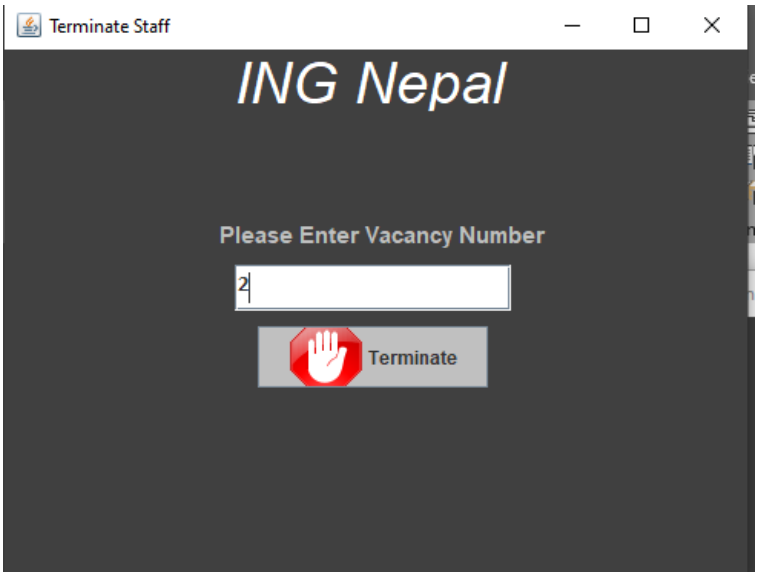


Figure 32 Entering vacancy number for termination

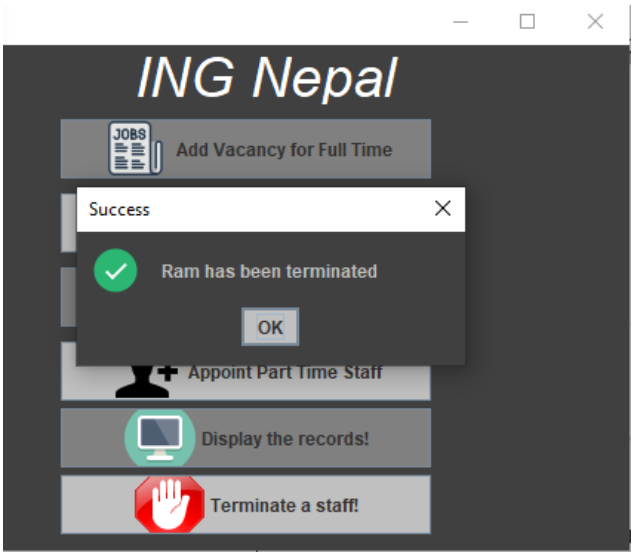


Figure 33 Staff terminate

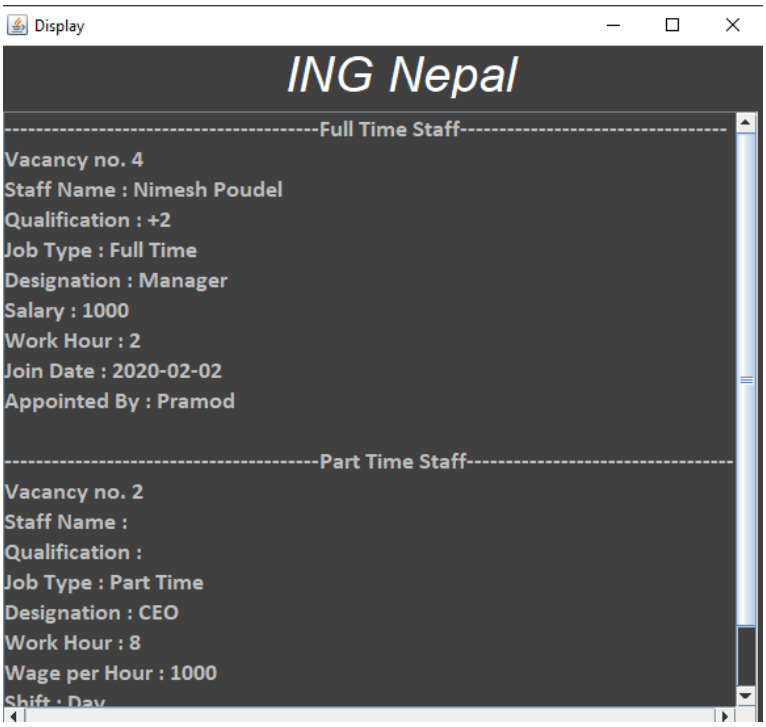


Figure 34 Display after termination

Figure 35 Display after termination

6.3 Test 3 Testing of dialog box when unsuitable values were entered for vacancy number

Objective	To test the appropriate dialog boxes appear when unsuitable values are entered for the vacancy number
Action	At first the program is run, and Add vacancy for full time staff is clicked and the form for add vacancy for full time was appear and the following values were entered Vacancy Number=one Designation=Cleaner Salary=500 Working Hour=1

Expected Result	As vacancy number should integer and here string values in entered the result should appear of dialog box with message number format exception for input string”one”.
Actual Result	Dialog box appear with message number format exception for input string”one”.
Conclusion	The test was successfully performed and screenshot was given below

Table 3 Testing 3

OUTPUT

Figure 36 Clicked in Add vacancy Full time

Figure 37 Form add full time vacancy

Figure 38 Form filling for add vacancy for full time

Figure 39 dialog box when string value is entered

7. Error detection and correction

7.1 Run type Error while user input String value in vacancy number of terminate

This error occurred while the program was fully compiled and add vacancy for both type of job and hire but there is error in terminating staff. When user want to terminate the hired staff if user inputs String value in place of integer then exception was seen and after than exception user cannot terminate other staff too. Terminate button didn't work after the exception

In order to overcome this run time error try catch is used with number exception in termination. By the helps of try catch if user input string value in text fields the . instead of throwing exception in console, the dialog box will appear with message and after the dialog box appear in further the terminate button will work and user can terminate the hired staff

Screen shot of error and correction is given below

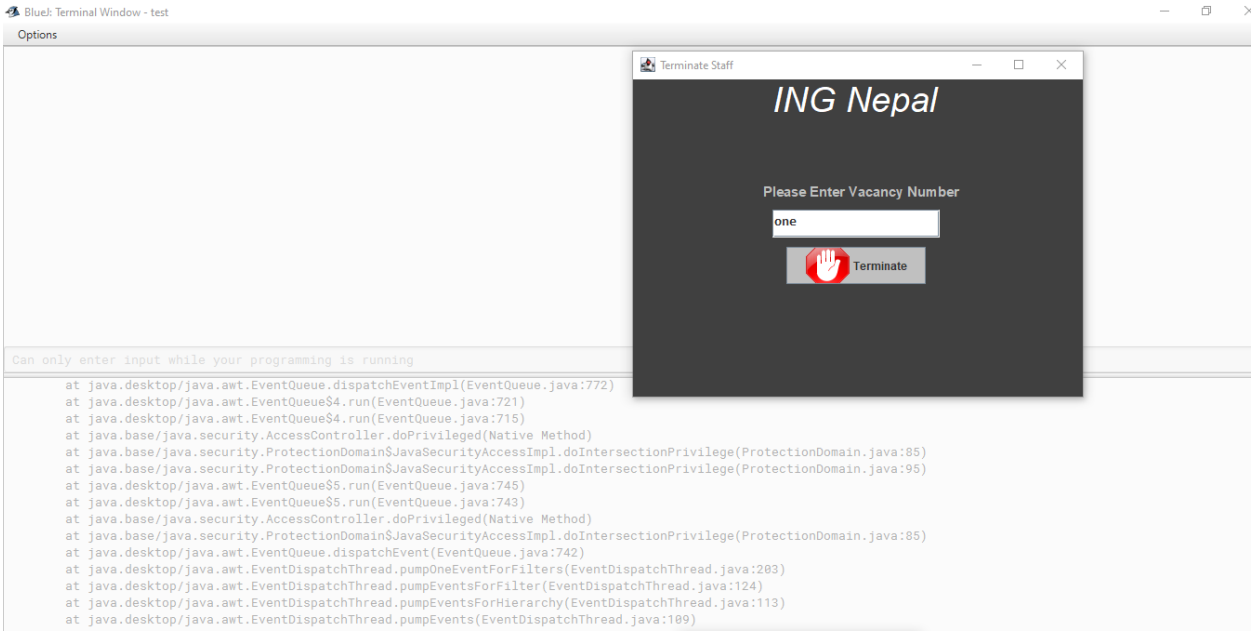


Figure 40 Error while inputting String value

```java
public void backendterminat(){
        boolean flagInTerminate = false;

        for (StaffHire termi : ingArray){
            if (termi instanceof PartTimeStaffHire) {
                 PartTimeStaffHire objTem = (PartTimeStaffHire) termi;
                 if (termi.getVacancyNo() == inputtemret()){
                    if (objTem.getjoined() == true) {
                        flagInTerminate = true;
                        objTem.terminate();     //This will invoke terminate method in PartTime StaffHire method.
                    } else {
                        JOptionPane.showMessageDialog(frame, "No Staff has been appointed", "Error!", JOptionPane.ERROR_MESSAGE);
                        terbo = false;
                    }
                }
            }
        }
        if (flagInTerminate==false && terbo==true){
                if(inputtem.getText().equals("")){
                 JOptionPane.showMessageDialog(frametm,"Please fill out the Text Field","Error",JOptionPane.ERROR_MESSAGE);
                  }
                else {
            JOptionPane.showMessageDialog(frametm,"Sorry record not found","Error!",JOptionPane.WARNING_MESSAGE);
                }}

    }
```

Figure 41Terminate code

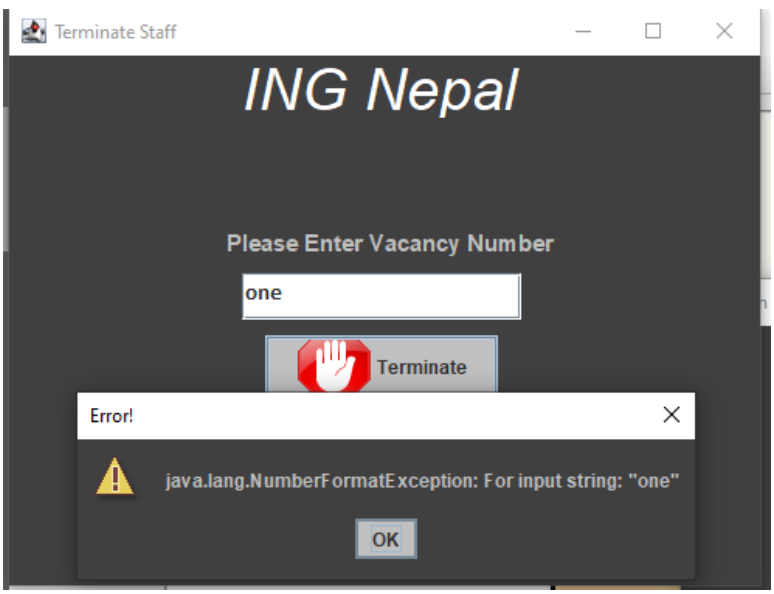


Figure 42 After correction terminate

```java
public void backendterminat(){
        boolean flagInTerminate = false;
        try {
        for (StaffHire termi : ingArray){
            if (termi instanceof PartTimeStaffHire) {
                PartTimeStaffHire objTem = (PartTimeStaffHire) termi;
                if (termi.getVacancyNo() == inputtemret()){
                    if (objTem.getjoined() == true) {
                        flagInTerminate = true;
                        objTem.terminate();     //This will invoke terminate method in PartTime StaffHire method.
                    } else {
                        JOptionPane.showMessageDialog(frame, "No Staff has been appointed", "Error!", JOptionPane.ERROR_MESSAGE);
                        terbo = false;
                    }
                }
            }
        }
        if (flagInTerminate==false && terbo==true){
                if(inputtem.getText().equals("")){
                 JOptionPane.showMessageDialog(frametm,"Please fill out the Text Field","Error",JOptionPane.ERROR_MESSAGE);
                  }
                else {
            JOptionPane.showMessageDialog(frametm,"Sorry record not found","Error!",JOptionPane.WARNING_MESSAGE);
                }}

        }catch (NumberFormatException ex) {
                if(inputtem.getText().equals("")){
                JOptionPane.showMessageDialog(frametm,"Please fill out the Text Field","Error",JOptionPane.ERROR_MESSAGE);
            }
                else {
            JOptionPane.showMessageDialog(frametm,ex,"Error!",JOptionPane.WARNING_MESSAGE);
```

Figure 43 After correction terminate code

7.2 Syntax Error in Submit1 missing semi-colon ";"

Inside the action listener method of submit1 button semi colon was missed while passing the value of job type to realJob type

In order to correct this error, the Submit1 event is checked clearly and found that there is syntax error, the semicolon was missing .semicolon is added and program is compiled and run successfully.

The screen shot of error and error correction are given below

```java
if (e.getSource()==submit1){ //This code will be run when appointSubmit1 is pressed
    try {
        boolean falgINbtn = false;    //creating a flag which will check if the vacancy no. has already been used in the arraylist
        for (StaffHire temp : ingArray) {       //iterates within the arraylist
            if (temp.getVacancyNo() == Integer.parseInt(inputvcba.getText()) && temp.getJobType().equals("Full Time")) {
                //If the user input vacancy no. is equals to the vacancy no. in the arraylist, it will open a new frame which will let to hire
                falgINbtn = true;
                realVacancy = temp.getVacancyNo();
                realDesignation = temp.getDesignation();
                realJob = temp.getJobType()
                FullTimeStaffHire objBtnfull = (FullTimeStaffHire) temp;
               realSalary = objBtnfull.getsalary();
                realWorkHour = objBtnfull.getworkHour();
                dataCheck();
            }
        }
        if (falgINbtn == false) {
            //If the user input vacancy no. is not equals to the vacancy no. in the arraylist, a message box will be shown
                if(inputvcba.getText().equals("")){
```

Figure 44 Error missing semicolon

```
if (e.getSource()==submit1){ //This code will be run when appointSubmit1 is pressed
    try {
        boolean falgINbtn = false;    //creating a flag which will check if the vacancy no. has already been used in the arrayli
        for (StaffHire temp : ingArray) {       //iterates within the arraylist
            if (temp.getVacancyNo() == Integer.parseInt(inputvcba.getText()) && temp.getJobType().equals("Full Time")) {
                //If the user input vacancy no. is equals to the vacancy no. in the arraylist, it will open a new frame which wi
                falgINbtn = true;
                realVacancy = temp.getVacancyNo();
                realDesignation = temp.getDesignation();
                realJob = temp.getJobType();
                FullTimeStaffHire objBtnfull = (FullTimeStaffHire) temp;
               realSalary = objBtnfull.getsalary();
                realWorkHour = objBtnfull.getworkHour();
                dataCheck();
            }
```

Figure 45 Error correction by adding semicolon

7.3 Error while displaying the value.

This error occur while displaying the data of part time staff hire. Here it say can find the method.

Actually the error was there no method qualification ,working hour , joining date, appointed By inside the StaffHire class. They are in inside of PartTimeStaffHire class. As temp is the variable of StaffHire class so it cannot call the methods of PartTimeStaffHire. so to call the new variable is declared displayobj and the method of part time staff hire and the methods are called in order to display the values of part time staff.

The Screen Shot of error and error correction are given below

```
public String partTimeData(){
    String com = "<br>----------------------------------------Part Time Staff----------------------------------";
    for( StaffHire temp : ingArray){
        if(temp instanceof PartTimeStaffHire){

            com = com +"<br>Vacancy no. "+temp.getVacancyNo();

            com = com +"<br>Staff Name : "+temp.getstaffName();
            com = com +"<br>Qualification : "+ temp.getqualification();
            com = com +"<br>Job Type : "+temp.getJobType();
            com = com +"<br>Designation : " + temp.getDesignation();

            com = com +"<br>Work Hour : "+temp.getworkHour();
            com = com +"<br>Wage per Hour : "+ temp.getwagePerHour();
            com = com +"<br>Shift : "+ temp.getshifts();

            com = com +"<br>Join Date : "+temp.getjoinDate();

            com = com +"<br>Appointed By : "+temp.getappointedBy();
            com = com +"<br>";

        }
    }
    return com+"</html>";
}
```

cannot find symbol - method getqualification()

Figure 46 Error calling in method

```java
public String partTimeData(){
    String com = "<br>----------------------------------------Part Time Staff----------------------------------";
    for( StaffHire temp : ingArray){
        if(temp instanceof PartTimeStaffHire){
            PartTimeStaffHire displayobj = (PartTimeStaffHire) temp;
            com = com +"<br>Vacancy no. "+temp.getVacancyNo();

            com = com +"<br>Staff Name : "+displayobj.getstaffName();
            com = com +"<br>Qualification : "+ displayobj.getqualification();
            com = com +"<br>Job Type : "+temp.getJobType();
            com = com +"<br>Designation : " + temp.getDesignation();

            com = com +"<br>Work Hour : "+displayobj.getworkHour();
            com = com +"<br>Wage per Hour : "+ displayobj.getwagePerHour();
            com = com +"<br>Shift : "+ displayobj.getshifts();

            com = com +"<br>Join Date : "+displayobj.getjoinDate();

            com = com +"<br>Appointed By : "+displayobj.getappointedBy();
            com = com +"<br>";

        }
    }
    return com+"</html>";
}
```

Figure 47 Error correction during calling method

8.Conscluscion

Coursework was satisfied and finished in time. The project was testing and nearly pushed us through our points of confinement in this restricted measure of time. All over this project, large blunders and false impressions were experienced which were altogether handled by assurance and diligent work in appropriate research on the different topic. Many discussion course teacher and research about each topic was also done for completing this coursework.

