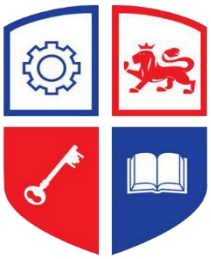
LONDON METROPOLITAN UNIVERSITY



ITAHARI INTERNATIONAL COLLEGE

Module Code & Module Title

(CS400NT) Programming

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2019 Spring

Student Name: Girija Tamang

London met ID:18030995

Assignment Due Date:15 July 2019

Submission Date:09 August 2019

I confirm that I understand my coursework needs to be submitted online via Local Server 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 marks of zero will be awarded.

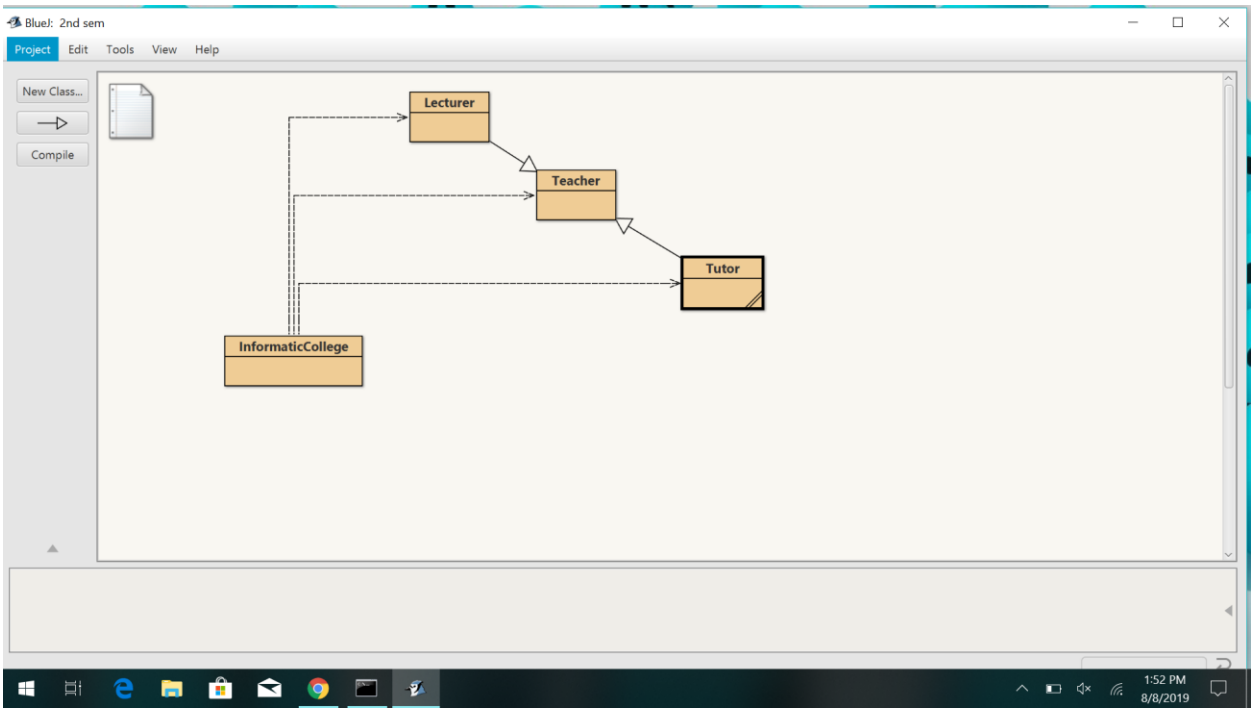
Table of Contents

Introduction

The coursework is based on java programming languages. It helps us to develop GUI teacher appointing system. This helps to store the information of teacher and its details. After developing this project, we will able to know about graphics and its importance. The project will able to find well experienced teacher using GUI.

This project will help us to be familiar with graphics coding and solving problems in java programming languages. It gives knowledge to develop other related projects in java that are based on graphics.

Class Diagram



Class Diagram of InformaticCollege

InformaticCollege
~ JFrame frame; ~ JTextField txtSubject txtInterviewerName txtClassPerDay txtSalary txtAppointedByn txtTerminationDate txtTeacherNumber txtTeacherName txtAppointedDate TerminationDatetxt txtQualification fldSubject fldInterviewerName fldClassPerDay fldSalary fldDailyWorkingHour fldTeacherNumber fldTeacherName JTextField fldStartingDate fldAdvanceSalary fldAcademicBlockNumber ~ JLabel lblTutor subjectLbl interviewerLbl appointLbl classLbl salaryLbl terminationDateLbl teacherNoLbl appointDateLbl qualificationLbl teacherLbl terminationLbl lecturerLbl lblSubject lblInterviewer lblDailyWorkingHour lblClass lblSalary lblTeacherNo lblStartingDate lblAcademicBlockNumber lblTeacherName lblAdvanceSalary ~JButton btnAddSubjectTutor btnHireTutor btnAddSubjectLecturer btnHireLecturer btnTermination btnDisplay btnClear
+actionPerformed(ActionEvent ea):void +main informaticCollege:void

Pseudocode

DEFINE InformaticCollege

DO

declare JFrame JTextField and JButton which are necessary for developing GUI

CREATE ArrayList name Data

CREATE object InformaticCollege()

CREATE frame and setname”Teacher Appointment System”

SET Size, DefaultCloseOperation, Resizable, Layout

CREATE JLabel for Tutor Appointment

SET Bounds and Add in frame

CREATE JLabel for subject for tutor

SET Bounds and Add in frame

CREATE JTextField for subject

SET Bounds and Add in frame

CREATE JLabel for interviewer name for tutor

SET Bounds and Add in frame

CREATE JTextField for interviewer name

SET Bounds and Add in frame

Code Sample: JLabel subjectLbl=new JLabel("Subject:");

```
subjectLbl.setBounds(15,65,300,40);
frame.add(subjectLbl);
txtSubject=new JTextField();
txtSubject.setBounds(200,65,350,35);
frame.add(txtSubject);
```

Likewise create other JLabel and JTextField for other necessary items and data and add in frame for creating GuI

CREATE JButton AddSubjectTutor

SET Bounds and Add in frame

ADD actionListener And actionperformed

DO

CHECK if text field are empty or not

IF text field are empty then give error message

Else check the data are correct or not

IF correct then add data in array list and display a successful message

END DO

CREATE JButton HireTutor

SET Bounds and Add in frame

ADD actionListener And actionperformed

DO

CHECK if text field are empty or not

IF text field are empty then give error message

Else check the teacher number are correct or not

IF correct then add data and display a successful hire message

END DO

For lecturer follow the above steps for adding subject and hiring

Sample of code: btnAddSubjectTutor = new JButton();

```
btnAddSubjectTutor.setText("Add Subject");
btnAddSubjectTutor.setBounds(950,180,200,40);
frame.add(btnAddSubjectTutor);
btnAddSubjectTutor.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent e){
   String subject=txtSubject.getText();
   String interviewerName= txtInterviewerName.getText();
   String classPerDay=txtClassPerDay.getText();
   String salary=txtSalary.getText();
   String terminationDate=txtTerminationDate.getText();
   String appointedBy=txtAppointedBy.getText();
```

```
if(subject.equals("")|| interviewerName.isEmpty()|| appointedBy.isEmpty()||
terminationDate.isEmpty()|| classPerDay.isEmpty() || salary.isEmpty()){
   JOptionPane.showMessageDialog(frame,"** Please input data in  fields completedly. **",
   "NonValid!!!",JOptionPane.WARNING_MESSAGE);
}
else{
    try{
       int ClassPerDay = Integer.parseInt(txtClassPerDay.getText());
       double Salary = Double.parseDouble(txtSalary.getText());
       Tutor tutor = new Tutor(subject,interviewerName,ClassPerDay,Salary,appointedBy,terminationDate);
       Data.add(tutor);
       JOptionPane.showMessageDialog(frame,"Tutor Successfully added in ArrayList.","Congratulation!!!",
       JOptionPane.INFORMATION_MESSAGE);
       txtSubject.setText(null);
       txtInterviewerName.setText(null);
       txtClassPerDay.setText(null);
       txtSalary.setText(null);
       txtAppointedBy.setText(null);
       txtTerminationDate.setText(null);
    }
    catch(NumberFormatException n){
        JOptionPane.showMessageDialog(frame,"** Please Enter Suitable Data Format ** .","Invalid!!",
        JOptionPane.WARNING_MESSAGE);
   }     } }     });
```

CREATE JButton Termination

SET Bounds and Add in frame

ADD actionListener And actionperformed

DO

CHECK if text field are empty or not

IF text field are empty then give error message

Else check the teacher number are correct or not

IF correct then check teacher number is same class or not

IF number is from same class and remove data and display a successful terminate message

END DO

CREATE JButton Display

SET Bounds and Add in frame

ADD actionListener And actionperformed

DO display all the data from array list

END DO

CREATE JButton Clear

SET Bounds and Add in frame

ADD actionListener And actionperformed

DO

Clear all the values from text field

END DO

SET the frame Visibility to True

END DO

Method Description

Main method ()

This method is used to call the main class constructor. The purpose of this method is to call constructor where constructor set the GUI components.

actionPerformed(ActionEvent e)

This method links the button with its respective method. During this method if and else if statement work to allow the functionality of adding ,hiring or terminating subject tutor or lecturer for respective class .it also help to proving platform for clearing data and displaying related data when button is pressed.

