LONDON METROPOLITAN UNIVERSITY

ITAHARI INTERNATIONAL COLLEGE

Module Code & Module Title

30% INDIVIDUAL

Assessment Weightage & Type

Year and Semester

2019 Spring

Student Name: Girija Tamang

London Met ID: 18030995

College ID:NP05CP4S190007

Assignment Due Date: week 8

Assignment Submission Date: week 12

Word Count (Where Required):

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.

Contents

1.Introduction

The project is based on java programming languages. It helps us to develop teacher appointing system. It helps to find the information of teacher and its details. After develop this project we will able to appoint teacher and gain various knowledge on java programming. The project will able to find well experienced teacher.

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

2.Class Diagram

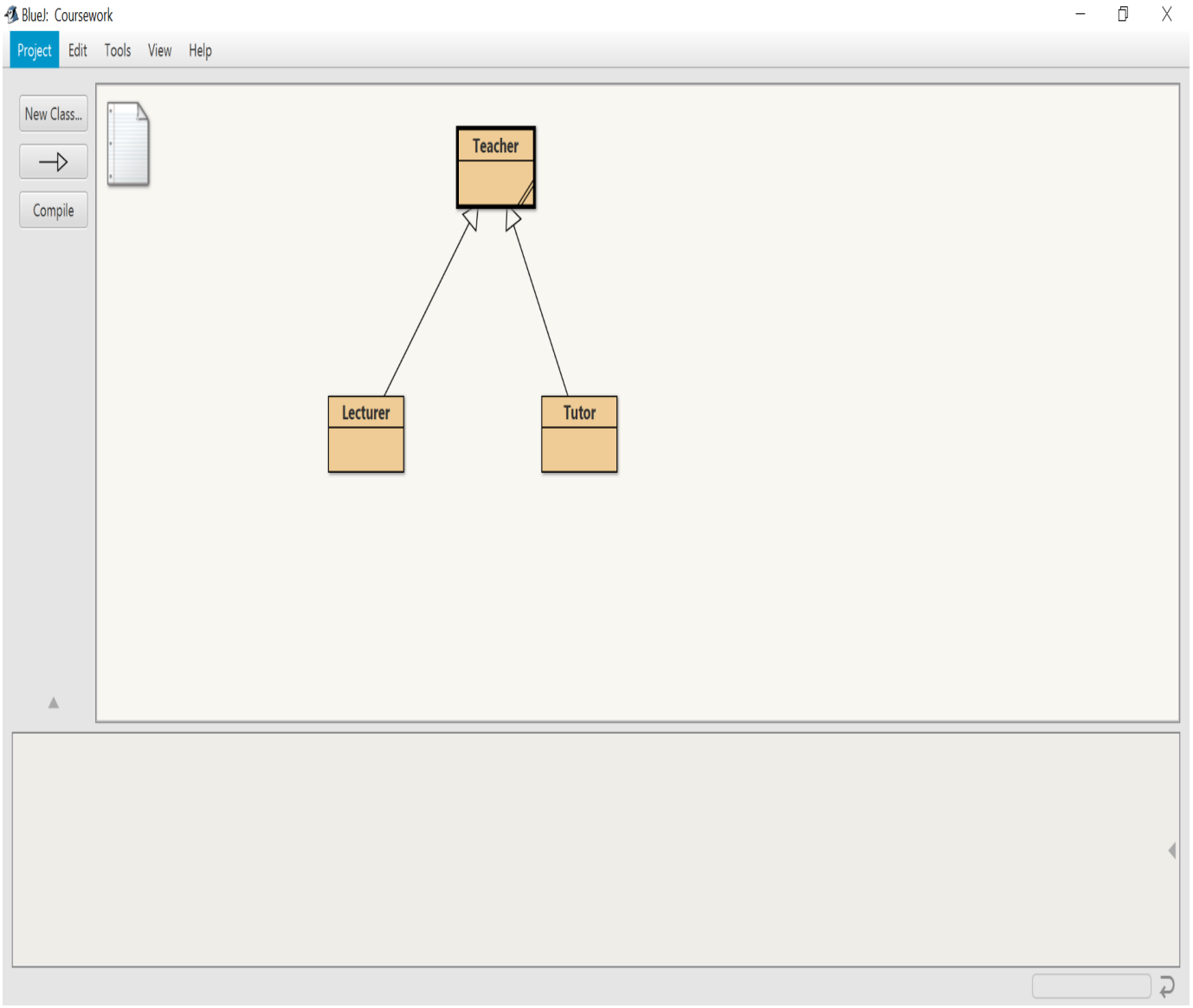
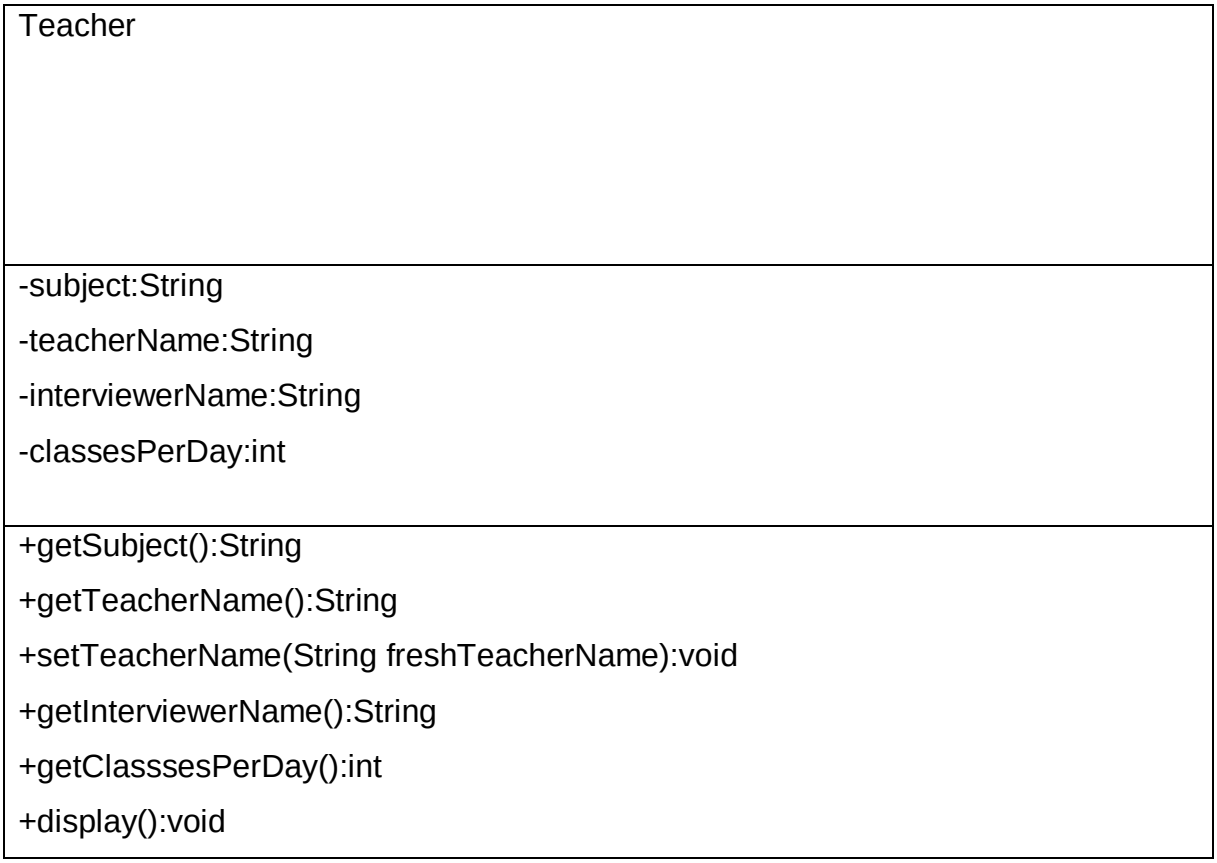


