CS4001NI Programming

30% Individual Coursework

2023-24 Autumn

Student Name: Nikita Bhandari

London Met ID: 23047392

College ID: NP01NT4A230092

Group: N5

Assignment Due Date: Friday, May 10, 2024

Assignment Submission Date: Friday, May 10, 2024

Table of Contents

Table of Figure

Table of Figure

1. Introduction

Figure 1: Java

Java is a high-level class-based and object-oriented programming language. Java was released in 1995 by James Gosling at Sun Microsystem. Security, portable, robustness, reliability, platform independence, and simplicity make Java stand out and popular for learners and enterprises. Code of Java can be compiled and run on any or the same operating system. Java is used for the development of web, desktop, and mobile applications, the scalation of cloud applications. Java helps users to write apps and games on Android. Therefore, Java is a very simple and popular programming language.

1.1 About the Coursework

This is the coursework of the module Programming which is all about creating a new class called “TeacherGUI”. According to the requirement, a new class is requested to be added to the previous part of the coursework to make a GUI (Graphical User Interface) for the system where teachers' details are stored, and those details are stored in an Array List. The main method will be made in the class. Testing will be done using a command prompt to ensure the unsalability and functionality. The main goal of the coursework is to integrate GUI where one can integrate with elements such as buttons, text fields, labels, and panels and get information about teachers stored in Array List. This coursework is an individual coursework, and it carries 30% of the module. A well-structured report must be made.

1.2 Tools used

During the completion of the project various tools were used and those tools improved the overall quality of the project. They were instrumental for the smooth development as well as documentation part. The used toots are mentioned below:

1.3 Blue J

Blue J, a Java Integrated Development language was developed in 1999 by Michael Kolling and John Rosenberg. Blue is user friendly software that is used for compiling, writing, and debugging of Java code. Blue J is simple, portable, innovative and allows users to interact graphically with object. It offers advanced features and is designed to learn and teach Java Programming in a simpler way and more efficient way for beginners.

Figure 2: Blue J

1.4 MS- Word

Microsoft Word (MS- Word) is a very known processor which was developed in 1983 published by Microsoft. It allows user to create high-content documentations, cv, letter and reports. It has many features like checking grammatical errors, spelling checking, text and font formatting. Microsoft is a user-friendly tool that allows user to make their documents spotless and error free. Therefore, they are very versatile and are used by any age groups for personal, educational to industrial purpose.

Figure 3: MS-Word

1.5 Draw.io

Draw.io is a free-to-use online diagramming tool that allows its user to create flowcharts, class diagrams, mind maps, UML and education related diagrams. Draw.io acts like a digital whiteboard where one can visualize their idea and adding it in the report makes it interactive. It is good option for professionals, students which has gain popularity among the users.

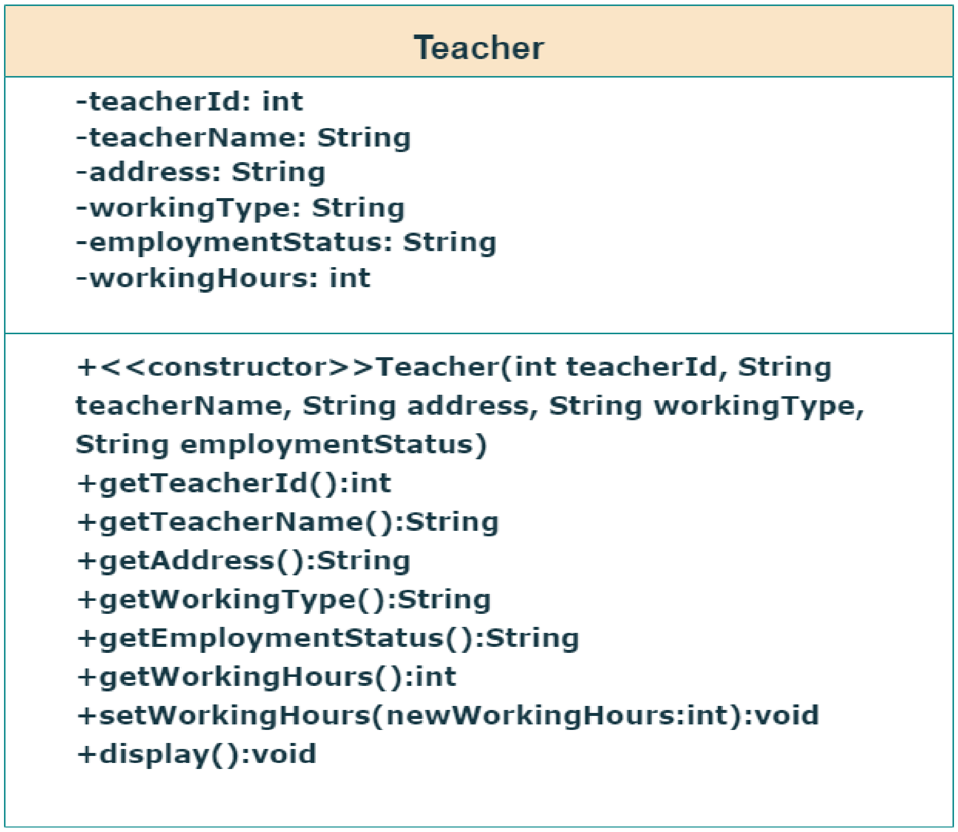
Figure 4: Draw.io

2. Class Diagram

2.1 Introduction

Class diagram is a most popular UML (Unified Modelling Language) that visualizes, documents, describes different aspects of the system. It analyses the static view of an application and attributes of the classes reducing the time of maintenance as before coding it shows an overview of how the application is structured.

2.2 Teacher Class



```
Teacher
-teacherId: int
-teacherName: String
-address: String
-workingType: String
-employmentStatus: String
-workingHours: int
+<<constructor>>Teacher(int teacherId, String
teacherName, String address, String workingType,
String employmentStatus)
+getTeacherId():int
+getTeacherName():String
+getAddress():String
+getWorkingType():String
+getEmploymentStatus():String
+getWorkingHours():int
+setWorkingHours(newWorkingHours:int):void
+display():void
```

Figure 5: Class Diagram of Teacher Class

2.3 Lecture Class

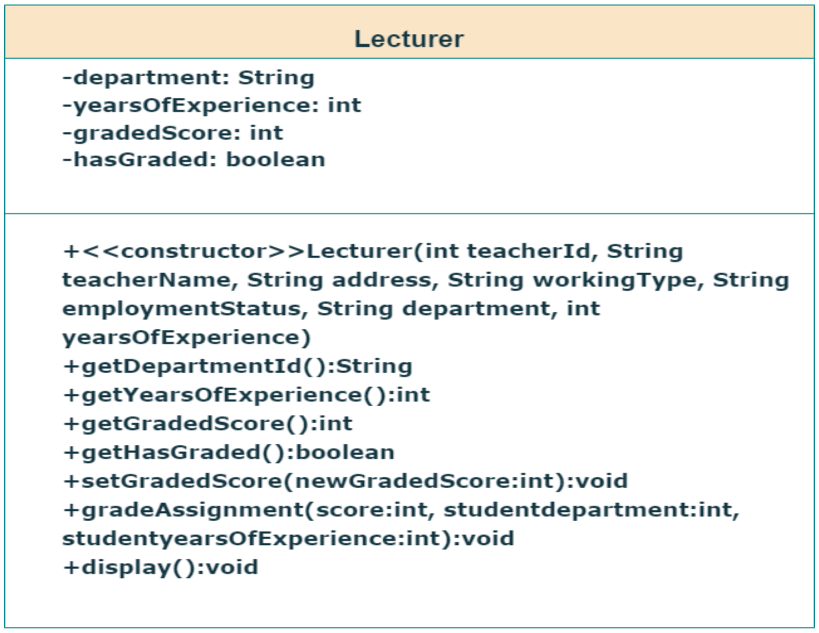

Lecturer
-department: String
-yearsOfExperience: int
-gradedScore: int
-hasGraded: boolean
+<<constructor>>Lecturer(int teacherId, String teacherName, String address, String workingType, String employmentStatus, String department, int yearsOfExperience)
+getDepartmentId():String
+getYearsOfExperience():int
+getGradedScore():int
+getHasGraded():boolean
+setGradedScore(newGradedScore:int):void
+gradeAssignment(score:int, studentdepartment:int, studentyearsOfExperience:int):void
+display():void

