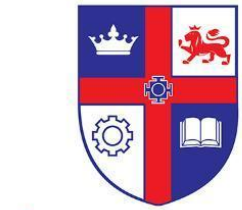
LONDON METROPOLITAN UNIVERSITY



islington college
(इस्लिङ्टन कलेज)

CS4001NI Programming

30% Individual Coursework

2023-24 Autumn

Student Name: Roshan Thapa

London Met ID: 23047462

College ID: NP01NT4A230135

Group: N5

Assignment Due Date: Friday, January 26, 2024

Assignment Submission Date: Friday, January 26, 2024

I confirm that I understand my coursework needs to be submitted online via MySecondTeacher 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

1. Introduction 1
2. 1.1 About the coursework.....1
3. 1.2 Tools Used 1
4. 2.Class Diagram 2
5. 2.1 Introduction 2
6. 2.2 Combined Class Teacher 3
7. 2.3 Class diagram of Student 4
8. 2.4 Class diagram of Lecturer 4
9. 2.5 Class diagram of Tutor.....5
10. 3. Pseudocode 6
11. 3.1 Introduction 6
12. 3.2 Pseudocode for class Teacher 7
13. 3.3 Pseudocode for class Lecturer..... 10
14. 3.4 Pseudocode for class Tutor 13
15.
16. 4. Method description 16
17. 4.1 Method description of class Teacher..... 16
18. 4.2 Method description of subclass lecturer 17
19. 4.3 Method description of subclass Tutor..... 18
20. 5. Testing..... 19

21. 5.1 Test 1
..... 19
22. 5.2 Test 2
..... 22
23. 5.3 Test 3
..... 25
24. 5.4 Test 4
..... 26
25. 6. Error Detection and Correction
..... 28
26. 6.1 Syntax error
..... 28
27. .....
..... 29
28. 6.2 Semantic error
..... 30
29. 7. Conclusion
..... 32
30. 8. References
..... 33
31.

9.Appendix……34

1.Introduction

Java is a programming language and computing platform first released by Sun Microsystems (which is now the subsidiary of Oracle) in 1995. Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. New, innovative products and digital services designed for the future continue to rely on Java, as well. Before java its name was Oak. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to java. (javatpoint, n.d.)

1.1 About the coursework

This is an individual coursework which carries 30% of the overall module. The coursework contains codes in java. The aim of this coursework is to check the understanding of the student about the java. In the coursework, we had to develop the program to set the working hours of the teachers. The project is done by using Blue J the project has three java files: Teacher and its subclasses Lecturer and Tutor.

1.2 Tools Used

During the coursework, we need to use different tools for the development of the project. The different type of tools used are:

- BlueJ: 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.This software application helps to provide a more precies interface for creating projects and coding in java. (opensource.com, n.d.)

- MS-Word: Microsoft Word is a computer application program written by Microsoft and a word processing software. It is developed by Microsoft and is part of Microsoft Office Suite. It enables you to create, edit and save professional documents like letters and reports.It is mainly used to design text for presentation. (javatpoint, n.d.)

- Draw.io: Designed by Seibert Media, draw.io is proprietary software for making diagrams and charts. The software lets you choose from an automatic layout function or create a custom layout. They have a large selection of shapes and hundreds of visual elements to make your diagram or chart one-of-a-kind. The drag-and-drop feature makes it simple to create a great looking diagram or chart. (computerhope., n.d.)

2.Class Diagram

2.1 Introduction

Class diagram is the Unified Modeling Language (UML) which 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, n.d.)