Figure 1 :class diagram

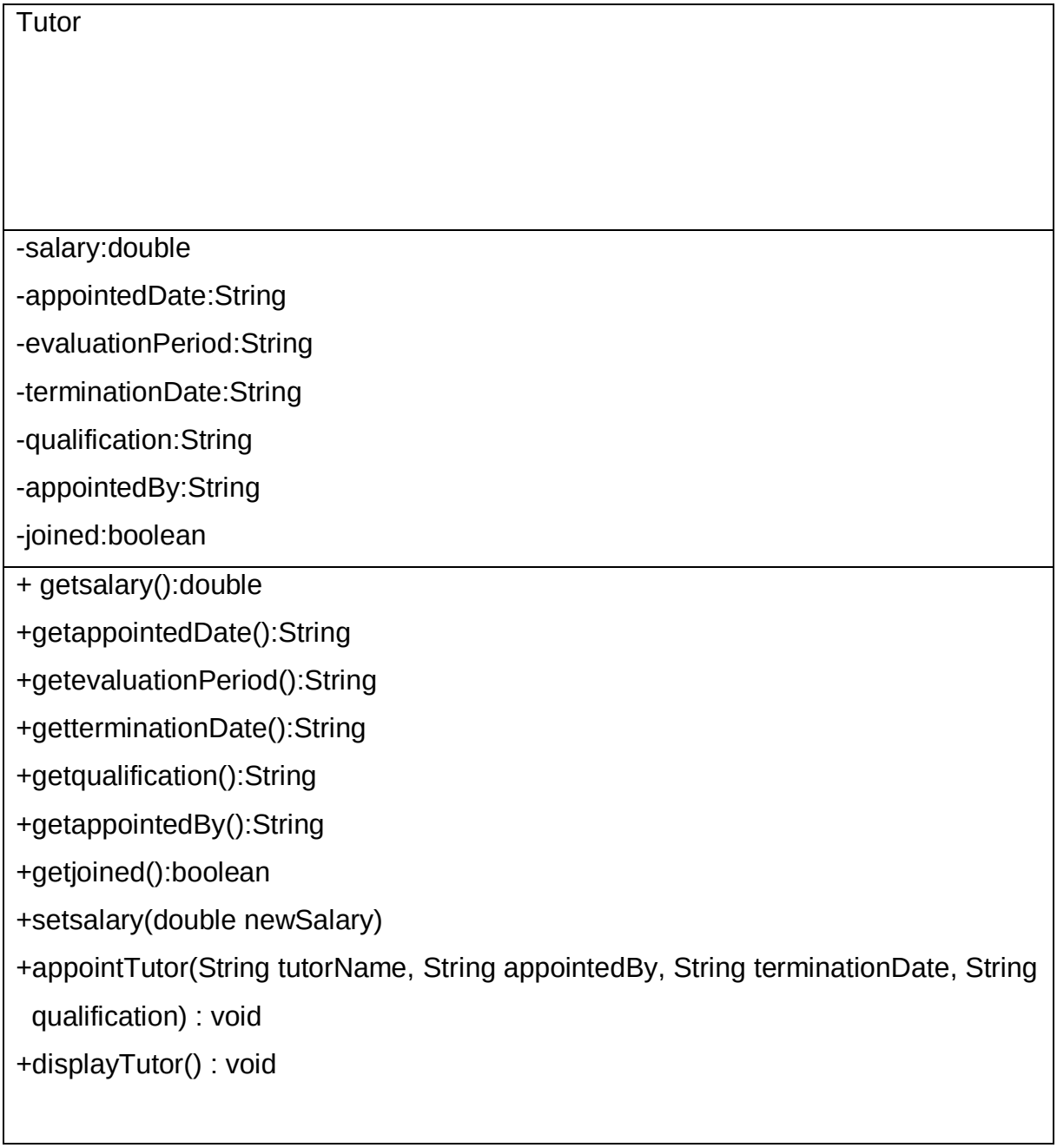
2.1 Class Diagram for “Teacher” Class



2.2 Class Diagram for “Lecturer” Class

Lecturer
-salary:double -startingFrom:String - academicBlockNumber:String - dailyWorkingHour:int - advanceSalary:double -joined:boolean - terminated:boolean
+getsalary():double +setSalary(double newsalary):void + getstartingFrom():String + getacademicBlockNumber():String +getdailyWorkingHour():int +setdailyWorkingHour(int newworkingHour):void +getadvanceSalary():double +getjoined():boolean +getterminated():boolean +setlecturer(lecturerName:String,startingFrom:String,advanceSalary:double, academicBlockNumber:String) : void + lecturerTermination():void + display():void + displayLecturer():void

2.3 Class Diagram for “Tutor” Class



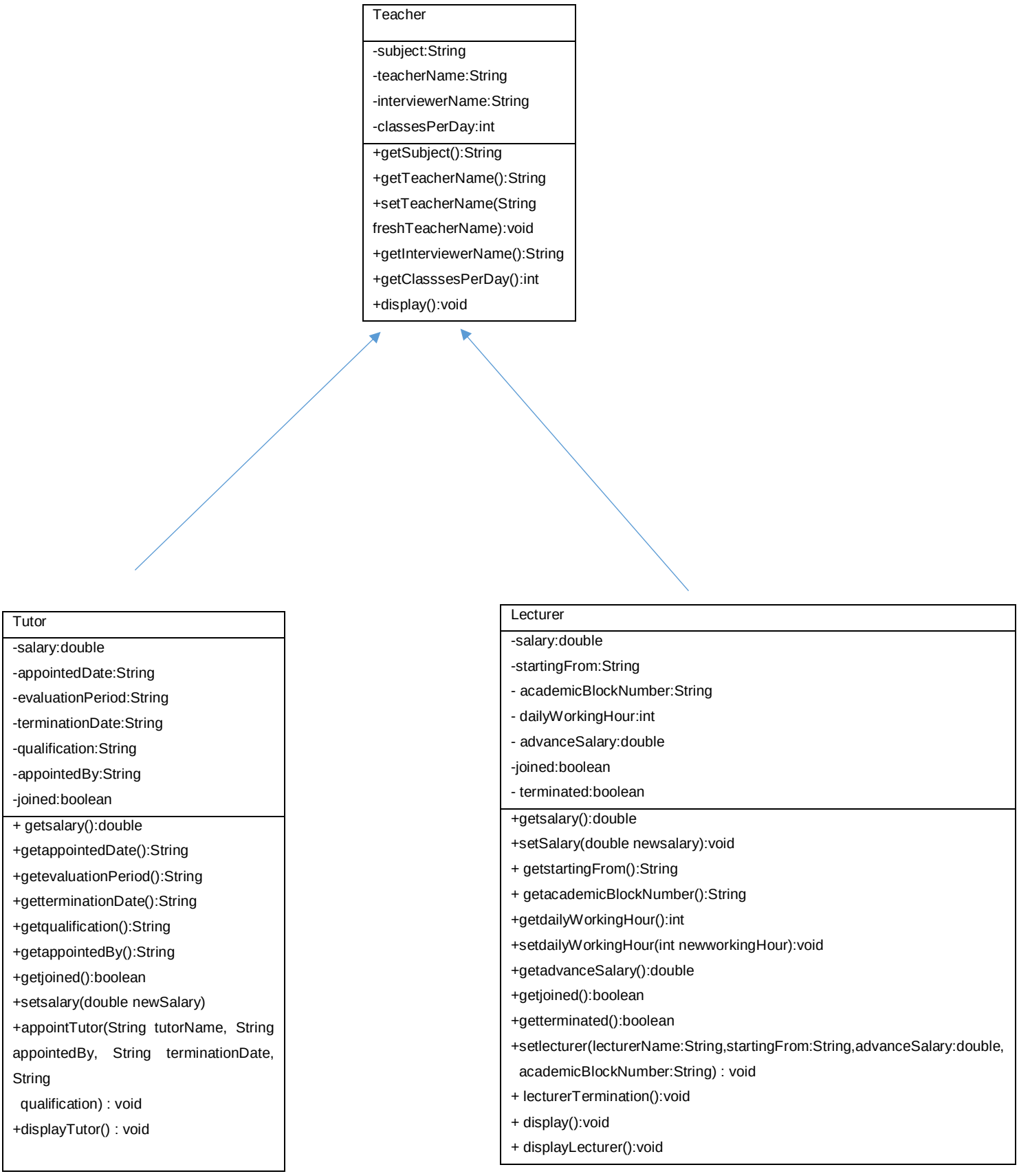


Fig:class diagram

Girija Tamang

3.Pseudocode

3.1Pseudocode for" Teacher" class:

```
DEFINE Teacher(String subject,int classPerDay,String interviewerName)
        DO
        INITIALIZE value to the variable salary
        This.salary=salary;
        INITIALIZE value to the variable teacherName
        This.teacherName=("");
        INITIALIZE value to the variable interviewerName
        This.interviewerName=interviewerName;
        INITIALIZE value to the variable classesPerDay
        This.classesPerDay=classesPerDay;
        END DO
DEFINE String getSalary()
        DO
        Return value from variable salary
        END DO
DEFINE String getTeacherName()
        DO
        Return value from variable TeacherName
        END DO
DEFINE String getInterviewerName()
        DO
        Return value from variable InterviewerName
        END DO
DEFINE int ClassesPerDay()
        DO
        Return value from variable ClassesPerDay
```

END DO

DEFINE setTeacherName(String freshteacherName)

DO

INITIALIZE value to the variable teachername

This.teacherName=freshteacherName

END DO

DEFINE display ()

DO

DISPLAY”getSalary()method”

DISPLAY”getClassesPerDay()method”

DISPLAY”getInterviewerName()method”

IF (teacherName is not an empty String)

DO

DISPLAY”getTeacherName()method”

END DO

END DO

3.2Pseudocode for” Lecturer” Class

DEFINE Lecturer (String subject,int classPerDay,String interviewerName,String className,double salary,int dailyWorkingHours)

DO

INITIALIZE value of (subject,classPerDay,interviewerName) from parent class (teacher).