Figure 6: Class Diagram of Lecture Class

2.4 Tutor Class

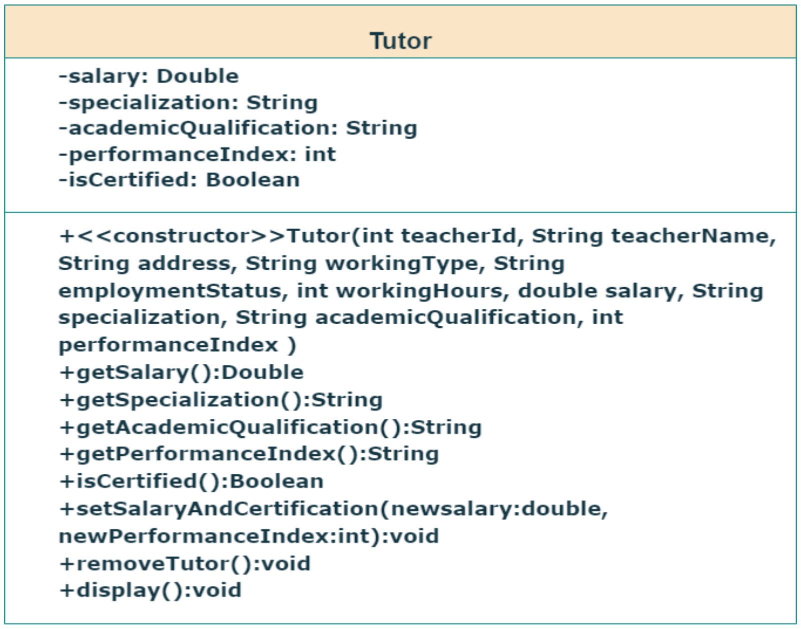

Tutor
-salary: Double
-specialization: String
-academicQualification: String
-performanceIndex: int
-isCertified: Boolean
+<<constructor>>Tutor(int teacherId, String teacherName, String address, String workingType, String employmentStatus, int workingHours, double salary, String specialization, String academicQualification, int performanceIndex)
+getSalary():Double
+getSpecialization():String
+getAcademicQualification():String
+getPerformanceIndex():String
+isCertified():Boolean
+setSalaryAndCertification(newsalary:double, newPerformanceIndex:int):void
+removeTutor():void
+display():void

Figure 7: Class Diagram of Tutor Class

2.5 TeacherGUI class

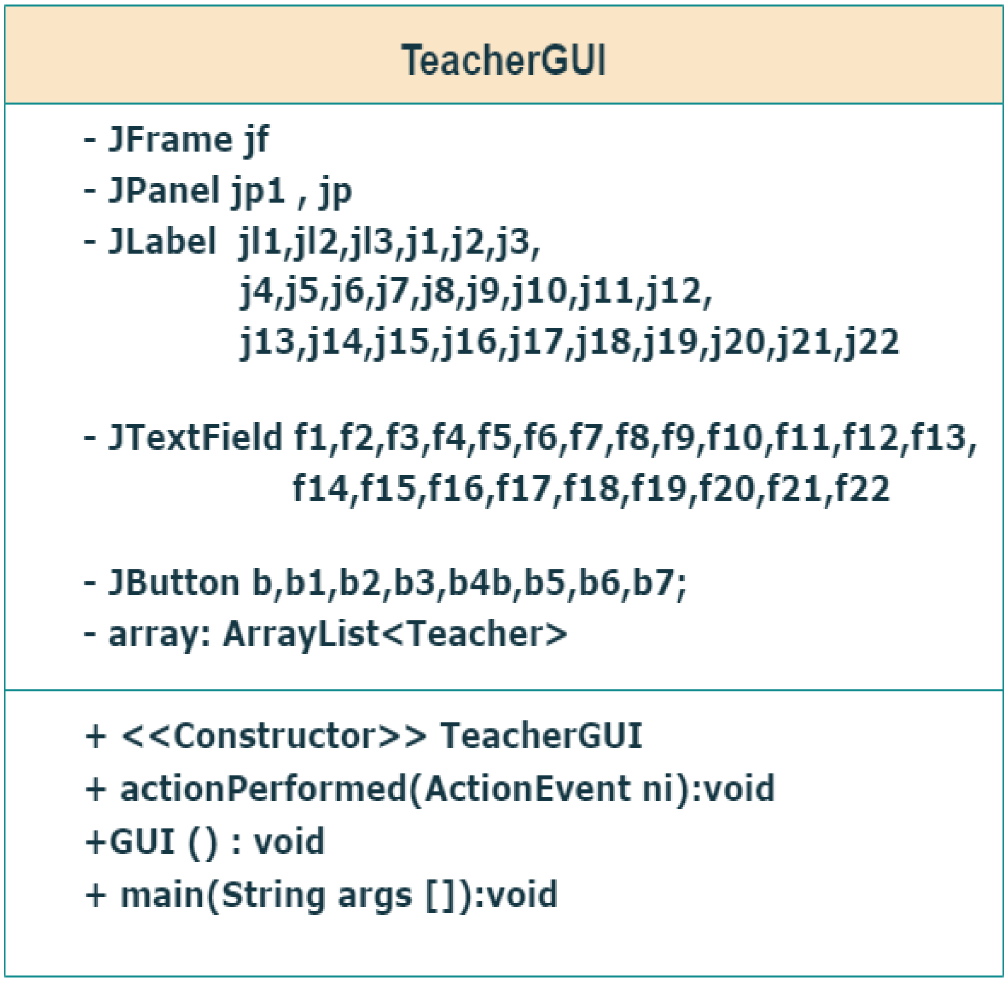
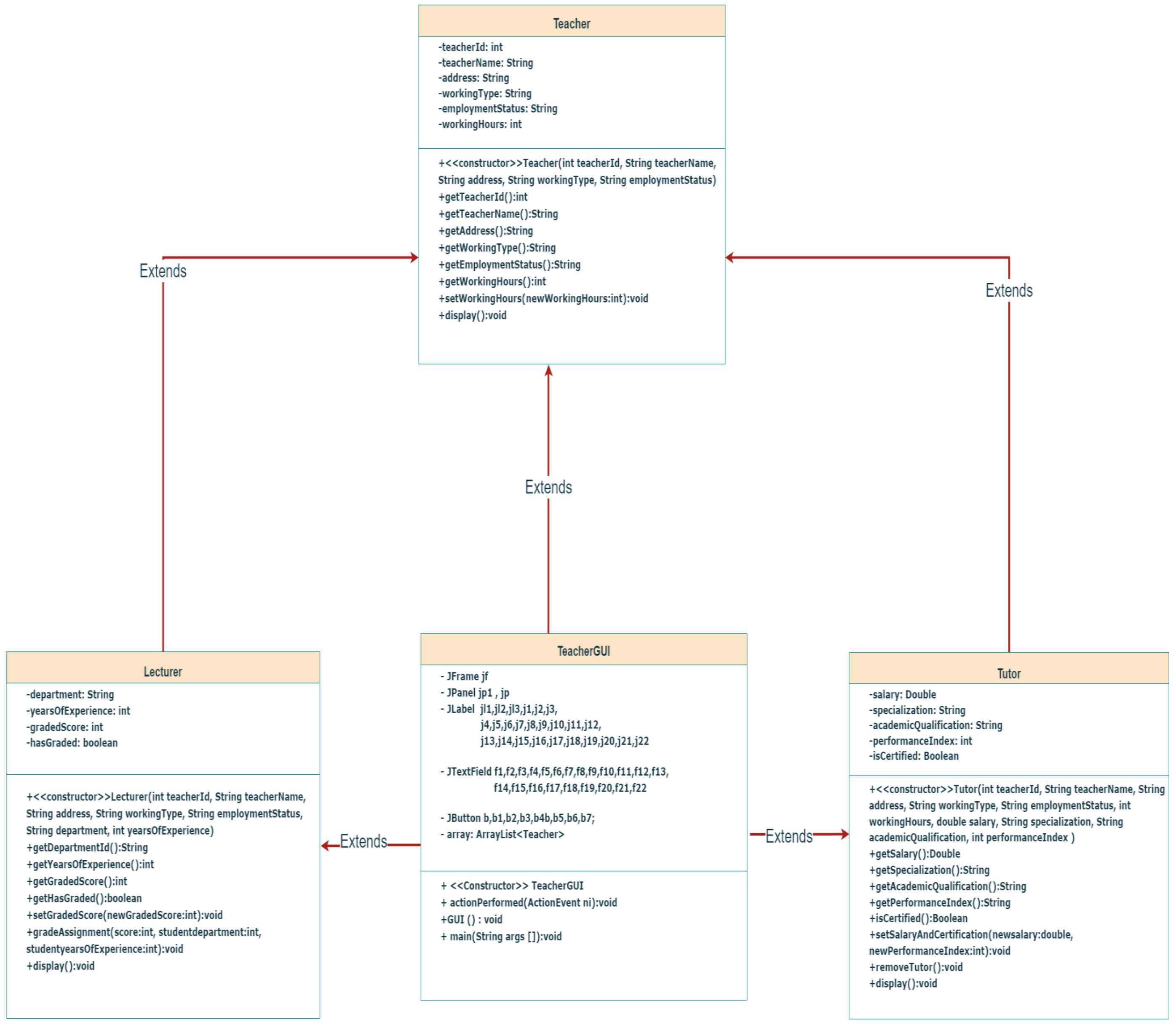