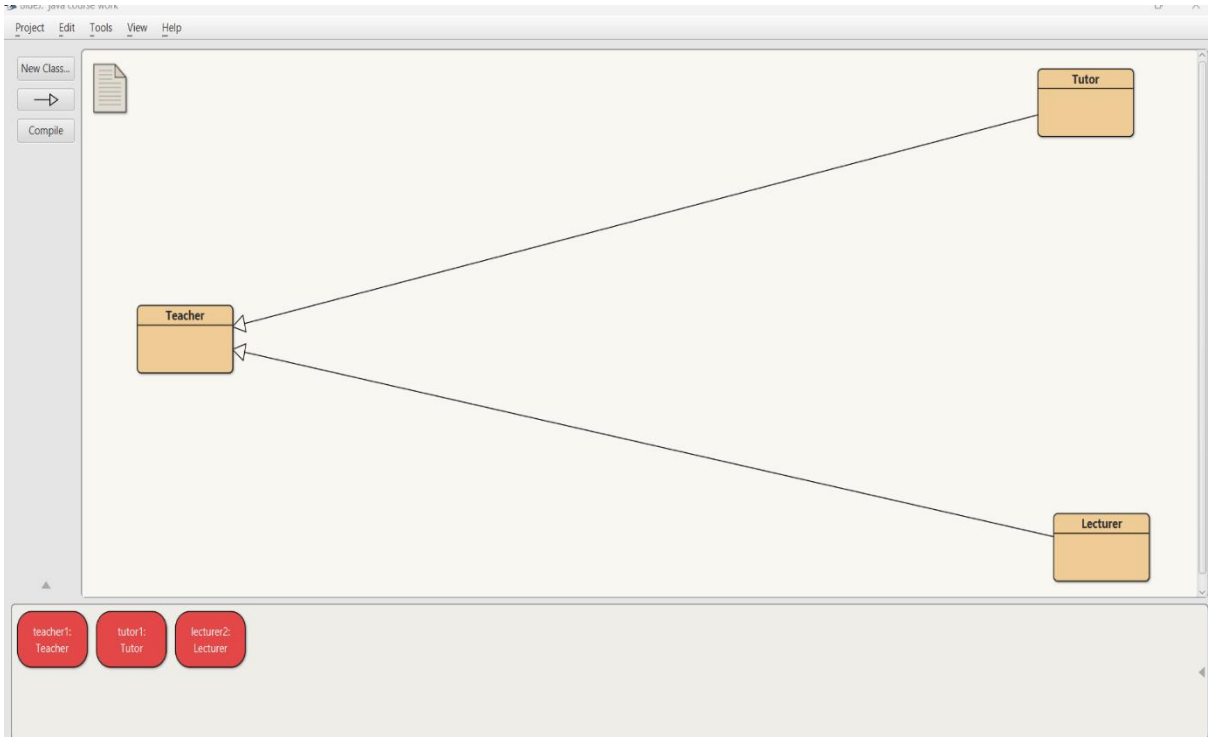


Figure 1: Screenshot from BlueJ

2.2 Combined Class Diagram:

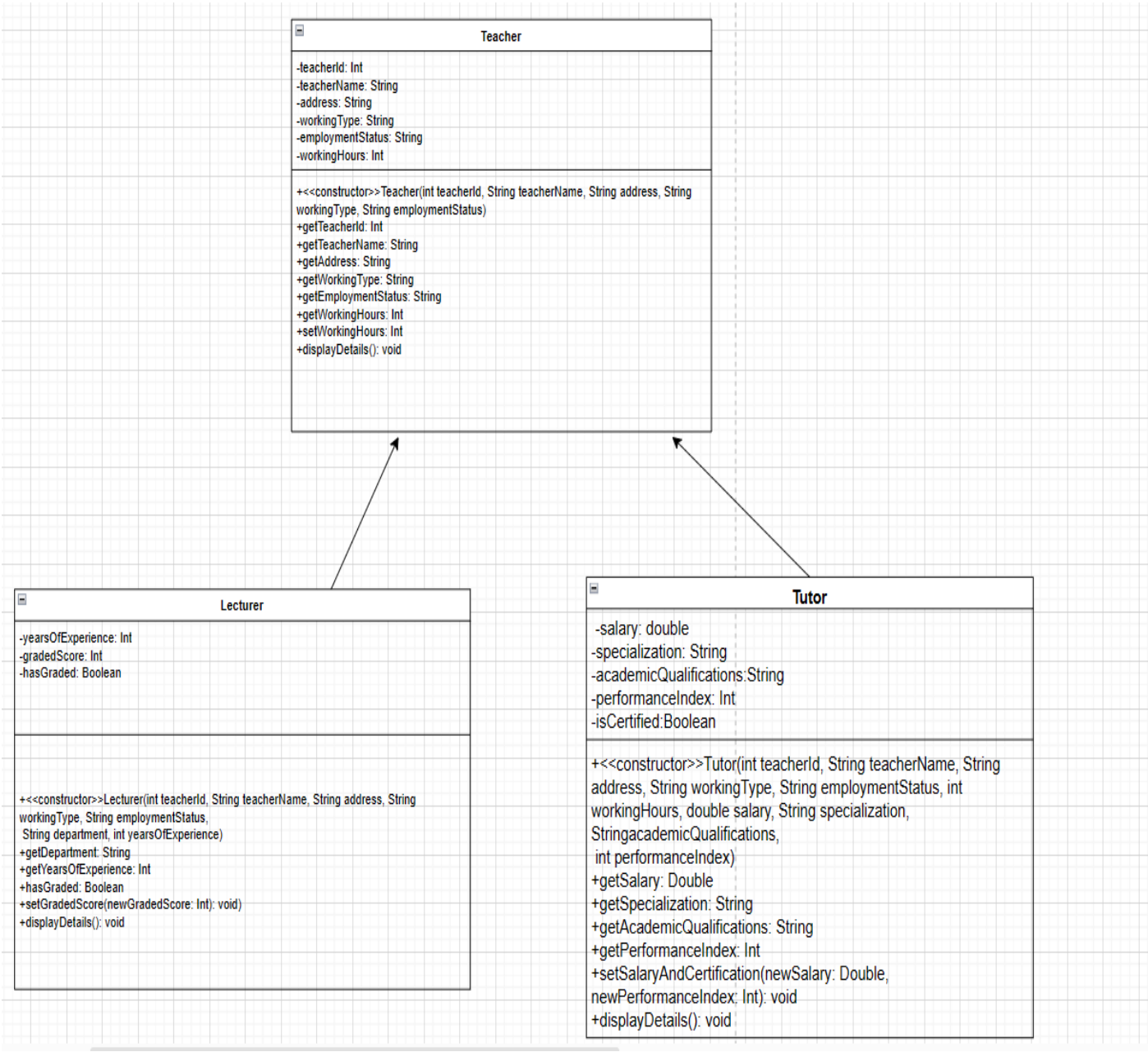


Figure2: Combined class diagram

2.3 Class diagram of lecture:

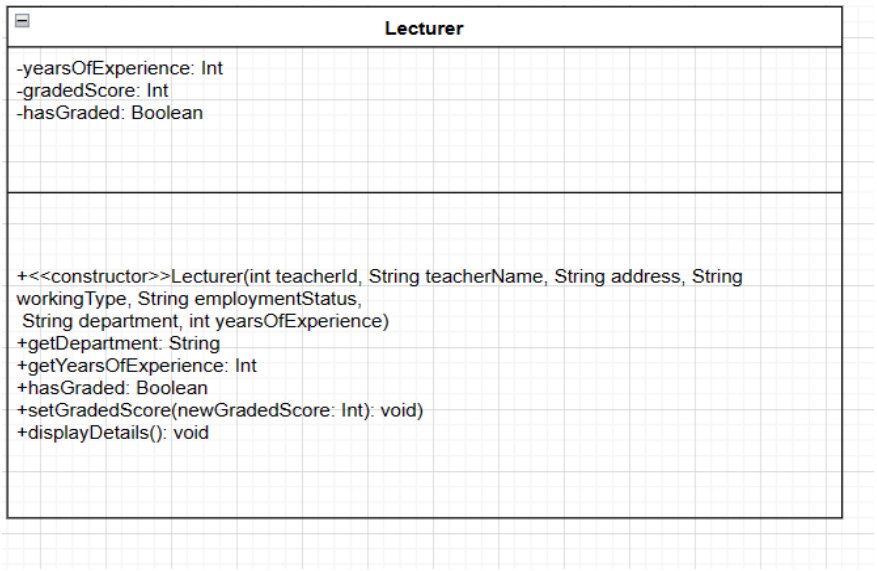


Figure 3: Class diagram of lecture

2.4 Class diagram of Tutor:

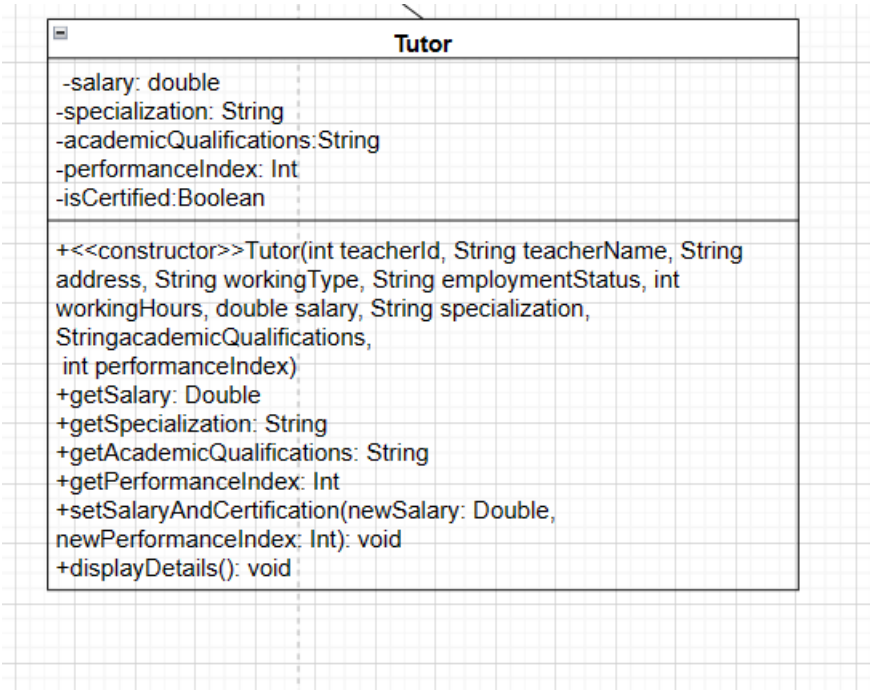


Figure 4:Class diagram of Tutor

2.5 Class diagram of Teacher:

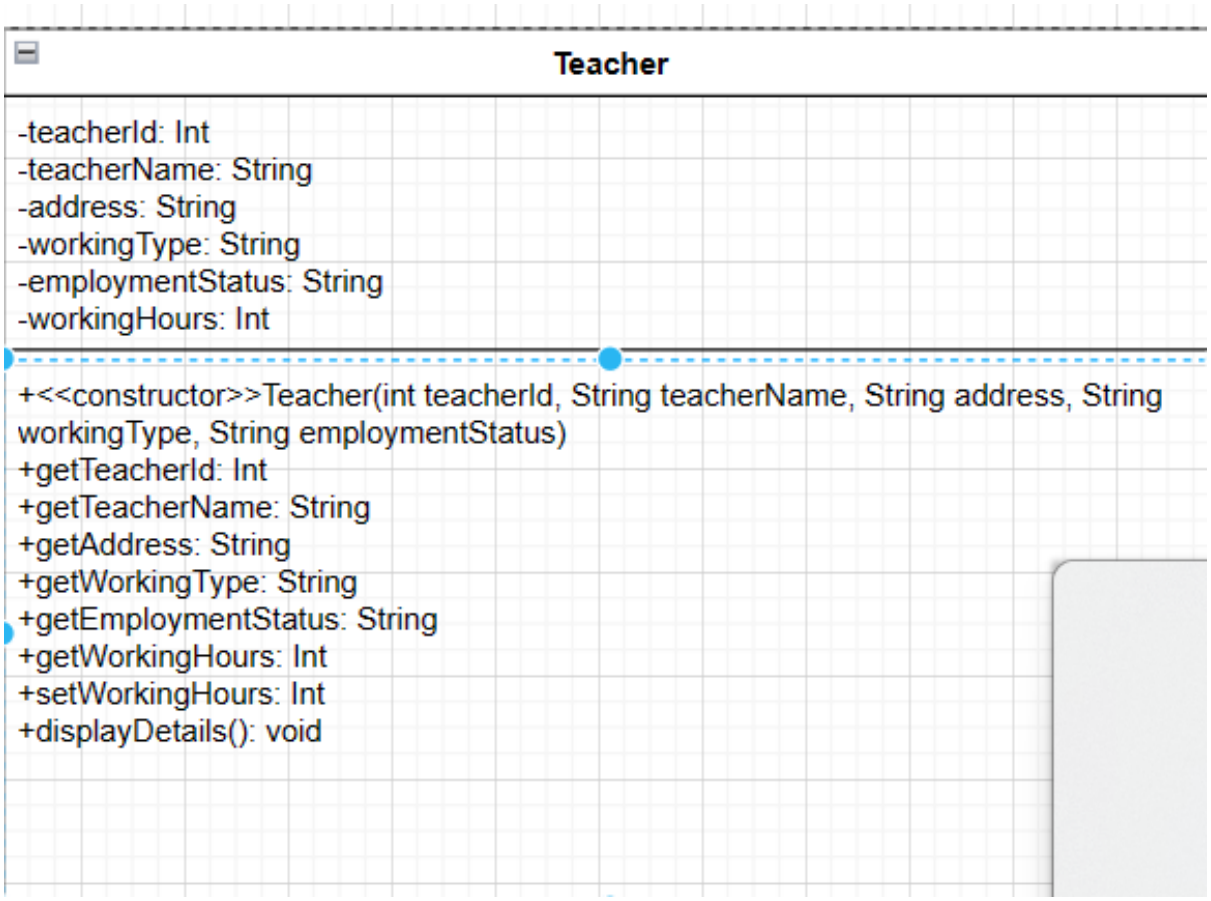


Figure 5: Class diagram of teacher

3.Pseudocode

3.1 Introduction

Pseudocode is a detailed yet readable description of what a computer program or ormatting so it can be easily understood by programmers and others involved in the development process. Pseudocode is not a programming language and cannot be compiled into an executable program. Instead, it serves as a blueprint for translating the code's logic into an actual programming language. . It uses short terms or simple English language syntaxes to write code for programs before it is actually converted into a specific programming language. This is done to identify top level flow errors, and understand the programming data flows that the final program is going to use.