Testing

Test 1:

Objective	Compile and Run program using command prompt
Action	• First start Command Prompt • Then find the file location of java program • Then compile the program using Javac InformaticCollege.java • Use java InformaticCollege
Expected Result	Program InformaticCollege GUI gets executed
Actual Result	Gui gets excuted of information class
Conclusion	Test done successful

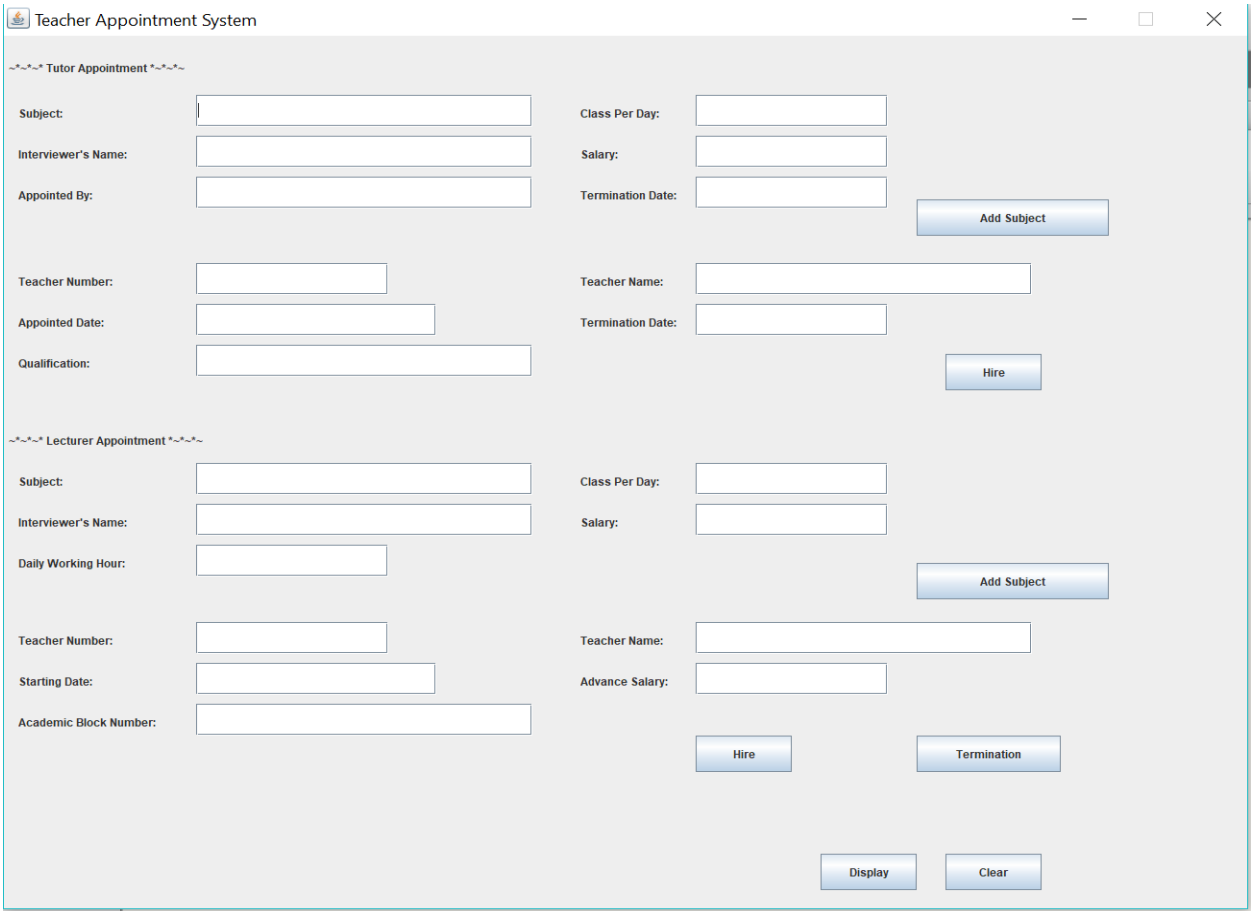


Figure 1:Program compile and run using command prompt.

Test 2:

Objective	Adding Subject for tutor list
Action	First add Subject, ClassPerDay and Interviewer's Name,Salary,AppointedBy and TerminationDate for tutor and press add subject button Data given were: • Subject=BIT • ClassPerDay=4 • Interviewer'sName=RAM • Salary=40000 • Appointed By=LUCKY • Termination Date=2020
Expected Result	The subject for tutor should add without any error
Actual Result	Adding subject for tutor done successfully without any error
Conclusion	Adding subject for tutor done successfully

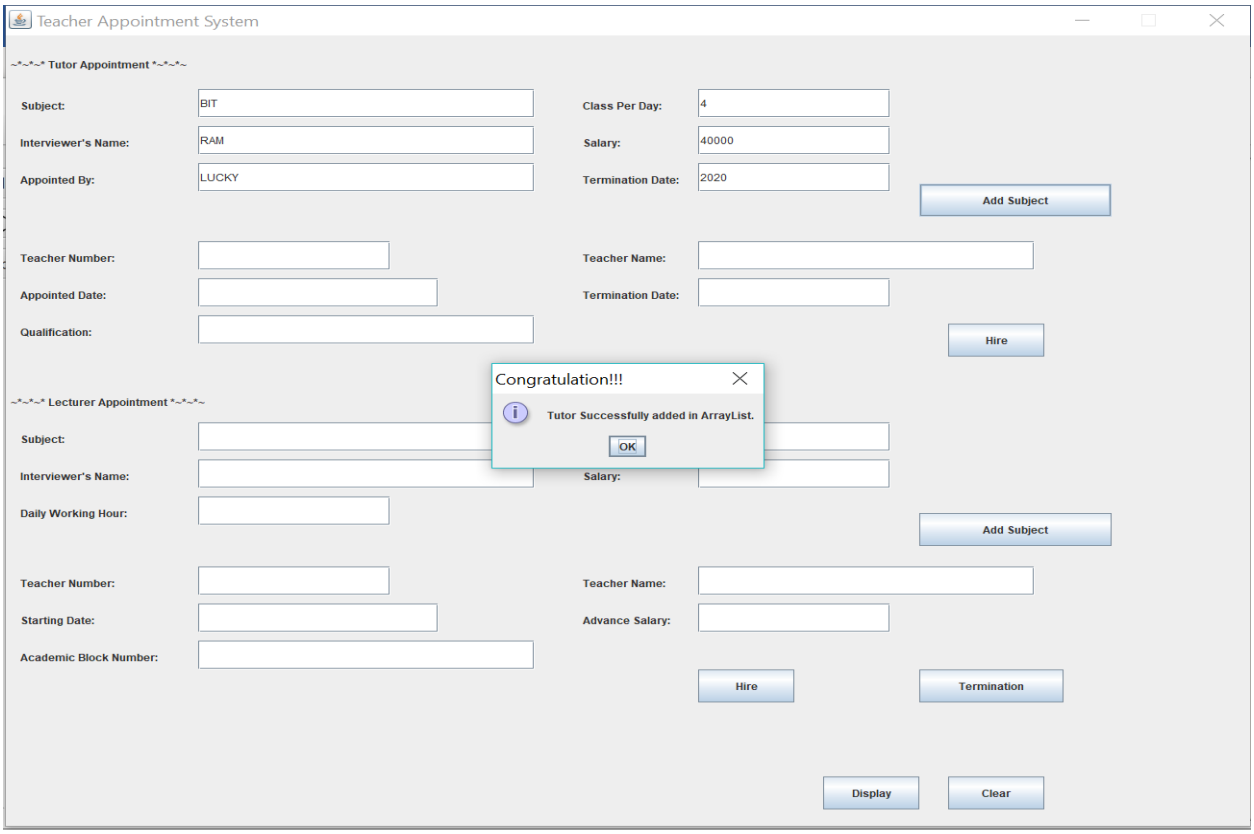


Figure 2:Adding subject for tutor list

Test 3:

Objective	Adding Subject for Lecturer
Action	First give data for Subject, ClassPerDay and Interviewer'sName, Salary, DailyWorkingHour and press add subject button Data given were: • Subject= Nepali • ClassPerDay=6 • Interviewer'sName=Hari • Salary=20000 • Daily Working Hour=6
Expected Result	The subject for Lecturer should add without any error
Actual Result	Without any error Adding subject for Lecturer done successfully
Conclusion	Adding subject for Lecturer done successfully

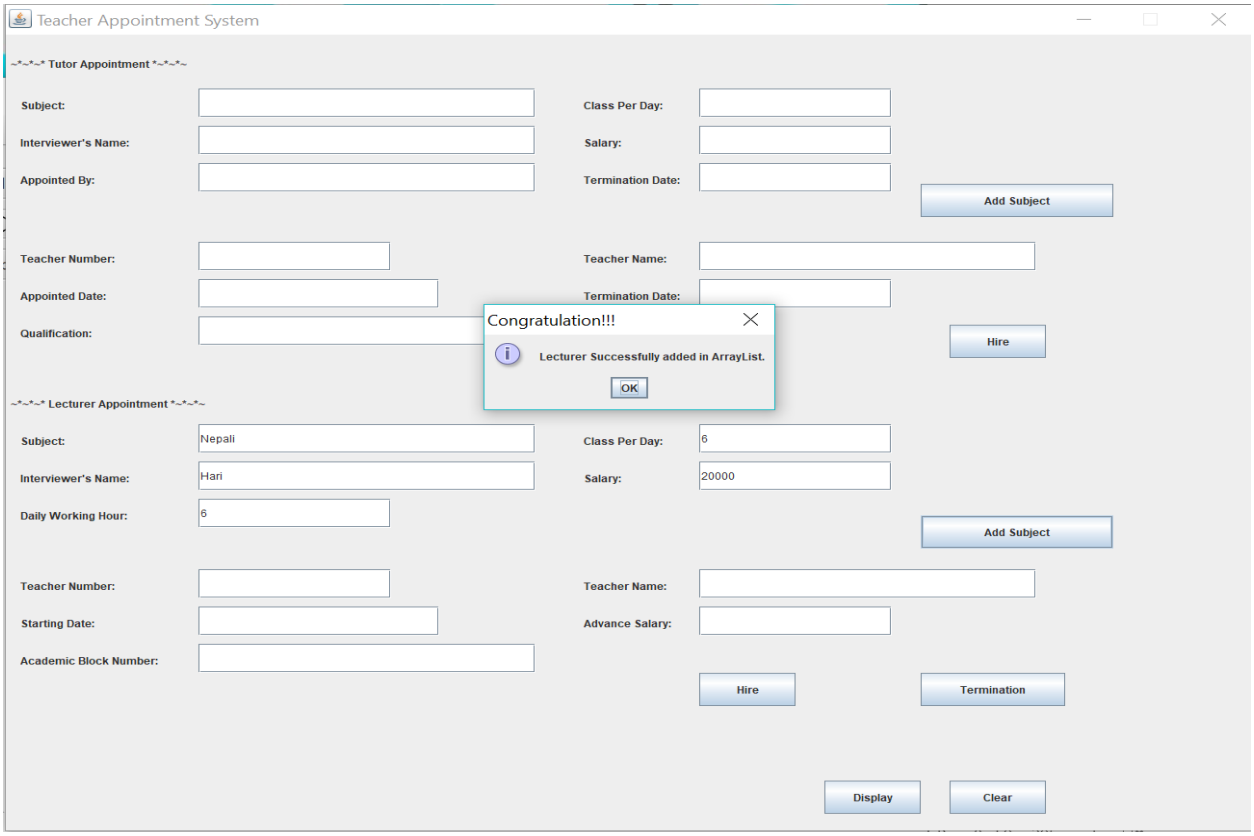


Figure 3:Adding subject for lecturer list

Test 4:

Objective	For Hiring Lecturer
Action	Enter data in TeacherNumber, TeacherName,StartingDate, Advance Salary and Academic Block Number and click hire button Data: TeacherNumber=1 TeacherName=Ram StartingDate=2018 AdvanceSalary=44444 AcademicBlockNumber=11
Expected Result	Hiring Should Be done
Actual Result	Hire lecturer is done and no error was found.
Conclusion	Hire lecturer is done successfully

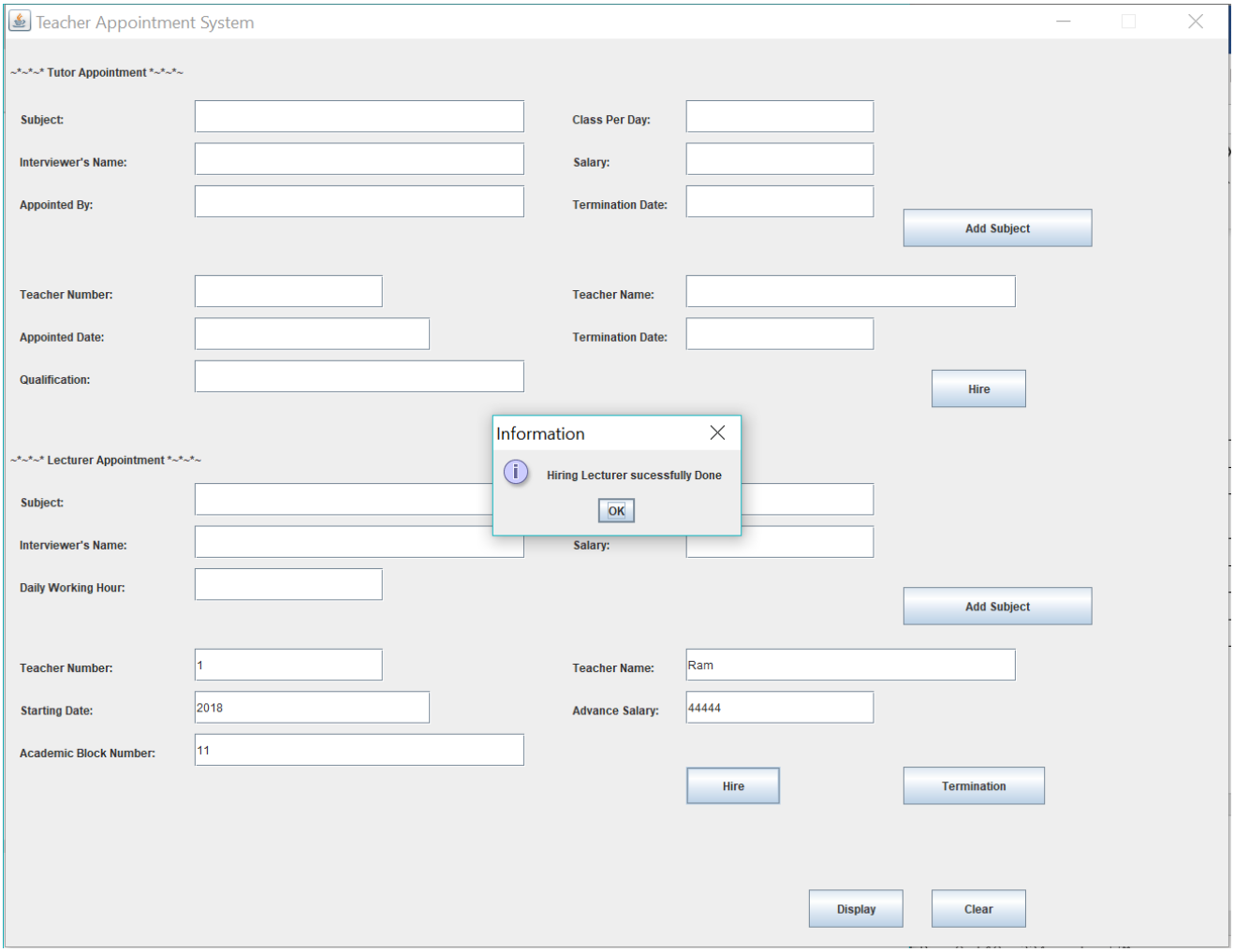


Figure 4:Hiring Lecturer

Test 5:

Objective	For Hiring Tutor
Action	Enter data in TeacherNumber,TeacherName, AppointDate,TerminationDate,Qualification and press hire button Data: TeacherNumber=1 TeacherName=Thapa AppointedDate=2019 TerminationDate=2020 Qualification=Master in Arts
Expected Result	Hiring Should Be done
Actual Result	Hire of tutor done without any error
Conclusion	Tutor hire successfully

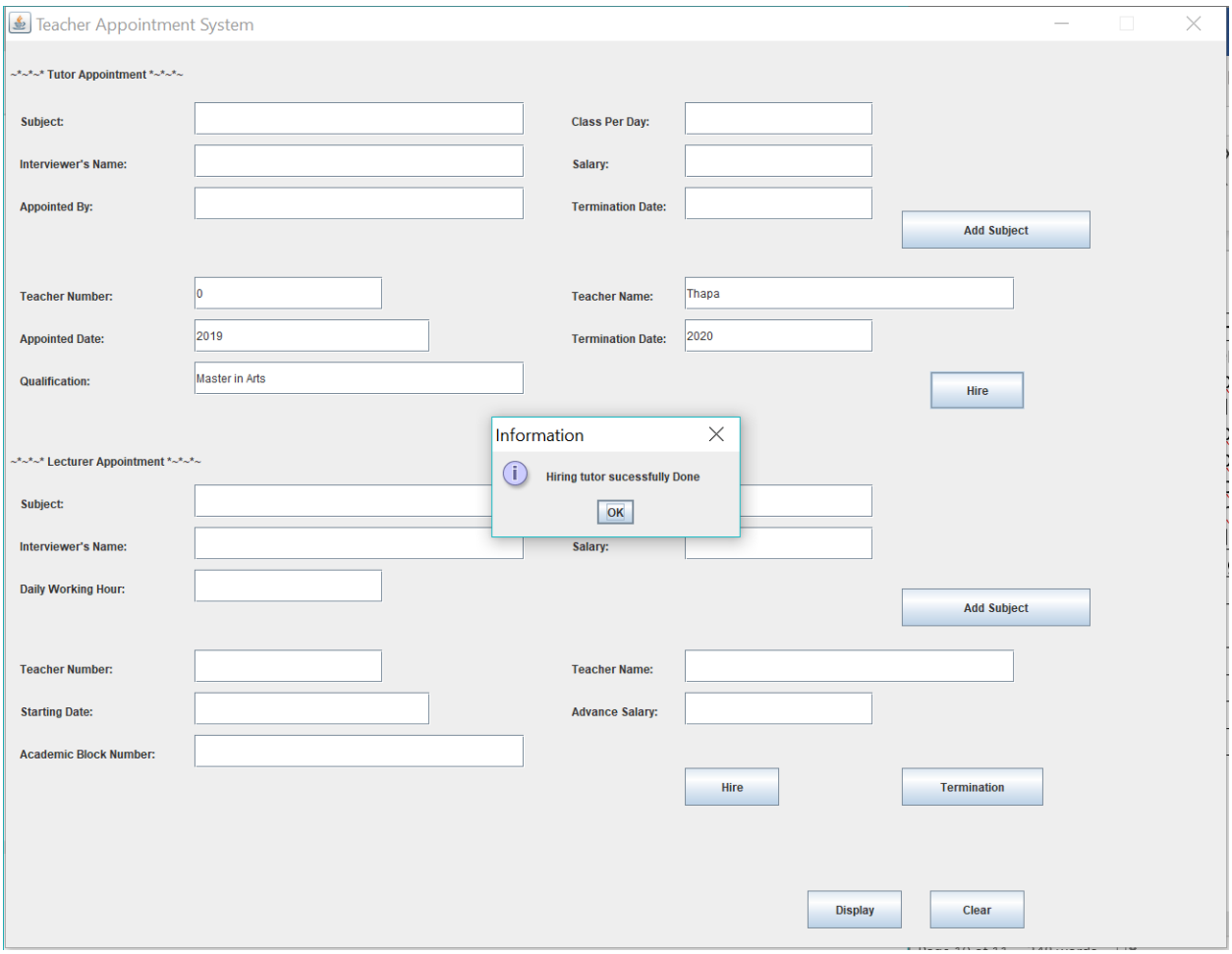


Figure 5:Hiring Tutor

Test 6:

Objective	Termination
Action	Enter Data and press terminate button Data entered: TeacherNumber=1 Teacher Name=ToriLal StartingDate=2019 Advance Salary=5000 AcademicBlockNumber=22AC
Expected Result	Terminate will be done
Actual Result	Termination of lecturer done without any error
Conclusion	Terminated done sucessfully

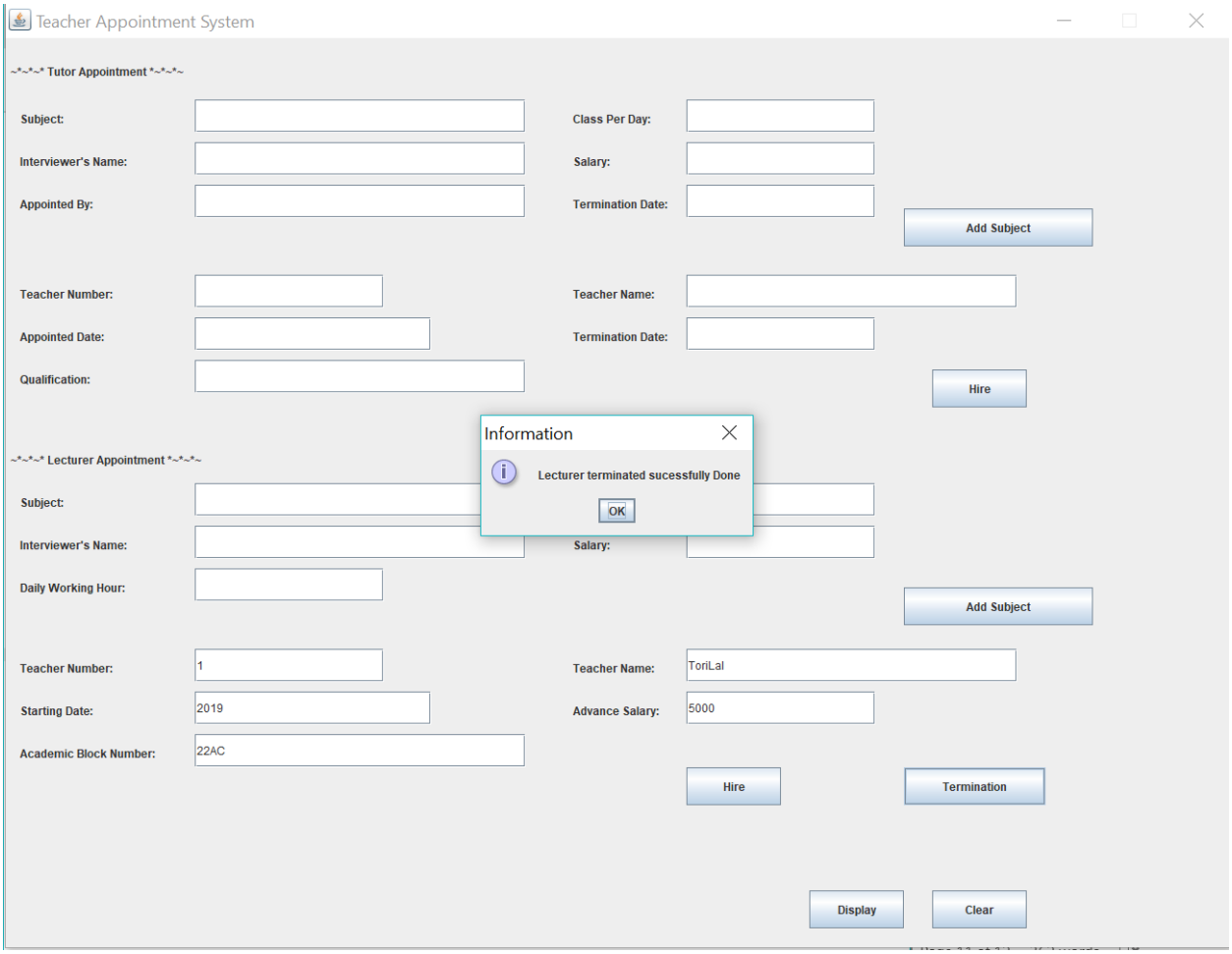


Figure 6:Terminating lecturer from array list

Test 7:

Objective	Appropriate dialog boxes appear or not when unsuitable values entered
Action	I have entered wrong teacher number
Expected Result	Wrong dialog box appears
Actual Result	Error dialog box appears
Conclusion	Testing successful

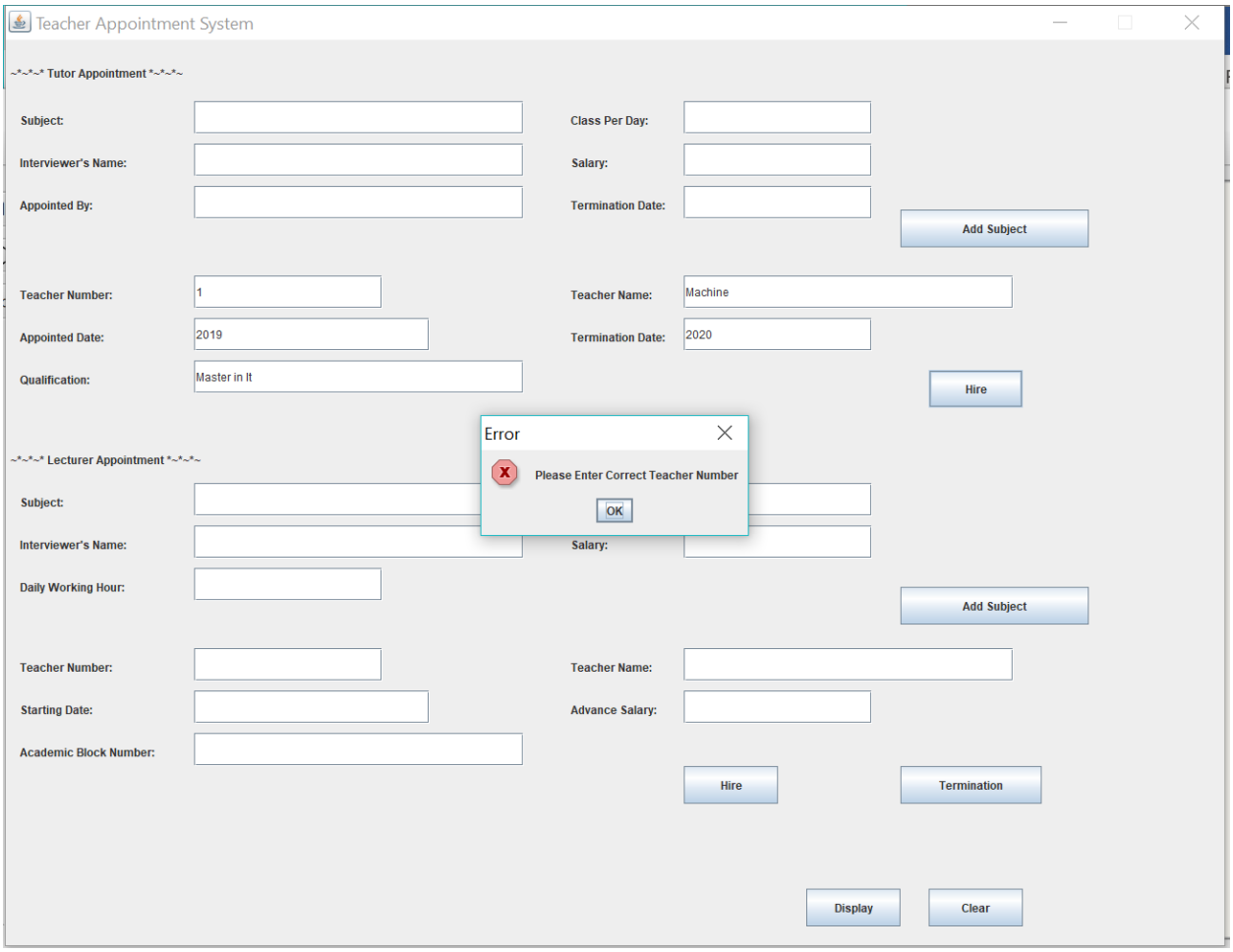


Figure 7:Showing error dialog box while entering wrong teacher number

Test 8:

Objective	Appropriate dialog boxes appear or not when unsuitable values entered
Action	Tutor class teacher number entered
Expected Result	Propriate Error dialog box appears
Actual Result	Error dialog box appears with propriate message
Conclusion	Testing done successful

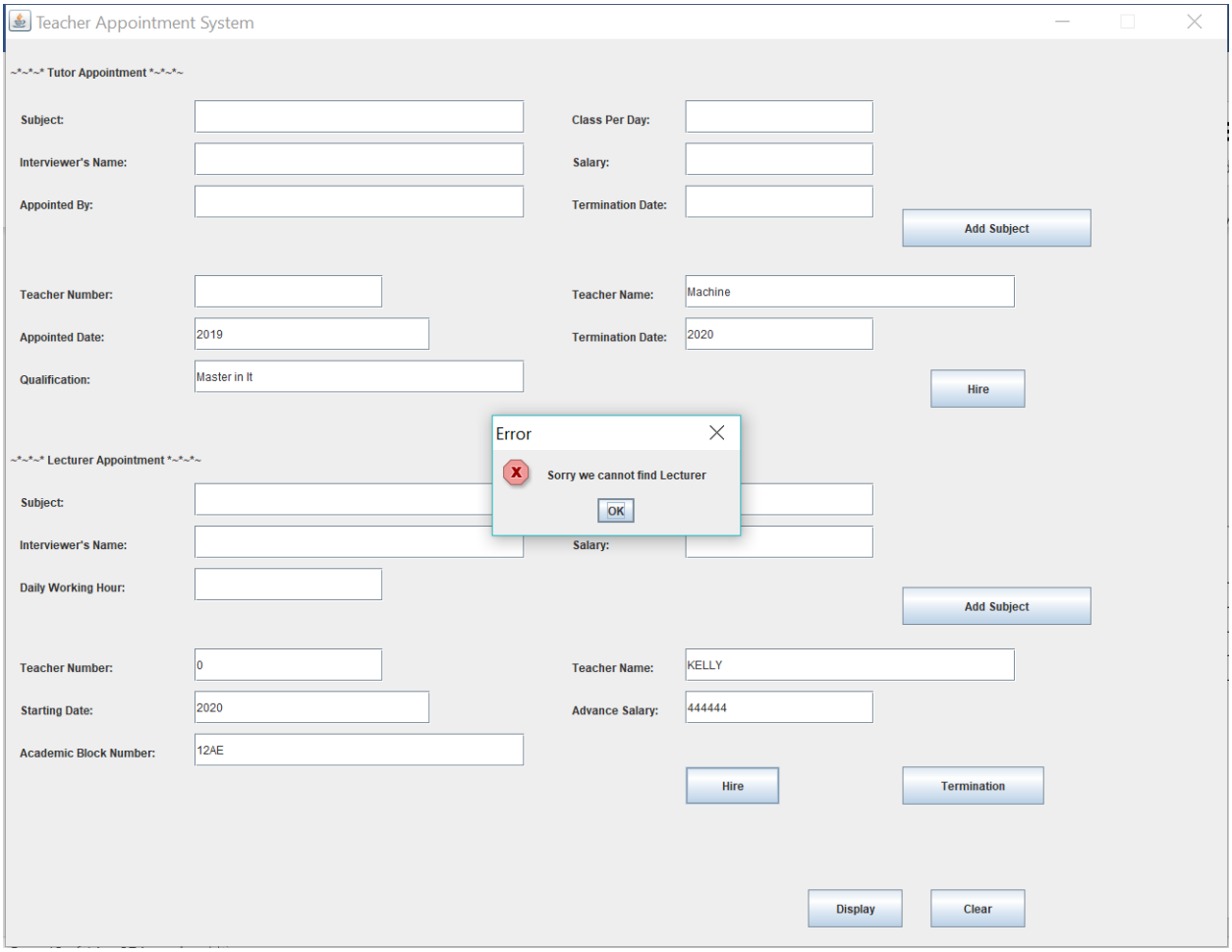


Figure 8:Showing error dialog box when wrong class teacher number used

Error Detection and Correction

Syntax Error

A syntax error is an error in which the language you use to create your code is incorrect.