TeacherGUI
- JFrame jf
- JPanel jp1 , jp
- JLabel jl1,jl2,jl3,j1,j2,j3, j4,j5,j6,j7,j8,j9,j10,j11,j12, j13,j14,j15,j16,j17,j18,j19,j20,j21,j22
- JTextField f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13, f14,f15,f16,f17,f18,f19,f20,f21,f22
- JButton b,b1,b2,b3,b4b,b5,b6,b7;
- array: ArrayList<Teacher>
+ <<Constructor>> TeacherGUI
+ actionPerformed(ActionEvent ni):void
+GUI () : void
+ main(String args []):void

Figure 8: Class Diagram of Teacher GUI

3.4 Combine Class Diagram


Teacher
-teacherId: int
-teacherName: String
-address: String
-workingType: String
-employmentStatus: String
-workingHours: int
+<<constructor>>Teacher(int teacherId, String teacherName, String address, String workingType, String employmentStatus)
+getTeacherId():int
+getTeacherName():String
+getAddress():String
+getWorkingType():String
+getEmploymentStatus():String
+getWorkingHours():int
+setWorkingHours(newWorkingHours:int):void
+display():void
Extends
Extends
Extends
Lecturer
-department: String
-yearsOfExperience: int
-gradedScore: int
-hasGraded: boolean
+<<constructor>>Lecturer(int teacherId, String teacherName, String address, String workingType, String employmentStatus, String department, int yearsOfExperience)
+getDepartmentId():String
+getYearsOfExperience():int
+getGradedScore():int
+getHasGraded():boolean
+setGradedScore(newGradedScore:int):void
+gradeAssignment(score:int, studentdepartment:int, studentyearsOfExperience:int):void
+display():void
Extends
TeacherGUI
- JFrame jf
- JPanel jp1 , jp
- JLabel jl1,jl2,jl3,j1,j2,j3, j4,j5,j6,j7,j8,j9,j10,j11,j12, j13,j14,j15,j16,j17,j18,j19,j20,j21,j22
- JTextField f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13, f14,f15,f16,f17,f18,f19,f20,f21,f22
- JButton b,b1,b2,b3,b4b,b5,b6,b7;
- array: ArrayList<Teacher>
+ <<Constructor>> TeacherGUI
+ actionPerformed(ActionEvent ni):void
+GUI () : void
+ main(String args []):void
Extends
Tutor
-salary: Double
-specialization: String
-academicQualification: String
-performanceIndex: int
-isCertified: Boolean
+<<constructor>>Tutor(int teacherId, String teacherName, String address, String workingType, String employmentStatus, int workingHours, double salary, String specialization, String academicQualification, int performanceIndex)
+getSalary():Double
+getSpecialization():String
+getAcademicQualification():String
+getPerformanceIndex():String
+isCertified():Boolean
+setSalaryAndCertification(newsalary:double, newPerformanceIndex:int):void
+removeTutor():void
+display():void
Figure 9: Combine Class Diagram