In pseudocode,you don't have to thnink about the semi-colons,curlybraces,the syntax for arrow function,how to define promises and other core language principles.You just have to be able to explain what you are thinking and doing. (techtarget, n.d.)

3.2 Pseudocode for class Teacher

```
Class Teacher:
    // Attributes
    teacherId: int
    teacherName: String
    address: String
    workingType: String
    employmentStatus: String
    workingHours: int

    // Constructor
    Teacher(teacherId, teacherName, address, workingType,
employmentStatus):
        Set this.teacherId to teacherId
        Set this.teacherName to teacherName
        Set this.address to address
        Set this.workingType to workingType
        Set this.employmentStatus to employmentStatus
        Set this.workingHours to 0 // Initialize workingHours to 0

    // Accessor methods
    getTeacherId():
        Return this.teacherId

    getTeacherName():
```

```
    Return this.teacherName

getAddress():
    Return this.address

getWorkingType():
    Return this.workingType

getEmploymentStatus():
    Return this.employmentStatus

getWorkingHours():
    Return this.workingHours

// Mutator method for working hours
setWorkingHours(newWorkingHours):
    Set this.workingHours to newWorkingHours

// Display method
displayDetails():
    Print "Teacher ID: " + this.teacherId
    Print "Teacher Name: " + this.teacherName
    Print "Address: " + this.address
    Print "Working Type: " + this.workingType
    Print "Employment Status: " + this.employmentStatus

    If this.workingHours == 0:
        Print "Working Hours: Not assigned"
```

Else:

Print "Working Hours: " + this.workingHours

3.3 Pseudocode for class Lecturer:

```
Class Lecturer extends Teacher:
    // Additional attributes
    department: String
    yearsOfExperience: int
    gradedScore: int
    hasGraded: boolean

    // Constructor
    Lecturer(teacherId, teacherName, address, workingType,
employmentStatus, department, yearsOfExperience):
        Call superclass constructor with parameters (teacherId,
teacherName, address, workingType, employmentStatus)
        Set this.department to department
        Set this.yearsOfExperience to yearsOfExperience
        Set this.gradedScore to 0
        Set this.hasGraded to true

    // Accessor methods
    getDepartment():
        Return this.department

    getYearsOfExperience():
        Return this.yearsOfExperience

    getGradedScore():
```

```
        Return this.gradedScore

    hasGraded():
        Return this.hasGraded

    // Mutator method for graded score
    setGradedScore(newGradedScore):
        Set this.gradedScore to newGradedScore

    // Method to grade an assignment
    gradeAssignment(score, studentDepartment,
studentYearsOfExperience):
        If not this.hasGraded and this.yearsOfExperience >= 5 and
this.department equals studentDepartment:
            If score >= 70:
                Set this.gradedScore to score
            Else if score >= 60:
                Set this.gradedScore to 60
            Else if score >= 50:
                Set this.gradedScore to 50
            Else if score >= 40:
                Set this.gradedScore to 40
            Else:
                Set this.gradedScore to 0

            Set this.hasGraded to false
        Else:
            Print "Assignment not graded yet or conditions not met."
```

```
// Display method
displayDetails():
    Call superclass displayDetails method
    Print "Department: " + this.department
    Print "Years of Experience: " + this.yearsOfExperience
    If this.hasGraded:
        Print "Graded Score: " + this.gradedScore
    Else:
        Print "Graded Score: Not Graded yet"
```

3.4 Pseudocode for class Tutor:

```
Class Tutor extends Teacher:
    // Additional attributes
    salary: double
    specialization: String
    academicQualifications: String
    performanceIndex: int
    isCertified: boolean

    // Constructor
    Tutor(teacherId, teacherName, address, workingType, employmentStatus,
          workingHours, salary, specialization, academicQualifications, performanceIndex):
        Call superclass constructor with parameters (teacherId, teacherName, address, workingType, employmentStatus)
        SetWorkingHours(workingHours)
        Set this.salary to salary
        Set this.specialization to specialization
        Set this.academicQualifications to academicQualifications
        Set this.performanceIndex to performanceIndex
        Set this.isCertified to false

    // Accessor methods
    getSalary():
        Return this.salary
```

```
    getSpecialization():
        Return this.specialization

    getAcademicQualifications():
        Return this.academicQualifications

    getPerformanceIndex():
        Return this.performanceIndex

    isCertified():
        Return this.isCertified

    // Method to set salary and certification status
    setSalaryAndCertification(newSalary, newPerformanceIndex):
        If newPerformanceIndex > 5 and GetWorkingHours() > 20:
            Declare appraisalPercentage
            If newPerformanceIndex >= 5 and newPerformanceIndex <= 7:
                Set appraisalPercentage to 0.05
            Else If newPerformanceIndex >= 8 and newPerformanceIndex
<= 9:
                Set appraisalPercentage to 0.1
            Else:
                Set appraisalPercentage to 0.2

            Set this.salary to newSalary + (appraisalPercentage *
newSalary)
            Set this.isCertified to true
        Else:
            Print "Salary cannot be approved. Tutor is not certified yet."
```

```
// Method to remove tutor
removeTutor():
    If not this.isCertified:
        Set this.salary to 0
        Set this.specialization to ""
        Set this.academicQualifications to ""
        Set this.performanceIndex to 0
        Set this.isCertified to false

// Display method
displayDetails():
    Call superclass displayDetails method
    If this.isCertified:
        Print "Salary: " + this.salary
        Print "Specialization: " + this.specialization
        Print "Academic Qualifications: " + this.academicQualifications
        Print "Performance Index: " + this.performanceIndex
```

4.Method description of subclass lecturer:

4.1 Method dewscription of class student

Method	Description
getDepartment()	This method returns the department name as string data type.
getYearsOfExperience()	This method returns the valueOf years of expriences as an int data type.
getGradedScore()	This method returns the value of Grade score as int data type.
hasGraded()	This method returns thevalue of grade as Boolean data type.
setGradedScore()	This method assign the paratemeter value to graded score.
displayDetails()	This method is used to display the values of attributes of the class.

Table1: Method description of class Lecturer

4.2 Method description of subclass Tutor:

Method	Description
getSalary()	This method returns the value of salary as a double data type.
getSpecilization()	This method returns the value of specilization as a String data type.
getAcademicQualification()	This method returns the value of academicqualification as a String data type.
getPerformanceIndex()	This method returns the value of performanceindex as an int data type.
isCertified()	This method returns the value of certified as a boolean data type.
setSalaryAndCertification(double newSalary,intPerformanceIndex)	This method assigns the paratemeter value to salary and certification status.
displayDetails()	This method is used to display the values of attributes of the class.

Table2: Method description of class tutor.

4.3 Method description of class Teacher:

getTeacherId()	This method returns the value of teacherid as a Int data type.
getTeacherName()	This method returns the value of teachername as a String data type.
getAddress()	This method returns the value of address as a String data type.
getWorkingType()	This method returns the value of workingtype as a String data type.
getEmploymentStatus()	This method returns the value of employmentstatus as a String data type.
getWorkingHours()	This method returns the value of workinghours as a int data type.
setWorkingHours(int newWorkingHours)	This method assigns the paratemeter value toworkinghours.
displayDetails()	This method is used to display the values of attributes of the class.

Table3:Method description of class Teacher.

5.Testing

5.1 Test 1

Objective	Inspect the lecturer class,grade the assigment,and re-inspect the lecture class.
Action	• The lecture class is called with following arguments: teacherId=45614 teacherName: Anmol Sapkota workingType: Fulltime employmentStatus:30 • Inspection of lecture class. • Granding the assigment . gradedScore: 75 studentDepartment: Networking studentYearOfExperience:5 • Re-inspect Lecturer class.
Expected Result:	The grade score would change to assigned number.

Actual Result:	The grade score was changed to assigned value.
Conclusion:	The test is successful.