During this coursework I learned many things . which is much more important for a developer to develop the application for any organization. Creating ArrayList, Using ArrayList, Extracting te value of Arraylist. Proper use of GUI, Designing the software, Error Handling use of Try catch, Proper use of flags, Concept of swings and awt like JPanel,JFrame, JButton, JScrollpane, UIManager, JOptionpane ,Border Layout, Java color , Action Listener are the main topic which I learnt from this coursework. Not only Converting String to Integer and Integer to String ,loop and how to make software user friendly and clean code writing were also learned. From this coursework I have also learned self learning, and remembering the topic which I have learn in previous days. I have also learn to solve the problem related java by very easy way. Lastly, after

completing this coursework I came to know to understand the mechanism of code and to write a code in java's program in a suitable manner.

After starting coursework problem regrading use of Try catch, Displaying multiple values in same vacancy number, capital letter and small letter in a code and proper implementation of code were faced. Mainly in using Try catch and flags Some of the symbols like semicolons, equals to were missed and some of the result were wrong. During developing program some of the methods were wrongly implement.

Heaps of reading, practice, and testing were performed. Tons of analysis was drained the thought of programming coming up with and therefore the development of program. The development of program was glanced through altogether in this part I had learn from course teacher and done lots of research also help me. I had also learned from internet regarding problem which I have been face in this coursework and practiced and research was done with last those wrong result became right which helps to completing this coursework in a time with successful result.

9. Appendix 1

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Image;
import java.awt.Panel;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.ConcurrentModificationException;
```

```
import java.util.PrimitiveIterator.OfDouble;

import javax.swing.BorderFactory;
    import javax.swing.BoxLayout;
    import javax.swing.ImageIcon;
    import javax.swing.JButton;
    import javax.swing.JFrame;
    import javax.swing.JLabel;
    import javax.swing.JOptionPane;
    import javax.swing.JPanel;
    import javax.swing.JScrollBar;
    import javax.swing.JScrollPane;
    import javax.swing.JTable;
    import javax.swing.JTextField;
    import javax.swing.UIManager;

    public class INGNepal implements ActionListener {
        public JFrame frame,frame1,framein,frame2,frameck,frametm,framedis;
        private String choice,realDesignation,realJob,realShift;
        private JPanel cointainfu,cointain,cointainfuap,InformationStaff,terminatepanel;
        private JButton Fullva,Fullap,Partva,Partap,display,terminate,hire,
        cancel,submit1,submit2,infoClear,infoSave,ckbtn,btnterminate,ckcl,exportButton;
        private JTextField inputvano,inputde,inputjb,inputsa,inputwh,inputwg,
        inputsh,inputvcba,vivac,vijobtype,videsign,viappol,vishift,vistaff,viworkingHour,
        vijoindate,viquali,viwageper,visalary,inputtem;
        private int realVacancy,realSalary,realWorkHour,realWage;
        boolean copyCheck=false;
        boolean toAddinside;
```

```
boolean terbo=true;
ArrayList<StaffHire> ingArray = new ArrayList<>(); //creating an arraylist of StaffHire ingArray
public static void main(String[] args) { //main method
INGNepal obj1=new INGNepal(); //creating object  of INGNepal
obj1.GuiBox(); //Calling GuiBox method

UIManager.put("OptionPane.background", Color.darkGray); // changing background color of JOptionPane
UIManager.put("Panel.background", Color.darkGray); // changing panel color of JOptionPane
UIManager.put("Button.background", Color.lightGray); // changing button color of JOptionPane

UIManager.put("OptionPane.messageForeground",Color.lightGray); // changing background color of JOptionPane

}

private void GuiBox() { // creating method for main GUI home page
frame=new JFrame("Main");
frame.setSize(500, 270);
frame.setPreferredSize(new Dimension(500,380));

cointain=new JPanel(null);
cointain.setBorder(BorderFactory.createEmptyBorder(10,10,10,10));
cointain.setBackground(Color.DARK_GRAY);

//creaating jpanel
```

```
ImageIcon img = new ImageIcon("../image/6.png");
ImageIcon img1 = new ImageIcon("../image/4.png");
ImageIcon img2 = new ImageIcon("../image/7.png");
ImageIcon img3 = new ImageIcon("../image/8.png");
JLabel titJLabel=new JLabel("ING Nepal");
titJLabel.setBounds(150,0,250,40);
titJLabel.setFont(new Font("SansSeri", Font.ITALIC, 38));

titJLabel.setForeground(Color.white);
 Fullva = new JButton("Add Vacancy for Full Time",img);
Fullva.setBounds(100,50,250,40);

Fullva.setBackground(Color.GRAY);
Fullap  = new JButton("Appoint Full Time Staff",img1);
   Fullap.setBounds(100,150,250,40);
   Fullap.setBackground(Color.GRAY);
 Partva = new JButton("Add Vacancy for Part Time",img);
Partva.setBounds(100,100,250,40);
Partva.setBackground(Color.LIGHT_GRAY);

 Partap = new JButton("Appoint Part Time Staff",img1);
Partap.setBounds(100,200,250,40);
Partap.setBackground(Color.LIGHT_GRAY);
 display = new JButton("Display the records!",img2);
display.setBounds(100,245,250,40);
display.setBackground(Color.GRAY);
 terminate = new JButton("Terminate a staff!",img3);
```

```
        terminate.setBounds(100,290,250,40);
        terminate.setBackground(Color.lightGray);
        //adding button in panel
        cointain.add(titJLabel);
        cointain.add(Fullva);
        cointain.add(Fullap);
        cointain.add(Partva);

        cointain.add(Partap);
        cointain.add(display);
        cointain.add(terminate);
        //action listener
        Fullva.addActionListener(this);
        Partva.addActionListener(this);
        Fullap.addActionListener(this);
        Partap.addActionListener(this);
        terminate.addActionListener(this);
        display.addActionListener(this);
       // adding panel in frame
        frame.add(cointain);
        frame.pack();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // to exit whole program when x is clicked
        frame.setVisible(true);
        frame.setLocationRelativeTo(null);



    }
    //method for add vacancy GUI
```

```
private void Fulltimeva() {
    frame1=new JFrame("Add Vacancy");
    frame1.setSize(500, 250);
    JLabel titJLabel=new JLabel("ING Nepal");
    titJLabel.setBounds(150,0,250,40);
    titJLabel.setFont(new Font("SansSeri", Font.ITALIC, 38));
    titJLabel.setForeground(Color.white);
    frame1.setPreferredSize(new Dimension(500,380));
    ImageIcon img = new ImageIcon("../image/9.png");
    ImageIcon img1 = new ImageIcon("../image/10.png");
    //creaating jpanel
    cointainfu=new JPanel(null);
    cointainfu.setBorder(BorderFactory.createEmptyBorder(10,10,10,10));
    cointainfu.setBackground(Color.DARK_GRAY);
    JLabel lbl1 =new JLabel("Vacancy Number:");
    lbl1.setBounds(20, 60, 150, 20);
    lbl1.setFont(new Font("SansSeri", Font.BOLD,15));
    lbl1.setForeground(Color.LIGHT_GRAY);
     inputvano=new JTextField(5);
     inputvano.setFont(new Font("Calibri",Font.BOLD,15));
    inputvano.setBounds(260, 60, 180, 25);
    JLabel lbl2 =new JLabel("Designation :");
    lbl2.setBounds(20, 95, 150, 20);
    lbl2.setFont(new Font("SansSeri", Font.BOLD,15));
    lbl2.setForeground(Color.LIGHT_GRAY);
     inputde=new JTextField(12);
    inputde.setBounds(260, 95, 180, 25);
    inputde.setFont(new Font("Calibri",Font.BOLD,15));
```

```
JLabel lbl3 =new JLabel("Job Type :");
lbl3.setBounds(20, 130, 150, 20);
lbl3.setFont(new Font("SansSeri", Font.BOLD,15));
lbl3.setForeground(Color.LIGHT_GRAY);
 inputjb=new JTextField(15);
 inputjb.setText(choice);
 inputjb.setBounds(260, 130, 180, 25);
 inputjb.setFont(new Font("Calibri",Font.BOLD,15));
 inputjb.setEditable(false);
JLabel lbl4 =new JLabel("Salary :");
lbl4.setBounds(20, 165, 150, 20);
lbl4.setFont(new Font("SansSeri", Font.BOLD,15));
lbl4.setForeground(Color.LIGHT_GRAY);
 inputsa=new JTextField(5);
inputsa.setBounds(260, 165, 180, 25);
JLabel lbl5 =new JLabel("Working Hour :");
lbl5.setBounds(20, 200, 150, 20);
lbl5.setFont(new Font("SansSeri", Font.BOLD,15));
lbl5.setForeground(Color.LIGHT_GRAY);
 inputwh=new JTextField(4);
inputwh.setBounds(260, 200, 180, 25);
JLabel lbl6=new JLabel("Wages Per Hour");
lbl6.setBounds(20, 165, 150, 20);
lbl6.setFont(new Font("SansSeri", Font.BOLD,15));
lbl6.setForeground(Color.LIGHT_GRAY);
 inputwg=new JTextField(10);
inputwg.setBounds(260, 165, 180, 25);
JLabel lbl7=new JLabel("Shift");
```

```
lbl7.setBounds(20, 235, 150, 20);
lbl7.setFont(new Font("SansSeri", Font.BOLD,15));
lbl7.setForeground(Color.LIGHT_GRAY);
 inputsh=new JTextField(10);
inputsh.setBounds(260, 235, 180, 25);
inputsh.setFont(new Font("Calibri",Font.BOLD,15));
inputwg.setFont(new Font("Calibri",Font.BOLD,15));
inputwh.setFont(new Font("Calibri",Font.BOLD,15));
inputsa.setFont(new Font("Calibri",Font.BOLD,15));

 hire = new JButton("Ok",img);
hire.setBounds(80, 280, 120, 40);
hire.setBackground(Color.lightGray);
hire.addActionListener(this);// action listener for hire button

 cancel=new JButton("Clear",img1);
cancel.setBounds(230, 280, 120, 40);
cancel.setBackground(Color.lightGray);
cancel.addActionListener(this); // aaction listener for cancel button
// adding component in panel
cointainfu.add(lbl1);
cointainfu.add(inputvano);
cointainfu.add(lbl2);
cointainfu.add(inputde);
cointainfu.add(lbl3);
cointainfu.add(inputjb);

cointainfu.add(lbl5);
```

```
        cointainfu.add(inputwh);
        cointainfu.add(hire);
        cointainfu.add(cancel);
        cointainfu.add(titJLabel);
        // adding panel in frame
        frame1.add(cointainfu);
        frame1.pack();
        frame1.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);//      dispose
when x is clicked
        frame1.setVisible(true);
        //i f conditon to look form different for full time and part time
        if(choice.equals("Full Time")){
            cointainfu.add(lbl4);
            cointainfu.add(inputsa);
        }
        if(choice.equals("Part Time")){
            cointainfu.add(lbl6);
            cointainfu.add(inputwg);
            cointainfu.add(lbl7);
            cointainfu.add(inputsh);
        }


    }
    //methof for apppointing staff
    public void Fulltimeapoint() {
        frame2=new JFrame("Appoint Staff");
        frame2.setSize(500, 250);
        JLabel titJLabel=new JLabel("ING Nepal");
        titJLabel.setBounds(150,0,250,40);
```

```
titJLabel.setFont(new Font("SansSeri", Font.ITALIC, 38));
titJLabel.setForeground(Color.white);
frame2.setPreferredSize(new Dimension(500,380));
ImageIcon img1= new ImageIcon("../image/4.png");


cointainfuap=new JPanel(null);
cointainfuap.setBorder(BorderFactory.createEmptyBorder(10,10,10,10));
cointainfuap.setBackground(Color.DARK_GRAY);

JLabel lbl1 =new JLabel("Please Enter Vacancy Number");
lbl1.setBounds(140, 110, 300, 20);
lbl1.setFont(new Font("SansSeri", Font.BOLD,15));
lbl1.setForeground(Color.LIGHT_GRAY);

inputvcba=new JTextField(5);
inputvcba.setBounds(150, 140, 180, 30);
inputvcba.setFont(new Font("Calibri",Font.BOLD,15));
// ading component in panel
cointainfuap.add(titJLabel);
cointainfuap.add(lbl1);
cointainfuap.add(inputvcba);


// adding panel in frame
frame2.add(cointainfuap);
frame2.pack();
frame2.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
        frame2.setVisible(true);
        // if condition for button . to add part time and full time staff
        if(choice.equals("Full Time")){
            submit1=new JButton("Hire",img1);
            submit1.setBounds(180, 180, 120, 40);
            submit1.setBackground(Color.lightGray);
            cointainfuap.add(submit1);
            submit1.addActionListener(this);
        }
        if(choice.equals("Part Time")){
            submit2=new JButton("Hire",img1);
            submit2.setBounds(180, 180, 120, 40);
            submit2.setBackground(Color.lightGray);
            cointainfuap.add(submit2);
            submit2.addActionListener(this);
        }

    }
    //form for appointing staff staffname,qualification etc
    public void InsideHiring() {
        framein=new JFrame("Staff apointment");
        framein.setSize(500, 250);
        JLabel titJLabel=new JLabel("ING Nepal");
        titJLabel.setBounds(250,0,250,40);
        titJLabel.setFont(new Font("SansSeri", Font.ITALIC, 38));
        titJLabel.setForeground(Color.white);
        ImageIcon img1= new ImageIcon("../image/10.png");
        ImageIcon img= new ImageIcon("../image/save.png");
```

```
InformationStaff=new JPanel(null);
InformationStaff.setBorder(BorderFactory.createEmptyBorder(10,10,10,10));
InformationStaff.setBackground(Color.DARK_GRAY);
JLabel lblvacancy = new JLabel("Vacancy Number:");
JLabel lbljobtype = new JLabel("Job Type:");
JLabel lblDesigna = new JLabel("Designation:");
JLabel lblstaffName = new JLabel("Staff Name :");
JLabel lblquali= new JLabel("Qualification :");
JLabel lblworkingHour = new JLabel("Working Hour :");
JLabel lbljoiningDate = new JLabel("Joining Date :");
JLabel lblappol = new JLabel("Appointed By :");
JLabel lblshift = new JLabel("Shift :");
JLabel lblwageper = new JLabel("Wage per Hour :");
JLabel lblsalary = new JLabel("Salary :");
// changing integer to string
String salar = Integer.toString(realSalary);
String wage = Integer.toString(realWage);
String hour = Integer.toString(realWorkHour);
String vac = Integer.toString(realVacancy);
vivac=new JTextField();
vijobtype=new JTextField();
videsign=new JTextField();
vistaff=new JTextField();
viquali=new JTextField();
viworkingHour=new JTextField();
vijoindate=new JTextField();
viappol=new JTextField();
vishift=new JTextField();
```

```
viwageper=new JTextField();
visalary=new JTextField();
//displaying in text field

viworkingHour.setText(hour);
viwageper.setText(wage);
vivac.setText(vac);
vijobtype.setText(realJob);
videsign.setText(realDesignation);
vishift.setText(realShift);

lblvacancy.setBounds(20, 60, 150, 20);
lblvacancy.setFont(new Font("SansSeri", Font.BOLD,15));
lblvacancy.setForeground(Color.LIGHT_GRAY);

lbljobtype.setBounds(200, 60, 150, 20);
lbljobtype.setFont(new Font("SansSeri", Font.BOLD,15));
lbljobtype.setForeground(Color.LIGHT_GRAY);

lblDesigna.setBounds(360, 60, 150, 20);
lblDesigna.setFont(new Font("SansSeri", Font.BOLD,15));
lblDesigna.setForeground(Color.LIGHT_GRAY);

lblappol.setBounds(520, 130, 150, 20);
lblappol.setFont(new Font("SansSeri", Font.BOLD,15));
lblappol.setForeground(Color.LIGHT_GRAY);

lblstaffName.setBounds(20, 130, 150, 20);
```

```
lblstaffName.setFont(new Font("SansSeri", Font.BOLD,15));
lblstaffName.setForeground(Color.LIGHT_GRAY);

lblquali.setBounds(200, 130, 150, 20);
lblquali.setFont(new Font("SansSeri", Font.BOLD,15));
lblquali.setForeground(Color.LIGHT_GRAY);

lblworkingHour.setBounds(520, 60, 150, 20);
lblworkingHour.setFont(new Font("SansSeri", Font.BOLD,15));
lblworkingHour.setForeground(Color.LIGHT_GRAY);

lbljoiningDate.setBounds(360, 130, 150, 20);
lbljoiningDate.setFont(new Font("SansSeri", Font.BOLD,15));
lbljoiningDate.setForeground(Color.LIGHT_GRAY);

lblshift.setBounds(20, 200, 150, 20);
lblshift.setFont(new Font("SansSeri", Font.BOLD,15));
lblshift.setForeground(Color.LIGHT_GRAY);

lblwageper.setBounds(200, 200, 150, 20);
lblwageper.setFont(new Font("SansSeri", Font.BOLD,15));
lblwageper.setForeground(Color.LIGHT_GRAY);

lblsalary.setBounds(20, 200, 150, 20);
lblsalary.setFont(new Font("SansSeri", Font.BOLD,15));
lblsalary.setForeground(Color.LIGHT_GRAY);

vivac.setBounds(20, 95, 140, 26);
```

```
vivac.setFont(new Font("Calibri",Font.BOLD,15));
vijobtype.setBounds(200, 95, 140, 26);
vijobtype.setFont(new Font("Calibri",Font.BOLD,15));
videsign.setBounds(360, 95, 140, 26);
videsign.setFont(new Font("Calibri",Font.BOLD,15));
viappol.setBounds(520, 165, 140, 26);
viappol.setFont(new Font("Calibri",Font.BOLD,15));
vistaff.setBounds(20, 165, 140, 26);
vistaff.setFont(new Font("Calibri",Font.BOLD,15));
viquali.setBounds(200, 165, 140, 26);
viquali.setFont(new Font("Calibri",Font.BOLD,15));
viworkingHour.setBounds(520, 95, 140, 26);
viworkingHour.setFont(new Font("Calibri",Font.BOLD,15));
vijoindate.setBounds(360, 165, 140, 26);
vijoindate.setFont(new Font("Calibri",Font.BOLD,15));
vishift.setBounds(20, 235, 140, 26);
vishift.setFont(new Font("Calibri",Font.BOLD,15));
viwageper.setBounds(200, 235, 140,26);
viwageper.setFont(new Font("Calibri",Font.BOLD,15));
visalary.setBounds(20, 235, 140,26);
visalary.setFont(new Font("Calibri",Font.BOLD,15));

infoSave=new JButton("Save",img);
infoClear=new JButton("Clear",img1);
infoClear.setBackground(Color.lightGray);
infoClear.setBounds(560, 290, 120, 40);
infoSave.setBackground(Color.lightGray);
infoSave.setBounds(420, 290, 120, 40);
```