3. Pseudocode

A pseudocode is code written in a simple human-understandable way.

```
CREATE a class named TeacherGUI
DO
    DECLARE jf AS JFrame
    DECLARE jf AS JFrame
    DECLARE jp1, jp AS JPanel
    DECLARE Jl1, Jl2, Jl3 AS JLabel
    DECLARE l1-l24 AS JLabel
    DECLARE f1-f24 AS JTextField
    DECLARE b1-b8 AS JButton
    DECLARE arrayy AS ArrayList<Teacher>

    CREATE a METHOD named GUI( )
    DO
        INITIALIZE jf AS new JFRAME(“23047392_NIKITAGUI”)
        SET the layout OF jf TO null
        SET the size OF jf TO (850, 775)

        CREATE a JPanel named jp1
        SET  the bounds of jp1 TO (0, 0, 850, 350)
        SET the background color OF jp1 TO c
        SET the layout of jp1 TO null
        ADD jp1 TO jf

        CREATE a JLabel named Jl1
        SET the text of Jl1 TO "Lecturer"
        SET the bounds OF Jl1 TO (370, 5, 50, 30)
        ADD jp1 TO l
```

```
**CREATE** a JLabel named l
**SET** the text **OF** Jl1 **TO** "Teacher ID:"
**SET** the bounds **OF** l **TO** (10, 80, 100, 25)
**ADD** l1 **TO** jp1

**CREATE** a JTextField named f
**SET** the bounds **OF** f1 **TO** (160, 50, 150, 25)
**ADD** f1 **TO** jp1

**CREATE** a JLabel named l1
**SET** the name **OF** l1 **TO** "Teacher Name:"
**SET** the bounds OF l1 **TO** (10, 80, 100, 25)
**ADD** l1 **TO** jp1

**CREATE** a JTextField named f1
**SET** the bounds OF f1 **TO** (160, 80, 150, 25)
**ADD** f1 **TO** jp1

**CREATE** a JLabel named l2
**SET** the name **OF** l2 **TO** "Address:"
**SET** the bounds **OF** l2 **TO** (10, 110, 100, 25)
**ADD** l2 **TO** jp1

**CREATE** a JTextField f2
**SET** the bounds **OF** f2 **TO** (160, 110, 150, 25)
**ADD** f2 **TO** jp1
```

```
**CREATE** a JLabel l3
**SET** the name **OF** l3 **TO** "Working Type:"
**SET** the bounds **OF** l3 **TO** (10, 140, 100, 25)
**ADD** l3 **TO** jp1

**CREATE** a JTextField f3
**SET** the bounds **OF** f3 **TO** (160, 140, 150, 25)
**ADD** f3 **TO** jp1

**CREATE** a JLabel l4
**SET** the name **OF** l4 **TO** "Working Hours"
**SET** the bounds **OF** l4 **TO** (500, 140, 150, 25)
**ADD** l4 **TO** jp1

**CREATE** a JTextField f4
**SET** the bounds **OF** f4 **TO** (650, 140, 150, 25)
**ADD** f4 **TO** jp1

**CREATE** a JLabel l5
**SET** the name **OF** l5 **TO** "Employment Status:"
**SET** the bounds **OF** l5 **TO** (500, 50, 150, 25)
**ADD** l5 **TO** jp1

**CREATE** a JTextField f5
**SET** the bounds **OF** f5 **TO** (650, 50, 150, 25)
**ADD** f5 **TO** jp1
```

```
CREATE a JLabel l6
SET the name OF l6 TO "Department:"
SET the bounds OF l6 TO (500, 80, 100, 25)
ADD l6 TO jp1

CREATE a JTextField f6
SET the bounds OF f6 TO (650, 80, 150, 25)
ADD f6 TO jp1

CREATE a JLabel l7
SET the name OF l7 TO "Year Of Experience:"
SET the bounds OF l7 TO (500, 110, 150, 25)
ADD l7 TO jp1

CREATE a JTextField f7
SET the bounds OF f7 TO (650, 110, 150, 25)
ADD f7 TO jp1

CREATE a JLabel Jl3
SET the name OF Jl3 TO "Grade Assignment"
SET the bounds OF Jl3 TO (350, 200, 150, 30)
ADD Jl3 TO jp1

CREATE JLabel l8
SET the name OF l8 TO "Graded Score"
SET the bounds OF l8 TO (500, 230, 150, 25)
ADD l8 TO jp1
```

```
CREATE a JTextField f8
SET the bounds OF f8 TO (650, 230, 150, 25)
ADD f8 TO jp1

CREATE a JLabel l9
SET the name OF l9 TO "New Teacher Id"
SET the bounds OF l9 TO (10, 230, 150, 25)
ADD l9 TO jp1

CREATE a JTextField f9
SET the bounds OF f9 TO (160, 230, 150, 25)
ADD f9 TO jp1

CREATE a JLabel l23
SET the name OF l23 TO "YearsOfExperience"
SET the bounds OF l23 TO (500, 270, 150, 25)
ADD l23 TO jp1

CREATE a JTextField f23
SET the bounds OF f23 TO (650, 270, 150, 25)
ADD f23 TO jp1

CREATE a JLabel l24
SET the name OF l24 TO "department"
SET the bounds OF l24 TO (10, 270, 150, 25)
ADD l24 TO jp1
```

```
CREATE a JTextField f24
SET the bounds OF f24 TO (160, 270, 150, 25)
ADD f24 TO jp1

CREATE a JButton b
SET the name OF b TO "Add Lecturer"
SET the bounds OF b TO (160, 180, 150, 20)
SET the background color OF b TO bc
ADD b TO jp1

CREATE a JButton b2
SET the name OF b2 TO "gradeAssignment"
SET the bounds OF b2 TO (160, 320, 150, 20)
SET the background color OF b2 TO bc
ADD b2 TO jp1

CREATE a JButton b1
SET the name OF b1 TO "Display"
SET the bounds OF b1 TO (500, 180, 100, 20)
SET the background color OF b1 TO bc
ADD b1 TO jp1

CREATE a JButton b3
SET the name OF b3 TO "Clear"
SET the bounds OF b3 TO (500, 320, 100, 20)
SET the background color OF b3 TO bc
ADD b3 TO jp1
```

```
CREATE ActionListener for b4
DO
    WHEN actionPerformed event occurs
    DO
        Try:
        DO
            IF any OF the text fields are left empty:
            DO
                Set the button background color TO indicate
                error
                Display a message asking TO fill in all
                text fields
            END DO
            ELSE IF any of the text field contains invalid
            characters:
            DO
                Set the button background color TO indicate
                error
                Display a message indicating only letters are
                allowed in certain fields
            END DO
            ELSE
            DO
                SET the button background color TO indicate
                success
                GET values FROM TEXT fields (teacher ID,
                name, address, etc.)
                Create a new Lecturer object WITH these
                values
                Add the new Lecturer object TO the array list
```

```
                    Display a success message indicating data is
                    saved
                END DO
            END
            Catch NumberFormatException:
            DO
                Display an error message indicating a number format
                exception occurred
            END DO
        END DO
    END DO

    CREATE ActionListener for b1
    DO
        WHEN actionPerformed event occurs
        DO
            SET button background color TO pink
            FOR each teacher in the array list:
            DO
            IF the teacher is an instance of Lecturer:
            DO
                Cast the teacher to a Lecturer object
                Display the details of the Lecturer
                Show a message indicating that data is displayed
                Print a newline
            END DO
        END DO
    END DO
```

```
CREATE ActionListener for b2
DO
    WHEN actionPerformed event occurs
    DO
        If any OF the required text fields are empty:
        DO
        SET the background color OF the button to indicate an error
        Show a message dialog prompting TO  fill in the text fields
        END DO
        ELSE
            Extract the values from the text fields for new teacher
            ID, score, years of experience, and department
            Iterate over the array of teachers:
            IF the teacher is a lecturer, and their ID matches the
            new teacher ID and department matches:
            DO
                Set the background color of the button to
                indicate success
                Cast the teacher to a Lecturer object
                Call the gradeAssignment method of the
                Lecturer object with the score, department, and
                years of experience as arguments
                Show a success message dialog
            END DO
            Else:
            DO
                Show a message dialog indicating that the new
                input values must match those in the array
            END DO
        IF any exception occurs during the process:
                DO
                    Clear the text fields related to the exception
                    Show a message dialog indicating a number formate
                    exception
                END DO
        END DO
END DO
CREATE ActionListener for b6
DO
        WHEN actionPerformed event occurs
        DO
                Set the background color of the button to pink
                SET text of f TO ""
                SET text of f1 TO ""
                SET text of f2 TO ""
                SET text of f3 TO ""
                SET text of f4 TO ""
                SET text of f5 TO ""
                SET text of f6 TO ""
                SET text of f7 TO ""
                SET text of f8 TO ""
                SET text of f9 TO ""
                SET text of f23 TO ""
                SET text of f24 TO ""
                Show a message dialog indicating that all data has been
                cleared
        END DO
END DO
```

```
CREATE a JPanel jp
SET the bounds OF jp TO (0, 340, 850, 400)
SET the background color OF jp TO Color.LIGHT_GRAY
SET the layout OF jp TO null
ADD jp TO jf

CREATE a JLabel Jl2
SET the text OF Jl2 TO "Tutor"
SET the bounds OF Jl2 TO (360, 5, 100, 30)
ADD Jl2 TO jp

CREATE a JLabel l10
SET the text OF l10 TO "Teacher ID:"
SET the bounds OF l10 TO (30, 70, 100, 25)
ADD l10 TO jp

CREATE a JTextField f10
SET the bounds OF f10 TO (180, 70, 150, 25)
ADD f10 TO jp

CREATE a JLabel l11
SET the text OF l11 TO "Teacher Name:"
SET the bounds OF l11 TO (30, 100, 100, 25)
ADD l11 TO jp

CREATE JTextField f11
SET bounds OF f11 TO (180, 100, 150, 25)
ADD f11 TO jp
```

```
CREATE a JLabel l12
SET the text OF l12 TO "Address:"
SET the bounds OF l12 TO (30, 130, 100, 25)
ADD l12 TO jp

CREATE a  JTextField f12
SET the bounds OF f12 TO (180, 130, 150, 25)
ADD f12 TO jp

CREATE a JLabel l13
SET the text OF l13 TO "Working Type:"
SET the bounds OF l13 TO (30, 160, 100, 25)
ADD l13 TO jp

CREATE a JTextField f13
SET the bounds OF f13 TO (180, 160, 150, 25)
ADD f13 TO jp

CREATE a JLabel l14
SET the text OF l14 TO "Employment Status :"
SET the bounds OF l14 TO (30, 190, 150, 25)
ADD l14 TO jp

CREATE a JTextField f14
SET bounds OF f14 TO (180, 190, 150, 25)
ADD f14 TO jp
```

```
CREATE JLabel l15
SET the text OF l15 TO "Salary:"
SET the bounds OF l15 TO (460, 70, 100, 25)
ADD l15 TO jp

CREATE JTextField f15
SET bounds OF f15 TO (625, 70, 150, 25)
ADD f15 TO jp

CREATE a JLabel l16
SET the text OF l16 TO "Specialization:"
SET the bounds OF l16 TO (460, 100, 100, 25)
ADD l16 TO jp

CREATE a JTextField f16
SET the bounds OF f16 TO (625, 100, 150, 25)
ADD f16 TO jp

CREATE a JLabel l17
SET the text OF l17 TO "Academic Qualification:"
SET the bounds OF l17 TO (460, 130, 150, 25)