Syntax errors are a type of compiler error.

Error:

```
public InformaticCollege(){
    frame= new JFrame("Teacher Appointment System");
    frame.setSize(1300,1000);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setResizable(false);
    frame.setLayout(null);

    JLabel lblTutor=new JLabel("~*~*~* Tutor Appointment *~*~*~");
    lblTutor.setBounds(5,15,500,40)
    frame.add(lblTutor);
```

Correction:

```
JLabel lblTutor=new JLabel("~*~*~* Tutor Appointment *~*~*~");
lblTutor.setBounds(5,15,500,40);
frame.add(lblTutor);
```

Run Time Error

A runtime error is a program error that occurs while running the program. Unlike other types of program errors, such as syntax errors and compile time errors, the term is often used.

Error

```
btnTermination.setBounds(950,770,150,40);
frame.add(btnTermination);
btnTermination.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent ae){
    teacherno=fldTeacherNumber.getText();
    String teachername= fldTeacherName.getText();
    String startingdate=fldStartingDate.getText();
    String advancesalary=fldAdvanceSalary.getText();
    String academicblockno=fldAcademicBlockNumber.getText();
    if(teacherno.isEmpty()||teachername.isEmpty()|| startingdate.isEmpty()|| advancesalary.isEmpty()
    || academicblockno.isEmpty()){
        JOptionPane.showMessageDialog(frame,"** Please input data in  fields  completedly. **","NonValid!!!",
        JOptionPane.WARNING_MESSAGE);
    }
```

Correction

```
btnTermination.setBounds(950,770,150,40);
frame.add(btnTermination);
btnTermination.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent ae){
    String teacherno=fldTeacherNumber.getText();
    String teachername= fldTeacherName.getText();
    String startingdate=fldStartingDate.getText();
    String advancesalary=fldAdvanceSalary.getText();
    String academicblockno=fldAcademicBlockNumber.getText();
    if(teacherno.isEmpty()||teachername.isEmpty()|| startingdate.isEmpty()|| advancesalary.isEmpty()
    || academicblockno.isEmpty()){
        JOptionPane.showMessageDialog(frame,"** Please input data in  fields  completedly. **","NonValid!!!",
```

Semantic error

The most prevalent semantic error is that the code utilizes a variable that is not correctly initialized. Luckily, in most instances, the compiler discovers this specific semantic error.

Error

```
btnHireLecturer = new JButton("Hire");
btnHireLecturer.setBounds(720,770,100,40);
frame.add(btnHireLecturer);
btnHireLecturer.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent ae){
    String teacherNumber==fldTeacherNumber.getText();
    String teachername= fldTeacherName.getText();
    String startingdate=fldStartingDate.getText();
    String advancesalary=fldAdvanceSalary.getText();
    String academicblockno=fldAcademicBlockNumber.getText();
    if(teacherNumber.isEmpty()||teachername.isEmpty()|| startingdate.isEmpty()|| advancesalary.isEmpty()||
    academicblockno.isEmpty()){
```

Correction

```
        btnHireLecturer = new JButton("Hire");
        btnHireLecturer.setBounds(720,770,100,40);
        frame.add(btnHireLecturer);
        btnHireLecturer.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent ae){
            String teacherNumber=fldTeacherNumber.getText();
            String teachername= fldTeacherName.getText();
            String startingdate=fldStartingDate.getText();
            String advancesalary=fldAdvanceSalary.getText();
            String academicblockno=fldAcademicBlockNumber.getText();
```

Conclusion

This coursework is based on java programming languages for developing teacher appointment system where we must create GUI. This coursework is developed for making GUI system for the first part of coursework that stores details of teacher.

GUI is fun in learning but hard to develop. While developing GUI, Creating the frame adding label, text field and button was quite easy but while adding action listener, action performed and setbounds are difficult and confusing for me. while developing this project I had faced many problems like syntax error, runtime errors, logical error, and many problem while implementing action listener action performed and linking with first coursework .By the help of the cooperative module teacher, friends, elder I have able to tackle the problem and develop working GUI project.

This course work helped me a lot and made familiar on graphics. Now I can create my own on different system using GUI.