```
//adding component in panel
InformationStaff.add(infoSave);
InformationStaff.add(infoClear);
//action lsnr for buton
infoClear.addActionListener(this);
infoSave.addActionListener(this);
//adding component in panel
InformationStaff.add(lblvacancy);
InformationStaff.add(lbljobtype);
InformationStaff.add(lblDesigna);
InformationStaff.add(lblappol);
InformationStaff.add(lblstaffName);
InformationStaff.add(lblquali);
InformationStaff.add(lblworkingHour);
InformationStaff.add(lbljoiningDate);



InformationStaff.add(vivac);
InformationStaff.add(vijobtype);
InformationStaff.add(videsign);
InformationStaff.add(viappol);
InformationStaff.add(vistaff);
InformationStaff.add(viquali);
InformationStaff.add(viworkingHour);
InformationStaff.add(vijoindate);
```

```
InformationStaff.add(titJLabel);
// if condition to look different form for different jot type
 if(choice.equals("Full Time")){
     InformationStaff.add(lblsalary);
        visalary.setText(salar);
        InformationStaff.add(visalary);
    }

 if(choice.equals("Part Time")){
        InformationStaff.add(lblshift);
        InformationStaff.add(lblwageper);
        InformationStaff.add(vishift);
        InformationStaff.add(viwageper);
    }
 // making some text non editiable
  vivac.setEditable(false);
  vijobtype.setEditable(false);
  videsign.setEditable(false);
  viwageper.setEditable(false);
  viworkingHour.setEditable(false);
  visalary.setEditable(false);
  vishift.setEditable(false);
    //adding panel in frame
framein.add(InformationStaff);
framein.setPreferredSize(new Dimension(700,380));
framein.pack();
framein.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        framein.setVisible(true);
    }
    // termination gui
    public void termGui(){
       frametm=new JFrame("Terminate Staff");
        frametm.setSize(500, 250);
        JLabel titJLabel=new JLabel("ING Nepal");
        titJLabel.setBounds(150,0,250,40);
        titJLabel.setFont(new Font("SansSeri", Font.ITALIC, 38));
        titJLabel.setForeground(Color.white);
        frametm.setPreferredSize(new Dimension(500,380));
        ImageIcon img1= new ImageIcon("../image/8.png");

        terminatepanel=new JPanel(null);
        terminatepanel.setBorder(BorderFactory.createEmptyBorder(10,10,10,10));
        terminatepanel.setBackground(Color.DARK_GRAY);

        JLabel lbltem =new JLabel("Please Enter Vacancy Number");
        lbltem.setBounds(140, 110, 300, 20);
        lbltem.setFont(new Font("SansSeri", Font.BOLD,15));
        lbltem.setForeground(Color.LIGHT_GRAY);

        btnterminate=new JButton("Terminate",img1);
        btnterminate.setBounds(165, 180, 150, 40);
        btnterminate.setBackground(Color.lightGray);
        terminatepanel.add(btnterminate);
        btnterminate.addActionListener(this);// action listener for button
```

```
        inputtem=new JTextField();
        inputtem.setBounds(150, 140, 180, 30);
        inputtem.setFont(new Font("Calibri",Font.BOLD,15));
        frametm.add(terminatepanel);// adding panel in frame
        terminatepanel.add(titJLabel);
        terminatepanel.add(lbltem);
        terminatepanel.add(inputtem);
        frametm.pack();
        frametm.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        frametm.setVisible(true);
        }
    // display Gui
    public void displaymethod(){

      framedis=new JFrame("Display");
      framedis.setSize(800, 380);

     JPanel panelDisplayJPanel=new JPanel(null);
      panelDisplayJPanel.setBackground(Color.darkGray);
      //calling fullTimeData and PartTimedata method and adding in jlabel
      JLabel titJLabel=new JLabel(fullTimeData() + partTimeData());
      JScrollPane scroll = new JScrollPane(null);
        scroll.setViewportView(titJLabel);
      titJLabel.setFont(new Font("Calibri", Font.BOLD, 18));
      titJLabel.setForeground(Color.lightGray);
      titJLabel.setOpaque(true);
      titJLabel.setBackground(Color.DARK_GRAY);
     // adding scroll bar
```

```
scroll.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS
);

scroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
        scroll.getVerticalScrollBar().setBackground(Color.DARK_GRAY);




        scroll.setBounds(5, 50, 575, 470);
        JLabel DisplayLabel=new JLabel("ING Nepal");
        DisplayLabel.setBounds(220, 0, 250, 40);
        DisplayLabel.setFont(new Font("SansSeri", Font.ITALIC, 38));
        DisplayLabel.setForeground(Color.white);
        //adding component in panel
           panelDisplayJPanel.add(scroll);

          panelDisplayJPanel.add(DisplayLabel);

         //adding panel in frame
        framedis.setPreferredSize(new Dimension(600,580));
        framedis.add(panelDisplayJPanel);
       framedis.pack();
       framedis.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        framedis.setVisible(true);
     }
   // displaying full time staff data
     public String fullTimeData(){
           //creating com in order to store and return value
```

```
        String com = "<html> ----------------------------------------Full Time Staff--------------------
--------------";
        for( StaffHire temp : ingArray){ // calling arraylist
            if(temp instanceof FullTimeStaffHire){
                FullTimeStaffHire displayobj = (FullTimeStaffHire) temp;// getting data from
arraylist
                com = com +"<br>Vacancy no. "+temp.getVacancyNumber()+"";
                com = com +"<br>Staff Name : "+ displayobj.getstaffName();
                com = com +"<br>Qualification : "+ displayobj.getqualification();
                com = com +"<br>Job Type : "+ temp.getJobType();
                com = com +"<br>Designation : " + temp.getDesignation();

                com = com +"<br>Salary : "+ displayobj.getsalary();
                com = com +"<br>Work Hour : "+ displayobj.getworkingHour();

                com = com +"<br>Join Date : "+ displayobj.getjoiningDate();

                com = com +"<br>Appointed By : "+ displayobj.getappointedBy();
                com = com +"<br>";
            }
        }
        return com;// retuning data of full time staff
    }
  // method to display part time data
    public String partTimeData(){
          //creaating string com for store and return value for part time
        String com = "<br>----------------------------------------Part Time Staff-----------------------
----------";
        for( StaffHire temp : ingArray){// calling arraylist
```

```
            if(temp instanceof PartTimeStaffHire){
                PartTimeStaffHire displayobj = (PartTimeStaffHire) temp;
                com = com +"<br>Vacancy no. "+temp.getVacancyNumber();//getting value
from arraylist


                com = com +"<br>Staff Name : "+displayobj.getstaffName();
                com = com +"<br>Qualification : "+ displayobj.getqualification();
                com = com +"<br>Job Type : "+temp.getJobType();
                com = com +"<br>Designation : " + temp.getDesignation();


                com = com +"<br>Work Hour : "+displayobj.getworkingHour();
                com = com +"<br>Wage per Hour : "+ displayobj.getwagePerHour();
                com = com +"<br>Shift : "+ displayobj.getshifts();


                com = com +"<br>Join Date : "+displayobj.getjoiningDate();


                com = com +"<br>Appointed By : "+displayobj.getappointedBy();
                com = com +"<br>";


            }
        }
        return com+"</html>"; // returning data of part time
    }


    //


        private void vacancyforward(){
            boolean flag = false;   // to check wether the text box is empty or not if empty
show message
```

```
            if(choice.equals("Full Time")) {
                if (inputvano.getText().equals("") || inputde.getText().equals("") || inputsa.getText().equals("") || inputwh.getText().equals("")) {
                    JOptionPane.showMessageDialog(frame1, "Please fill out all the text fields", "Error", JOptionPane.ERROR_MESSAGE);
                    flag=true;

                }
            }
            else if(choice.equals("Part Time")) {
                if (inputvano.getText().equals("") || inputde.getText().equals("") || inputwg.getText().equals("") || inputsh.getText().equals("")
                || inputwh.getText().equals("")){
                    JOptionPane.showMessageDialog(frame1, "Please fill out all the text fields", "Error", JOptionPane.ERROR_MESSAGE);
                    flag=true;

                }

            }
            if(flag==false) {

                // if textbocy is not empty and vacancy number is avaialbe then this code will excutive
                try {
                    ImageIcon img1= new ImageIcon("../image/suc.png");
                    if (choice.equals("Full Time")) {
                        FullTimeStaffHire objF = new FullTimeStaffHire(vacancyFrom(), designationFrom(), choice, salaryFrom(), workingHourFrom());
                        ingArray.add(objF);
```

```
                    JOptionPane.showMessageDialog(frame1,"Sucessful \n vacancy Number "
                    +inputvano.getText()+" Added","Sucess",JOptionPane.PLAIN_MESSAGE,img1);
                }
                if (choice.equals("Part Time")) {
                    PartTimeStaffHire objP = new PartTimeStaffHire(vacancyFrom(), designationFrom(), choice,
                    workingHourFrom(), wagePHourFrom(), shiftFrom());
                    ingArray.add(objP);
                    JOptionPane.showMessageDialog(frame1,"Sucessful \n vacancy Number "+inputvano.getText()
                    +" Added","Sucess",JOptionPane.PLAIN_MESSAGE,img1);
                }
            } catch (NumberFormatException exe) {
                JOptionPane.showMessageDialog(frame1,exe ,"Error!", JOptionPane.WARNING_MESSAGE);
            }
        }
    }

    //actionperformed method when some component is clicked
    @Override
    public void actionPerformed(ActionEvent e)throws ConcurrentModificationException {
        if (e.getSource()==Fullva){ // if fullva button is clicked then string value is set and methos will be open
            choice = "Full Time";
            Fulltimeva();

        }
```

```
        if (e.getSource()==Partva){// if partva  button is clicked then string value is set and methos will be open
            choice = "Part Time";
            Fulltimeva();

        }

        if (e.getSource()==Fullap){ // if fullap button is clicked then string value is set and methos will be open
            choice = "Full Time";
            Fulltimeapoint();
        }
        if (e.getSource()==Partap){ // if partap button is clicked then string value is set and methos will be open
            choice = "Part Time";
            Fulltimeapoint();

        }
        if (e.getSource()==ckcl){// if ckcl is clicked the frameck will dispose
            frameck.dispose();

        }
        if (e.getSource()==terminate){// when terminate button is pressed ther termGyi method will open
          termGui();
```

```
        }
        if (e.getSource()==btnterminate){// action performed for btn terminate if it is empty the show message
            if (inputtem.getText().equals("")){
                JOptionPane.showMessageDialog(frametm,"Please fill out the Text Field","Error",JOptionPane.ERROR_MESSAGE);
            }else {
                backendterminat();
            }
        }
        if(e.getSource()==display){// when display buttonis clicked then displaymethd will invoke
            displaymethod();
        }
        if (e.getSource()==hire){// when hire button is cliked then repetation staff will opened
            repetationStaff();
        }


        if(e.getSource()==cancel){// when cancel button is clicked then value of text filed will empty
            inputvano.setText("");
           inputsh.setText("");
            inputwh.setText("");
            inputde.setText("");
            inputwg.setText("");
            inputsa.setText("");
        }
        if(e.getSource()==infoClear){ // when infoClear button is clicked then value of text filed will empty
```

```
            viappol.setText("");
            viquali.setText("");
            vistaff.setText("");
            vijoindate.setText("");
        }
        if(e.getSource()==infoSave){// to save data from hiring form if empty to show message
            if (vistaff.getText().equals("")||viquali.getText().equals("")||viappol.getText().equals("")||vijoindate.getText().equals("")){
                JOptionPane.showMessageDialog(framein,"Please fill out all the text fields.","Error",JOptionPane.ERROR_MESSAGE);
            }else {
                framein.dispose();
                dataStore();
          }
            }

        if (e.getSource()==submit1){ //This code will be run when submit1 is clicked
            try {
                boolean falgINbtn = false;    // checking whether vacancy number is vacant or not  if vacant then add other show message
                for (StaffHire temp : ingArray) {
                    if (temp.getVacancyNumber() == Integer.parseInt(inputvcba.getText()) && temp.getJobType().equals("Full Time")) {

                        falgINbtn = true;
                        realVacancy = temp.getVacancyNumber();
                        realDesignation = temp.getDesignation();
                        realJob = temp.getJobType();
```

```
                    FullTimeStaffHire objBtnfull = (FullTimeStaffHire) temp;
                   realSalary = objBtnfull.getsalary();
                    realWorkHour = objBtnfull.getworkingHour();
                    dataCheck();
                }
            }
            if (falgINbtn == false) {
                // checking if vacany number enter by user is equals to vacancy number present in array list or not
              // if text box is empty the show messageg
                if(inputvcba.getText().equals("")){
                    JOptionPane.showMessageDialog(frame2,"Please fill out the Text Field","Error",JOptionPane.ERROR_MESSAGE);
                }
                else {
                JOptionPane.showMessageDialog(frame2, "Not found!! Please enter correct vacancy number", "Error", JOptionPane.ERROR_MESSAGE);
            }}
        }catch(Exception exe){
            if(inputvcba.getText().equals("")){
                JOptionPane.showMessageDialog(frame2,"Please fill out the Text Field","Error",JOptionPane.ERROR_MESSAGE);
            }
            else{
                JOptionPane.showMessageDialog(frame2,exe);
            }
        }
    }
    if (e.getSource()==submit2){
        try {
```

```
                // checking whether vacancy number is vacant or not  if vacant then add other show message
            boolean falgINbtn = false;
            for(StaffHire temp: ingArray) {
              if (temp.getVacancyNumber() == Integer.parseInt(inputvcba.getText()) ) {
                if (temp.getJobType()=="Part Time") {
                  falgINbtn = true;
                  realVacancy = temp.getVacancyNumber();
                  realDesignation = temp.getDesignation();
                  realJob = temp.getJobType();
                  PartTimeStaffHire objBtnpart = (PartTimeStaffHire) temp;
                  realWorkHour = objBtnpart.getworkingHour();
                  realWage = objBtnpart.getwagePerHour();
                  realShift = objBtnpart.getshifts();
                  dataCheck();
                }
              }

            }
            if ( falgINbtn == false){
                //if text box is empty to show message.
                if(inputvcba.getText().equals("")){
                  JOptionPane.showMessageDialog(frame2,"Please fill out the Text Field","Error",JOptionPane.ERROR_MESSAGE);
              }
                else {// checking if vacany number enter by user is equals to vacancy number present in array list or not
              JOptionPane.showMessageDialog(frame2,"Not found!! Please enter correct vacancy number" ,"Error",JOptionPane.ERROR_MESSAGE);
                } }
```

```
            }catch(Exception exe){
                if(inputvcba.getText().equals("")){
                    JOptionPane.showMessageDialog(frame2,"Please fill out the Text Field","Error",JOptionPane.ERROR_MESSAGE);
                }
                else{
                    JOptionPane.showMessageDialog(frame2,exe);
                }
            }
        }
        if(e.getSource()==ckbtn){// checking wether the staff is appoint for particular vacancy or not
            frame2.dispose();
            if (choice=="Full Time") {
                for (StaffHire temp : ingArray) {
                    if (temp instanceof FullTimeStaffHire) {  // checkcing instance of temp is object is or not
                        FullTimeStaffHire obful = (FullTimeStaffHire) temp;
                        if (obful.getVacancyNumber()==realVacancy) {
                            if (obful.getjoined() == false) {
                                InsideHiring();
                                frameck.dispose();
                            } else {

                                JOptionPane.showMessageDialog(frame, " Sorry Staff has already been appointed \n Try for next vacancy",
                                "Message", JOptionPane.ERROR_MESSAGE);
                                frameck.dispose();
                            }
                        }
```

```
                }
            }
        }
        if (choice=="Part Time") {
            for (StaffHire temp : ingArray) {
                if (temp instanceof PartTimeStaffHire) {
                    PartTimeStaffHire obpar = (PartTimeStaffHire) temp;
                    if (obpar.getVacancyNumber()==realVacancy) {
                    if (obpar.getjoined() == false) {
                        InsideHiring();
                        frameck.dispose();
                    } else {
                        JOptionPane.showMessageDialog(frame, "Sorry staff has been
already appointed \n Try for next vacancy",
                        "Message", JOptionPane.ERROR_MESSAGE);
                    }
                    }
                }
            }
        }
    }

}
// returning methods
public int vacancyFrom(){
    return Integer.parseInt(inputvano.getText());
}

public String designationFrom(){
```

```
        return inputde.getText();
    }

    public int workingHourFrom(){
        return Integer.parseInt(inputwh.getText());
    }

    public int salaryFrom(){
        return Integer.parseInt(inputsa.getText());
    }

    public String shiftFrom(){
        return inputsh.getText();
    }

    public int wagePHourFrom(){
        return Integer.parseInt(inputwg.getText());
    }

    public String staffNameFrom(){
        return vistaff.getText();
    }

    public String appointeByFrom(){
        return viappol.getText();
    }

    public String joingDateFrom(){
```

```
        return vijoindate.getText();
    }

    public String qualificationFrom(){
        return viquali.getText();
    }
    public int inputtemret(){
        return Integer.parseInt(inputtem.getText());
    }//
    public void backendterminat(){
            boolean flagInTerminate = false;
            try {
            for (StaffHire termi : ingArray){
                if (termi instanceof PartTimeStaffHire) {
                    PartTimeStaffHire objTem = (PartTimeStaffHire) termi;
                    if (termi.getVacancyNumber() == inputtemret()){
                      if (objTem.getjoined() == true) {
                          flagInTerminate = true;
                          objTem.terminate();     //calling terminate method from part time staff
hire.
                      } else {
                          JOptionPane.showMessageDialog(frame, "No Staff has been
appointed", "Error!", JOptionPane.ERROR_MESSAGE);
                          terbo = false;
                      }
                  }
                }
            }
```