ADD l17 TO jp

CREATE a JTextField f17
SET the bounds OF f17 TO (625, 130, 150, 25)
ADD f17 TO jp
```

```
CREATE a JLabel l18
SET the text OF l18 TO "Performance Index:"
SET the bounds OF l18 TO (460, 160, 150, 25)
ADD l18 TO jp

CREATE a JTextField f18
SET the bounds OF f18 TO (625, 160, 150, 25)
ADD f18 TO jp

CREATE a JLabel l19
SET the text OF l19 TO "Working Hours:"
SET the bounds OF l19 TO (460, 190, 150, 25)
ADD l19 TO jp

CREATE a JTextField f19
SET the bounds OF f19 TO (625, 190, 150, 25)
ADD f19 TO jp

CREATE JLabel l20
SET text OF l20 TO "New Salary:"
SET bounds OF l20 TO (580, 275, 150, 25)
ADD l20 TO jp

CREATE a JTextField f20
SET the bounds OF f20 TO (655, 275, 150, 25)
ADD f20 TO jp
```

```
CREATE JLabel l21
SET the text OF l21 TO "New Performance Index:"
SET the bounds OF l21 TO (20, 275, 150, 25)
ADD l21 TO jp

CREATE a JTextField f21
SET the bounds OF f21 TO (170, 275, 150, 25)
ADD f21 TO jp

CREATE a JLabel l22
SET the text OF l22 TO "New Teacher ID:"
SET the bounds OF l22 TO (300, 310, 150, 25)
ADD l22 TO jp

CREATE JTextField f22
SET bounds OF f22 TO (420, 310, 150, 25)
ADD f22 TO jp

CREATE a JButton b4
SET the text OF b4 TO "Add Tutor"
SET the bounds OF b4 TO (140, 230, 120, 20)
SET the background color OF b4 TO bc
ADD b4 TO jp

CREATE a JButton b5
SET the text OF b5 TO "Display"
SET the bounds OF b5 TO (560, 230, 120, 20)
SET the background color OF b5 TO bc
```

```
ADD b5 TO jp

CREATE a JButton b6
SET the text OF b6 TO "Remove Tutor"
SET the bounds OF b6 TO (140, 350, 120, 20)
SET the background color OF b6 TO bc
ADD b6 TO jp

CREATE a JButton b7
SET the text OF b7 TO "Clear"
SET the bounds OF b7 TO (370, 350, 100, 20)
SET the background color OF b7 TO bc
ADD b7 TO jp

CREATE a JButton b8
SET the text OF b8 TO "Set Slary"
SET the bounds OF b8 TO (560, 350, 120, 20)
SET the background color OF b8 TO bc
ADD b8 TO jp

CREATE ActionListener for b4 button
DO
        WHEN actionPerformed event occurs
        DO
                Try:
                DO
                        IF any OF the text fields are left empty:
                        DO
                                Set the button background color TO indicate
                                error

                                Display a message asking TO fill in all
                                text fields

                        END DO

                        ELSE IF any of the text field contains invalid
                        characters:

                        DO

                                Set the button background color TO indicate
                                error

                                Display a message indicating only letters are
                                allowed In certain fields

                        END DO

                        ELSE

                        DO

                                SET the button background color TO indicate
                                success

                                 PARSE Integer from f10 and assign to
                                teacherId

                                GET text from f11 and assign to teacherName

                                GET text from f12 and assign to address

                                GET text from f13 and assign to workingType

                                GET text from f14 and assign to
                                employmentStatus

                                PARSE Integer from f15 and assign to salary

                                GET text from f16 and assign to specialization

                                GET text from f17 and assign to
                                academicQualification

                                PARSE Integer from f18 and assign to
                                performanceIndex

                                 PARSE Integer from f19 and assign to
                                 workingHours
```

```
                    CREATE Tutor object tobj with parameters
                    (teacherId,teacherName, address,
                    workingType,employmentStatus,workingHours,
                    salary,specialization, academicQualification,
                    performanceIndex)
                    Add the new Tutor object TO the array list
                    Display a success message indicating data is
                    saved
                END DO
            END DO
            Catch NumberFormatException:
            DO
                Display an error message indicating a number format
                exception occurred
            END DO
        END DO
    END DO


    CREATE ActionListener for b5
    DO
        WHEN actionPerformed event occurs
        DO
            SET button background color TO pink
            FOR each teacher in the array list:
            DO
             IF the teacher is an instance of Tutor:
            DO
                Cast the teacher to a Tutor object
                Display the details of the Tutor
                Show a message indicating that data is displayed
                Print a newline
            END DO
        END DO
    END DO

    CREATE ActionListener for b6
    DO
        WHEN actionPerformed event occurs
        DO
            TRY
            DO
                IF any of the text fields (f10, f15, f16, f17, f18, f22) is
                 empty
                DO
                    SET background color of b6 TO no
                    SHOW "Please fill in the text field" message
                    using  JOptionPane
                END DO
                DO
                IF f16 or f17 does not match the pattern "[a-zA-Z ]+"
                    SET background color of b6 TO no
                    SHOW "Invalid input. Only letters are allowed."
                    message using JOptionPane
                END DO
                ELSE
                DO
                    SET background color OF b6 TO Color.PINK
                    PARSE teacherId from f10 as integer
                    PARSE salary from f15 as double
                    GET specialization from f16
                    GET academicQualification from f17
                    PARSE performanceIndex from f18 as integer
                    PARSE newteacherId from f22 as integer
                    FOR each Teacher t in array
                    DO
                        SET background color of b6 TO Color.PINK
                        IF t is an instance of Tutor AND teacherId is equal to newteacherId
                        DO
                            CAST t to Tutor and store it in t1
                            CALL removeTutor method on t1
                            SHOW "Removal is done" message using JOptionPane
                        END DO
                        ELSE
                        DO
                            SHOW "Teacher id must be same" message using JOptionPane
                        END DO
                CATCH NumberFormatException e
                DO
                    SHOW "Number Format Exception is found" message using JOptionPane
                END DO
            END DO
        END DO
```

```
CREATE ActionListener for b8
DO
    WHEN actionPerformed event occurs
    DO
        TRY
        DO
             IF any OF the text fields (f10, f20, f21, f22) is empty
            DO
                SET background color of b8 TO no
                SHOW "Please fill in the text field" message
                using JOptionPane
            END DO
            IF f16 or f17 does not match the pattern "[a-zA-Z
            ]+"
            DO
                SET background color of b8 TO no
                SHOW "Invalid input. Only letters are
                allowed." message using JOptionPane
            END DO
            ELSE
            DO
                SET background color of b8 TO Color.PINK
                PARSE newsalary from f20 as double
                PARSE newPerformanceIndex from f21 as
                integer
                PARSE newteacherId from f22 as integer
                FOR each Teacher t in array
                DO
```

```
                        IF t is an instance of Tutor AND t's
                        teacherId is equal to newteacherId
                        DO
                        `   CAST t to Tutor and store it in t1
                            CALL setSalaryAndCertification
                            method on t1 with parameters
                            (newsalary,
                            newPerformanceIndex)
                             SHOW "Salary is set" message
                            using JOptionPane
                        END DO
                        ELSE
                        DO
                            SHOW "Teacher id must be
                            same" message using
                            JOptionPane
                        END DO
                    END DO
                END DO
        END DO
        CATCH NumberFormatException e
        DO
            SHOW "Number Format Exception is found" message
            using JOptionPane
        END DO
    END DO
END DO

    CREATE ActionListener for b6
    DO
        WHEN actionPerformed event occurs
```

```
                DO
                    Set the background color of the button to pink
                    SET text of f10 TO ""
                    SET text of f11 TO ""
                    SET text of f12 TO ""
                    SET text of f13 TO ""
                    SET text of f14 TO ""
                    SET text of f15 TO ""
                    SET text of f16 TO ""
                    SET text of f17 TO ""
                    SET text of f18 TO ""
                    SET text of f19 TO ""
                    SET text of f20 TO ""
                    SET text of f21 TO ""
                    SET text of f22 TO ""
                    Show a message dialog indicating that all data has been
                    cleared
                END DO
            END DO
            Make visibility of jf To true
        END DO
        CREATE function main taking String array args as parameter
        DO
            CREATE TeacherGUI object named obj
             CALL GUI function OF obj
        END DO
    END DO
```