INITIALIZE value to the variable salary

This.salary=salary;

INITIALIZE value to the variable dailyWorkingHour

This.dailyWorkingHour=dailyWorkingHour;

INITIALIZE value to the variable startingFrom to empty String

INITIALIZE value to the variable academicBlockNumber to empty String

INITIALIZE value to the variable advanceSalary to 0.0

INITIALIZE value to the variable joined to Boolean value “false”

INITIALIZE value to the variable terminated to Boolean value “false”

END DO

DEFINE double getSalary()

DO

Return value from variable salary

END DO

DEFINE String getStartingFrom()

DO

Return value from variable startingFrom

END DO

DEFINE String getAcademicBlockNumber()

DO

Return value from variable academicBlockNumber

END DO

DEFINE int getDailyWorkingHour()

DO

Return value from variable dailyWorkingHour

END DO

DEFINE double getAdvanceSalary()

DO

Return value from variable advanceSalary

END DO

DEFINE boolean getJoined()

DO

Return value from variable joined

END DO

```
DEFINE boolean getTerminated()
        DO
        Return value from variable terminated
        END DO
DEFINE setSalary(double newSalary)
        DO
        INITIALIZE value to the variable newsalary
        This.salary=newSalary
        END DO
DEFINE setworkingHour(int newworkingHour)
        DO
        INITIALIZE value to the variable newworkingHour
        This.workingHour=newWorkingHour
        END DO



DEFINE Lecturer (String lecturerName,String startingFrom,double advanceSalary,String
academicBlockNumber)
DO
        IF(joined is True)
                DO
                        DISPLAY lecturerName
                        DISPLAY academicBlockNumber
                END DO
        ELSE
                DO
                        super.setTeacherName(lecturerName);
                         this.startingFrom=startingFrom;
                          this.academicBlockNumber=academicBlockNumber;
```

```
                    this.advanceSalary=advanceSalary;
                   this.joined=true;
                    this.terminated=false;
        END DO
DEFINE lecturerTermination()
DO
IF(terminated is True)
        DO
        DISPLAY lecturer is terminated
        END DO
ELSE
        DO
        super.setTeacherName("");
        this.startingFrom=("");
        this.advanceSalary=0.0;
        this.joined=false;
        this.terminated=true;
         END DO
DEFINE display ()
        DO
        DISPLAY”getSubject()method”
        DISPLAY”getTeacherName()method”
        DISPLAY”salary”
DEFINE displayLecturer()
DO
super.display();
        IF(joined is True)
                DO
```

DISPLAY terminated

DISPLAY startingFrom

DISPLAY advanceSalary

DISPLAY”getTeacherName()method”

END DO

END DO

3.3Pseudocode for” Tutor “Class:

DEFINE Tutor (String subject,String interviewerName,int classPerDay,double Salary,String appointedBy,int terminationDate)

DO

super(subject,classPerDay,interviewerName);

this.salary=salary;

this.appointedBy=appointedBy;

this.appointedDate=("");

this.evaluationPeriod=("");

this.qualification=("");

this.joined=false;

END DO

DEFINE double getSalary()

DO

Return value from variable salary

END DO

DEFINE String getAppointeddate()

DO

Return value from variable appointeddate

END DO

DEFINE String getEvaluationPeriod()

DO

Return value from variable evaluationPeriod

```
        END DO
DEFINE String getTerminatedDate()
        DO
        Return value from variable terminatedDate
        END DO
DEFINE String getQualification()
        DO
        Return value from variable qualification
        END DO
DEFINE String getAppointedBy()
        DO
        Return value from variable appointedBy
        END DO
DEFINE boolean getJoined()
        DO
        Return value from variable joined
        END DO
DEFINE setSalary(double newSalary)
        DO
        IF(joined is False)
                DO
                Salary=newsalary
                END DO
        ELSE
                DO
                PRINT we cannot change salary
                END DO
        END DO
```

```
DEFINE setappointTutor (String tutorName,String appointedBy,String
terminationDate,String qualification)
        DO
        IF(joined is False)
                super.setTeacherName(tutorName);
                 joined=true;
        ELSE
                DO
                PRINT The tutor has already appointed
                this.appointedDate=appointedDate;
                 this.terminationDate=terminationDate;
                this.qualification=qualification;
                END DO
        END DO
DEFINE displayTutor()
DO
super.display();
IF(joined is true)
        DO
                DISPLAY appointedDate
                DISPLAY getTeacherName()
                DISPLAY evaluationPeriod
                DISPLAY terminationDate
                DISPLAY salary
                DISPLAY qualification
                DISPLAY appointedBy
        END DO
END DO
```

4.Method Description

4.1Method Description for “Teacher” class

- getsubject():This method is used to obtain the variable subject instance value and return it to getsubject .
- getTeacherName():This method is used to obtain the variable teacherName instance value and return it to getTeacherName.
- getInterviewerName():This method is used to obtain the variable interviewerName instance value and return it to getInterviewerName.
- getclassesPerDay():This method is used to obtain the variable classesPerDay instance value and return it to getclassesPerDay.
- setTeacherName():This is a method that defines the name of the teacher before t he method is given and returns it.
- Display():This method is used to display the required information of teacher class.

4.2 Method description for “Lecturer” class

- getSalary():This method is used to obtain the variable salary instance value and return it to getsalary.
- setSalary():This is a method that defines the salary before the method is given and returns it.
- getstartingFrom():This method is used to obtain the variable starting from instance value and return it to getstartingFrom.
- getacademicBlockNumber():This method is used to obtain the variable academicblockNumber instance value and return it to getacademicBlockNumber.
- getdailyWorkingHour():This method is used to obtain the variable dailyworkingHour instance value and return it to getdailyworkingHour.
- setdailyWorkingHour():This is a method that defines the name of the dailyworking hour before the method is given and returns it
- getadvanceSalary():This method is used to obtain the variable advanceSalary instance value and return it to getadvancesalary
- getjoined():This method is used to obtain the variable joined instance value and return it to getjoined
- getterminated():This method is used to obtain the variable terminated instance value and return it to getterminated.

- setlecturer():This is a method that defines the name of the lecturer before the method is given and returns it
- set lecturerTermination():
- display():This method is used to display the required information of lecturer class.

4.3 Method description for "Tutor"

- getSalary():This method is used to obtain the variable salary instance value and return it to getsalary.
- getappointeddate():This method is used to obtain the variable appointeddate instance value and return it to getappointeddate.
- getevaluationPeriod():This method is used to obtain the variable evaluationperiodinstance value and return it to getevaluationperoid.
- getterminationDate():This method is used to obtain the variable terminationDate instance value and return it to getterminationdate.
- setsalary():This is a method that defines the value of the salary before the method is given and returns it
- getqualification():This method is used to obtain the variable qualification instance value and return it to getqualification
- getjappointedby():This method is used to obtain the variableappointedby instance value and return it to getappointedby
- getjoined():This method is used to obtain the variable joined instance value and return it to getjoined.
- appointutor():This method is used to display the required information of tutor class
- display():This method is used to display the requirement information of tutor.

5.Testing

5.1 Test 1

Objective	Inspect Lecturer Class, appoint lecturer for any particular subject, and re-inspect the Lecturer Class.
Action	I. Inspect Lecturer class Subject=”BIT” ClassesperDay=2 className=”p” salary=22222 dailyworkingHour=6 II. Appointing lecturer for any particular subject lecturerName=”pesal” startingFrom=2019 advanceSalary=222 academicBlockNumber=L1C5
Expected Result	Object will be created and value will be assign
Actual Result	Creates object and assign value
Conclusion	Test done succesfully