```
            if (flagInTerminate==false && terbo==true){
                if(inputtem.getText().equals("")){
                  JOptionPane.showMessageDialog(frametm,"Please fill out the Text Field","Error",JOptionPane.ERROR_MESSAGE);
               }
                else {
               JOptionPane.showMessageDialog(frametm,"Sorry record not found","Error!",JOptionPane.WARNING_MESSAGE);
                }}


            }catch ( Exception ex) {
                if(inputtem.getText().equals("")){
                  JOptionPane.showMessageDialog(frametm,"Please fill out the Text Field","Error",JOptionPane.ERROR_MESSAGE);
               }
                else {

JOptionPane.showMessageDialog(frametm,ex,"Error!",JOptionPane.WARNING_MESSAGE);
                } }


          }


     private void repetationStaff(){
        boolean nimcheck = false;





         try {
```

```
            for (StaffHire temp : ingArray) {
                if (temp.getVacancyNumber() == vacancyFrom()) {   // checking vacancy number is used or not
                    JOptionPane.showMessageDialog(frame1, "Sorry vacancy number is not available", "Error!", JOptionPane.ERROR_MESSAGE);

                    nimcheck = true;
                    break;
                }

                }

            if(nimcheck) {     // if vacancy number is not used then to add vacancy

            } else {

                vacancyforward();

                frame1.dispose();
                frame.dispose();
                GuiBox();
            }} catch (Exception e) {
                if (inputvano.getText().equals("") || inputde.getText().equals("") || inputwg.getText().equals("") || inputsh.getText().equals("")|| inputwh.getText().equals("")){
                vacancyforward();
                }
                else {
```

```
JOptionPane.showMessageDialog(frame1,e,"Error!",JOptionPane.ERROR_MESSAGE)
;

                }
            }
              }

    // method to store ata in arraylist
    public void dataStore(){
        ImageIcon img1= new ImageIcon("../image/suc.png");
      if(choice.equals("Full Time")){
        for(StaffHire obj:ingArray){
            if(obj instanceof FullTimeStaffHire){
                FullTimeStaffHire sto = (FullTimeStaffHire) obj;
                if(sto.getVacancyNumber()==realVacancy) {
                if(sto.getjoined()==false){

sto.fullhire(staffNameFrom(),joingDateFrom(),qualificationFrom(),appointeByFrom());
                    JOptionPane.showMessageDialog(framein,"Thank you !!!
"+staffNameFrom()+" has been appointed \n for
"+realDesignation,"Sucess",JOptionPane.PLAIN_MESSAGE,img1);
                }
                }
            }
        }
      }
      if (choice=="Part Time"){
        for(StaffHire obj:ingArray){
            if(obj instanceof PartTimeStaffHire){
```

```
                PartTimeStaffHire sto = (PartTimeStaffHire) obj;
                if(sto.getVacancyNumber()==realVacancy) {
                if(sto.getjoined()==false){

sto.partTimehire(staffNameFrom(),joingDateFrom(),qualificationFrom(),appointeByFrom
());
                    JOptionPane.showMessageDialog(framein,"Thank you !!!
"+staffNameFrom()+" has been appointed \n for
"+realDesignation,"Sucess",JOptionPane.PLAIN_MESSAGE,img1);
                }
              }
            }
          }
        }
      }

      //method for cheking vacacny is appointrd or not and are you sure box gui
      public void dataCheck(){
        for (StaffHire datas : ingArray) {
          if (datas.getVacancyNumber() == Integer.parseInt(inputvcba.getText())) {
            frameck = new JFrame("Check");
            JPanel panelck = new JPanel(null);
            JLabel titJLabel=new JLabel("ING Nepal");
          titJLabel.setBounds(95,0,250,40);
          titJLabel.setFont(new Font("SansSeri", Font.ITALIC, 38));
          titJLabel.setForeground(Color.white);
          ImageIcon img1= new ImageIcon("../image/cancel.png");
          ImageIcon img= new ImageIcon("../image/9.png");
          JLabel AreJLabel=new JLabel("Are You Sure?");
          AreJLabel.setBounds(120,100,250,40);
```

```
    AreJLabel.setFont(new Font("SansSeri", Font.BOLD,25));
    AreJLabel.setForeground(Color.white);
    frameck.setPreferredSize(new Dimension(400,300));

    panelck.setBackground(Color.DARK_GRAY);

      ckbtn = new JButton("Confirm",img);
      ckbtn.setBounds(75,150,120,40);
      ckbtn.setBackground(Color.LIGHT_GRAY);
      ckbtn.addActionListener(this);
      panelck.add(ckbtn);

      ckcl = new JButton("Cancel",img1);
      ckcl.setBounds(210,150,110,40);
      ckcl.setBackground(Color.LIGHT_GRAY);
      ckcl.addActionListener(this);
      panelck.add(ckcl);
      panelck.add(titJLabel);
      panelck.add(AreJLabel);

      frameck.add(panelck);
     frameck.setVisible(true);



      frameck.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
      frameck.pack();
  }
```

```
        }
    }
  // end

}
```

10.Appendix 2

1.Introduction

1.1 Java

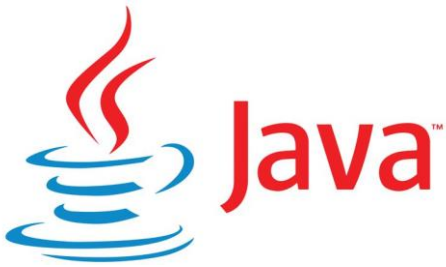


Figure 48 java

(lifewire.com, n.d.)

Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible. It is intended to let application developers write once, run anywhere (WORA),[15] meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.[16] Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them. As of 2019, Java was one of the most popular programming languages in use according to GitHub,[17][18] particularly for client-server web applications, with a reported 9 million developers. (wikipedia.org, 2019)

1.2 Blue J

BlueJ is an integrated development environment (IDE) for the Java programming language, developed mainly for educational purposes, but also suitable for small-scale software development. It runs with the help of JDK (Java Development Kit). (wikipedia.org, 2020)

1.3 Introduction of project

This project was given as a first coursework for the modules CS4001NI programming. In this coursework students should make object oriented program by the helps of java. The main aims of this project is to create the three program Staff hire , Full time staff hire and part time staff hire where staff hire is the class and other two are sub class and to check the programming ability of the students. And this project is done in bluej by installing java virtual machine.

This program was developed for the company which helps in hiring staff. This program show all the details information of vacancy post, designation, vacancy number and job type as well. And in sub classes it provide the information about working hour, qualification ,wages and appointment too. By the helps of this program company can keeps record of the staff and it also can terminated if any this happened and again new staff can be hire.

In this project getter (get), is used for the obtain the value and setter(set) is used for store the value and supper class is used for call parent class in sub classes method and constructor is used to to run the program. When once method is defined it can used many time ans constructor helps to intilized the object in this program. And display is also used in this project in order to show the details of staff and vacancy.

2. CLASS DIAGRAM

StaffHire
-vacancy_number int -designation String -job_type String
+getVacancyNumber() int +setVacancyNumber(int vacancy_number) void + getDesignation() String +setDesignation(String designation) void +getJobType() String +setJobType(String job_type) void +displayInfo() void

Table 4 class diagram staff hire

FullTimeStaffHire
-salary int -workingHour int -staffName String -joiningDate String -qulafication String -appointedBy string -joined boolean
+ getsalary() int + getworkingHour() int + getstaffName() string + getjoinigDate() string + getqulafication(String + getappointedBy string + getjoined() Boolean + setsalary(int salary) void + setWorkingHour(int workingHour) void + fullhire (String staffName, String joinigDate, String qulafication, String appointedBy) +displayInfo() void

Table 5 class diagram of FullTimeStaff

PartTimeStaffHire
- workingHour int -wagePerHouri nt -staffName String -joiningDate String -qulafication String -appointedBy string -shifts string -joined boolean -terminated boolean
+getworkingHour() int +getwagePerHour() int +getstaffName ()String +getjoinigDate() string +getqulafication() String +getappointedBy() string +getshifts() string +getjoined() boolean +getterminated() Boolean +setshifts(String shifts) void +setWagesPerHour(int wagePerHour) void +setWorkingHour(int workingHour) void +partTimehire(String staffName, String joinigDate, String qulafication, String appointedBy) void +terminate() void +displayInfo() void

Table 6 class diagram of PartTimeStaffHire

3. Pseudo Code and method description

3.1 StaffHire Class

- **Method Name**:- getVacancyNumber()

Pseudocode:-

```
CREATE METHOD  getVacancyNumber()
DO
  RETURN vacancy_number;
END DO
```

Method Description : This method return integer value assigned to instance variable vacancy_ number and does not allow to change the value store in the private instance

- **Method name**:- setVacancyNumber(int vacancy_number)

Pseudo code:-

```
CREATE METHOD setVacancyNumber(int vacancy_number)
DO
   this.vacancy_number=vacancy_number;
END DO
```

Method Description : This method accept the values from user and the value will be stored variable vacancy_number but it didn't return any value.

- **Method name**:- getDesignation()

Pseudo code:-

```
CREATE METHOD getDesignation()
DO
      return designation;
END DO
```

Method Description : This method return value assigned to instance variable designation and does not allow to change the value store in the private instance.

.

- **Method name:-** setDesignation(String designation)

Pseudo code:-

```
CREATE METHOD setDesignation(String designation)
DO
    this.designation=designation;
END DO
```

Method Description : This method accept the values from user and the value will be stored in variable designation but it didn't return any value.

- **Method name:-** getJobType ()

Pseudo code:-

```
CREATE METHOD getJobType ()
DO
    return job_type;
END DO
```

Method Description : This method return value assigned to instance variable job_type and does not allow to change the value store in the private instance

- **Method name:-** setJobType(String job_type)

Pseudo code:-

```
CREATE METHOD setJobType(String job_type)
DO
    this.job_type=job_type;
END DO
```

Method Description: This method accept the values from user and the value will be stored in variable job_type but it didn't return any value.

- **Method name**:- displayInfo()

Pseudo code:-

```
CREATE METHOD displayInfo()
DO
    print  "Vacancy Number : "CALL  getVacancyNumber()
```

```
        print ”Designation : “ CALL getDesignation()

        print “Job Type : “CALL getJobType()


END DO
```

Method Description: This Display VacancyNumber,Designation and JobType.

3.2 FullTimeStaffHire

- **Method name:-** getSalary()

Pseudo code:-

```
CREATE METHOD getSalary(){

DO

    return salary;

end do

}
```

Method Description: This method return value assigned to instance variable salary and does not allow to change the value store in the private instance

- **Method name:-** getWorkingHour()

Pseudo code:-

```
CREATE METHOD getWorkingHour(){

do

    return workingHour

end do

}
```

Method Description: This method return value assigned to instance variable working hour and does not allow to change the value store in the private instance

- Method name:- getStaffName()

Pseudo code:-

```
CREATE METHOD getStaffName(){
do
        return staffName
end do
}
```

Method Description: This method return value assigned to instance variable staffName and does not allow to change the value store in the private instance

- **Method name:-** getJoiningDate()

Pseudo code:-

```
CREATE METHOD getJoiningDate(){
do
        return joiningDate
end do
}
```

Method Description This method return value assigned to instance variable joiningDateand does not allow to change the value store in the private instance

- **Method name**:- getQualification()

Pseudo code:-

```
CREATE METHOD getQualification(){

Do

        return qualification

end do}
```

Method Description: This method return value assigned to instance variable qualification and does not allow to change the value store in the private instance

- **Method name**:- getAppointedBy()

Pseudo code:-

```
CREATE METHOD getAppointedBy(){

do

        return appointedBy

end do


}
```

Method Description: This method return value assigned to instance variable aapointedBy and does not allow to change the value store in the private instance

- **Method name**:- getJoined()

Pseudo code:-

```
CREATE METHOD getJoined(){

do

        return joined

end do}
```

Method Description: This method return value assigned to instance variable joined and does not allow to change the value store in the private instance

- **Method name:-** setSalary(int newSalary)

Pseudo code:-

```
CREATE METHOD setSalary(int newSalary ){

DO

                    if(joined==false){

DO

    this.salary=salary;

  }

END IF

  else{

   DISPLAY:("It is not possible to change the salary of hired staff for the post of "+ call getDesignation());

  }

END ELSE

End DO
```

Method Description: This method accept the values from user and the value will be stored in variable salary if and only joined is false if joined is true the the salary cannot be changed and it will display It is not possible to change the salary of hired staff for the post of "+ call getDesignation()

- **Method name:-** setWorkingHour(int workingHour)

Pseudo code:-

```
CREATE METHOD setWorkingHour(int workingHour){
do
        this.workingHour=workingHour;
end do
}
```

Method Description: This method accept the values from user and the value will be stored in variable workinghour . this method does not checked whether the joined is true or false. If joined is true then also it can changed the value of working hour

- **Method name:-** fullhire (String staffName, String joinigDate, String qulafication, String appointedBy)

Pseudo code:-

```
CREATE METHOD fullhire ( String staffName, String joinigDate, String qulafication,
String appointedBy){
DO
    if(joined==false){
    this.staffName=staffName;
    this.joinigDate=joinigDate;
    this.qulafication=qulafication;
    this.appointedBy=appointedBy;
    DISPLAY("Staff has been hired");
    joined=true;
   }
ENF IF
```

```
  else{
        DISPLAY:( getStaffName() + " has already been hired on date "+CALL getJoinigDate()+" with qulafiation "+CALL getQulafication()+" apppoint by "+CALL getAppointedBy());
END ELSE
      }
  }
END DO
```

Method Description: This method only works when object of this class is created. After the created object by the helps of this method user can input value of staffName,joiningDate,qulafilaction,AppointBy is Boolean joined is false he value will stored in local variables if Boolean joined is ture then it will display getStaffName() + " has already been hired on date "+CALL getJoinigDate()+" with qulafiation "+CALL getQulafication()+" apppoint by "+CALL getAppointedBy());

- Method name:- displayInfo()

PSEDUO CODE:-

```
CREATE METHOD displayInfo(){
DO
  super.displayInfo();
     if(joined==true){
        DISPLAY("----------------------------------------");
        DISPLAY("Staff Name = " + staffName);
```

```
        DISPLAY("Joined Date = " + joinigDate);

        DISPLAY("Salary = " + salary);

        DISPLAY("Working Hour= " + workingHour);

        DISPLAY("qulafication = " + qulafication );

        DISPLAY("Appointed By = " + appointedBy);

        DISPLAY("----------------------------------------");

    }

END IF

    else{

        DISPLAY("Staff has not been hired. please hire the staff for " +CALL getDesignation() );

    }

  }

END ELSE

END DO
```

Method Description: This method print staff name, joining date,salay,working hour,qulatication,appointment if boolean condition of joined is true other wise it will display Staff has not been hired. please hire the staff +CALL getDesignation().

3.3 PartTimeStaffHire

- **Method name:-** getWorkingHour()

Pseudo code:-

```
CREATE METHOD getWorkingHour(){
do
        return workingHour
end do
}
```

Method Description: This method return integer value assigned to instance variable working hour and does not allow to change the value store in the private instance

- **Method name:-** getWagePerHour()

Pseudo code:-

```
CREATE METHOD getWagePerHour(){
do
        return wagePerHour;
end do
}
```

Method Description: This method return integer value assigned to instance variable wagePerHour and does not allow to change the value store in the private instance

- **Method name:-** getStaffName()

Pseudo code:-

```
CREATE METHOD getStaffName(){
do
        return staffName
end do
}
```

Method Description: This method return value assigned to instance variable staffName and does not allow to change the value store in the private instance.

- **Method name**:- getJoiningDate()

Pseudo code:-

```
CREATE METHOD getJoiningDate(){

do

        return joiningDate

end do

}
```

Method Description This method return value assigned to instance variable joiningDateand does not allow to change the value store in the private instance.

- **Method name:-** getQualification()

Pseudo code:-

```
CREATE METHOD getQualification(){

do

        return qualification

end do}
```

Method Description: This method return value assigned to instance variable qualification and does not allow to change the value store in the private instance

- **Method name:-** getAppointedBy()

Pseudo code:-

```
CREATE METHOD getAppointedBy(){

do

        return appointedBy
```

```
end do


}
```

Method Description: This method return value assigned to instance variable aapointedBy and does not allow to change the value store in the private instance.

- **Method name:-** getJoined()

Pseudo code:-

```
CREATE METHOD getJoined(){

do

        return joined

end do}
```

Method Description: This method return value assigned to instance variable joined and does not allow to change the value store in the private instance

- **Method name:-** getShifts()

Pseudo code:-

```
CREATE METHOD getShifts(){

do

return shifts;

end do

}
```

Method Description: This method return value assigned to instance variable shifts and does not allow to change the value store in the private instance

- **Method name:-** getTerminated()

Pseudo code:-

```
CREATE METHOD getTerminated(){
```

```
do
return terminated;
end do
}
```

Method Description: This method return value assigned to instance variable Terminated and does not allow to change the value store in the private instance

- **Method name:-** setShifts(String shifts)

Pseudo code:-

```
CREATE METHOD setShifts( String shifts ){
DO
if(joined==false)
    {
        this.shifts=shifts;
    DISPLAY("Shift has been changed.");
   }
END IF
   else
   {
       DISPLAY("Shift cannot be changed as staff has already been appointed. ");
END ELSE
END DO
   }
```

Method Description: This method is worked then objected is created . if joined is false then its value assigned to instance variable by the helps of this keyword and shows shift

is changed. If value assigned to instance variable is true then it display shift has been already changed.

- **Method name:-** setWagesPerHour(int wagePerHour)

Pseudo code:-

```
CREATE METHOD setWagesPerHour(int wagePerHour){
DO
if (joined==false){
   this.wagePerHour=wagePerHour;
   DISPLAY("Wages Per Hour has been changed.");
   }
END IF
END DO
   }
```

Method Description: This method is worked then objected is created . if joined is false then its value assigned to instance variable by the helps of this keyword and shows Wages per day has been changed.

- **Method name:-** setWorkingHour(int workingHour){

Pseudo code:-