Output result of test1:

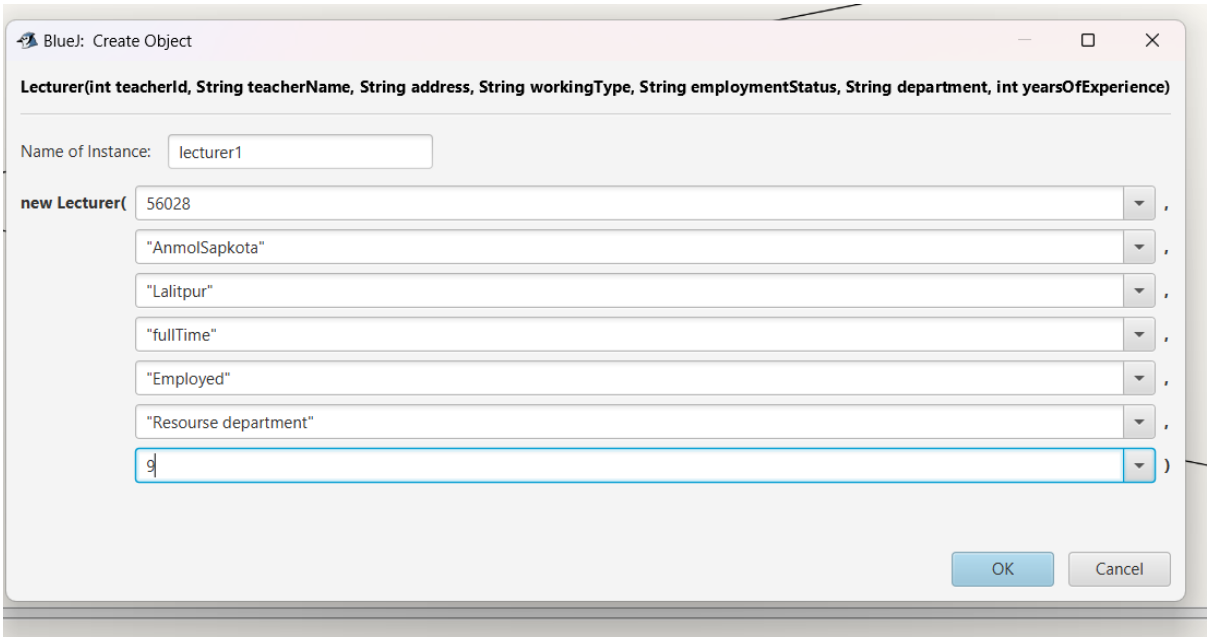


Figure 7 :Screenshot of assiging data in Lecturer class.

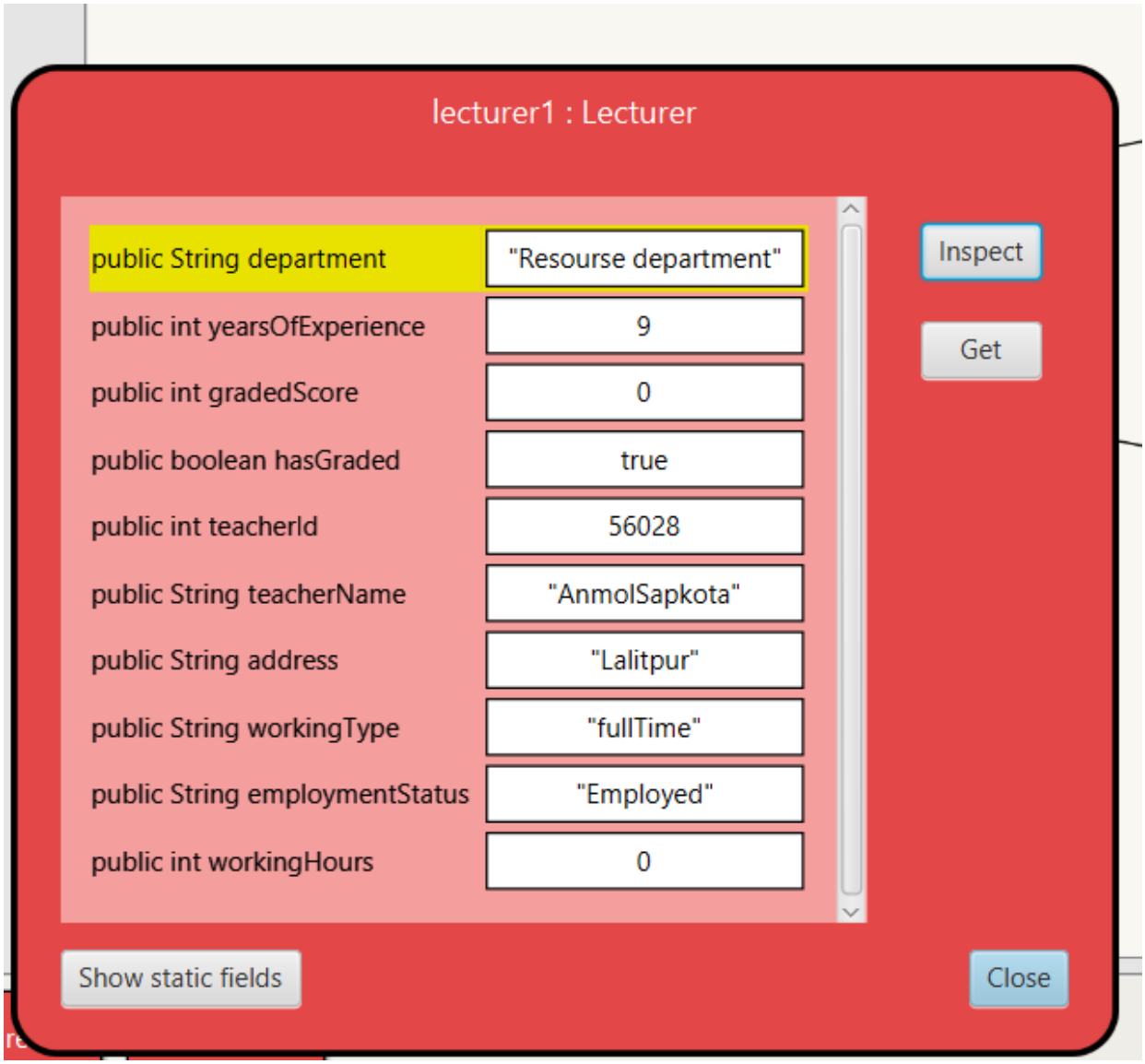


Figure 7 : Screenshot of Inspection of Lecturer class'

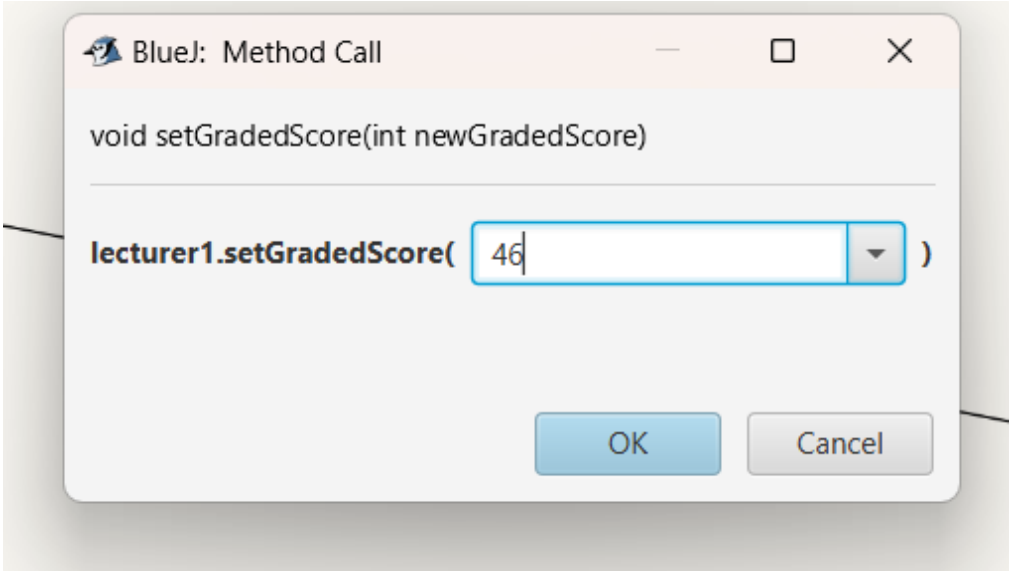


Figure 8: Screenshot of assigning the data of gradedScore for re-inspection.