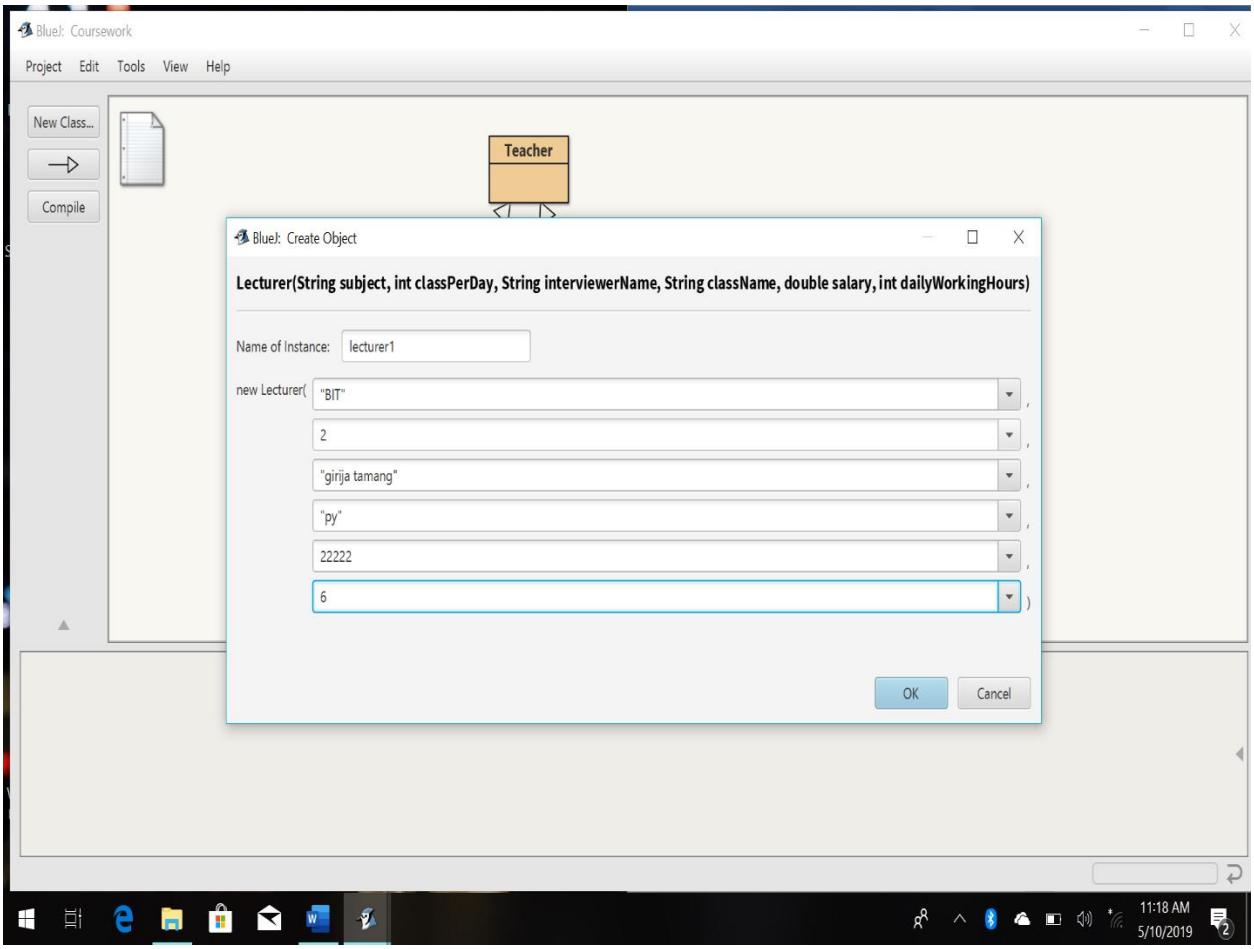


Figure 2:lecturer

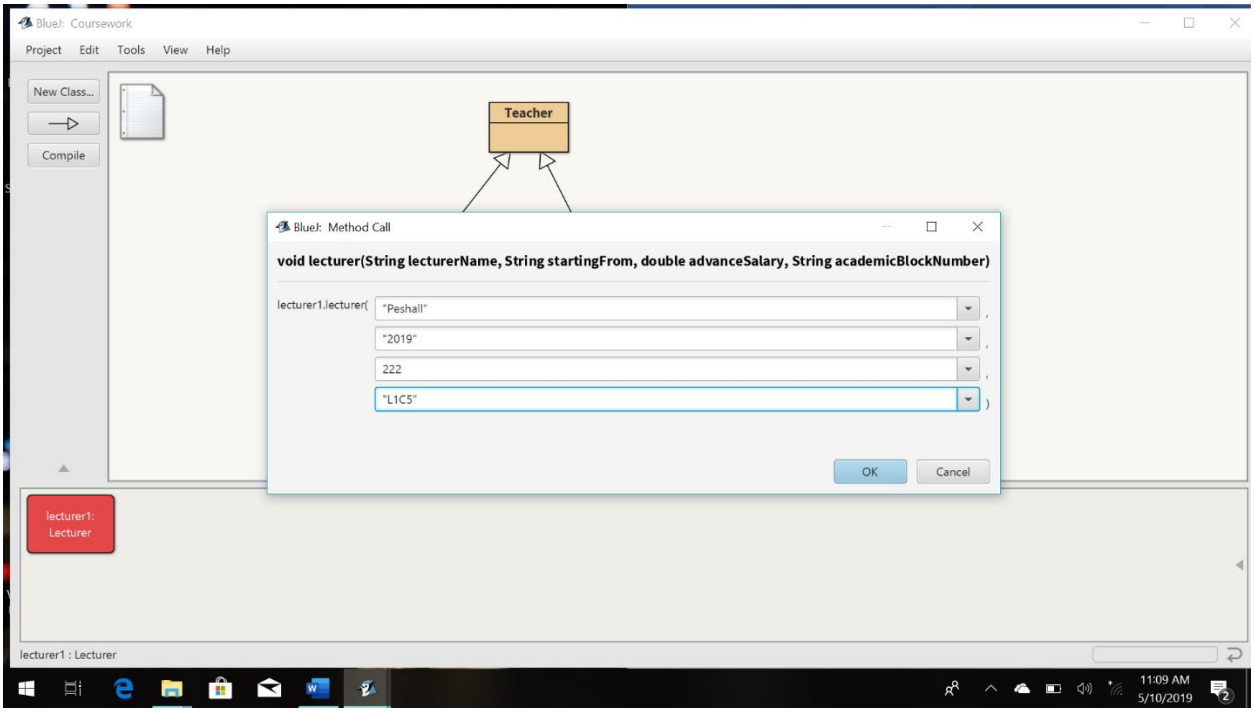


Figure 3:appoint lecturer in particular subject

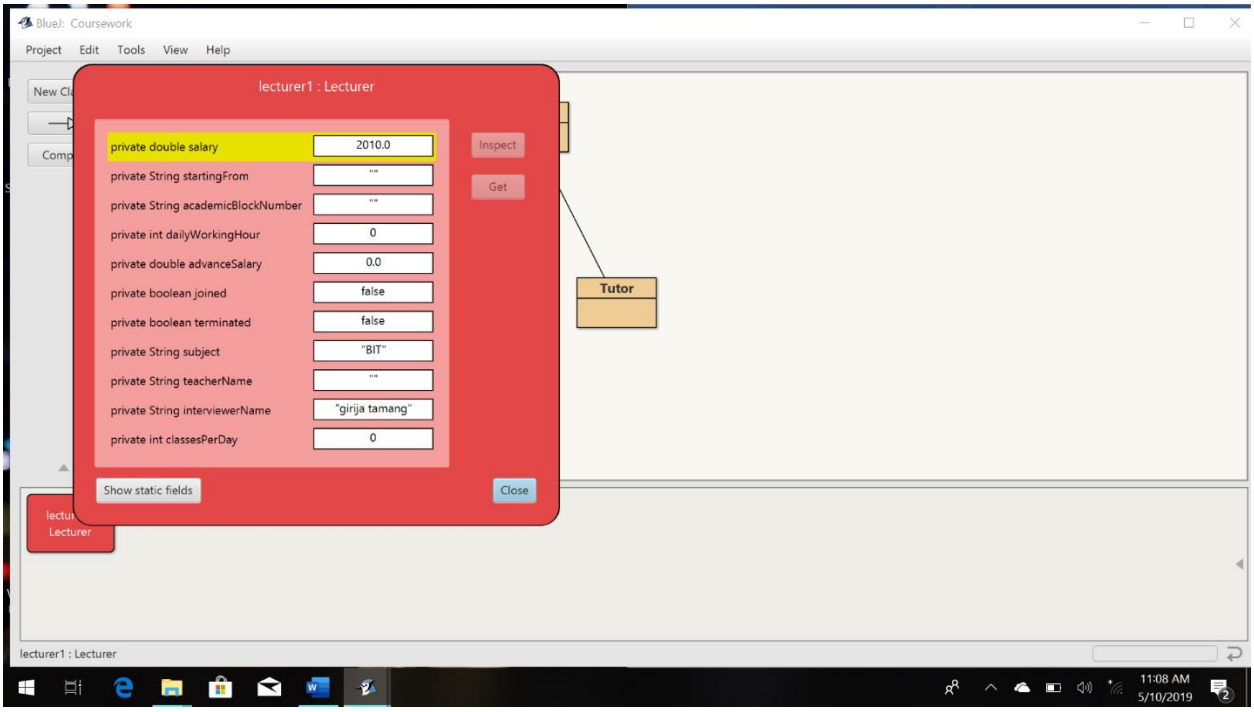
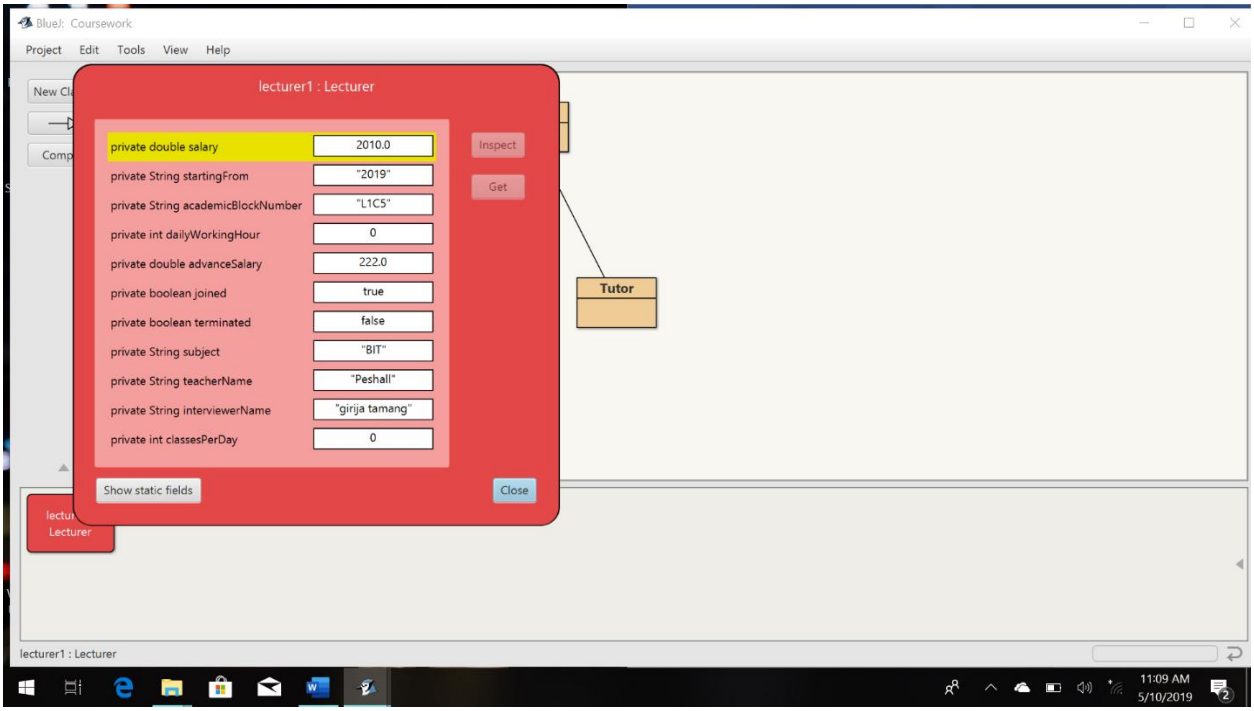