```
CREATE METHOD setWorkingHour(int workingHour){
DO
   if (joined==false){
   this.workingHour=workingHour;
   DISPLAY("Working Hour has been changed.");
```

END IF

END DO

}

Method Description: This method is worked then objected is created . if joined is false then its value assigned to instance variable by the helps of this keyword and working hour has been changed.

- **Method name:-** partTimehire(String staffName, String joinigDate, String qulafication, String appointedBy){

Pseudo code:-

CREATE METHOD partTimehire(String staffName, String joinigDate, String qulafication, String appointedBy){

DO

if(joined==false){

this.staffName=staffName;

this.joinigDate=joinigDate;

this.qulafication=qulafication;

this.appointedBy=appointedBy;

joined=true;

terminated=false;

DISPLAY(staffName+"Staff has been hired on"+CALLgetJoinigDate());

END IF

}

else{

```
        DISPLAY( CALLgetStaffName() + " has already been hired "+CALLgetJoinigDate() +" by " +CALLgetAppointedBy());
END ELSE
      }END DO
   }
```

Method Description: This method only works when object of this class is created. After the created object by the helps of this method user can input value of staffName,joiningDate,qulafilaction,AppointBy is Boolean joined is false the value will stored in local variables and display staff has been hired. if Boolean joined is ture then it will display staff has been has already been hired

- **Method name:-** Terminate()

Pseudo code:-

```
CREATE METHOD  Terminate()
DO
      if( terminated==true){
         DISPLAY("The Staff's record has already been terminated.");
       }
END IF
       else{
         DISPLAY( staffName + " has been terminated.");
         this.staffName="";
         this.joinigDate="";
         this.qulafication="";
         this.appointedBy="";
```

```
        joined=false;
        terminated=true;
      }
  }
END ELSE
      }END DO
   }
```

Method Description: This method is created in PartTimeStaffHire. It return integer value assigned to instance variable to delete the record of the staff when joined is false. When joined is true it display staff has be already terminate.

- **Method name:-** displayInfo()

Pseudo code:-

```
CREATE METHOD  displayInfo()
DO
super.displayInfo();
      if(joined==true){
      DISPLAY("----------------------------------------");
        DISPLAY("Staff Name = " + staffName);
        DISPLAY("Joined Date = " + joinigDate);
        DISPLAY("Working Hours = " + workingHour);
        DISPLAY("Wage Per Hour = " + wagePerHour);
        DISPLAY("Qualification = " + qulafication );
        DISPLAY("Appointed By = " + appointedBy);
```

```
        DISPLAY("Income Per Day= " + (workingHour*wagePerHour));
        DISPLAY("----------------------------------------");
    }
END IF
    else{
        DISPLAY("Staff has not been hired yet. please hire the staff for " +CALL getDesignation());
    }
  }
END ELSE
     }END DO
   }
```

Method Description: This method is created to display all the details of staff which has been hired. Super keyword is used in order to call super class Staffhire. If Boolean joined is true this instance variable data will be displayed . if joined is false then it wil show staff has not been hired yet.

4.TESTING

4.1 TEST 1 Inspect in PartTimeStaffHire Class and re-inspect the PartTimeStaffHire Class and display it

Objective	Inspect in PartTimeStaffHire Class and re-inspect the PartTimeStaffHire Class and displayind all
Action	PartTimeStaffHire object's is created and value are assigned in the constructor vacancyNumber=1 designation="Co-manager" jobType="full Time" workingHour=6 wagesPerHour=1200 shifts="Day" The object was created successfully and.inspected. Then the method of PartTimeStaff(String staffName, String joinigDate, String qualification, String appointedBy) following value are putted for the arguments. staffName="Nimesh Poudel" joiningDate="2019-02-02"

	qualification=”+2” appointedBy=”youson” The object was re-inspected.
Expected Result	Methods must accept the value entered by the user and assign them to the variable and display it.
Actual Result	The value entered by the user was and assign them to the variable and display it.
Conclusion	The test was successfully performed and screenshot was given below

Table 7 test i

output

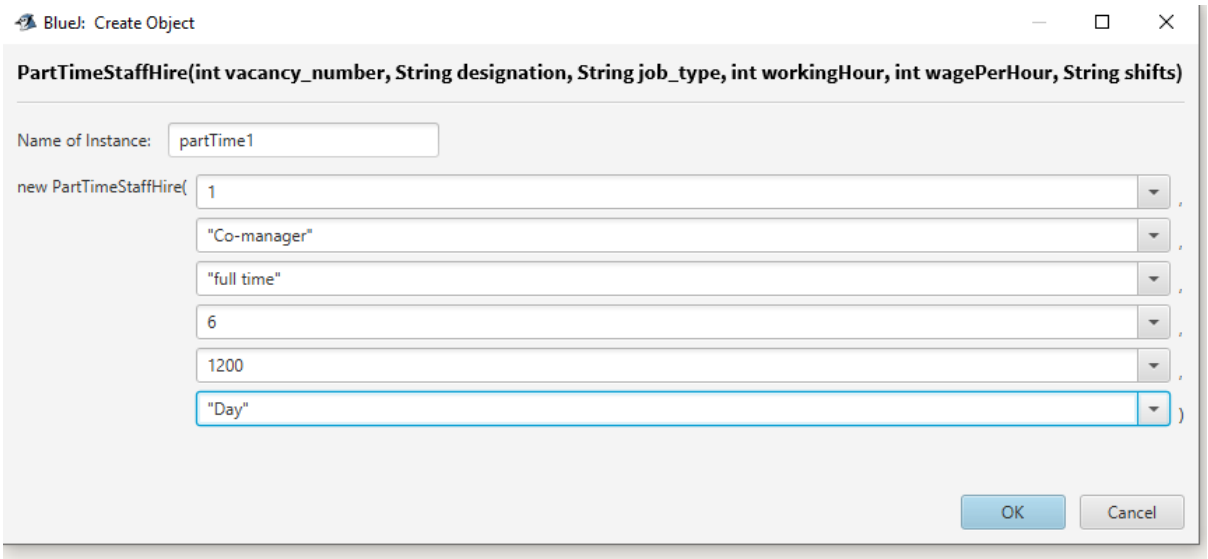


Figure 49 creating object for PartTimeStaffHire

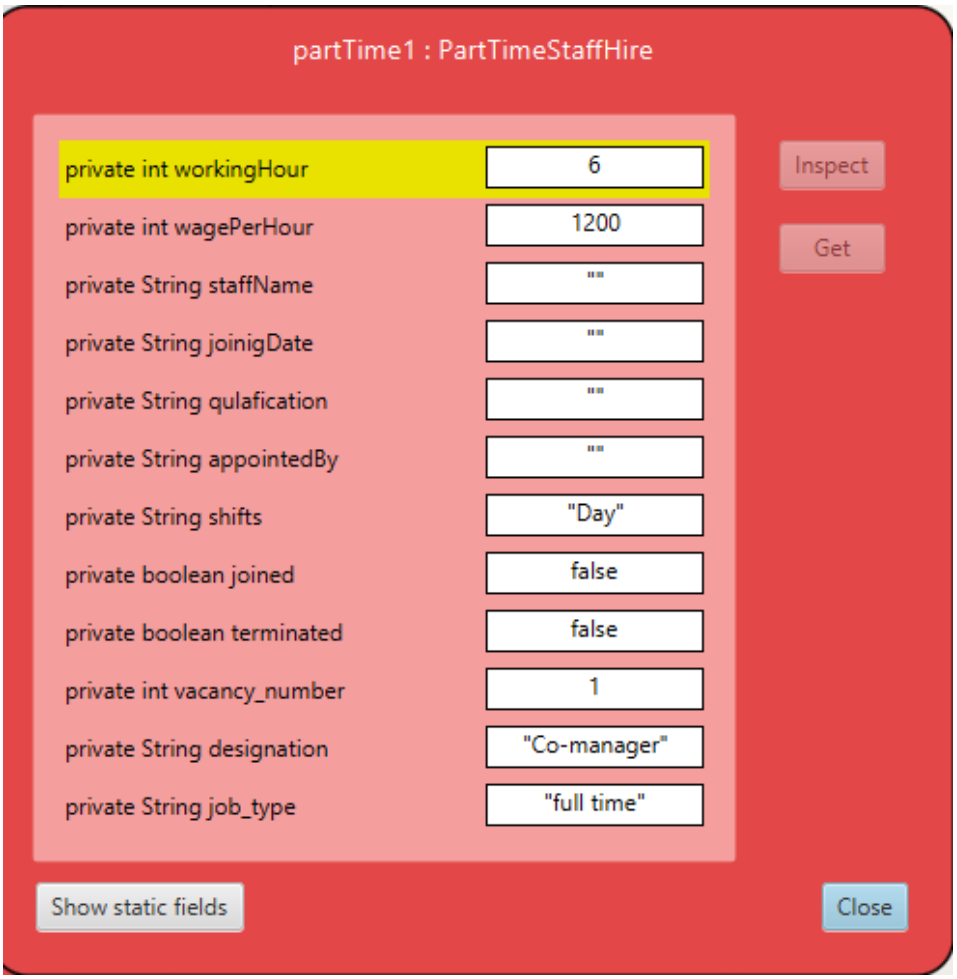


Figure 50 inspecting i for PartTimeStaffHire

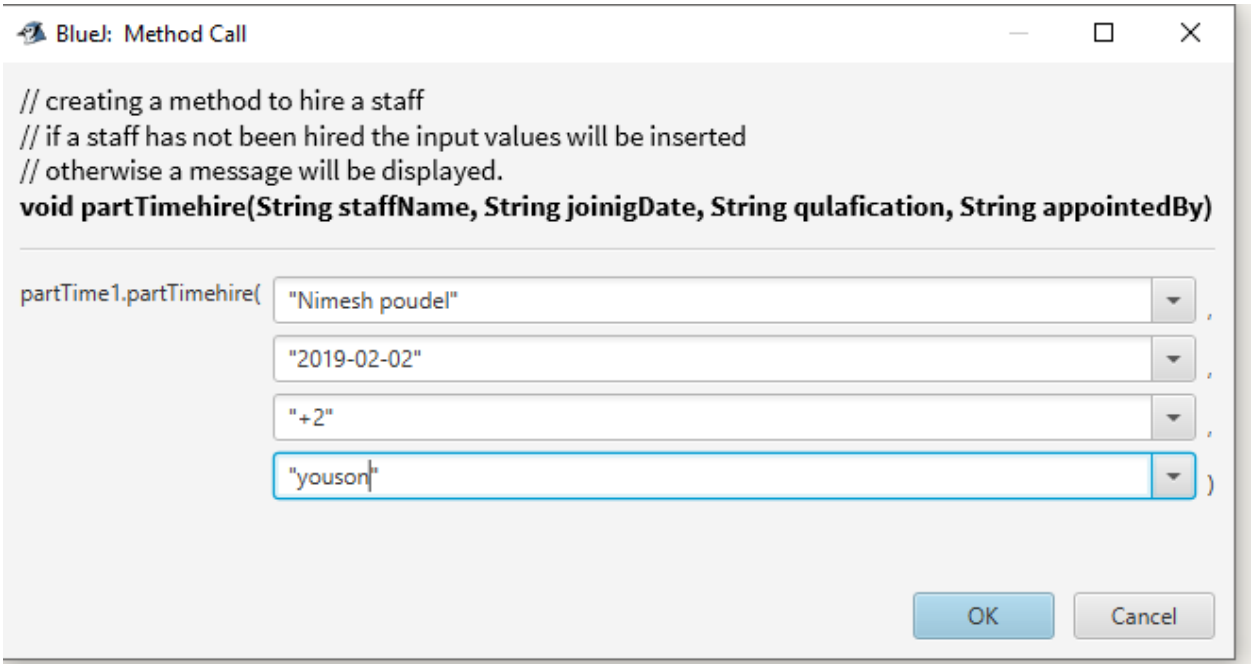


Figure 51 hiring staff for PartTimeStaffHire

Figure 52 inspecting ii for PartTimeStaffHire

BlueJ: Terminal Window - classwork

Options

```
Nimesh poudel has been hired on2019-02-02
```

Figure 53 message after hiring staff for PartTimeStaffHire

BlueJ: Terminal Window - classwork

Options