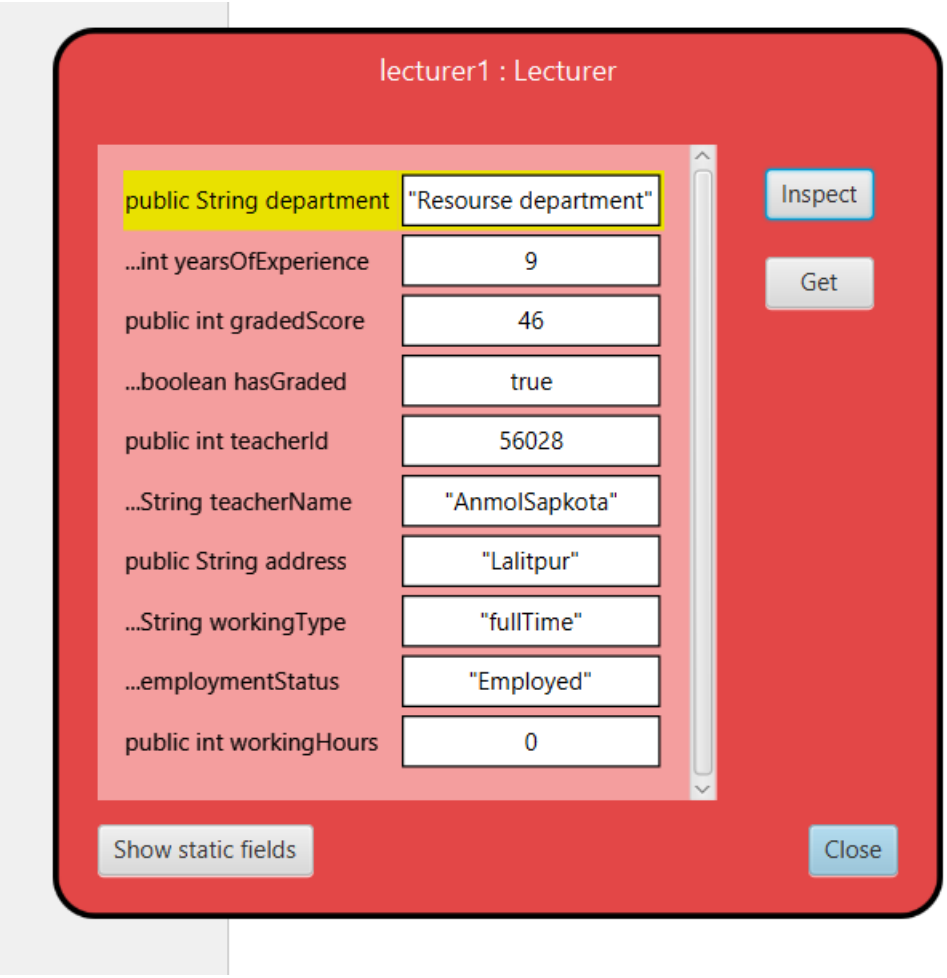


Fig 9: Screenshot of re-inspection of lecturer class.

Test 2.

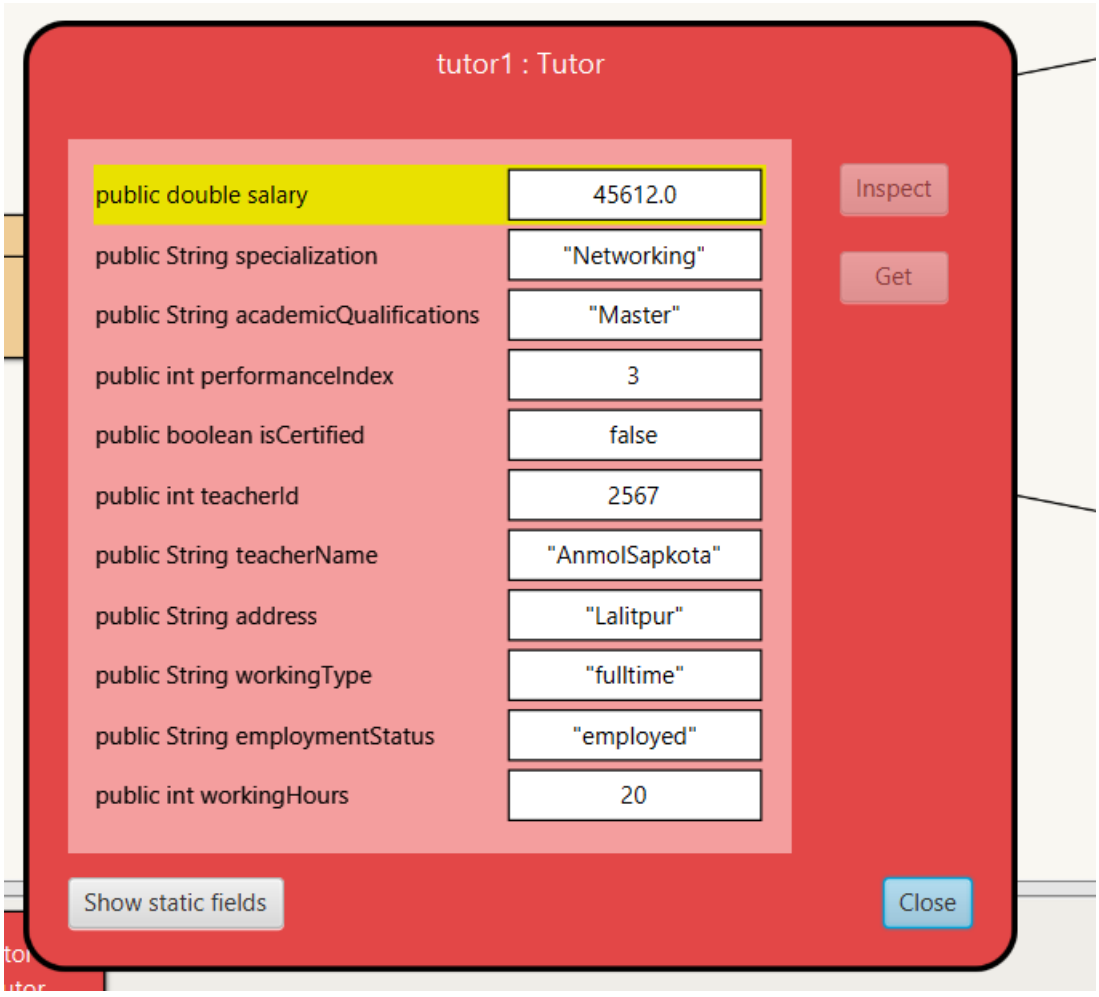


Figure 10: Screenshop of inspection class tutor

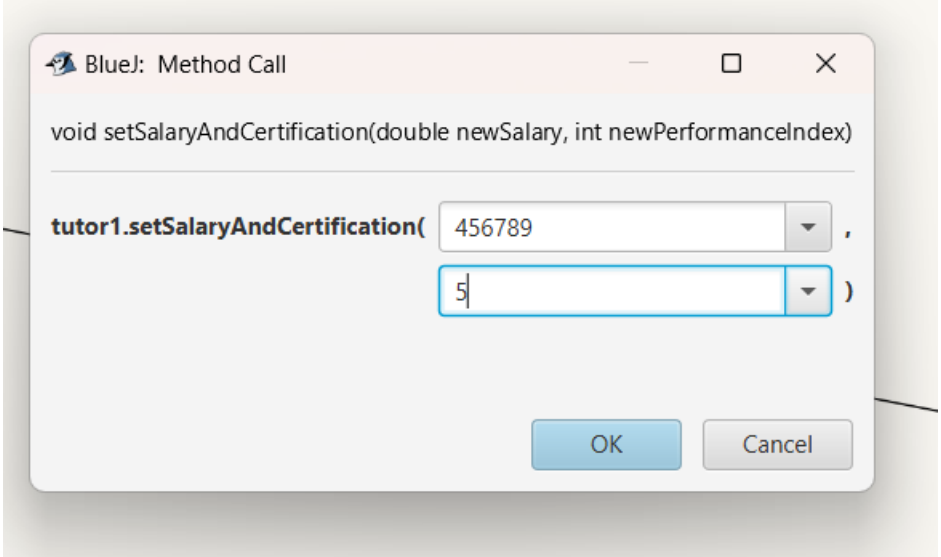


Fig 11. Screenshot of assiging the value of salary and performanceindex.