4. Method Description

4.1 Method Description as a whole

Method	Description
Input Validation ➢ if(f.getText().isEmpty() ||..etc) ➢ else(!f1.getText().matches("[a-zA-Z]+")||..etc)	➢ It checks if there are any of the fields like "f" are empty or not. If left empty it changes the color and displays a message to fill in the text field. ➢ It checks if these fields have string value only. If not it changes the color and displays
Message Confirmation	➢ When the data has been added successfully it displays a confirmation message
Catch (NumberFormatException)	➢ It displays a message when there is a string value instead of an integer.

Table 1: Table 1 of Method Description

4.2 Method Description of all the Buttons

Method	Description
Add lecturer	➢ When the button is pressed it converts text fields like f, f1, etc to integers. ➢ It then takes values from the corresponding text field i.e. teacher id, teacher name, address, working type, employment status, gradedScore, workingHours and YearsOfExperience and assign them to the variable. ➢ A new object Lecture is created using the data. ➢ It is added to an array of Teacher class.
Grade the assignment	➢ When the button is pressed it converts text fields into integers and assigns them to teacher id, score, department, and YearsOfExperience. ➢ It uses Lecture object and when the valid teacher ID and department are assigned these values are compared to the existing value. If they are valid they are used to assign the grade accordingly from the lecture class.

Add Tutor	➢ When the button is pressed it converts text fields like f10, f15, etc to integers. ➢ It then takes values from the corresponding text field i.e. teacher id, teacher name, address, working type, employment status, working hours, salary, specialization, academic qualifications, and performanceIndex, and assigns them to the variable. ➢ A new object Teacher is created using the data. It is added to an array of Teacher classes.
Set the salary of the Tutor	➢ When the button b8 is pressed it converts text fields like f20, f21, f22 to integers. ➢ It goes through the list of teachers (array) and compares the teacher ID with the existing teacher Id. ➢ It updates ans sets the salary and performance index with the new values provided in the text fields.
Remove the salary of Tutor	➢ When the button b6 is pressed it goes through the list of teachers (array) and compares the teacher ID with the existing teacher Id. If found, it removes the tutor from the system and displays a confirmation message
Display	➢ "When you press this button, the information related to the relevant class will be displayed."
Clear	➢ When the button is pressed all the values are cleared from the text fields.

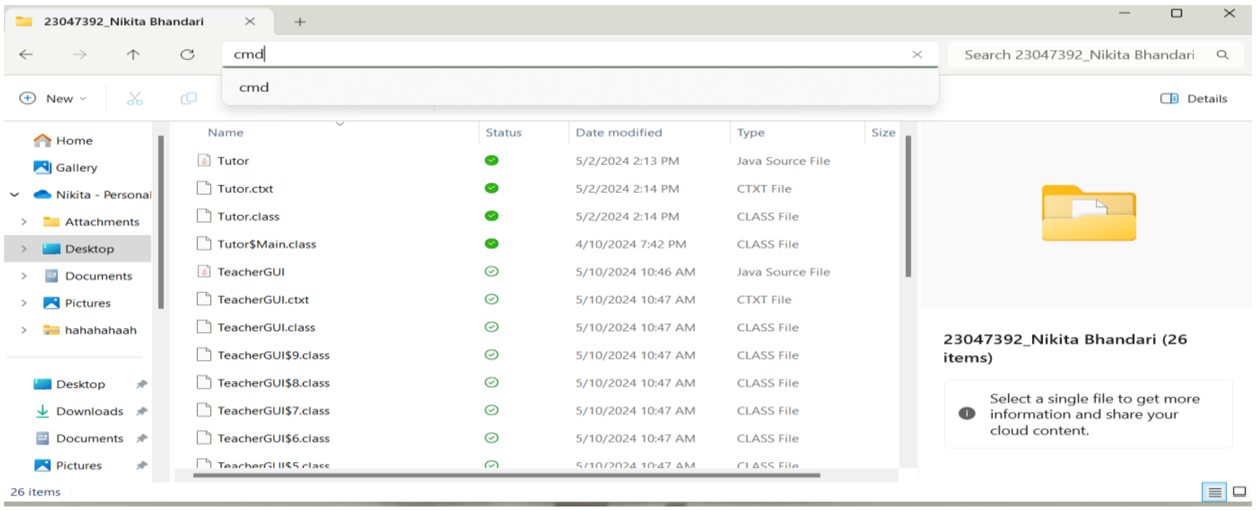
Table 2: Table 2 of Method Description

5. Testing

5.1 Testing in Command Prompt

Table 3: Table for Command Prompt

Objective	To run the program use the command prompt.
Action	Write cmd and then type Java TeacherGUI.
Expected Result	The GUI should appear.
Actual Result	The GUI did appear.
Conclusion	The test was successful.