```
Vacancy no:-1
Designation:-Co-manager
Job Type:-full time
----------------------------------------
Staff Name = Nimesh poudel
Joined Date = 2019-02-02
Working Hours = 6
Wage Per Hour = 1200
Qualification = +2
Appointed By = youson
Income Per Day= 7200
----------------------------------------
```

Figure 54 displaying detail of staff for PartTimeStaffHire

4.2 Test 2 Inspect in FullTimeStaffHire Class and re-inspect the FullTimeStaffHire Class and displaying all

Objective	Inspect in FullTimeStaffHire Class and re-inspect the FullTimeStaffHire Class and displaying all
Action	FullTimeStaffHire object's is created and value are assigned in the constructor vacancyNumber=2 designation="CEO" jobType="FullTime" salary=90000 workingHour=12 " The object was created successfully and.inspected. Then the method of FullTimeStaff(String staffName, String joinigDate, String qualification, String appointedBy) is called and the following value are putted for the arguments. staffName="Ram Shah"

	joiningDate="2020-01-03" qualification="Msc IT" appointedBy="Ramesh" The object was re-inspected.
Expected Result	Methods must accept the value entered by the user and assign them to the variable and display it.
Actual Result	The value entered by the user was and assign them to the variable and display it.
Conclusion	The test was successfully performed and screenshot was given below

Table 8 test ii

output

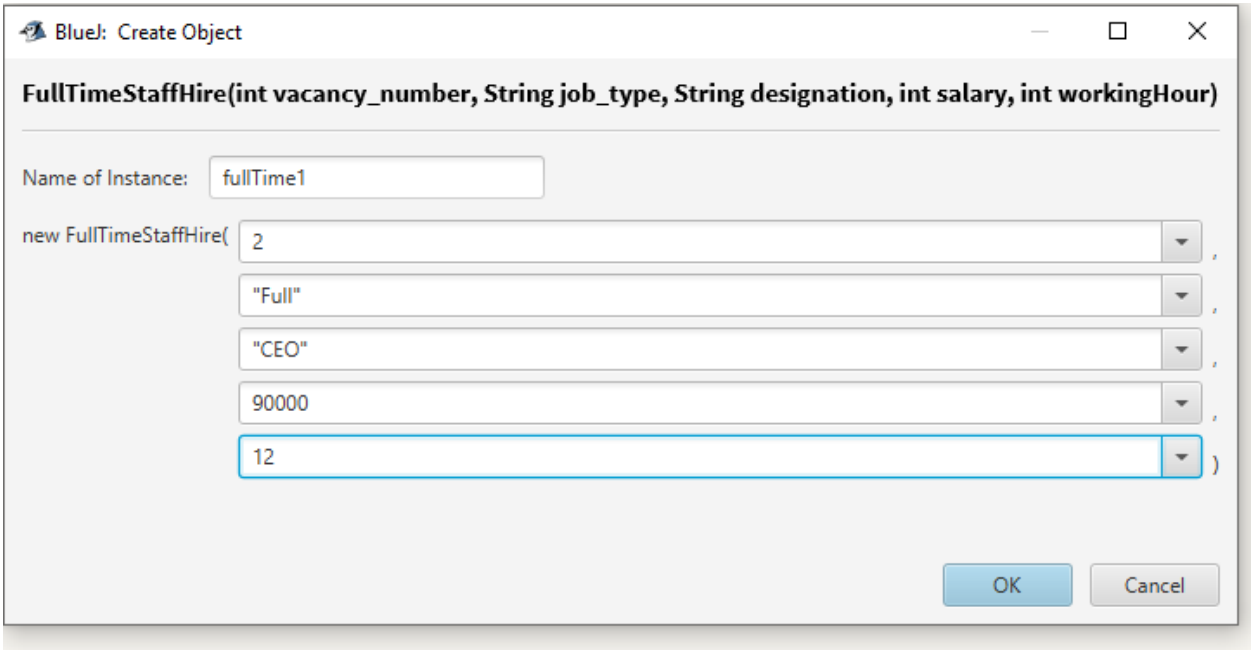


Figure 55 creating object for FullTimeStaffHire

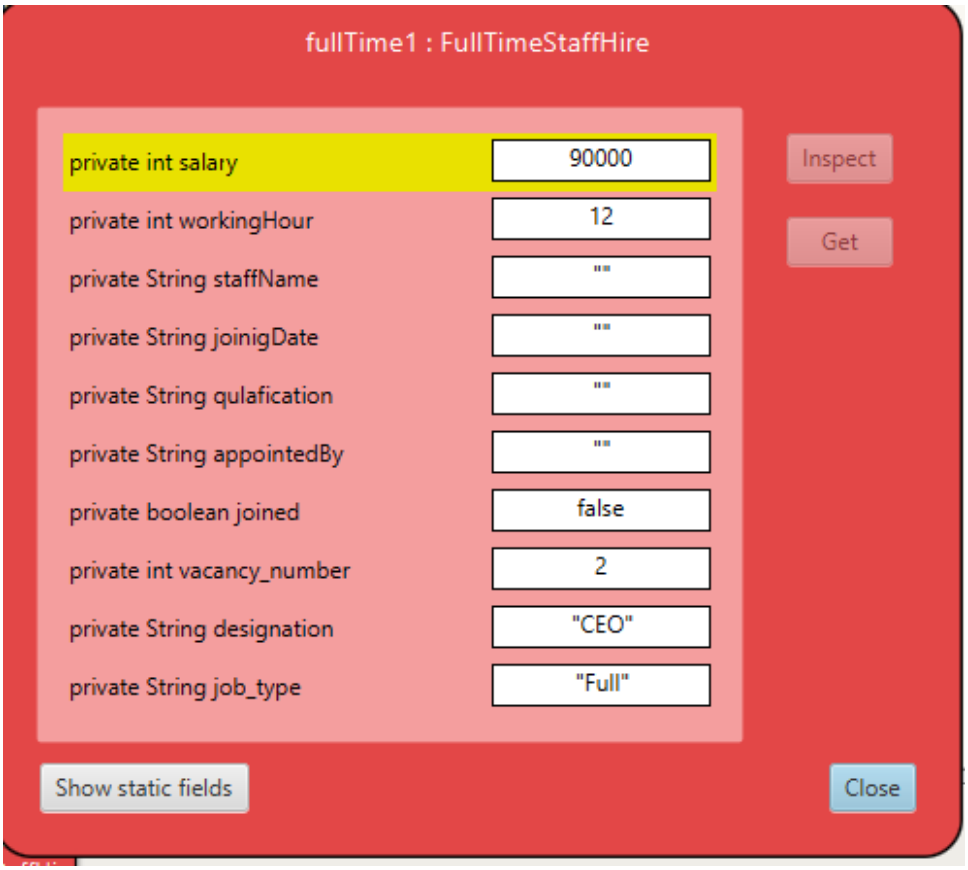


Figure 56 inspecting i for FullTimeStaffHire

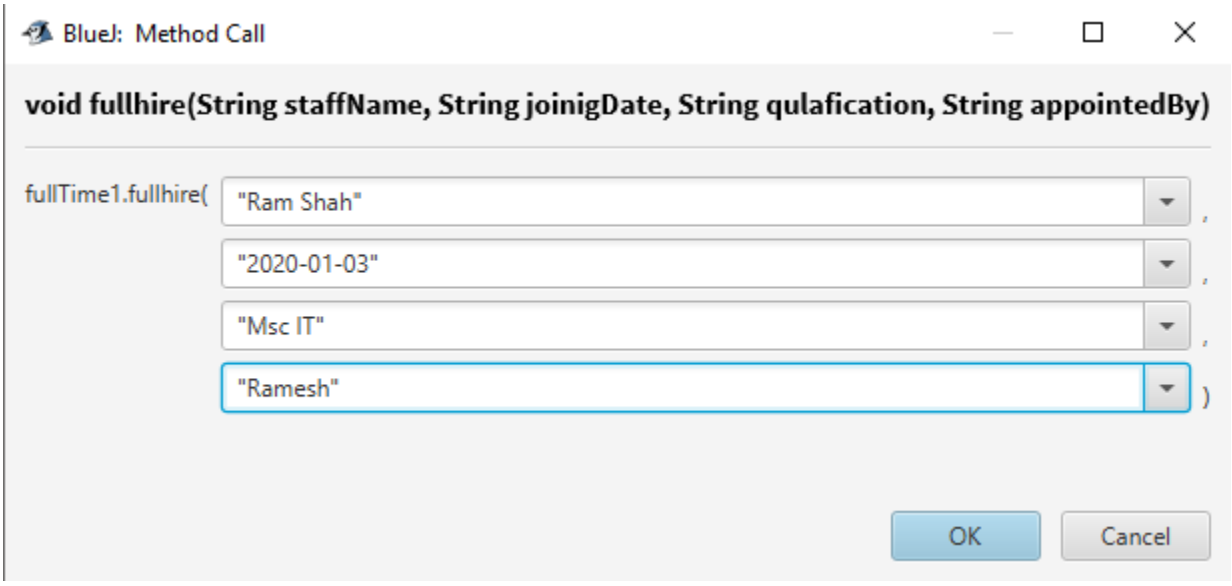


Figure 57 hiring staff for FullTimeStaffHire

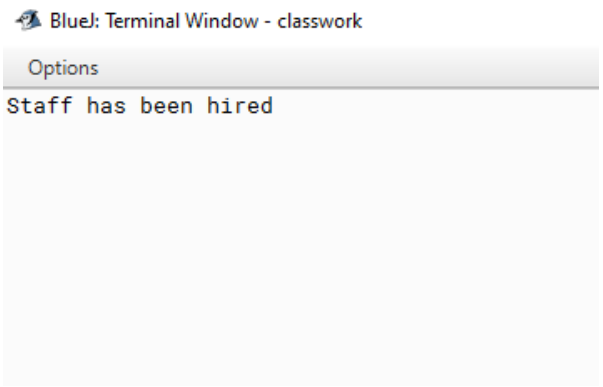


Figure 58 message after hiring staff for FullTimeStaffHire

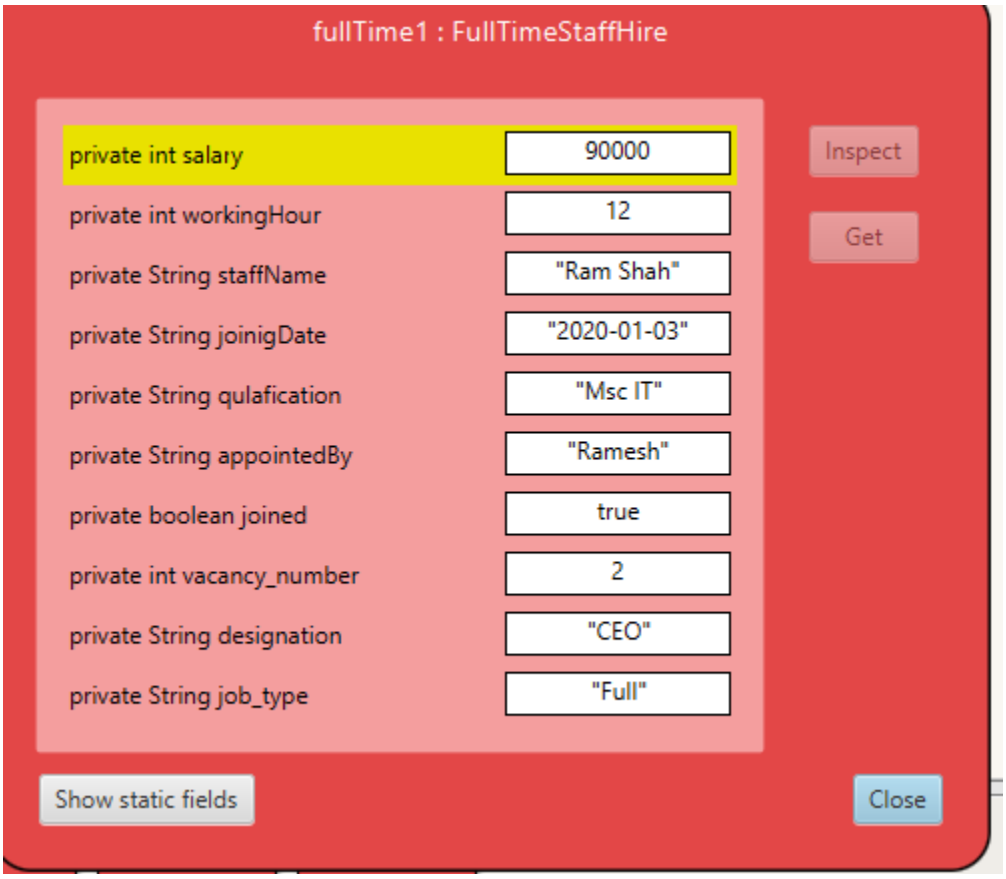


Figure 59 insecting ii for FullTimeStaffHire

BlueJ: Terminal Window - classwork

Options

```
Vacancy no:-2
Designation:-CEO
Job Type:-Full
----------------------------------------
Staff Name = Ram Shah
Joined Date = 2020-01-03
Salary = 90000
Working Hour= 12
qulafication = Msc IT
Appointed By = Ramesh
----------------------------------------
```

Figure 60 Details about staff for FullTImeStaffHire

4.3 TEST 3 Inspect in PartTimeStaffHire Class and to change the value of terminate and joined and re-inspect the PartTimeStaffHire Class and display all

Objective	Inspect in PartTimeStaffHire Class and to change the value of terminate and joined and re-inspect the PartTimeStaffHire Clas And display all
Action	PartTimeStaffHire object's is created and value are assigned in the constructor and again making terminate value true and joined false and display it vacancyNumber=1 designation="Co-manager" jobType="full Time" workingHour=6

	wagesPerHour=1200 shifts=”Day” The object was created successfully and.inspected. Then the method of PartTimeStaff(String staffName, String joinigDate, String qualification, String appointedBy) following value are putted for the arguments. staffName=”Nimesh Poudel” joiningDate=”2019-02-02” qualification=”+2” appointedBy=”youson” The object was re-inspected.
Expected Result	Methods must accept the value entered by the user and assign them to the variable and at first terminate value is false and joined is true but last terminate value should be false and joined is true and display it

Actual Result	The value entered by the user was and assign them to the variable at first terminate value is false and joined is true but last terminate value should be false and joined is true and display it.
Conclusion	The test was successfully performed and screenshot was given below

Table 9 test iii

output

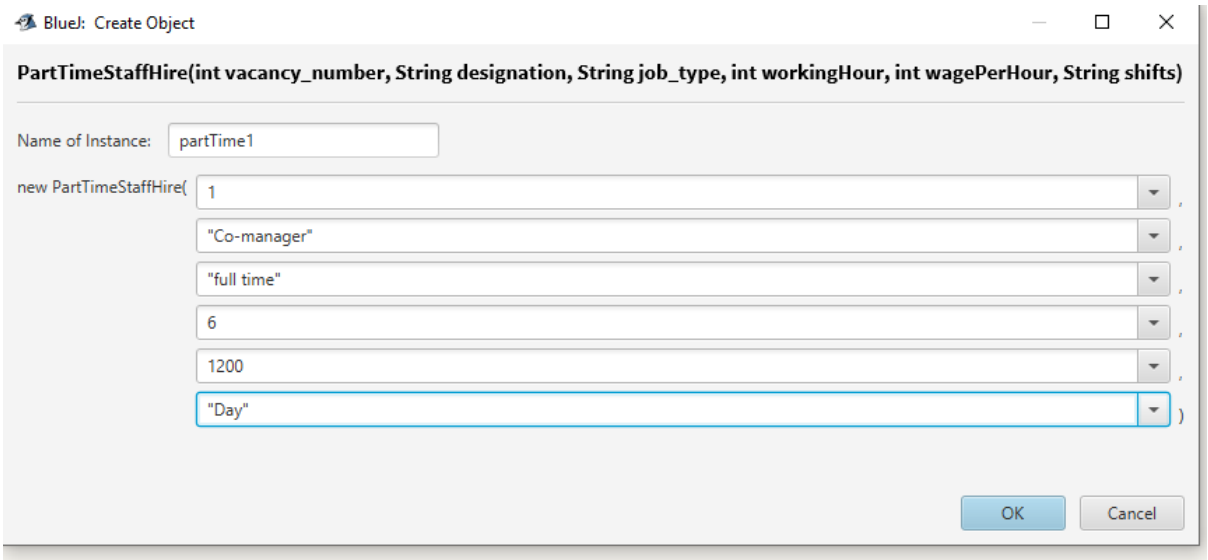


Figure 61 creating object for PartTimeStaffHire

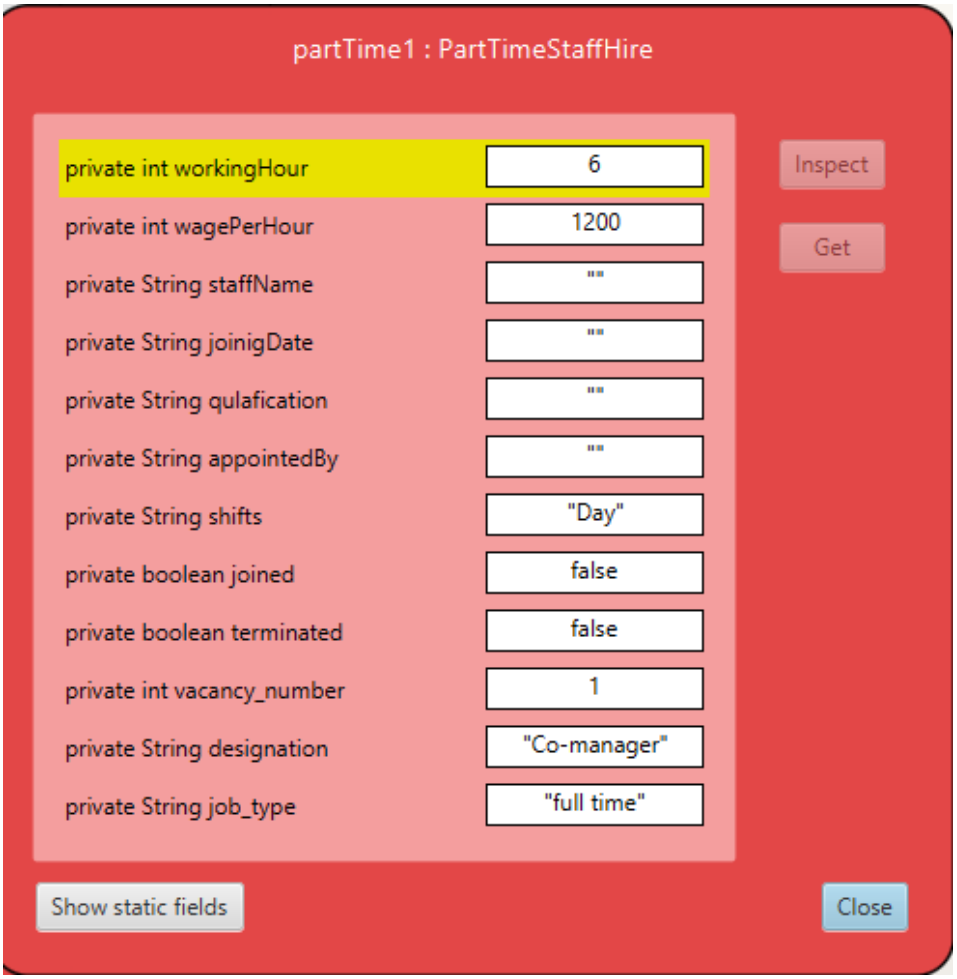


Figure 62 inspecting i for PartTimeStaffHire

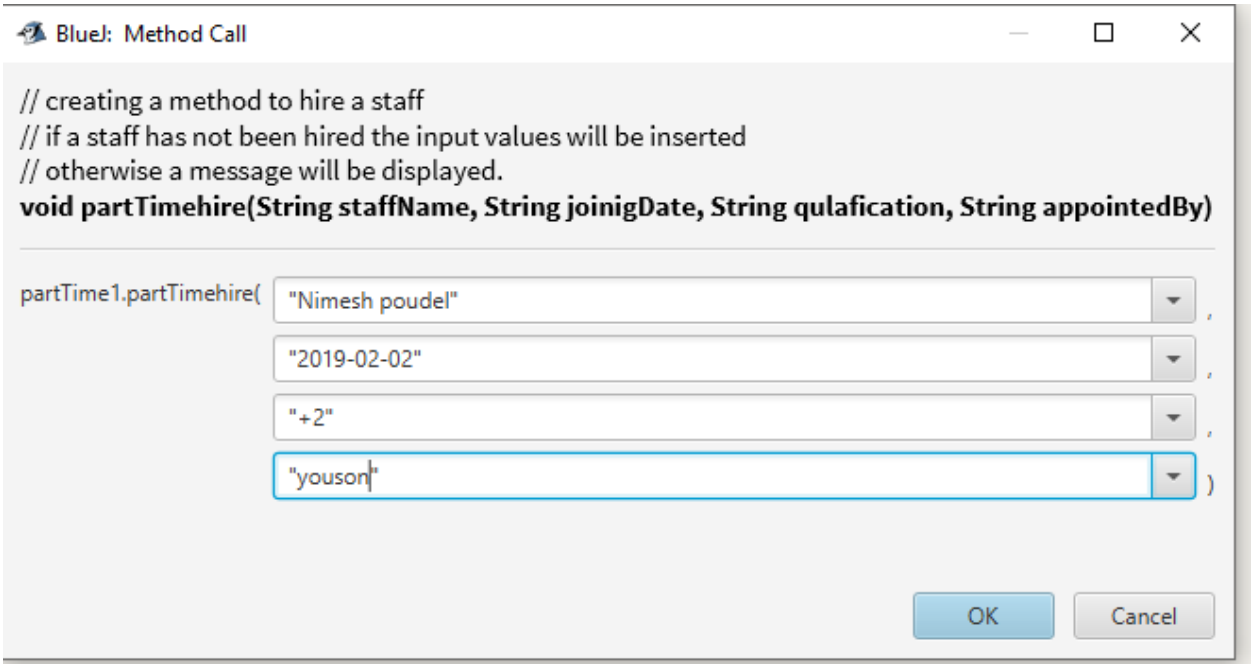


Figure 63hiring staff for PartTimeStaffHire

Figure 64 inspecting ii for PartTimeStaffHire

BlueJ: Terminal Window - classwork

Options

```
Nimesh poudel has been hired on2019-02-02
```

Figure 65 message after hiring staff

BlueJ: Terminal Window - classwork

Options