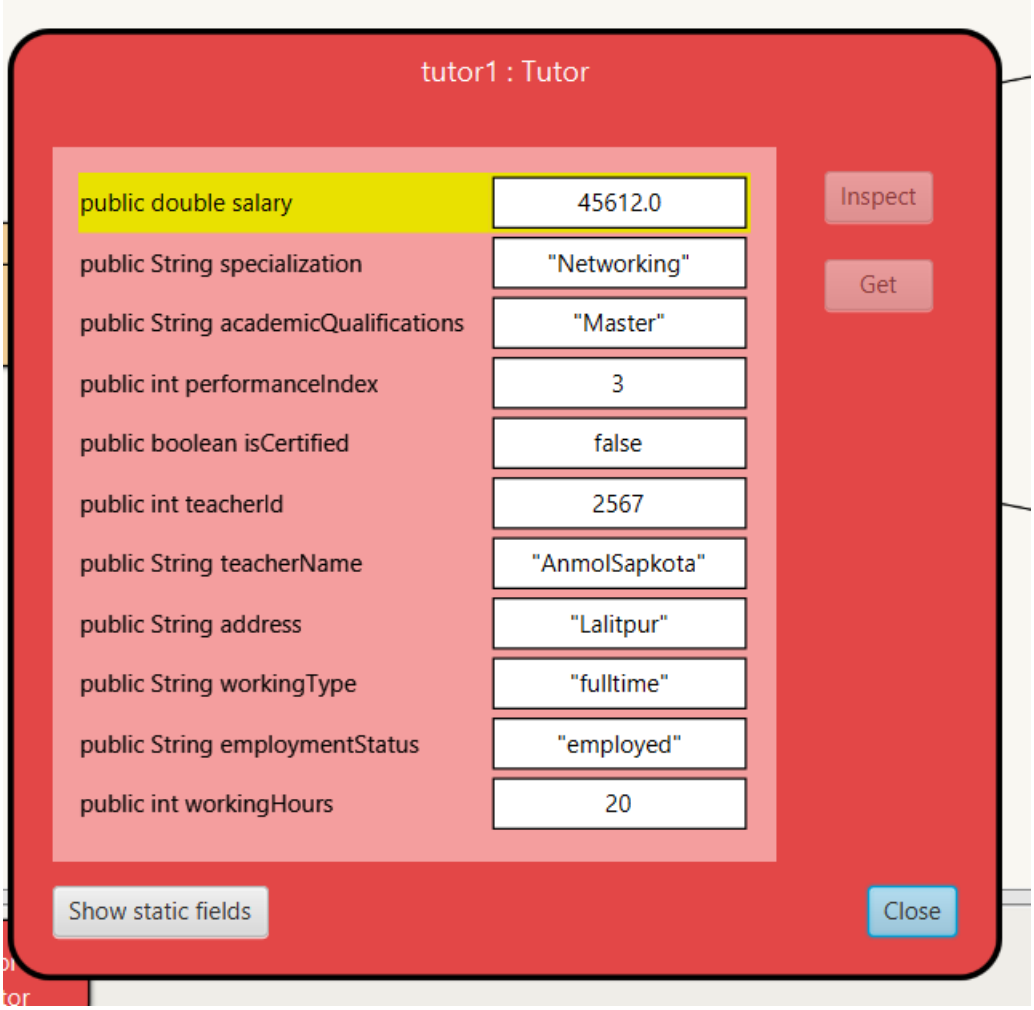


Figure 12:Screenshot Of reinspection of tutor class.

Testing 3:

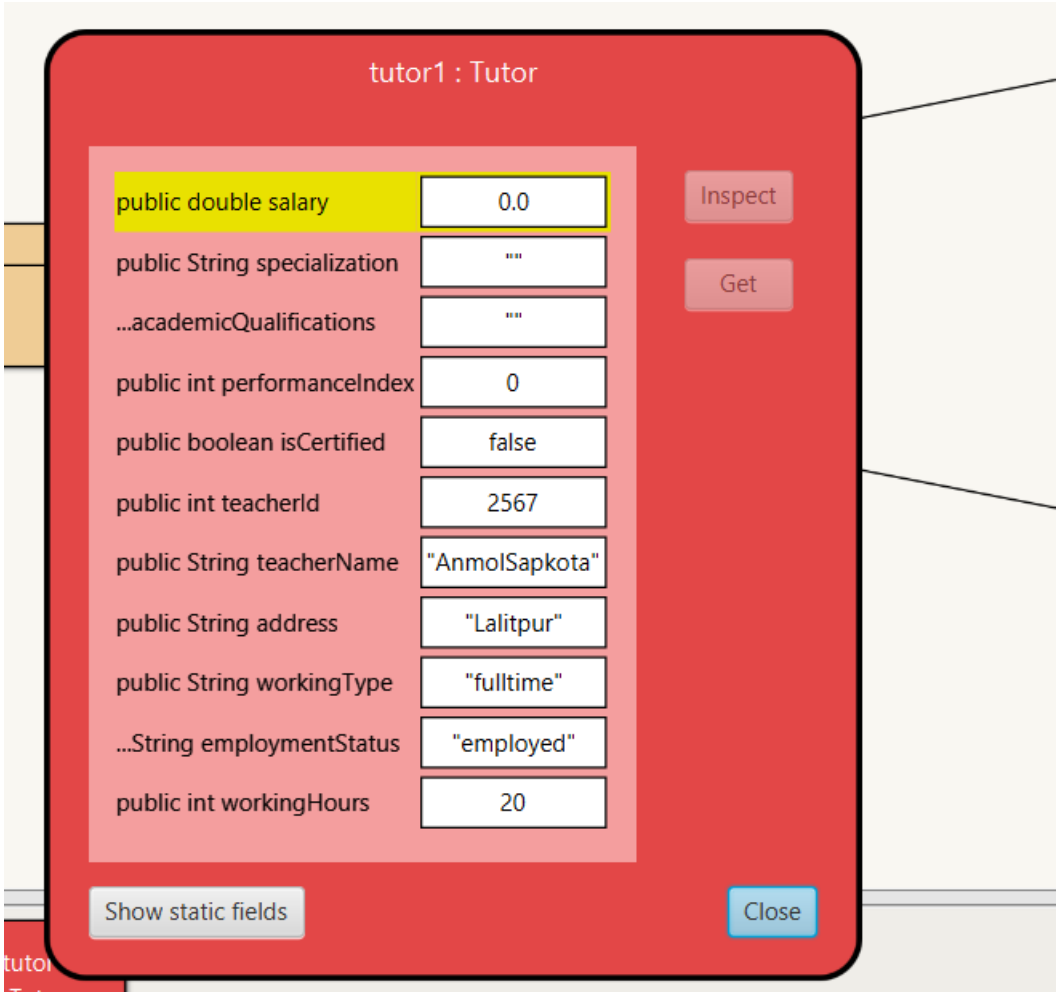


Figure 13 :Screenshot of inspect after remove of tutor.

Test 4.

```
Teacher ID: 56123
Teacher Name: Anmol Sapkota
Address: lalitpur
Working Type: full time
Employment Status: employed
Working Hours: Not assigned
Department: cybersecurity
Years of Experience: 9
Graded Score: 46
Salary cannot be approved. Tutor is not certified yet.
Teacher ID: 564123
Teacher Name: Anmol Sapkota
Address: Lalitpur
Working Type: full time
Employment Status: Employed
Working Hours: 20
```

Figure 14:ScreenShot of details of lecture class and the tutor class.

6.Error Detection and Correction

An error is a mistake made by programmers, whether on purpose or unintentionally, that causes difficulties with the program. The technique of recognizing or discovering errors is known as error detection. The process of fixing an error with the right code or solution is known as error correction.

I got to encounter syntax error, semantic error and logical error during the making of the coursework.

The detection and correction of each error is shown below with proofs.

6.1 Syntax error

A syntax error in computer science is an error in the syntax of a coding or programming language, entered by a programmer. Syntax errors are caught by a software program called a compiler, and the programmer must fix them before the program is compiled and then run. (netnut.io, n.d.)

Error Detection: Semi colon(;)

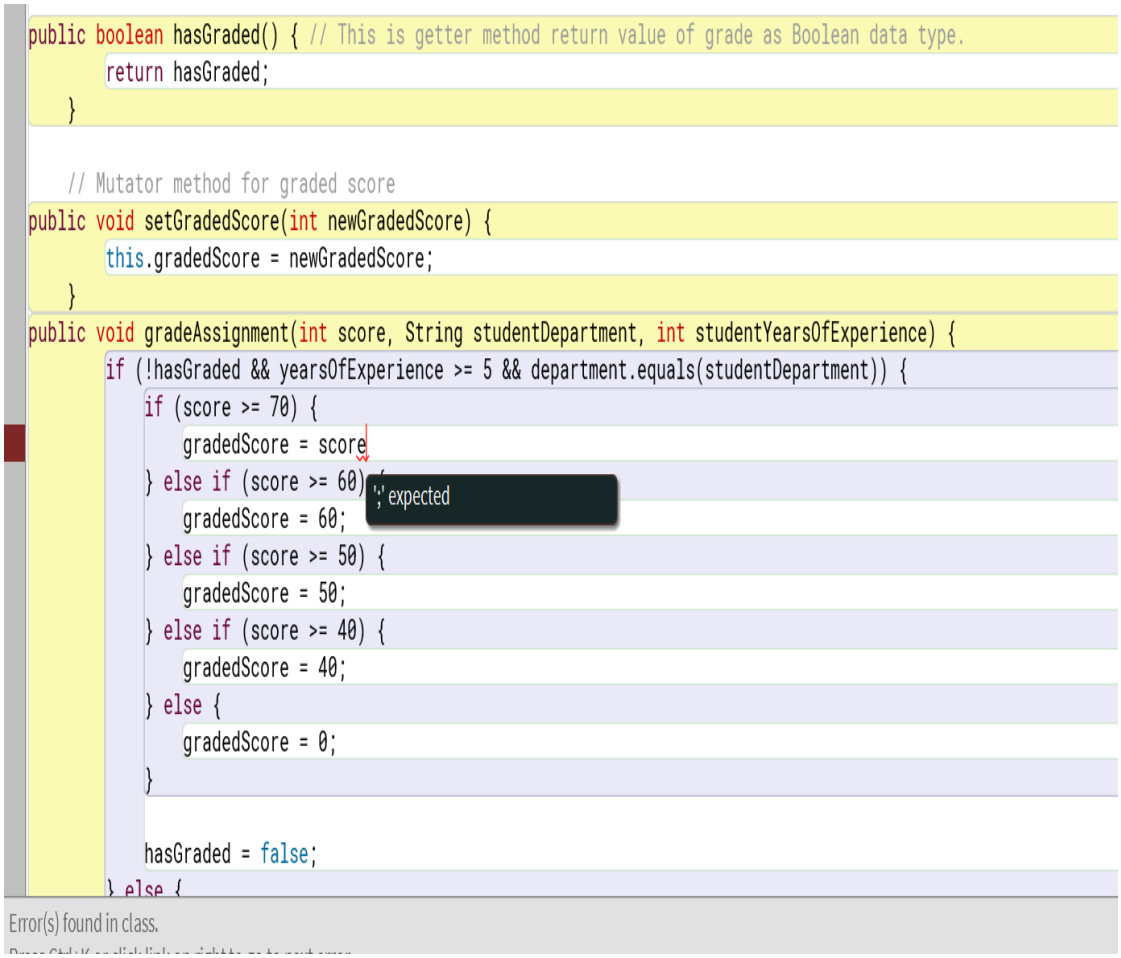


Figure 15: Detection of Syntax error.