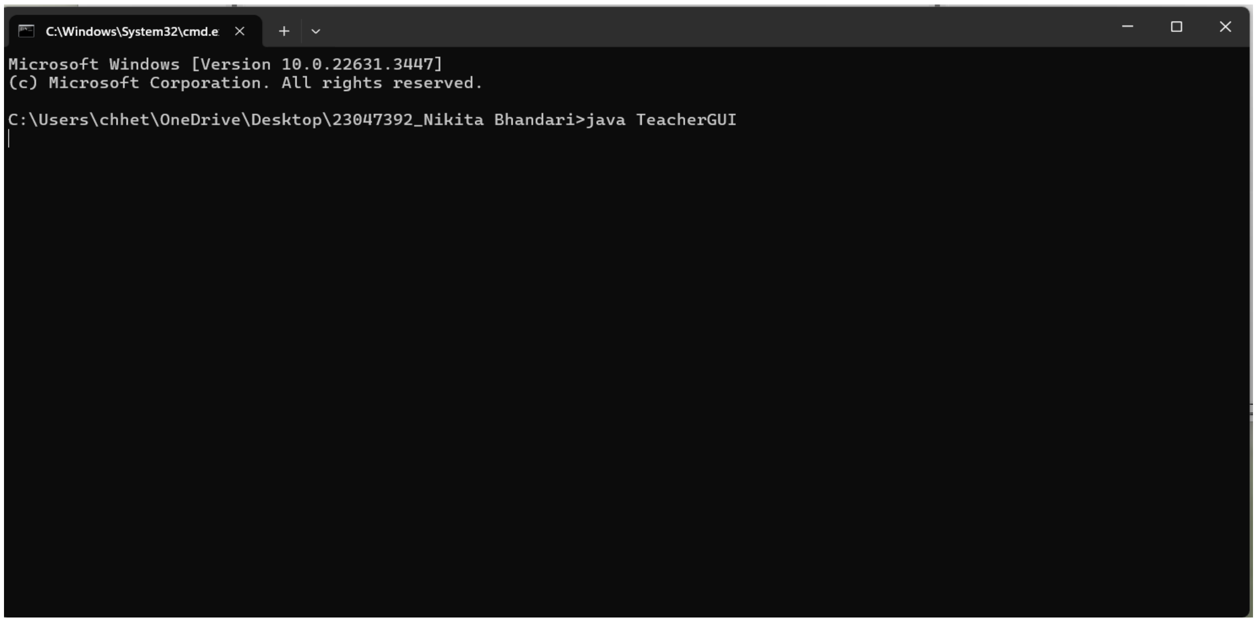
Name	Status	Date modified	Type	Size
Tutor	✓	5/2/2024 2:13 PM	Java Source File	
Tutor.ctxt	✓	5/2/2024 2:14 PM	CTXT File	
Tutor.class	✓	5/2/2024 2:14 PM	CLASS File	
Tutor$Main.class	✓	4/10/2024 7:42 PM	CLASS File	
TeacherGUI	✓	5/10/2024 10:46 AM	Java Source File	
TeacherGUI.ctxt	✓	5/10/2024 10:47 AM	CTXT File	
TeacherGUI.class	✓	5/10/2024 10:47 AM	CLASS File	
TeacherGUI$9.class	✓	5/10/2024 10:47 AM	CLASS File	
TeacherGUI$8.class	✓	5/10/2024 10:47 AM	CLASS File	
TeacherGUI$7.class	✓	5/10/2024 10:47 AM	CLASS File	
TeacherGUI$6.class	✓	5/10/2024 10:47 AM	CLASS File	
TeacherGUI$5.class	✓	5/10/2024 10:47 AM	CLASS File	

23047392_Nikita Bhandari (26 items)

Select a single file to get more information and share your cloud content.

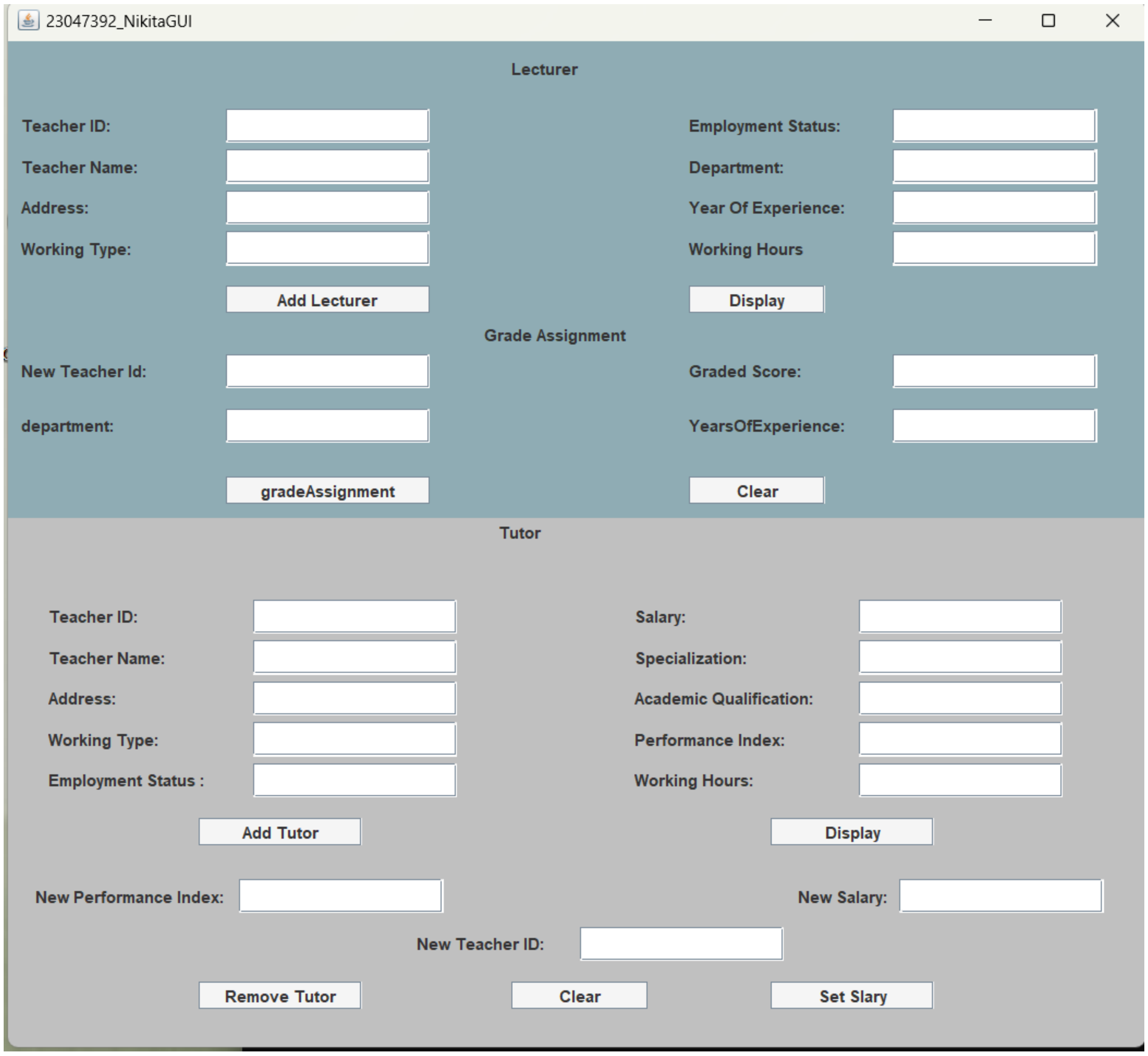
26 items

Figure 10: Writing cmd



```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\chhet\OneDrive\Desktop\23047392_Nikita Bhandari>java TeacherGUI
```