```
Vacancy no:-1
Designation:-Co-manager
Job Type:-full time
----------------------------------------
Staff Name = Nimesh poudel
Joined Date = 2019-02-02
Working Hours = 6
Wage Per Hour = 1200
Qualification = +2
Appointed By = youson
Income Per Day= 7200
----------------------------------------
```

Figure 66 details about staffs for PartTimestaffHire

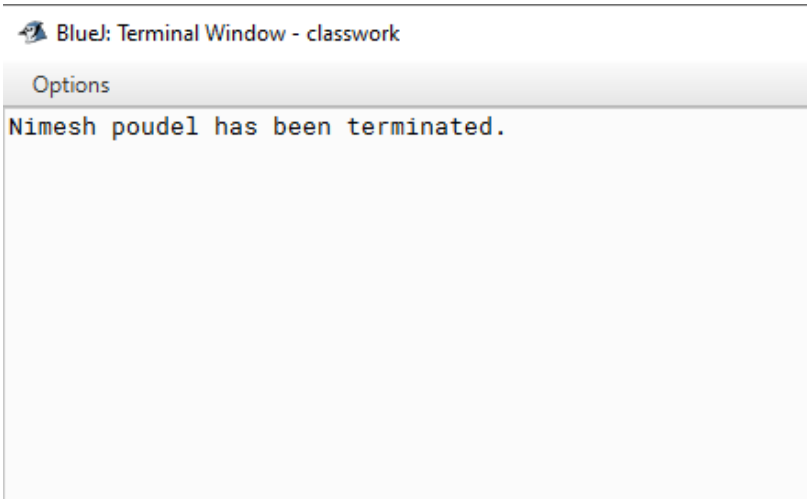


Figure 67 terminating staff for PartTimeStaffHire

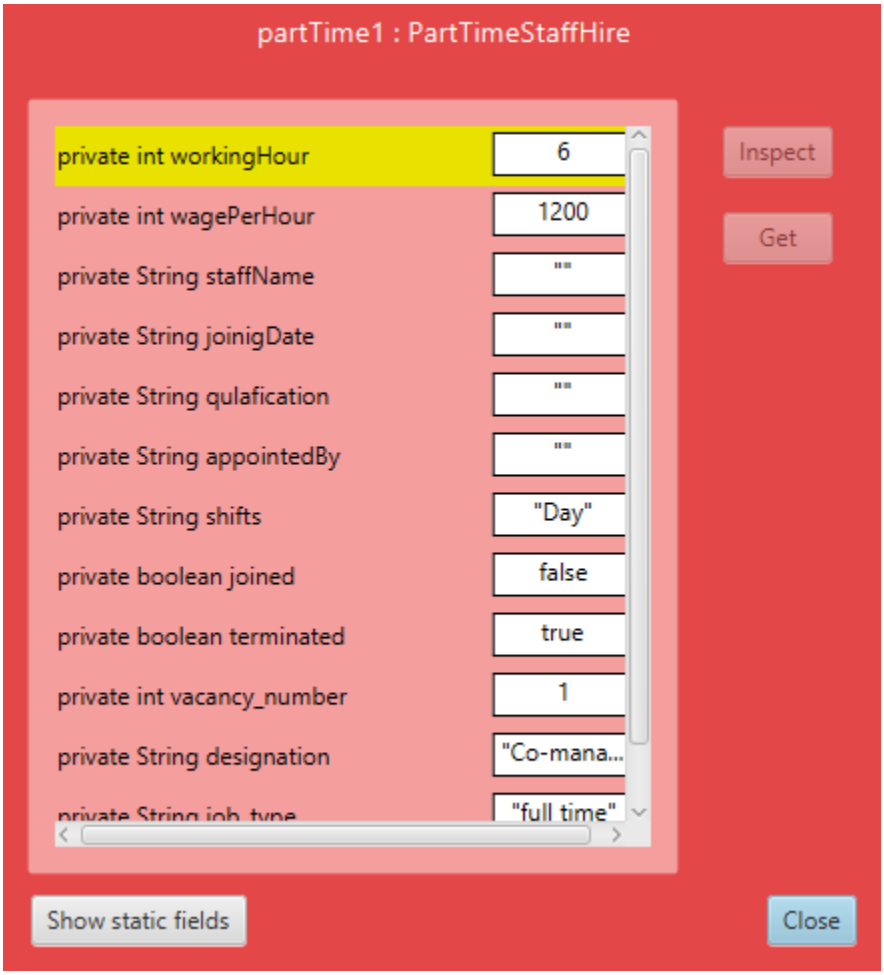


Figure 68 inspecting iii for PartTimeStaffHire

4.4 Test 4 Display all the details of staffHire

Objective	Display all the details of staffHire
Action	PartTimeStaffHire object's is created and value are assigned in the constructor and inspecting them and didplaying them all details vacancyNumber=3 designation="assistance" jobType="full Time" The object was created successfully and.inspected.
Expected Result	Methods must accept the value entered by the user and assign them to the variable
Actual Result	The value entered by the user was and assign them to the variable
Conclusion	The test was successfully performed and screenshot was given below

Table 10 test iv

output

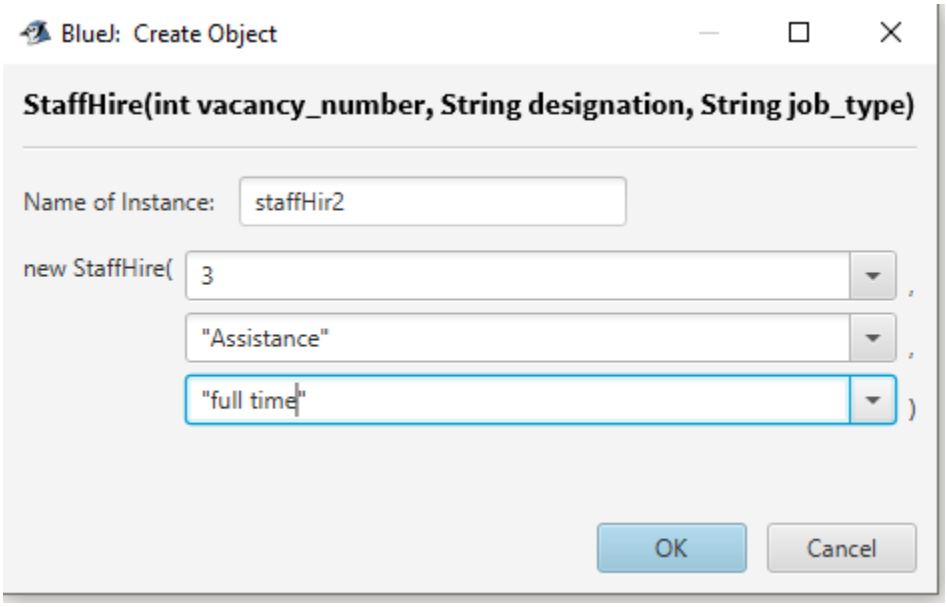


Figure 69 creating object for staffHire

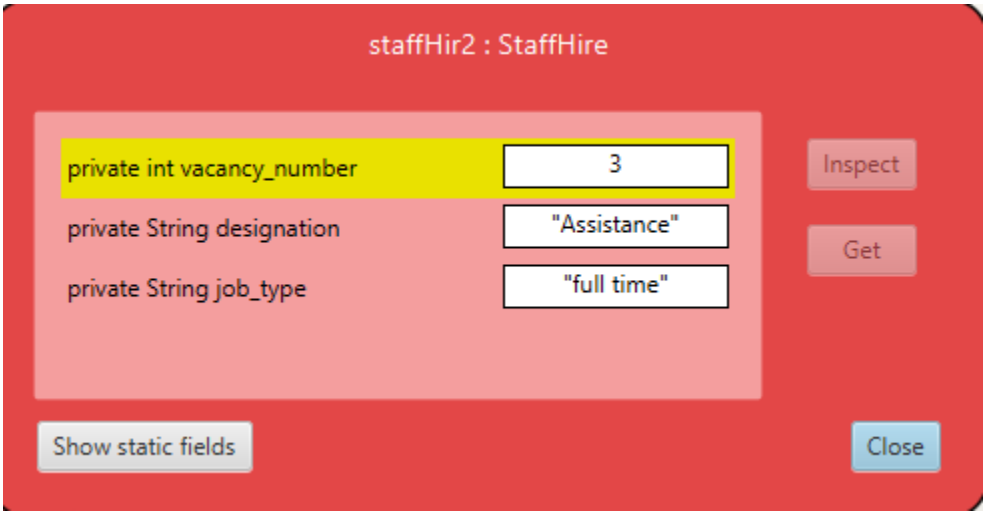


Figure 70 inspecting for StaffHire

BlueJ: Terminal Window - classwork

Options

```
Vacancy no:-3
Designation:-Assistance
Job Type:-full time
```

Figure 71 display about StaffHire

4.5 Test 5 Inspect in PartTimeStaffHire Class and changing the value of shift and re-inspect the PartTimeStaffHire Class

Objective	Inspect in PartTimeStaffHire Class and changing the value of shift and re-inspect the PartTimeStaffHire Class
Action	PartTimeStaffHire object's is created and value are assigned in the constructor vacancyNumber=4 designation="Waiter" jobType="full Time" workingHour=8 wagesPerHour=1000 shifts="Day" The object was created successfully and.inspected.

	Then the method of PartTimeStaff(String staffName, String joinigDate, String qualification, String appointedBy) following value are putted for the arguments. staffName=”Pramod Poudel” joiningDate=”2019-02-02” qualification=”SLC” appointedBy=”Nimesh” The object was re-inspected.
Expected Result	Methods must accept the value entered by the user and assign them to the variable and the value of shift should not changed it should be remain same as first inspected
Actual Result	The value entered by the user was and assign them to the variable and the value of shift was not changed it is remain same as first inspected
Conclusion	The test was successfully performed and screenshot was given below

Table 11 test v

output

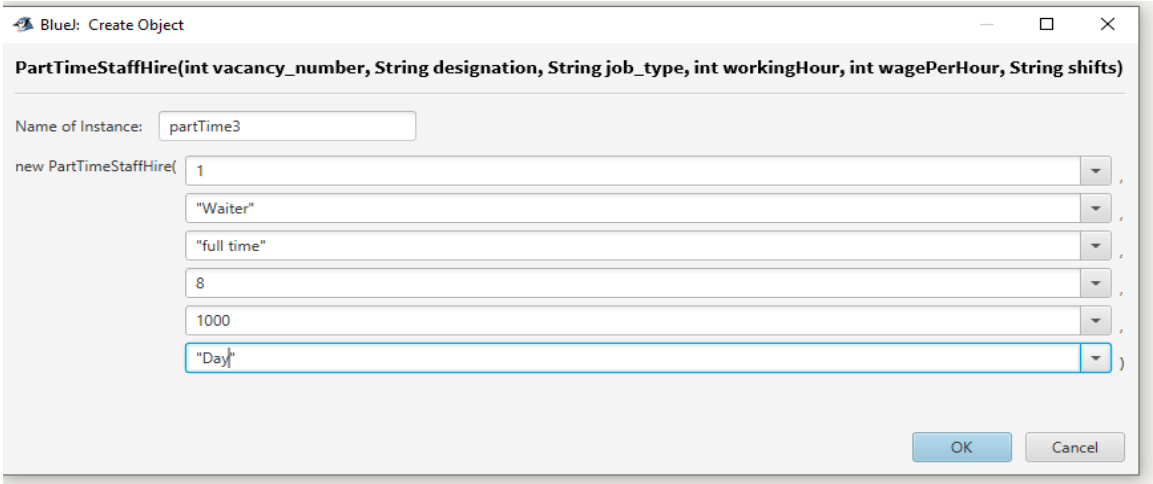


Figure 72 creating object for PartTimeStaffHire

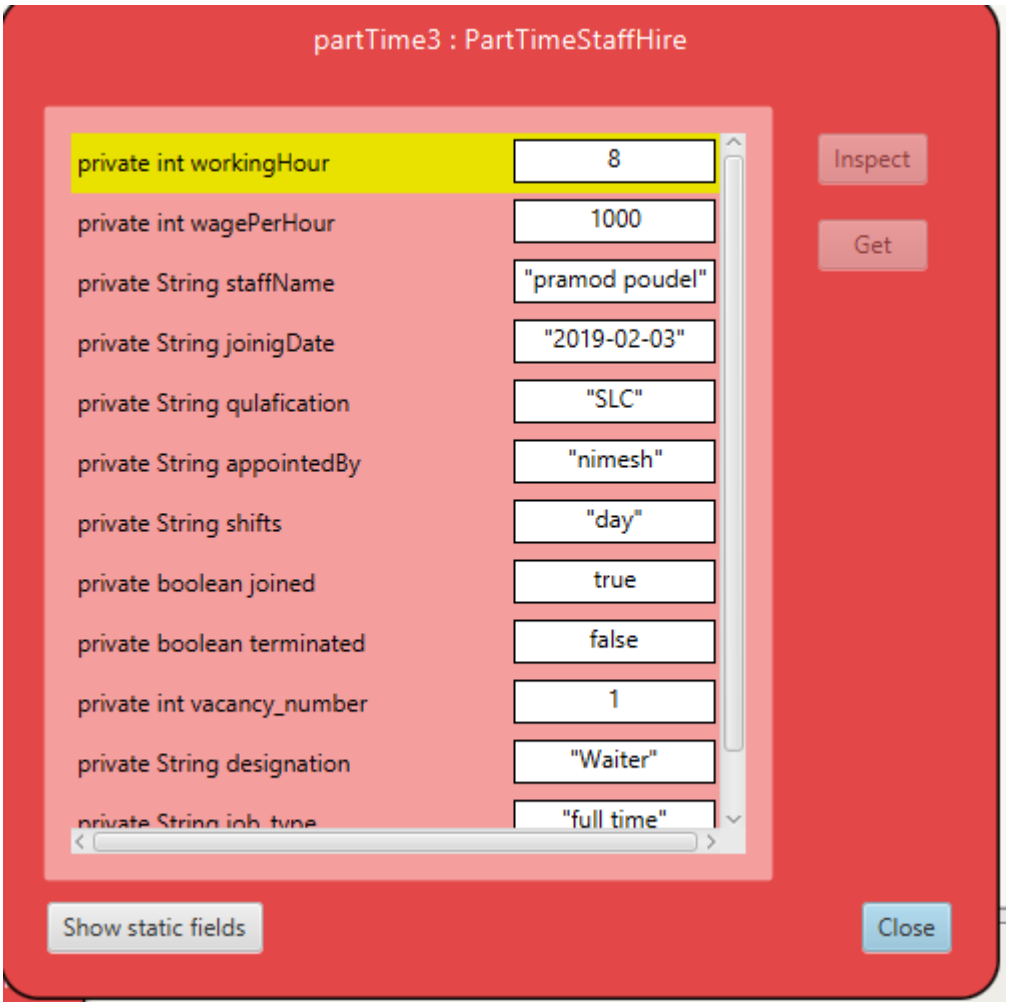


Figure 73 inspecting i for PartTimeStaffHire

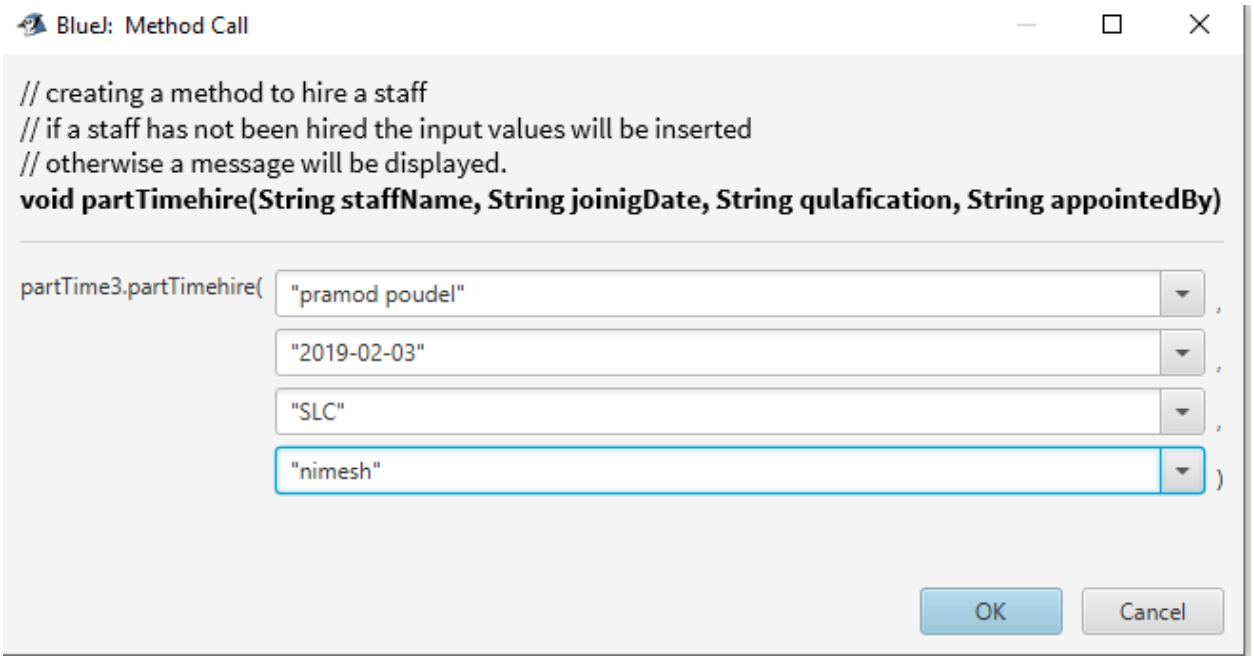


Figure 74 hiring staff for PartTimeStaffHire

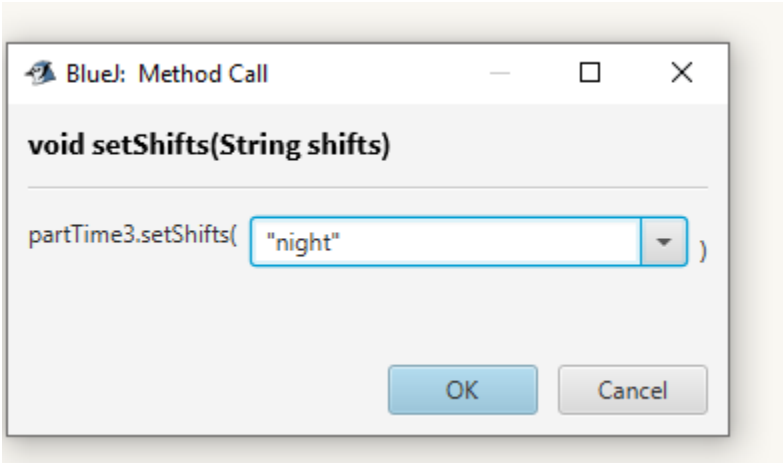


Figure 75 Changing value of shift

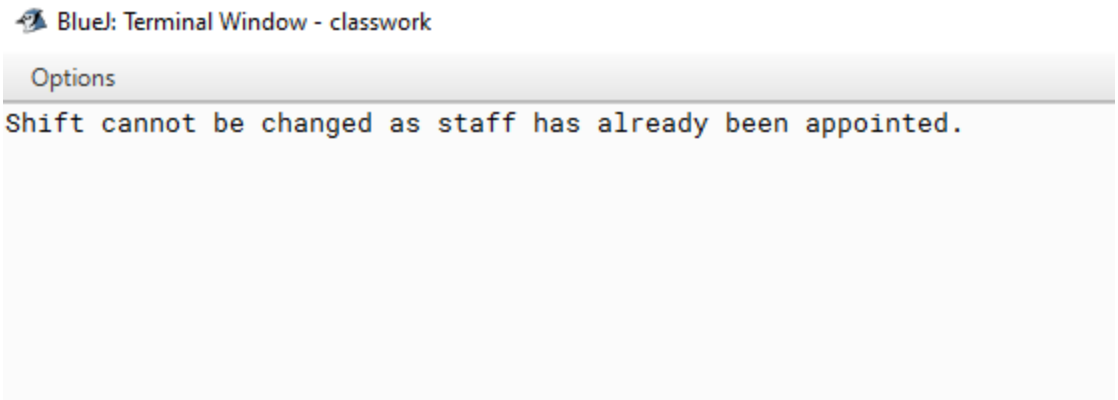


Figure 76 message after changing changing shift

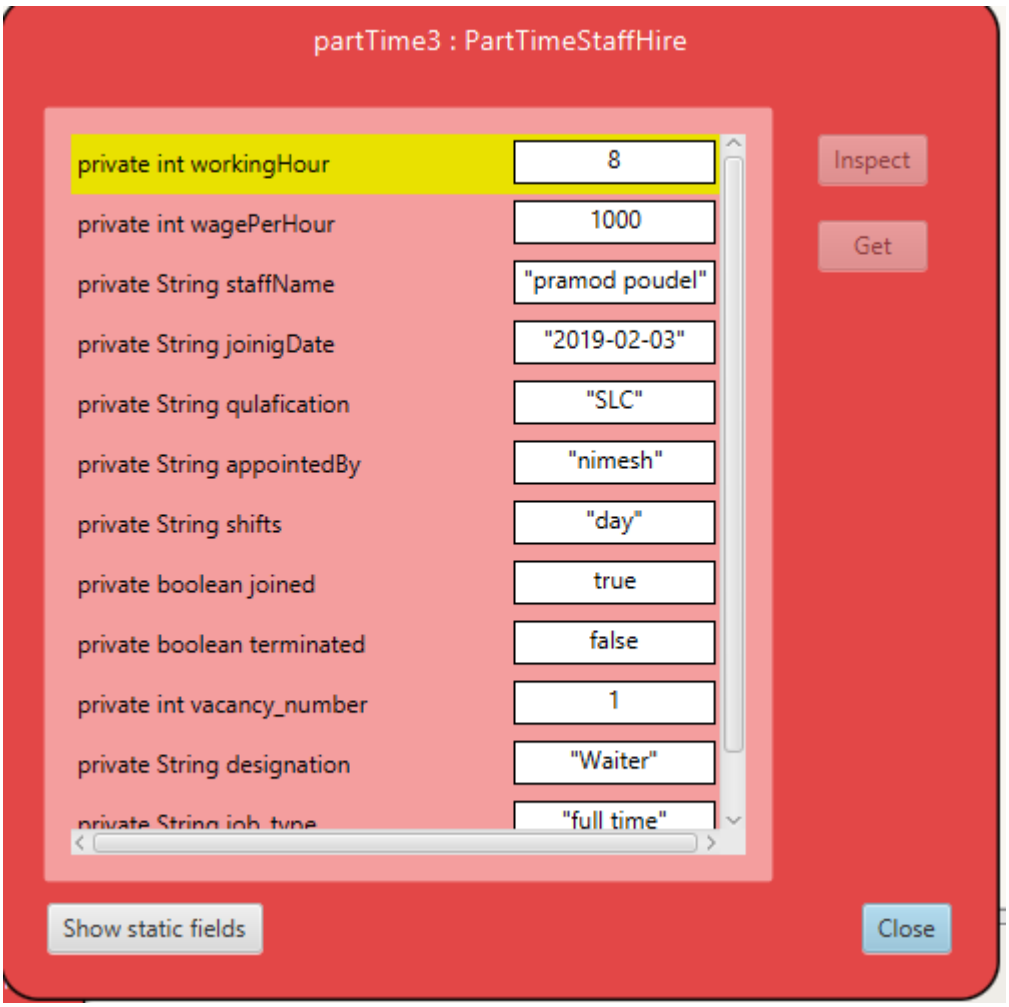


Figure 77 inspecting after changing shift(same)

5. Error detection and correction

5.1 Error in getter