Error Correction: A semi colon was missing so I added it and corrected the error.

```
public void setGradedScore(int newGradedScore) {
        this.gradedScore = newGradedScore;
    }
public void gradeAssignment(int score, String studentDepartment, int studentYearsOfExperience) {
        if (!hasGraded && yearsOfExperience >= 5 && department.equals(studentDepartment)) {
            if (score >= 70) {
                gradedScore = score;
            } else if (score >= 60) {
                gradedScore = 60;
            } else if (score >= 50) {
                gradedScore = 50;
            } else if (score >= 40) {
                gradedScore = 40;
            } else {
                gradedScore = 0;
            }

            hasGraded = false;
        } else {
lass compiled - no syntax errors
```

Figure 16: Screenshot of correction of syntax error.

Error Detection: Incompatible data type

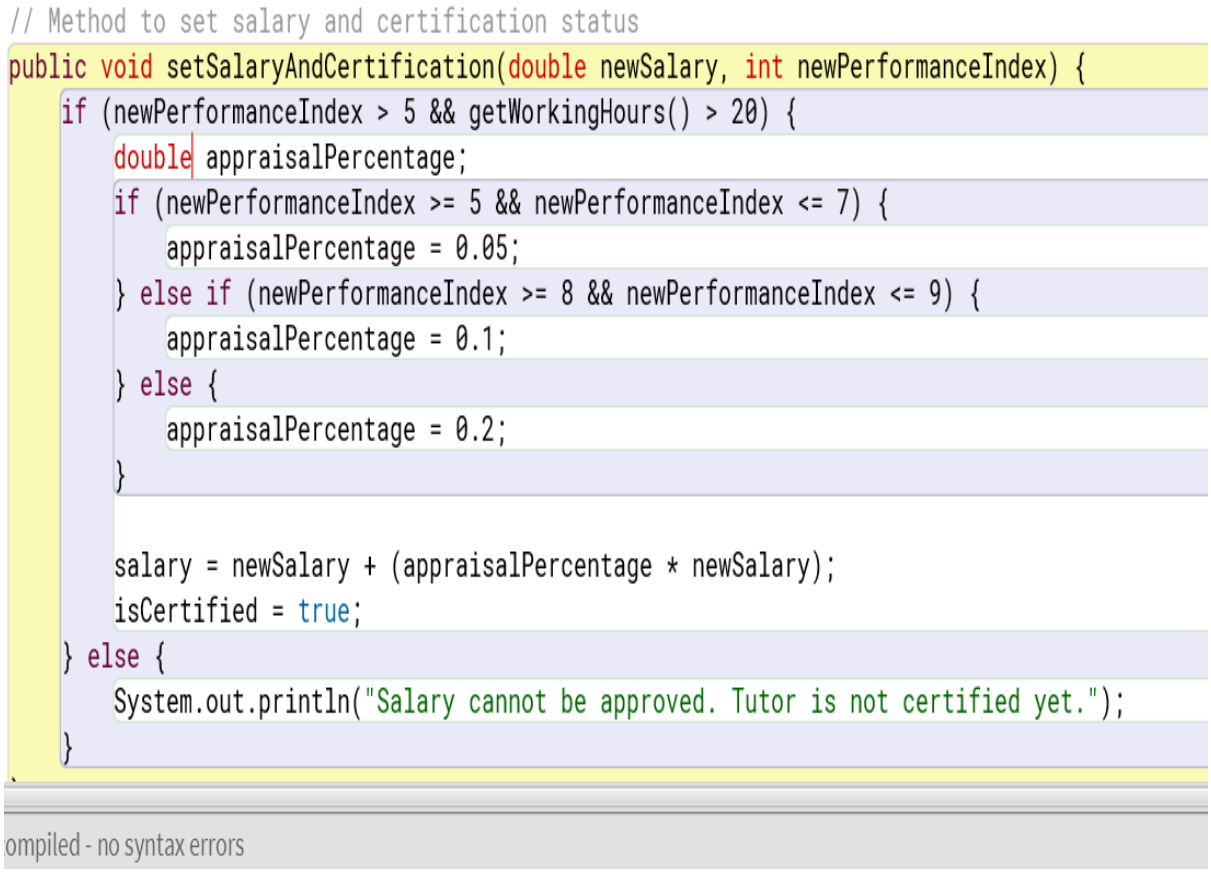
```
// Method to set salary and certification status
public void setSalaryAndCertification(double newSalary, int newPerformanceIndex) {
    if (newPerformanceIndex > 5 && getWorkingHours() > 20) {
        int appraisalPercentage;
        if (newPerformanceIndex >= 5 && newPerformanceIndex <= 7) {
            appraisalPercentage = 0.05;
        } else if (newPerformanceInd
            appraisalPercentage = 0.
        } else {
            appraisalPercentage = 0.2;
        }

        salary = newSalary + (appraisalPercentage * newSalary);
        isCertified = true;
```

incompatible types: possible lossy conversion from double to int

Figure17: detection of syntax error.

Error Correction:

A variable whose data type was supposed to be double was declared Int. So,I declared it to doble.



```
// Method to set salary and certification status
public void setSalaryAndCertification(double newSalary, int newPerformanceIndex) {
    if (newPerformanceIndex > 5 && getWorkingHours() > 20) {
        double appraisalPercentage;
        if (newPerformanceIndex >= 5 && newPerformanceIndex <= 7) {
            appraisalPercentage = 0.05;
        } else if (newPerformanceIndex >= 8 && newPerformanceIndex <= 9) {
            appraisalPercentage = 0.1;
        } else {
            appraisalPercentage = 0.2;
        }

        salary = newSalary + (appraisalPercentage * newSalary);
        isCertified = true;
    } else {
        System.out.println("Salary cannot be approved. Tutor is not certified yet.");
    }
```


Figure 18: Screenshot of syntax error correction.

6.2 Semantic error

Unlike a syntax error,a semantic error is all about meaning. If a program contains

This kids of error. It will sucessfully run, but won't output the correct result. (netnut.io, n.d.)

Error detection: Not equal to sign(!=) was found in place of to Comparison sign(==)

```
// Mutator method for working hours
public void setWorkingHours(int newWorkingHours) {
    this.workingHours = newWorkingHours;
}

/*
 * This is display method which displays the values of the instance variable, display suitable mes
 */
public void displayDetails() {
    System.out.println("Teacher ID: " + teacherId);
    System.out.println("Teacher Name: " + teacherName);
    System.out.println("Address: " + address);
    System.out.println("Working Type: " + workingType);
    System.out.println("Employment Status: " + employmentStatus);

    if (workingHours != 0) {
        System.out.println("Working Hours: Not assigned");
    } else {
        System.out.println("Working Hours: " + workingHours);
    }
}
```

Figure 19: detection of semantic error.

Here the output of this program is as expected.It doesnot displays the details of the workinghours.

```
Teacher ID: 45621
Teacher Name: roshan
Address: lalitpur
Working Type: fulltime
Employment Status: employed
Working Hours: Not assigned
```

FIGURE 20 : Screenshot of out put of semantic error.

Error correction:Comparison sign(==)was asignrd in place of not equals sign(!=).

```
public void displayDetails() {
    System.out.println("Teacher ID: " + teacherId);
    System.out.println("Teacher Name: " + teacherName);
    System.out.println("Address: " + address);
    System.out.println("Working Type: " + workingType);
    System.out.println("Employment Status: " + employmentStatus);

    if (workingHours == 0) {
        System.out.println("Working Hours: Not assigned");
    } else {
        System.out.println("Working Hours: " + workingHours);
    }
}
```

Figure 21: ScreenShot of error correction.

```
Teacher ID: 6564165
Teacher Name: Anmol Sapkota
Address: Lalitpur
Working Type: fulltime
Employment Status: employed
Working Hours: 25
```

Figure 22: ScreenShot of output.

7.Conclusion

After finishing this coursework, there are few things I'd like to discuss about. Because this was a major individual programming course with a 30% weightage assigned to us. We had to use BlueJ, Ms-Word and draw.io for the completion of the coursework and during the coursework, I got to discover so many tools that I was unknown of and that was the best part because discovering new things has always been my pleasure. Most of the work in this coursework is done in BlueJ, and writing program code was a challenging task. I haven't had any encounter with programming works set by teachers previously. So, it was challenging for me, but with the help from teachers and the reference to the proper guidelines provided to us for the coursework, I was able to successfully complete the assignment. I began working on reports right away as I finished writing the code. To be honest, reporting was much easier, but it was additionally far more exhausting and time demanding. Proof of work done, i.e. appropriate screenshots of what was done, were required to be included, which was a messy process for me because I had to recognize and locate the right one from many different screenshots.