Figure 4:inspect lecturer



Girija Tamang

5.2Test 2

Objective	Inspect Lecturer, change the status of lecturer to terminated, and re-inspect the Lecturer Class.
Action	I. Changing the status of lecturer termination
Expected Result	Object will create and assign value
Actual Result	Create object and value assign
Conclusion	Test done successfully

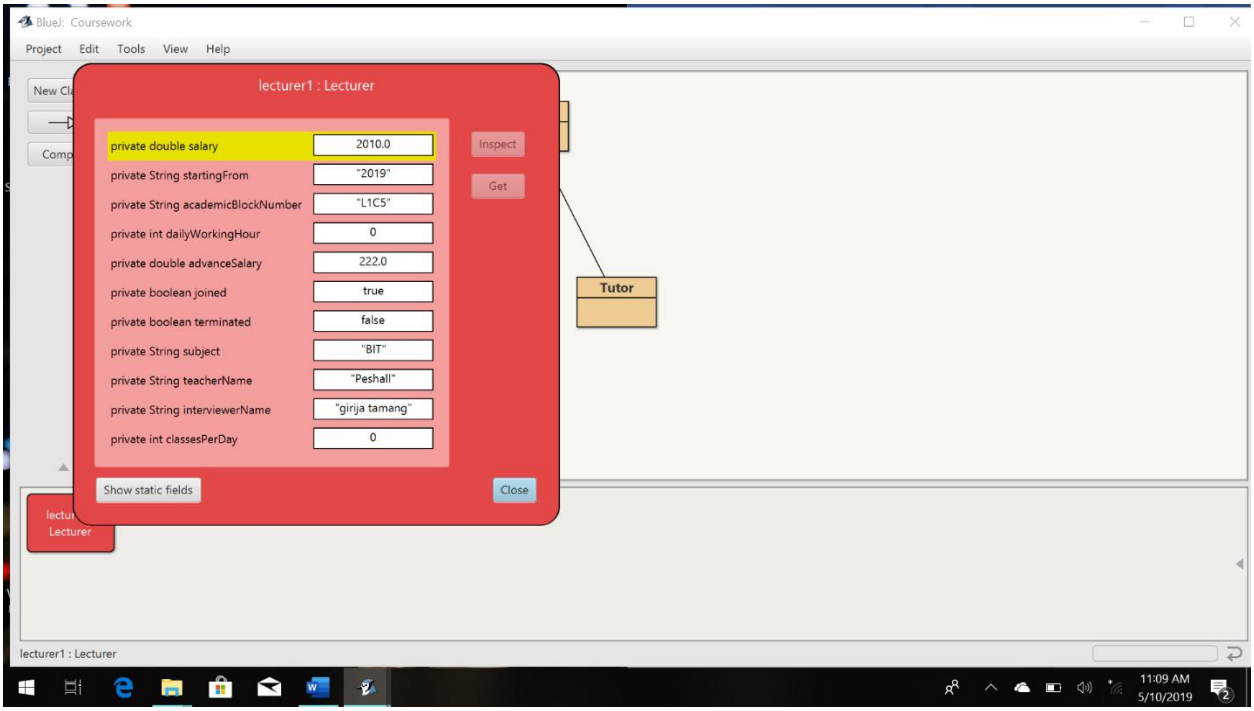


Figure 5 inspect lecture in test 2

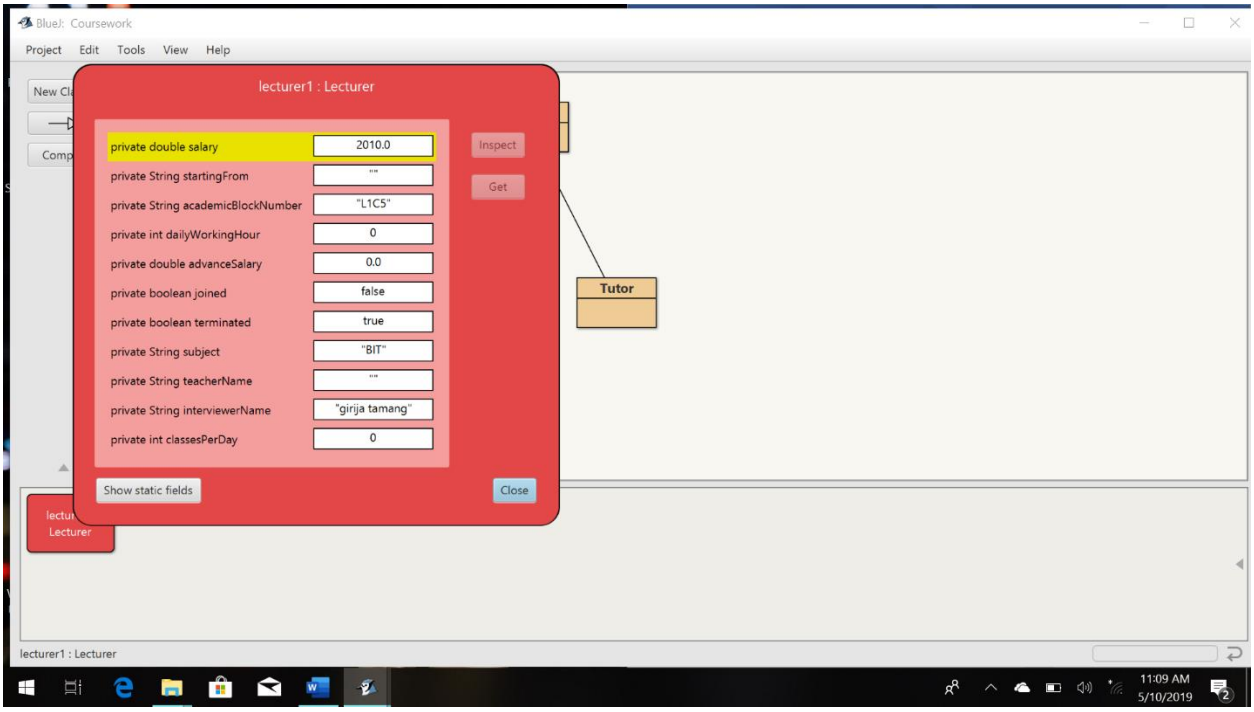


Figure 6 Changing the status of lecturer termination

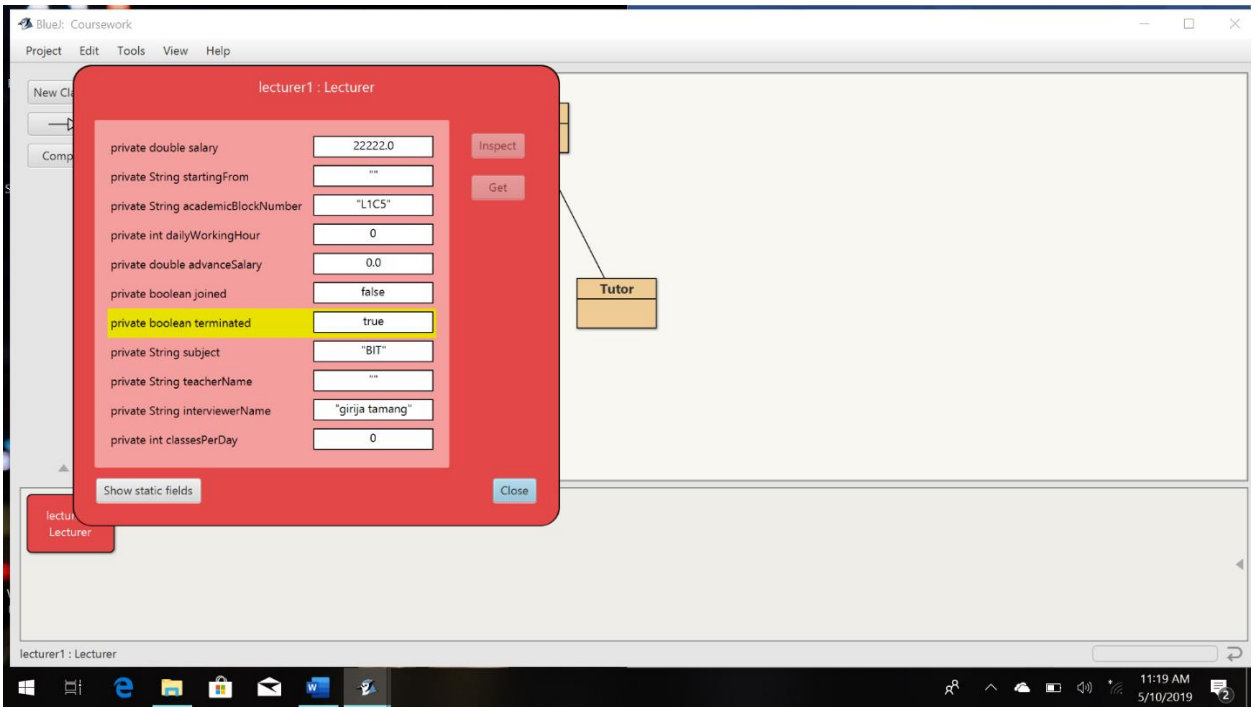


Fig 9: re-inspect the Lecturer Class.

5.3 Test 3

Objective	Inspect Tutor class, appoint tutor, and re-inspect.
Action	i. Inspect Tutor Class Subject=bba interviewerName=”girija tamang” classesPerDay=3 salary=33333 appointedby=peshal terminationdate=2220 ii. Appointing tutor=”mike hussy” Appointed by=”peshall” terminationDate=2222 qualification=MBA
Expected Result	Object will create
Actual Result	Objected created and assign value
Conclusion	Test done sucessfully