Appendix

```
/**
 * InformaticCollege develop a gui system that stores details of teachers .
 *
 * @author (Girija Tamang)
 * @Londonmet ID (18030995)
 */
//importing java utilities
import javax.swing.JFrame;
import javax.swing.JTextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.Iterator;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class InformaticCollege
{
    //declaration of frame ,textfield and button
    JFrame frame;

    JTextField txtSubject;
    JTextField txtInterviewerName;
    JTextField txtClassPerDay;
    JTextField txtSalary;
```

```
JTextField txtAppointedBy;
JTextField txtTerminationDate;
JTextField txtTeacherNumber;
JTextField txtTeacherName;
JTextField txtAppointedDate;
JTextField TerminationDatetxt;
JTextField txtQualification;

JTextField fldSubject;
JTextField fldInterviewerName;
JTextField fldClassPerDay;
JTextField fldSalary;
JTextField fldDailyWorkingHour;
JTextField fldTeacherNumber;
JTextField fldTeacherName;
JTextField fldStartingDate;
JTextField fldAdvanceSalary;
JTextField fldAcademicBlockNumber;

JButton btnAddSubjectTutor;
JButton btnHireTutor;
JButton btnAddSubjectLecturer;
JButton btnHireLecturer;
JButton btnTermination;
JButton btnDisplay;
JButton btnClear;

ArrayList<Teacher> Data= new ArrayList();//Array list  named Dataa created.
```

```
    public InformaticCollege(){
        frame= new JFrame("Teacher Appointment System");
        frame.setSize(1300,1000);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setResizable(false);
        frame.setLayout(null);

        JLabel lblTutor=new JLabel("~*~*~* Tutor Appointment *~*~*~");//it only show the tutor appoint label in gui
        lblTutor.setBounds(5,15,500,40);
        frame.add(lblTutor);

        JLabel subjectLbl=new JLabel("Subject:");//adding label name subject
        subjectLbl.setBounds(15,65,300,40);
        frame.add(subjectLbl);
        txtSubject=new JTextField();//addiing text field for subject
        txtSubject.setBounds(200,65,350,35);
        frame.add(txtSubject);


        JLabel interviewerLbl=new JLabel("Interviewer's Name:");
        interviewerLbl.setBounds(15,110,300,40);
        frame.add(interviewerLbl);
        txtInterviewerName=new JTextField();
        txtInterviewerName.setBounds(200,110,350,35);
        frame.add(txtInterviewerName);
```

```
JLabel appointLbl=new JLabel("Appointed By:");
appointLbl.setBounds(15,155,300,40);
frame.add(appointLbl);
txtAppointedBy=new JTextField();
txtAppointedBy.setBounds(200,155,350,35);
frame.add(txtAppointedBy);


JLabel classLbl=new JLabel("Class Per Day:");
classLbl.setBounds(600,65,300,40);
frame.add(classLbl);
txtClassPerDay=new JTextField();
txtClassPerDay.setBounds(720,65,200,35);
frame.add(txtClassPerDay);


JLabel salaryLbl=new JLabel("Salary:");
salaryLbl.setBounds(600,110,300,40);
frame.add(salaryLbl);
txtSalary=new JTextField();
txtSalary.setBounds(720,110,200,35);
frame.add(txtSalary);


JLabel terminationDateLbl=new JLabel("Termination Date:");
terminationDateLbl.setBounds(600,155,300,40);
frame.add(terminationDateLbl);
txtTerminationDate=new JTextField();
```

```
        txtTerminationDate.setBounds(720,155,200,35);
        frame.add(txtTerminationDate);

       //Adding button name Add Subject in Frame
        btnAddSubjectTutor = new JButton();
        btnAddSubjectTutor.setText("Add Subject");
        btnAddSubjectTutor.setBounds(950,180,200,40);
        frame.add(btnAddSubjectTutor);
        //adding action listener in button Tutor
        btnAddSubjectTutor.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
            String subject=txtSubject.getText();
            String interviewerName= txtInterviewerName.getText();
            String classPerDay=txtClassPerDay.getText();
            String salary=txtSalary.getText();
            String terminationDate=txtTerminationDate.getText();
            String appointedBy=txtAppointedBy.getText();
            if(subject.equals("")|| interviewerName.isEmpty()|| appointedBy.isEmpty()||
            terminationDate.isEmpty()|| classPerDay.isEmpty() || salary.isEmpty()){
                JOptionPane.showMessageDialog(frame,"** Please input data in  fields
completedly. **",
                "NonValid!!!",JOptionPane.WARNING_MESSAGE);
            }
            else{
                try{
                    int ClassPerDay = Integer.parseInt(txtClassPerDay.getText());
                    double Salary = Double.parseDouble(txtSalary.getText());
                    Tutor tutor = new
Tutor(subject,interviewerName,ClassPerDay,Salary,appointedBy,terminationDate);
```

```
                Data.add(tutor);
                JOptionPane.showMessageDialog(frame,"Tutor Successfully added in ArrayList.","Congratulation!!!",
                JOptionPane.INFORMATION_MESSAGE);
                txtSubject.setText(null);
                txtInterviewerName.setText(null);
                txtClassPerDay.setText(null);
                txtSalary.setText(null);
                txtAppointedBy.setText(null);
                txtTerminationDate.setText(null);
            }
            catch(NumberFormatException n){
                JOptionPane.showMessageDialog(frame,"** Please Enter Suitable Data Format ** .","Invalid!!",
                JOptionPane.WARNING_MESSAGE);
              }
        }


        }
    });

    JLabel teacherNoLbl=new JLabel("Teacher Number:");
    teacherNoLbl.setBounds(15,250,300,40);
    frame.add(teacherNoLbl);
    txtTeacherNumber=new JTextField();
    txtTeacherNumber.setBounds(200,250,200,35);
    frame.add(txtTeacherNumber);
```

```
JLabel appointDateLbl=new JLabel("Appointed Date:");
appointDateLbl.setBounds(15,295,300,40);
frame.add(appointDateLbl);
txtAppointedDate=new JTextField();
txtAppointedDate.setBounds(200,295,250,35);
frame.add(txtAppointedDate);


JLabel qualificationLbl=new JLabel("Qualification:");
qualificationLbl.setBounds(15,340,300,40);
frame.add(qualificationLbl);
txtQualification=new JTextField();
txtQualification.setBounds(200,340,350,35);
frame.add(txtQualification);


JLabel teacherLbl=new JLabel("Teacher Name:");
teacherLbl.setBounds(600,250,300,40);
frame.add(teacherLbl);
txtTeacherName=new JTextField();
txtTeacherName.setBounds(720,250,350,35);
frame.add(txtTeacherName);


JLabel terminationLbl=new JLabel("Termination Date:");
terminationLbl.setBounds(600,295,300,40);
frame.add(terminationLbl);
TerminationDatetxt=new JTextField();
```

```
        TerminationDatetxt.setBounds(720,295,200,35);
        frame.add(TerminationDatetxt);


      //Adding Hiring Button  for tutor
        btnHireTutor= new JButton("Hire");
        btnHireTutor.setBounds(980,350,100,40);
        frame.add(btnHireTutor);
        btnHireTutor.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent ae){
            String teachernumber=txtTeacherNumber.getText();
            String teachername= txtTeacherName.getText();
            String appointeddate=txtAppointedDate.getText();
            String terminationdate=TerminationDatetxt.getText();
            String qualification=txtQualification.getText();
            if(teachernumber.isEmpty()||teachername.isEmpty()||   appointeddate.isEmpty()||
terminationdate.isEmpty()||
            qualification.isEmpty()){
                JOptionPane.showMessageDialog(frame,"** Please input data in  fields
completedly. **","NonValid!!!",
                JOptionPane.WARNING_MESSAGE);
            }
           else{
                try{
                    int TeacherNumber=Integer.parseInt(teachernumber);
                    if(TeacherNumber>=Data.size() || TeacherNumber<0){
                        JOptionPane.showMessageDialog(frame ,"Please Enter Correct Teacher
Number","Error",
                        JOptionPane.ERROR_MESSAGE);
```

```
                }
                else{
                    Tutor tutor = (Tutor)Data.get(TeacherNumber);

tutor.appointTutor(teachername,appointeddate,terminationdate,qualification);
                    JOptionPane.showMessageDialog(frame,"Hiring tutor sucessfully Done
","Information",
                    JOptionPane.INFORMATION_MESSAGE);
                    txtTeacherNumber.setText(null);
                    txtTeacherName.setText(null);
                    txtAppointedDate.setText(null);
                    TerminationDatetxt.setText(null);
                    txtQualification.setText(null);
                }
            }

            catch(NumberFormatException Error){
                JOptionPane.showMessageDialog(frame,"Number Format Exception
","Error",JOptionPane.ERROR_MESSAGE);
            }
            catch(ClassCastException Error){
                JOptionPane.showMessageDialog(frame,"Tutor Not
Found","Error",JOptionPane.ERROR_MESSAGE);
            }
         }
         }
        });

        JLabel  lecturerLbl=new JLabel("~*~*~* Lecturer Appointment *~*~*~");
        lecturerLbl.setBounds(5,425,500,40);
```

```
frame.add(lecturerLbl);


JLabel lblSubject=new JLabel("Subject:");
lblSubject.setBounds(15,470,300,40);
frame.add(lblSubject);
fldSubject=new JTextField();
fldSubject.setBounds(200,470,350,35);
frame.add(fldSubject);


JLabel lblInterviewer=new JLabel("Interviewer's Name:");
lblInterviewer.setBounds(15,515,300,40);
frame.add(lblInterviewer);
fldInterviewerName=new JTextField();
fldInterviewerName.setBounds(200,515,350,35);
frame.add(fldInterviewerName);


JLabel lblDailyWorkingHour=new JLabel("Daily Working Hour:");
lblDailyWorkingHour.setBounds(15,560,300,40);
frame.add(lblDailyWorkingHour);
fldDailyWorkingHour=new JTextField();
fldDailyWorkingHour.setBounds(200,560,200,35);
frame.add(fldDailyWorkingHour);


JLabel lblClass=new JLabel("Class Per Day:");
```

```
        lblClass.setBounds(600,470,300,40);
        frame.add(lblClass);
        fldClassPerDay=new JTextField();