I get to learn a lot about writing code, running and executing it, and debugging any form of error that comes up and causes an obstacle to the program. The programming portion taught me that even a comma (,) or curly bracket ({ }) or semi colon(;) could hinder the execution of the program. I got much more familiar with the BlueJ than I had been through in my class sessions. The reporting section made me learn how to do any job calmly and correctly, and it helped me realize that no matter how hard the given task is, it can be completed if we come up with hardworking and patience.

In a nutshell, this programming coursework is finished, and I feel a little more confident with java programming because I learned a lot from it. I now have a solid foundation in Java programming up to some level that will help me with my upcoming works. I've learned a lot from this coursework, and I can't wait to use the skills and knowledge I've acquired to difficult programming problems.

7.Refernces:

1. javaT. (n.d.). JavaTpoint. Retrieved from ttps://www.javatpoint.com/java-tutorial

2.ComputerHope. (n.d.). Retrieved from https://www.computerhope.com/jargon/d/drawio.html

3.Sheldon, R. (2023, 6 27). pseudocode. Retrieved from https://www.techtarget.com/whatis/definition/pseudocode#:~:text=Pseudocode%20is%20a%20detailed%20yet,involved%20in%20the%20development%20process.
Times, T. E. (2024, 1 24). pseudocode. Retrieved from https://economictimes.indiatimes.com/definition/pseudocode.

4. w3school. (n.d.). Retrieved from w3school: https://www.w3schools.com/java/default.asp

5.Visualparadigrm. (n.d.). Retrieved from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/

9.Appendix
9.1 Code of Teacher

```
class Teacher {
    // Attributes
    public int teacherId;// a variable for the teacher id
    public String teacherName;// a variable for the teacher
name
    public String address;// a variable for the address
    public String workingType;// a variable for the
workingtype
    public String employmentStatus;// a variable for the
employment status
    public int workingHours;// a variable for the working
hours

    // Constructor that takes one int parameters (i.e
teacher id), four string parameter(
    public Teacher(int teacherId, String teacherName,
String address, String workingType, String
employmentStatus) {
        this.teacherId = teacherId;
        this.teacherName = teacherName;
        this.address = address;
        this.workingType = workingType;
        this.employmentStatus = employmentStatus;
    }

    // Accessor methods
    public int getTeacherId() { //This is getter method
which returns value of TeacherId as int data type.
        return teacherId;
    }
```

```
    public String getTeacherName() { //This is getter method which returns value of TeacherName as String data type.
        return teacherName;
    }

    public String getAddress() { //This is getter method which returns value of address as String data type.
        return address;
    }

    public String getWorkingType() { //This is getter method which returns value of working type as String data type.
        return workingType;
    }

    public String getEmploymentStatus() { //This is getter method which returns value of employment status as String data type.
        return employmentStatus;
    }

    public int getWorkingHours() {  //This is getter method which returns value of working hours as int data type.
        return workingHours;
    }

    // Mutator method for working hours
    public void setWorkingHours(int newWorkingHours) {
        this.workingHours = newWorkingHours;
    }
```

```
    /*
     * This is display method which displays the values of
the instance variable, display suitable message if
teachername is empty.
     */
    public void displayDetails() {
        System.out.println("Teacher ID: " + teacherId);
        System.out.println("Teacher Name: " +
teacherName);
        System.out.println("Address: " + address);
        System.out.println("Working Type: " +
workingType);
        System.out.println("Employment Status: " +
employmentStatus);

        if (workingHours == 0) {
            System.out.println("Working Hours: Not
assigned");
        } else {
            System.out.println("Working Hours: " +
workingHours);
        }
    }

}
```

9.2 Code of Lecturer

```
public class Lecturer extends Teacher {
    // Additional attributes
public String department;
public int yearsOfExperience;
public int gradedScore;
public boolean hasGraded;
public Lecturer(int teacherId, String teacherName, String
address, String workingType, String employmentStatus,
                    String department, int yearsOfExperience) {
        super(teacherId, teacherName, address,
workingType, employmentStatus); //super class setter is
called.
        this.department = department;
        this.yearsOfExperience = yearsOfExperience;
        this.gradedScore = 0;
        this.hasGraded = true;
    }

    // Accessor methods
public String getDepartment() { //This is getter method
returns value of department name as string data type.
        return department;
    }
public int getYearsOfExperience() {//This getis getter
method returns value of year of exprinces as int data
type.
        return yearsOfExperience;
    }

public int getGradedScore() { //This is getter method
returns value of Grade score as int data type.
```

```
        return gradedScore;
    }

public boolean hasGraded() { // This is getter method
return value of grade as Boolean data type.
        return hasGraded;
    }

    // Mutator method for graded score
public void setGradedScore(int newGradedScore) {
        this.gradedScore = newGradedScore;
    }
public void gradeAssignment(int score, String
studentDepartment, int studentYearsOfExperience) {
        if (!hasGraded && yearsOfExperience >= 5 &&
department.equals(studentDepartment)) {
            if (score >= 70) {
                gradedScore = score;
            } else if (score >= 60) {
                gradedScore = 60;
            } else if (score >= 50) {
                gradedScore = 50;
            } else if (score >= 40) {
                gradedScore = 40;
            } else {
                gradedScore = 0;
            }

            hasGraded = false;
        } else {
            System.out.println("Assignment not graded yet
or conditions not met.");
        }
```

```
    }

    // Display method
public void displayDetails() {
        super.displayDetails();//super class display method
was called(i.e lecturer display method)
        System.out.println("Department: " + department);
        System.out.println("Years of Experience: " +
yearsOfExperience);
        if (hasGraded) {
            System.out.println("Graded Score: " +
gradedScore);
        } else {
            System.out.println("Graded Score: Not graded
yet");
        }
    }

}
```

9.3 Code of Tutor

```
public class Tutor extends Teacher {
    // Additional attributes
    public double salary;
    public String specialization;
    public String academicQualifications;
    public int performanceIndex;
    public boolean isCertified;

    // Constructor
      /*
 * Constructor that takes three int parameters(i.e teacher
id, working hours and performance index), six String
parameter(i.e teacher name, address, working type,
employmentStatus,specialization,
academicQualifications  ),
 * and one double parameter(i.e salary). It also call
parent class. It also  initilize the value of variale.
 *
 */
    public Tutor(int teacherId, String teacherName, String
address, String workingType, String employmentStatus,
                int workingHours, double salary, String
specialization, String academicQualifications,
                int performanceIndex) {
        super(teacherId, teacherName, address,
workingType, employmentStatus);
        this.setWorkingHours(workingHours);
        this.salary = salary;
        this.specialization = specialization;
        this.academicQualifications =
academicQualifications;
```

```
        this.performanceIndex = performanceIndex;
        this.isCertified = false;
    }

    // Accessor methods
    public double getSalary() { // This is getter method
which returns value salary as a double data type
        return salary;
    }

    public String getSpecialization() {  // This is getter
method which returns value Specialization as a string
data type
        return specialization;
    }

    public String getAcademicQualifications() { // This is
getter method which returns value
AcademicQualifications as a string data type
        return academicQualifications;
    }

    public int getPerformanceIndex() { // This is getter
method which returns value PerformanceIndex as a int
data type
        return performanceIndex;
    }

    public boolean isCertified() {  // This is getter method
which returns value certified as a bollean data type
        return isCertified;
    }
```

```
    // Method to set salary and certification status
    public void setSalaryAndCertification(double
newSalary, int newPerformanceIndex) {
        if (newPerformanceIndex > 5 &&
getWorkingHours() > 20) {
            double appraisalPercentage;
            if (newPerformanceIndex >= 5 &&
newPerformanceIndex <= 7) {
                appraisalPercentage = 0.05;
            } else if (newPerformanceIndex >= 8 &&
newPerformanceIndex <= 9) {
                appraisalPercentage = 0.1;
            } else {
                appraisalPercentage = 0.2;
            }

            salary = newSalary + (appraisalPercentage *
newSalary);
            isCertified = true;
        } else {
            System.out.println("Salary cannot be approved.
Tutor is not certified yet.");
        }
    }

    // Method to remove tutor
    public void removeTutor() {
        if (!isCertified) {
            salary = 0;
            specialization = "";
            academicQualifications = "";
            performanceIndex = 0;
            isCertified = false;
```

```
        }
    }

    // Display method
    /*
   * This is display method which details according to the
value of isGranted.
   */
    public void displayDetails() {
        super.displayDetails(); // Call the displayDetails
method in the parent class

        if (isCertified) { //super class display method
            System.out.println("Salary: " + salary);
            System.out.println("Specialization: " +
specialization);
            System.out.println("Academic Qualifications: " +
academicQualifications);
            System.out.println("Performance Index: " +
performanceIndex);
        }
    }
}
```