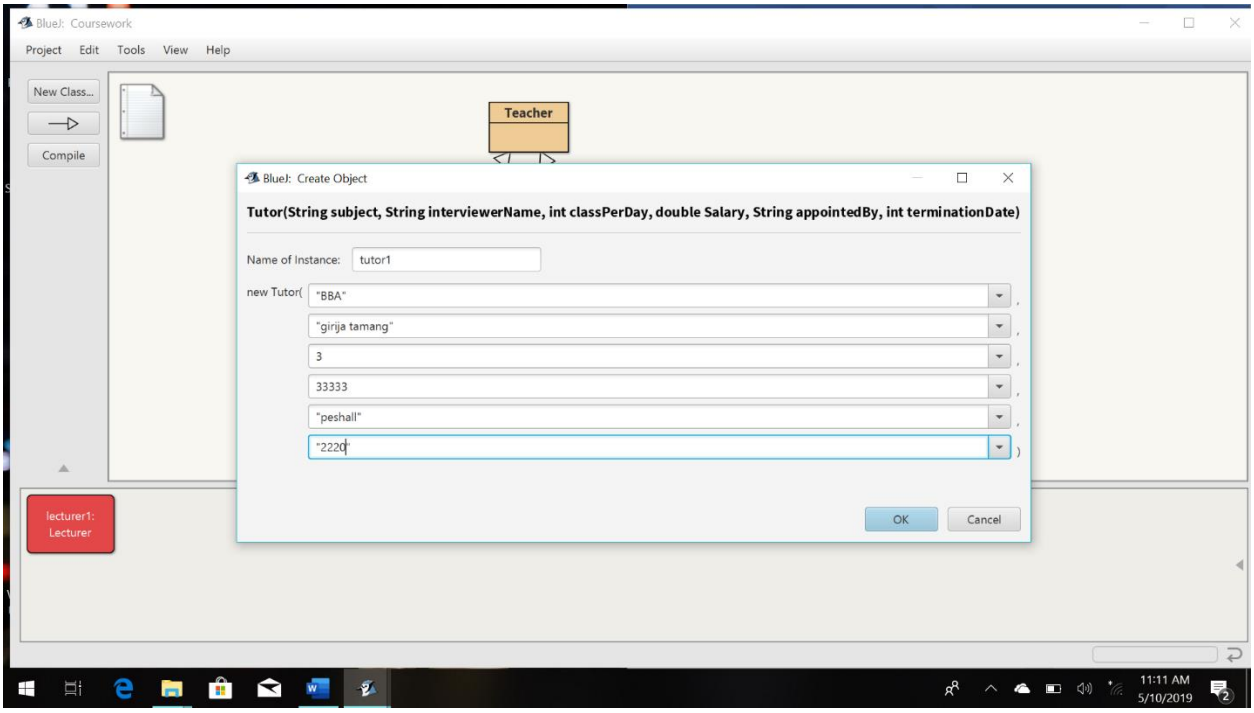


Fig 10: Inspect Tutor class,

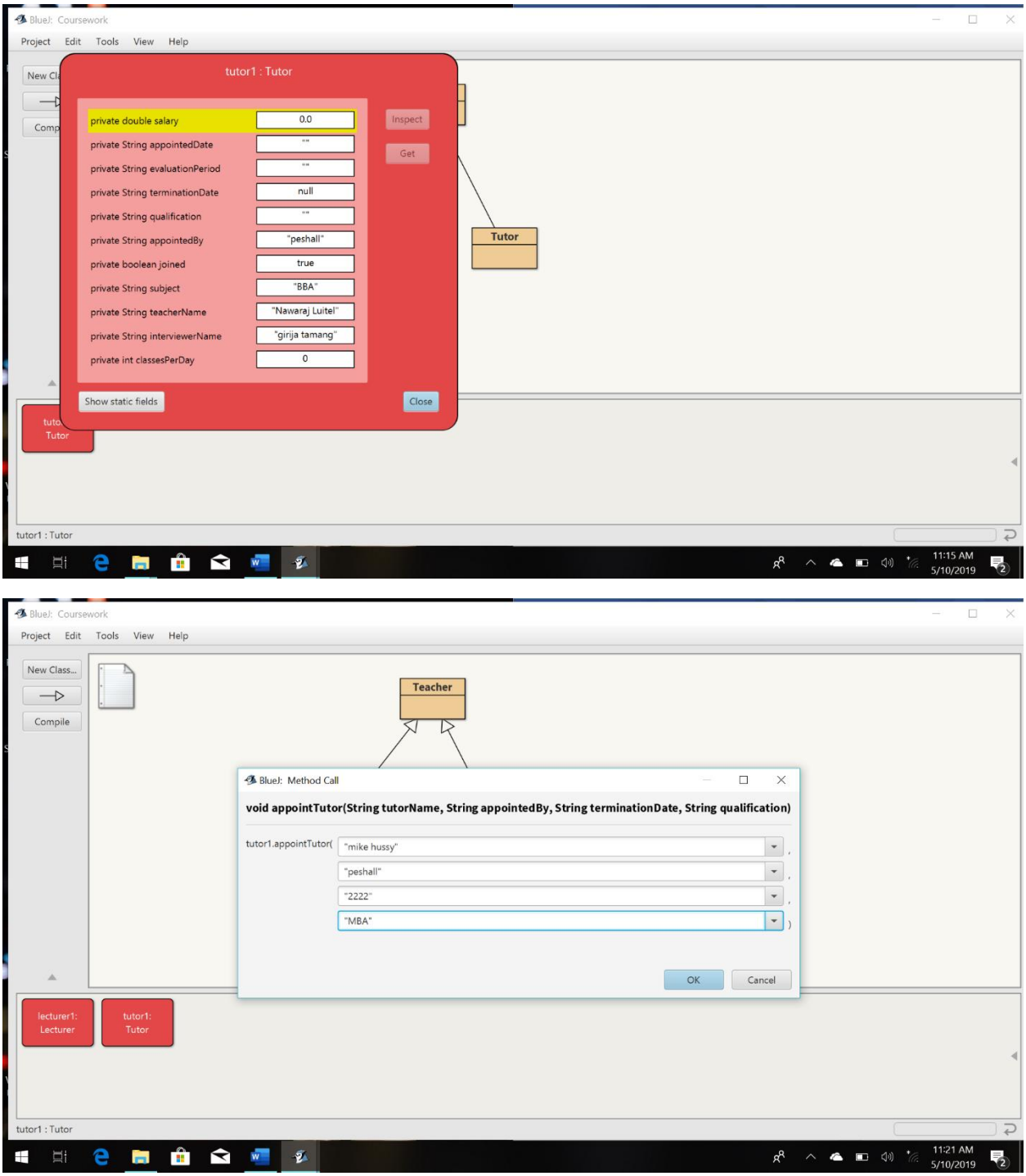


Fig 11: appoint tutor

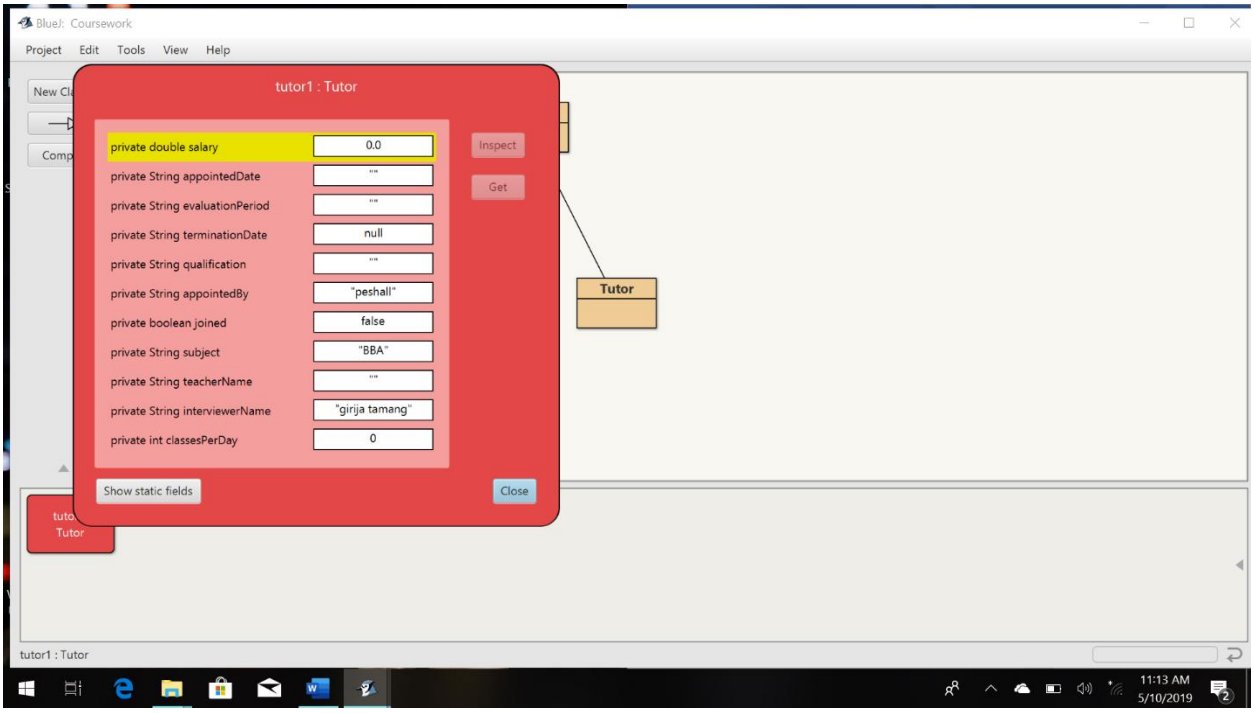
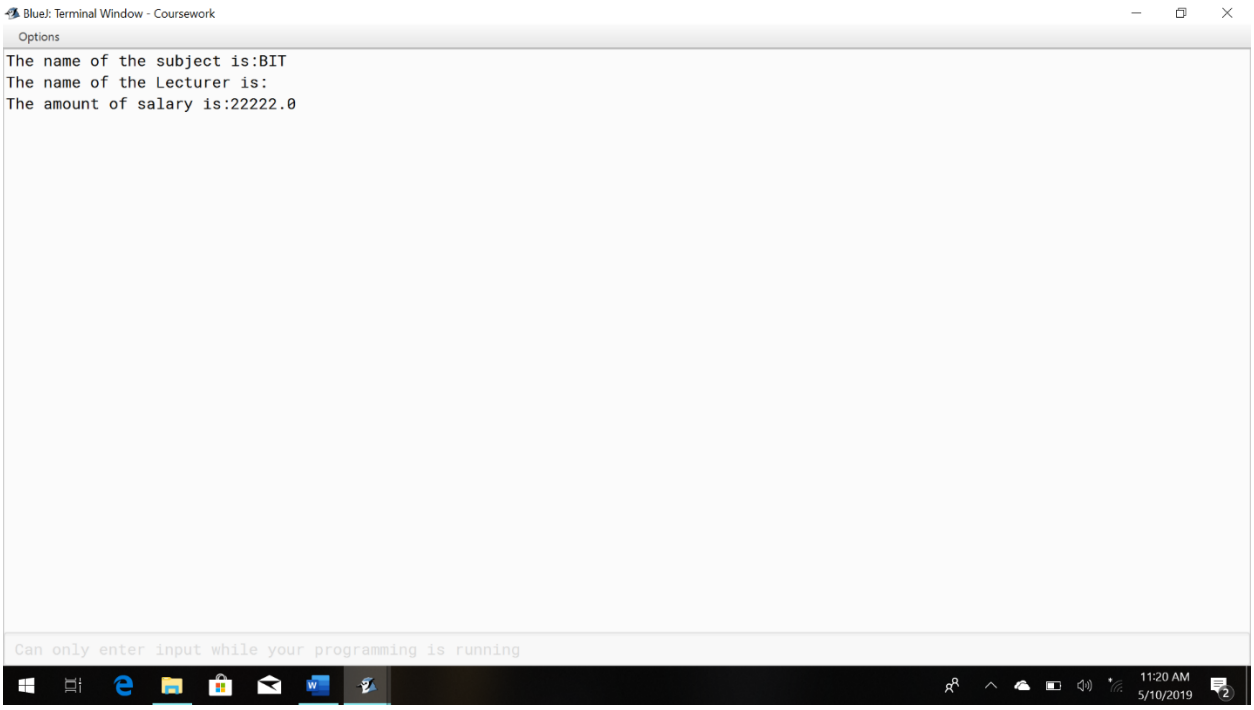


Fig 12:reinspect tutor

5.4Test 4

Objective	Display the detail of Lecturer and Tutor Class.
Action	i. Printing the details of lecture and tutor class