```
public int getWorkingHour(){
    return workingHour;
}
public String getstaffName(){
    return staffName;
}
public String getJoinigDate(){
    return joinigDate;
}
public String getQulafication(){
    return qulafication;
}
```

Figure 78 error in getter method

```
public void fullhire ( String staffName, String joinigDate, String qulafication, String appointedBy){
    if(joined==false){
    this.staffName=staffName;
    this.joinigDate=joinigDate;
    this.qulafication=qulafication;
    this.appointedBy=appointedBy;
    System.out.println("Staff has been hired");
    joined=true;
    }
    else{
      System.out.println( getStaffName() + " has already been hired on date "+getJoinigDate()+" with qulafia
cannot find symbol -   method getStaffName(); maybe you meant:
getstaffName
```

Figure 79 error in getter method

The first error was in getter method of staffname during calling it. The error was actually small size “s”. In method staffname it was small size s and durring calling staffname it was capital “S” get so it says cannot find symbol.

In order to correct this error first the getter method of staffname was checked because an error is shown in that field and in getter method s was made capital in staffname i.e in place of getstaffname() this getStaffname() was made. Correcting error screen short is below

Figure 80 correction error of getter method

```
}
public String getStaffName(){
    return staffName;
}
public String getJoinigDate(){
    return joinigDate;
}
public String getQulafication(){
    return qulafication;
}
```

5.2 Boolean datatype error in joined

```
        return workingHour;
    }
    public String getStaffName(){
        return staffName;
    }
    public String getJoinigDate(){
        return joinigDate;
    }
    public String getQulafication(){
        return qulafication;
    }
    public String getAppointedBy(){
        return appointedBy;
    }
    public String getJoined(){
        return joined;
    }
    incompatible types: boolean cannot be converted to java.lang.String
    public void setSalary(int Salary){
        if(joined==false){
        this.salary=salary;
    }
    else{
        System.out.println("It is not possible to change the salary of hired staff for the post of "+getDesignation());
```

Figure 81 error in datatype in joined

The error was data in getter method of joined for private instance variable.

In order to correct this error, the data type of the private instance variable joined was checked and found the datatype is String which was a error because getter method for variable joined return String type but variable joined was returning a boolean value. The return type of the getter method of joined was change to Boolean from String. Correcting error screen short is below

```
    }
    public String getAppointedBy(){
        return appointedBy;
    }
    public Boolean getJoined(){
        return joined;
    }

    public void setSalary(int Salary){
        if(joined==false){
        this.salary=salary;
    }
```

Figure 82 correcting error in datatype joined

5.3 Error in '=' symbols

```
public void fullhire ( String staffName, String joinigDate, String qulafication, String appointedBy){
    if(joined=false){
    this.staffName=staffName;
    this.joinigDate=joinigDate;
    this.qulafication=qulafication;
    this.appointedBy=appointedBy;
    System.out.println("Staff has been hired");
    joined=true;
    }
    else{
      System.out.println( getStaffName() + " has already been hired on date "+getJoinigDate()+" with qulafia
   }

  public void displayInfo(){
      super.displayInfo();
      if(joined=true){
      System.out.println("----------------------------------------");
      System.out.println("Staff Name = " + staffName);
      System.out.println("Joined Date = " + joinigDate);
      System.out.println("Salary = " + salary);
      System.out.println("Working Hour= " + workingHour);
      System.out.println("qulafication = " + qulafication );
      System.out.println("Appointed By = " + appointedBy);
      System.out.println("----------------------------------------");
    }
    else{
```

Figure 83 error in symbols in if else

```
BlueJ: Terminal Window - classwork
Options
has already been hired on date  with qulafiation  apppoint by
```

Figure 84 unwanted result due to error in if else

This error occurred while the program was fully compiled and running display method fo the class FullTimeStaffHire . The error was in if else statement. Inside of if.. else statement only one '=' was use in condition and the actual result was not come. By this error program skip this statement and directly print the unwanted output.

In order to correct this error if else statement of private instance variable of displayinfo and full hire was checked and found error. Two symbols '==' was made changed and compiled the application and test the program was done and the error correction was done successfully. Correcting error screen short is below

```
public void fullhire ( String staffName, String joinigDate, Stri
    if(joined==false){
    this.staffName=staffName;
    this.joinigDate=joinigDate;
    this.qulafication=qulafication;
    this.appointedBy=appointedBy;
    System.out.println("Staff has been hired");
    joined=true;
    }
    else{
      System.out.println( getStaffName() + " has already been h
    }

  public void displayInfo(){
      super.displayInfo();
      if(joined==true){
      System.out.println("-----------------------------------
      System.out.println("Staff Name = " + staffName);
      System.out.println("Joined Date = " + joinigDate);
      System.out.println("Salary = " + salary);
```

Figure 85 correcting error in if else statement

6.conscluscion

Coursework was satisfied and finished in time. The project were testing and nearly pushed us through our points of confinement in this restricted measure of time. All over this project, large blunders and false impressions were experienced which were altogether handled by assurance and diligent work in appropriate research on the different topic. Many discussion course teacher and research about each topic was also done for completing this coursework.

During this coursework I learned many this . which is much more important for a developer to develop the application for any organization. Creating methods like getter and setter, constructor , private instance variable , local variable are main topic which I learnt from this coursework. Not only this Creating super class and calling it, proper use of 'this' keyword, loop are clean code writing were also learned. From this coursework I have also learned self learning, and remembering the topic which I have learn in previous days. I have also learn to solve the problem related java by very easy way. Lastly, after completing this coursework I came to know to understand the mechanism of code and to write a code in java's program in a suitable manner

After starting coursework problem regrading capital letter and small letter in a code and proper implementation of code, calling super class were faced. Mainly in using super and this keyword. Some of the symbols like semicolons, equals to were missed and some of the result were wrong. During developing program some of the datatype were wrongly implement.

Heaps of reading, practice, and testing were performed. Tons of analysis was drained the thought of programming coming up with and therefore the development of program. The development of program was glanced through altogether in this part I had learn from course teacher and done lots of research also help me. I had also learn from internet regarding problem which I have been face in this coursework and practiced and research was done with last those wrong result became right which helps to completing this coursework in a time with successful result.

7.References

lifewire.com, n.d. *2019.* [Online]
Available at: https://www.lifewire.com/what-is-java-4172382

wikipedia.org, 2019. *java programming.* [Online]
Available at: https://en.wikipedia.org/wiki/Java_(programming_language)

wikipedia.org, 2020. *blue j.* [Online]
Available at: https://en.wikipedia.org/wiki/BlueJ

8.Appendix

8.1 Code of StaffHire

```
/**
 * Write a description of class StaffHire here.
 *Write a description of
 * @author (Nimesh poudel)
 * @version (0.01)
 */
public class StaffHire
{
  private int vacancy_number;
  private String designation;
  private String job_type;
  //creating a constructor for StaffHire
  public StaffHire(int vacancy_number,String designation,String job_type){
  this.vacancy_number=vacancy_number;
  this.designation=designation;
  this.job_type=job_type;
 }
  //creating accessor methods for each attribute
  public int getVacancyNumber(){
   return vacancy_number;
  }
 public void setVacancyNumber(int vacancy_number){
   this.vacancy_number=vacancy_number;
  }
 public String getDesignation(){
```

```
  return designation;
 }
 public void setDesignation(String designation){
  this.designation=designation;
 }
 public String getJobType(){
  return job_type;
 }
 public void setJobType(String job_type){
  this.job_type=job_type;
 }
 //creating a method to display info about a staffhire
    public void displayInfo(){
   System.out.println("Vacancy no:-"+getVacancyNumber());
   System.out.println("Designation:-"+getDesignation());
   System.out.println("Job Type:-"+getJobType());

    }
}
```

8.2 Code of FullTimeStaffHire

```
 /**
 * Write a description of class FullTimeStaffHire here.
 *
 * @author (Nimesh poudel)
 * @version (0.01)
 */
 public class FullTimeStaffHire extends StaffHire
  {
    private int salary;
    private int workingHour;
    private String staffName;
    private String joinigDate;
    private String qualification;
    private String appointedBy;
    private boolean joined;
    //creating a constructor for FullTimeStaffHire
    public FullTimeStaffHire(int vacancy_number,String job_type,String designation,int salary,int
workingHour)
    {
        super(vacancy_number,designation,job_type);
        this.workingHour=workingHour;
        this.salary=salary;
        staffName="";
        joinigDate="";
        qualification="";
        appointedBy="";
```

```
      joined=false;

    }
    //creating accessor methods for each attribute
     public int getsalary(){
       return salary;
     }
     public int getworkingHour(){
        return workingHour;
      }
     public String getstaffName(){
       return staffName;
     }
     public String getJoinigDate(){
       return joinigDate;
     }
     public String getqualification(){
       return qualification;
     }
     public String getappointedBy(){
       return appointedBy;
     }
     public Boolean getJoined(){
       return joined;
     }
       //creating a method to change the salary of a staff if staff is hire then cannot change salary
      public void setsalary(int salary){
         if(joined==false){
        this.salary=salary;
```

```
      }
        else{
      System.out.println("It is not possible to change the salary of hired staff for the post of
"+getDesignation());
      }
    }
      //creating a setter method the working hour of a staff
     public void  setworkingHour(int workingHour){
      this.workingHour=workingHour;
       }
       //creating a method to display hired staff if already hire on that post displaying already hired
      public void fullhire ( String staffName, String joinigDate, String qualification, String appointedBy){
        if(joined==false){
        this.staffName=staffName;
        this.joinigDate=joinigDate;
        this.qualification=qualification;
        this.appointedBy=appointedBy;
        System.out.println("Staff has been hired");
        joined=true;
        }
        else{
          System.out.println( getstaffName() + " has already been hired on date "+getJoinigDate()+" with
qulafiation "+getqualification()+" apppoint by "
          +getappointedBy());
       }
     }
     //creating a method to display info about a staff
      public void displayInfo(){
          super.displayInfo();
```

```
        if(joined==true){
        System.out.println("----------------------------------------");
        System.out.println("Staff Name = " + staffName);
        System.out.println("Joined Date = " + joinigDate);
        System.out.println("salary = " + salary);
        System.out.println("Working Hour= " + workingHour);
        System.out.println("qualification = " + qualification );
        System.out.println("Appointed By = " + appointedBy);
        System.out.println("----------------------------------------");
      }
      else{
        System.out.println("Staff has not been hired. please hire the staff for " +getDesignation() );
       }
    }

  }
```

8.3 Code of PartTimeStaffHire

```
import javax.swing.*;


/**
 * PartTimeStaffHire is a child class of StaffHire class.
 *
 * @Nimesh Poudel
 * @version v0.1
 */
public class PartTimeStaffHire extends StaffHire
{
   private int workHour;
   private int wagePerHour;
   private String staffName;
   private String joinDate;
   private String qualification;
   private String appointedBy;
   private String shifts;
   private boolean joined;
   private boolean terminated;
   INGNepal back = new INGNepal();
   //creating a constructor for PartTimeStaffHire
   public PartTimeStaffHire(int vacancyNo, String designation, String jobType,int workHour, int
wagePerHour, String shifts)
   {
      super(vacancyNo, designation, jobType);
      this.workHour=workHour;
```

```
        this.wagePerHour=wagePerHour;
        this.shifts=shifts;
        staffName="";
        joinDate="";
        qualification="";
        appointedBy="";
        joined= false;
        terminated=false;
    }
    //creating accessor methods for each attribute
    public int getworkHour(){
        return workHour;
    }
    public int getwagePerHour(){
        return wagePerHour;
    }
    public String getstaffName(){
        return staffName;
    }
    public String getjoinDate(){
        return joinDate;
    }
    public String getqualification(){
        return qualification;
    }
    public String getappointedBy(){
        return appointedBy;
    }
    public String getshifts(){
```

```
        return shifts;
    }
    public Boolean getjoined(){
        return joined;
    }
    public Boolean getterminated(){
        return terminated;
    }

    /*creating a method to hire a staff
     * if a staff has not been hired the input values will be inserted
     * otherwise a message will be displayed.
     */
    public void partTimehire( String staffName, String joinDate, String qualification, String appointedBy){
        if(joined==false){
            this.staffName=staffName;
            this.joinDate=joinDate;
            this.qualification=qualification;
            this.appointedBy=appointedBy;
            joined=true;
            terminated=false;
        }
        else{
            System.out.println( getstaffName() + " has already been hired on "+ getjoinDate());
        }
    }

    //creating a method to terminate a hired staff
    public void terminate(){
```

```
            ImageIcon img1= new ImageIcon("../image/suc.png");
        if( terminated==true){
            JOptionPane.showMessageDialog(back.frame,"The Staff's record has already been
terminated","Success",JOptionPane.ERROR_MESSAGE);


        }
        else{
            JOptionPane.showMessageDialog( back.frame,getstaffName() + " has been
terminated","Success",JOptionPane.PLAIN_MESSAGE,img1);
            staffName="";
            joinDate="";
            qualification="";
            appointedBy="";
            joined=false;
            terminated=true;


        }
    }
}
```

11.References

guru99, 2020. *guru99.* [Online]
Available at: https://www.guru99.com/uml-relationships-with-example.html

Java.com, 2020. *Java.com.* [Online]
Available at: https://java.com/en/download/faq/whatis_java.xml

lifewire.com, n.d. *2019.* [Online]
Available at: https://www.lifewire.com/what-is-java-4172382

Techopedia, 2020. *Techopedia.com.* [Online]
Available at: https://www.techopedia.com/definition/29530/bluej

Visual-paradigm, 2020. *Visual-paradigm.* [Online]
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/

wikipedia.org, 2019. *java programming.* [Online]
Available at: https://en.wikipedia.org/wiki/Java_(programming_language)

wikipedia.org, 2020. *blue j.* [Online]
Available at: https://en.wikipedia.org/wiki/BlueJ