        fldClassPerDay.setBounds(720,470,200,35);
        frame.add(fldClassPerDay);



        JLabel lblSalary=new JLabel("Salary:");
        lblSalary.setBounds(600,515,300,40);
        frame.add(lblSalary);
        fldSalary=new JTextField();
        fldSalary.setBounds(720,515,200,35);
        frame.add(fldSalary);

        //Adding Subject Button for Lecturer
       btnAddSubjectLecturer = new JButton("Add Subject");
       btnAddSubjectLecturer.setBounds(950,580,200,40);
       frame.add(btnAddSubjectLecturer);
       btnAddSubjectLecturer.addActionListener(new ActionListener(){
          public void actionPerformed(ActionEvent e){
          String subject=fldSubject.getText();
          String interviewerName= fldInterviewerName.getText();
          String classPerDay=fldClassPerDay.getText();
          String salary=fldSalary.getText();
          String dailyworkinghour=fldDailyWorkingHour.getText();

          if(subject.isEmpty()|| interviewerName.isEmpty()|| classPerDay.isEmpty()||
salary.isEmpty()||
          dailyworkinghour.isEmpty()){
```

```
            JOptionPane.showMessageDialog(frame,"** Please fiil up data in txt fields completedly. **",
              "NonValid!!!",JOptionPane.WARNING_MESSAGE);
          }
          else{
              try{
                 int ClassPerDay = Integer.parseInt(classPerDay);
                 double Salary = Double.parseDouble(salary);
                 int DailyWorkingHour=Integer.parseInt(fldDailyWorkingHour.getText());
                 Lecturer lecturer = new Lecturer(subject,ClassPerDay,interviewerName,Salary,DailyWorkingHour);
                 Data.add(lecturer);
                 JOptionPane.showMessageDialog(frame,"Lecturer Successfully added in ArrayList.","Congratulation!!!"
                 ,JOptionPane.INFORMATION_MESSAGE);
                 fldSubject.setText(null);
                 fldInterviewerName.setText(null);
                 fldClassPerDay.setText(null);
                 fldSalary.setText(null);
                 fldDailyWorkingHour.setText(null);

                }
                catch(NumberFormatException n){
                   JOptionPane.showMessageDialog(frame,"** Please Enter Suitable Data Format ** .","Invalid!!",
                   JOptionPane.WARNING_MESSAGE);
                }
          }
          }
       });
```

```
JLabel lblTeacherNo=new JLabel("Teacher Number:");
lblTeacherNo.setBounds(15,645,300,40);
frame.add(lblTeacherNo);
fldTeacherNumber=new JTextField();
fldTeacherNumber.setBounds(200,645,200,35);
frame.add(fldTeacherNumber);


JLabel lblStartingDate=new JLabel("Starting Date:");
lblStartingDate.setBounds(15,690,300,40);
frame.add(lblStartingDate);
fldStartingDate=new JTextField();
fldStartingDate.setBounds(200,690,250,35);
frame.add(fldStartingDate);


JLabel lblAcademicBlockNumber=new JLabel("Academic Block Number:");
lblAcademicBlockNumber.setBounds(15,735,300,40);
frame.add(lblAcademicBlockNumber);
fldAcademicBlockNumber=new JTextField();
fldAcademicBlockNumber.setBounds(200,735,350,35);
frame.add(fldAcademicBlockNumber);


JLabel lblTeacherName=new JLabel("Teacher Name:");
```

```
        lblTeacherName.setBounds(600,645,300,40);
        frame.add(lblTeacherName);
        fldTeacherName=new JTextField();
        fldTeacherName.setBounds(720,645,350,35);
        frame.add(fldTeacherName);


        JLabel lblAdvanceSalary=new JLabel("Advance Salary:");
        lblAdvanceSalary.setBounds(600,690,300,40);
        frame.add(lblAdvanceSalary);
        fldAdvanceSalary=new JTextField();
        fldAdvanceSalary.setBounds(720,690,200,35);
        frame.add(fldAdvanceSalary);

        //Adding Hire Button for lecturer
        btnHireLecturer = new JButton("Hire");
        btnHireLecturer.setBounds(720,770,100,40);
        frame.add(btnHireLecturer);
        btnHireLecturer.addActionListener(new ActionListener(){
          public void actionPerformed(ActionEvent ae){
          String teacherNumber=fldTeacherNumber.getText();
          String teachername= fldTeacherName.getText();
          String startingdate=fldStartingDate.getText();
          String advancesalary=fldAdvanceSalary.getText();
          String academicblockno=fldAcademicBlockNumber.getText();
          if(teacherNumber.isEmpty()||teachername.isEmpty()||      startingdate.isEmpty()||
advancesalary.isEmpty()||
          academicblockno.isEmpty()){
```

```
            JOptionPane.showMessageDialog(frame,"** Please input data in  fields completedly. **","NonValid!!!",
            JOptionPane.WARNING_MESSAGE);
          }
        else{
            try{
               int Teachernumber=Integer.parseInt(teacherNumber);
               double AdvanceSalary=Double.parseDouble(advancesalary);
               if(Teachernumber>=Data.size() || Teachernumber<0){
                  JOptionPane.showMessageDialog(frame ,"Please Enter Correct Teacher Number","Error",
                  JOptionPane.ERROR_MESSAGE);

               }
               else{
                   Lecturer lecturer = (Lecturer)Data.get(Teachernumber);

lecturer.appointLecturer(teachername,startingdate,AdvanceSalary,academicblockno);
                   JOptionPane.showMessageDialog(frame,"Hiring Lecturer sucessfully Done ","Information",
                   JOptionPane.INFORMATION_MESSAGE);
                   fldTeacherNumber.setText(null);
                   fldTeacherName.setText(null);
                   fldStartingDate.setText(null);
                   fldAdvanceSalary.setText(null);
                   fldAcademicBlockNumber.setText(null);
               }
            }
            catch(NumberFormatException Error){
```

```
                JOptionPane.showMessageDialog(frame,"Number Format Exception ","Error",JOptionPane.ERROR_MESSAGE);
            }
            catch(ClassCastException Error){
                JOptionPane.showMessageDialog(frame,"Sorry we cannot find Lecturer","Error",JOptionPane.ERROR_MESSAGE);
            }
        }
        }
    });



    btnTermination = new JButton("Termination");
    btnTermination.setBounds(950,770,150,40);
    frame.add(btnTermination);
    //this button terminates the lecture and remove the lecturer;
    btnTermination.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent ae){
        String teacherno=fldTeacherNumber.getText();
        String teachername= fldTeacherName.getText();
        String startingdate=fldStartingDate.getText();
        String advancesalary=fldAdvanceSalary.getText();
        String academicblockno=fldAcademicBlockNumber.getText();
        if(teacherno.isEmpty()||teachername.isEmpty()|| startingdate.isEmpty()|| advancesalary.isEmpty()
        || academicblockno.isEmpty()){
            JOptionPane.showMessageDialog(frame,"** Please input data in  fields completedly. **","NonValid!!!",
            JOptionPane.WARNING_MESSAGE);
        }
```

```
        else{
            try{
                int teacherNumber=Integer.parseInt(teacherno);


                if(teacherNumber>=Data.size() || teacherNumber<0){
                    JOptionPane.showMessageDialog(frame ,"Please Enter Correct Teacher
Number","Error",
                    JOptionPane.ERROR_MESSAGE);


                }
                else{
                    Lecturer terlecturer = (Lecturer)Data.get(teacherNumber);
                    terlecturer.lecturerTermination();
                    Data.remove(teacherNumber);
                    JOptionPane.showMessageDialog(frame," Lecturer terminated
sucessfully Done ","Information",
                    JOptionPane.INFORMATION_MESSAGE);
                    fldTeacherNumber.setText(null);
                    fldTeacherName.setText(null);
                    fldStartingDate.setText(null);
                    fldAdvanceSalary.setText(null);
                    fldAcademicBlockNumber.setText(null);
                }
            }
            catch(NumberFormatException Error){
                JOptionPane.showMessageDialog(frame,"Number Format Exception
","Error",JOptionPane.ERROR_MESSAGE);
            }
            catch(ClassCastException Error){
```

```
                JOptionPane.showMessageDialog(frame,"Lecturer Not
Found","Error",JOptionPane.ERROR_MESSAGE);
            }
        }
        }
    });



    btnDisplay = new JButton("Display");
    btnDisplay.setBounds(850,900,100,40);
    frame.add(btnDisplay);
    btnDisplay.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e){

            for (Teacher data:Data){
                data.display();
            }
        }
    });

    btnClear = new JButton("Clear");
    btnClear.setBounds(980,900,100,40);
    frame.add(btnClear);



    btnClear.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent ae){
                txtSubject.setText(null);
```

```
            txtInterviewerName.setText(null);
            txtClassPerDay.setText(null);
            txtSalary.setText(null);
            txtAppointedBy.setText(null);

            txtTerminationDate.setText(null);
            txtTeacherName.setText(null);
            txtTeacherNumber.setText(null);

            txtAppointedDate.setText(null);
            TerminationDatetxt.setText(null);

            txtQualification.setText(null);
            fldSubject.setText(null);
            fldInterviewerName.setText(null);
            fldClassPerDay.setText(null);
            fldSalary.setText(null);



            fldDailyWorkingHour.setText(null);
            fldTeacherName.setText(null);
            fldTeacherNumber.setText(null);

            fldStartingDate.setText(null);
            fldAdvanceSalary.setText(null);
            fldAcademicBlockNumber.setText(null);
        }
    });
```

```
        frame.setVisible(true);

    }
    public static void main(String[]args)
        {
            new InformaticCollege();
        }
}
```