Expected Result	Show all lecture and tutor class
Actual Result	All result shown
Conclusion	Test done successfully



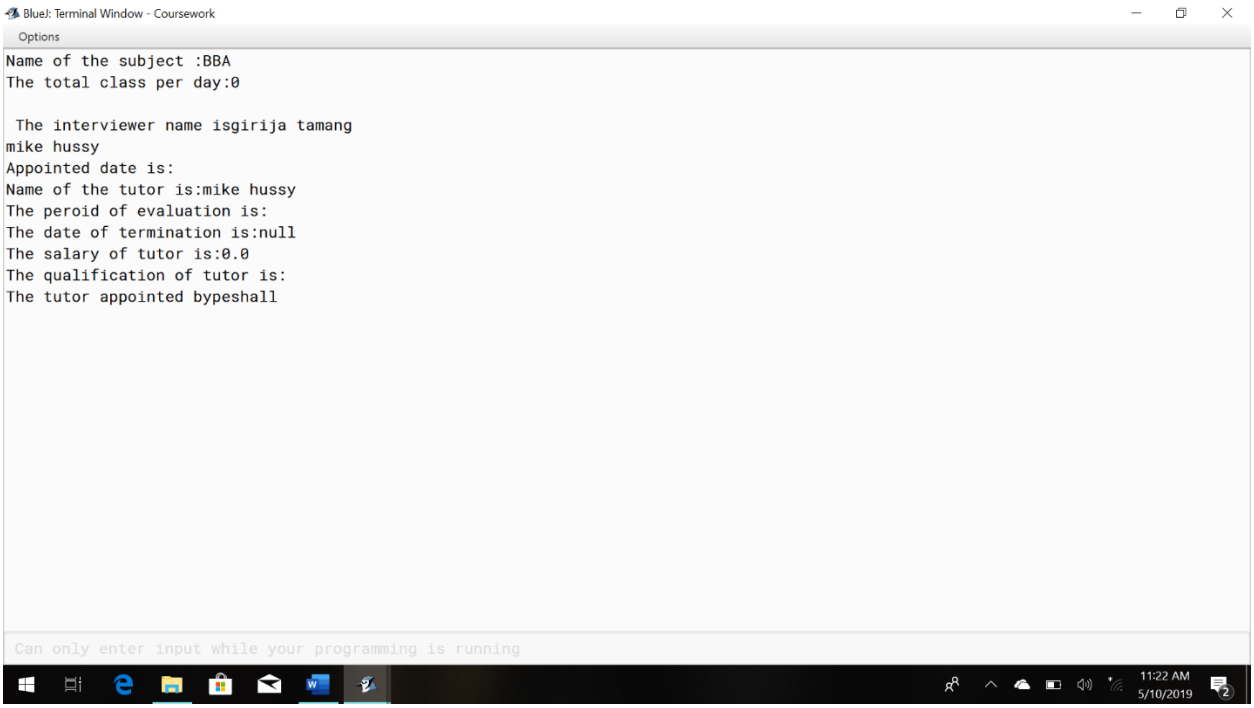


Fig 15:Display al information

6.Error Detection

6.1 Syntax Error

Java syntax errors refer to a programmer's mistake in implementing the Java language grammar. The logic of the program itself does not include any errors.

6.2 Logical Error

A logical error (or logical error) causes incorrect or unexpected behavior to a program's source code. It is a kind of runtime error which can just cause the wrong output or cause a program to crash while it is running.

6.3Run Time Error

An error occurring during program execution. In contrast, errors in compile time occur during the compilation of a program. Runtime errors point to program bugs or problems that were anticipated by designers, but could do nothing. Extending memory, for example, often causes an error in runtime.

7.Conclusion

This coursework is based on java programming language for developing teacher appointment system. The project should be done in Blue J which is an IDLE used in java. I have to develop three classes in Blue J script. It helped me to gain knowledge on coding.

Programming was unknown area for me in the beginning. Sometime it is really hard to understand the module as I am new to the module. With the help of friends and teachers I become more familiar to the programming and able to solve problems of programming. Developing the project was difficult for me in the beginning but after consulting with module teacher I was able to understand the basic and able to develop the project.

while developing this project I had faced many problems like syntax error, runtime errors, logical error, etc by the help of the cooperative module teacher, friends, elder I have able to tackle the problem and develop working project. This course work helped me a lot and made familiar on programming.

8.Appendix

8.1 Source Code of Teacher class

```
/**
 * Description of class Teacher .
 *
 * @author (Girija Tamang)
 * @londonmetId (18030995)
 */
public class Teacher
{
    private String subject;
    private String teacherName;
    private String interviewerName;
    private int classesPerDay;
    public Teacher(String subject,int classPerDay,String interviewerName)
    {
        this.subject=subject;
        this.teacherName=("");
        this.interviewerName=interviewerName;
        this.classesPerDay=classesPerDay;
    }
    //getSubject() method returns the subject
    public String getSubject()
    {
        return subject;
    }
    //getTeacherName() method returns the teachername
```

```
    public String getTeacherName()
    {
        return teacherName;
    }
    //getInterViewerName() method returns the interviewrName
    public String getInterviewerName()
    {
        return interviewerName;
    }
    //getClassesPerDay() method returns the classesperday
    public int getClassesPerDay()
    {
        return classesPerDay;
    }
    //setTeacherName() method sets the value of the name
    public void setTeacherName(String freshteacherName)
    {
        this.teacherName=freshteacherName;
    }
    public void display()
    {
        System.out.println("Name of the subject :"+subject+"\nThe total class per
day:"+classesPerDay);
        System.out.println("\n The interviewer name is"+interviewerName);
        if(teacherName!=(""))
        {
            System.out.println(teacherName);
        }
    }
```

```
}
```

8.2 Source Code of lecturer class

```
/**
 * Description of class Lecturer .
 *
 * @author (Girija Tamang)
 * @lodonmetId (18030995)
 */
public class Lecturer extends Teacher
{
    private double salary;
    private String startingFrom;
    private String academicBlockNumber;
    private int dailyWorkingHour;
    private double advanceSalary;
    private boolean joined;
    private boolean terminated;
    public Lecturer(String subject,int classPerDay,String interviewerName,String
className,double salary,int dailyWorkingHours)
    {
        super(subject,classPerDay,interviewerName);
        this.salary=salary;
        this.dailyWorkingHour=dailyWorkingHour;
        this.startingFrom=("");
        this.academicBlockNumber=("");
        this.advanceSalary=0.0;
        this.joined=false;
```

```
        this.terminated=false;
    }
    public double getSalary()
    {
        return salary;
    }
    public int getDailyWorkingHour()
    {
        return dailyWorkingHour;
    }
    public String getStartingForm()
    {
        return startingFrom;
    }
    public String getAcademicBlockNumber()
    {
        return academicBlockNumber;
    }
    public double getAdvanceSalary()
    {
        return advanceSalary;
    }
    public boolean getJoined()
    {
        return joined;
    }
    public boolean getTerminated()
    {
```

```
        return terminated;
    }
    //setSalary() method sets the value of the salary
    public void setsalary(double newsalary)
    {
        this.salary=newsalary;
    }
    //setWorkingHour() method sets the value of the WorkingHour
    public void setworkingHour(int newworkingHour)
    {
        this.dailyWorkingHour=newworkingHour;
    }
    public void lecturer(String lecturerName,String startingFrom,double
advanceSalary,String academicBlockNumber)
    {
        if (joined==true)
        {
            System.out.println("The name of the appointed lecturer
is:"+lecturerName+"\nThe academic block number is:"+academicBlockNumber);
        }
        else
        {
            super.setTeacherName(lecturerName);
            this.startingFrom=startingFrom;
            this.academicBlockNumber=academicBlockNumber;
            this.advanceSalary=advanceSalary;
            this.joined=true;
            this.terminated=false;
        }
```

```
    }
    public void lecturerTermination()
       {
          if(terminated==true)
          {
             System.out.println("The lecturer is terminated");
          }
          else
          {
             super.setTeacherName("");
             this.startingFrom=("");
             this.advanceSalary=0.0;
             this.joined=false;
             this.terminated=true;
          }
    }
    public void display()
    {
        System.out.println("The name of the subject is:"+getSubject()+"\nThe name of the
Lecturer is:"+getTeacherName()+"\nThe amount of salary is:"+salary);
    }
    public void displayLecturer()
    {
        super.display();
        if (joined=true)
        {
            System.out.println("termination status:"+terminated+"\nThe StartingFrom
is:"+startingFrom+"\nThe amount of advance salary is:"+advanceSalary+"\nThe name of
teacher is:"+getTeacherName());
```

```
        }
    }
}
```

8.3 Source Code o Tutor class

```
/**
 * Description of class Tutor.
 *
 * @author (Girija Tamang)
 * @londonmetId (18030995)
 */
public class Tutor extends Teacher
{
    private double salary;
    private String appointedDate;
    private  String evaluationPeriod;
    private String terminationDate;
    private String qualification;
    private String appointedBy ;
    private boolean joined;
    public Tutor(String subject,String interviewerName,int classPerDay,double
Salary,String appointedBy,int terminationDate)
    {
        super(subject,classPerDay,interviewerName);
        this.salary=salary;
        this.appointedBy=appointedBy;
        this.appointedDate("");
        this.evaluationPeriod("");
        this.qualification("");
```

```
        this.joined=false;
    }
    public double getSalary()
    {
        return salary;
    }
    public String getAppointedDate()
    {
        return appointedDate;
    }
    public String getEvaluationPeriod()
    {
        return evaluationPeriod;
    }
    public String getTerminationDate()
    {
        return terminationDate;
    }
    public String getQualification()
    {
        return qualification;
    }
    public String getAppointedBy()
    {
        return appointedBy;
    }
    public boolean getJoined()
    {
```

```
        return joined;
    }
    public void salary(double newSalary)
    {
        if(joined==false)
        {
            this.salary=newSalary;
        }
        else
        {
            System.out.println("We cannot change the salary");
        }
    }
    public void appointTutor(String tutorName,String appointedBy,String
terminationDate,String qualification)
    {
        if (joined==false)
        {
            super.setTeacherName(tutorName);
            joined=true;
        }
        else
        {
            System.out.println("The tutor has already appointed");
            this.appointedDate=appointedDate;
            this.terminationDate=terminationDate;
            this.qualification=qualification;
        }
    }
```

```
    public void displayTutor()
  {
    //calling parents class display
    super.display();
    if (joined==true)
    {
      System.out.println("Appointed date is:"+appointedDate);
      System.out.println("Name of the tutor is:"+getTeacherName());
      System.out.println("The peroid of evaluation is:"+evaluationPeriod);
      System.out.println("The date of termination is:"+terminationDate);
      System.out.println("The salary of tutor is:"+salary);
      System.out.println("The qualification of tutor is:"+qualification);
      System.out.println("The tutor appointed by"+appointedBy);
    }
  }
}
```