Figure 11: Typing in Command Prompt


23047392_NikitaGUI

Lecturer

Teacher ID:
Teacher Name:
Address:
Working Type:
Employment Status:
Department:
Year Of Experience:
Working Hours

Grade Assignment

New Teacher Id:
department:
Graded Score:
YearsOfExperience:

Tutor

Teacher ID:
Teacher Name:
Address:
Working Type:
Employment Status :
Salary:
Specialization:
Academic Qualification:
Performance Index:
Working Hours:

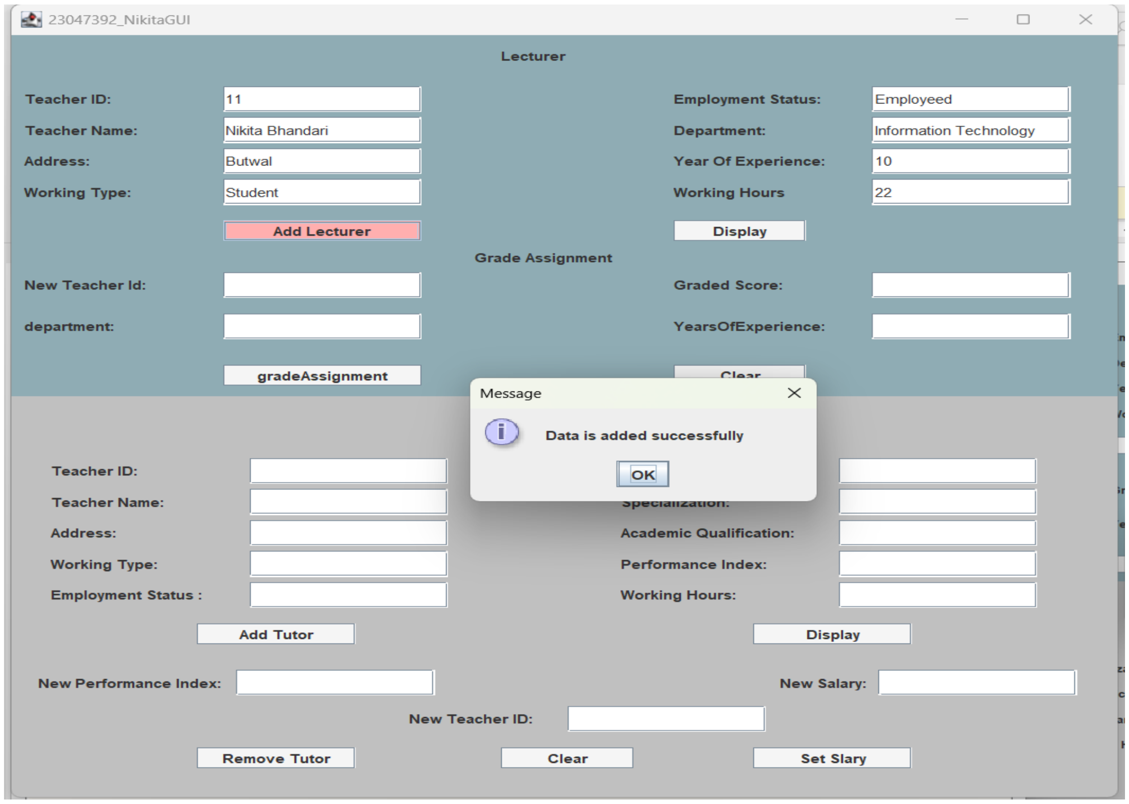
New Performance Index:
New Salary:
New Teacher ID:

Figure 12: Appereance Of GUI

5.2 Testing of Add and display Lecturer

Objective	Add lectuter
Action	The values are entered in the text field Teacher ID: 11 Teacher Name: Nikita Bhandari Address: Butwal Working Type: Student Employment Status: Employed Department: Information Technology Year of Experience : 10 Working Hours 22 Then the add button is pressed and finally the display.
Expected Result	All the buttons should be working properly.
Actual Result	The button did work properly.
Conclusion	The test was successful.

Table 4: Table for Add and Dsiplay of Lecture



23047392_NikitaGUI

Lecturer

Teacher ID: 11
Teacher Name: Nikita Bhandari
Address: Butwal
Working Type: Student
Employment Status: Employeed
Department: Information Technology
Year Of Experience: 10
Working Hours 22

Add Lecturer
Display

Grade Assignment

New Teacher Id:
department:
Graded Score:
YearsOfExperience:

gradeAssignment
Clear

Message
Data is added successfully
OK

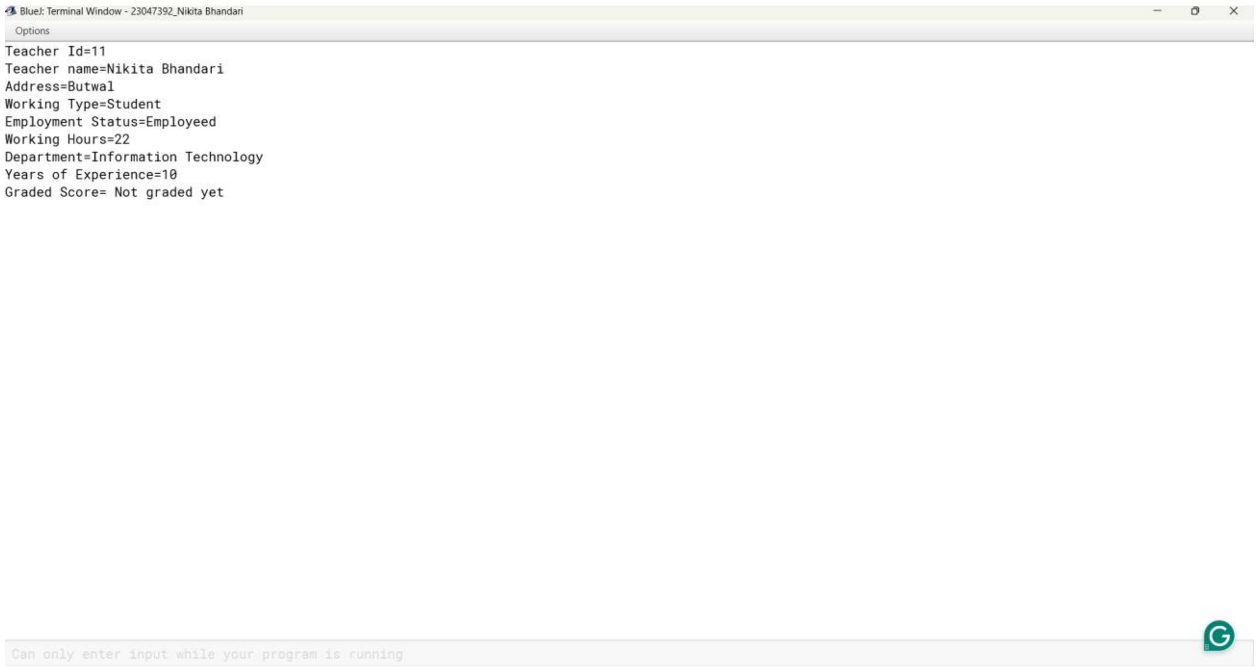
Teacher ID:
Teacher Name:
Address:
Working Type:
Employment Status :
Specialization:
Academic Qualification:
Performance Index:
Working Hours:

Add Tutor
Display

New Performance Index:
New Salary:
New Teacher ID:

Remove Tutor
Clear
Set Slary

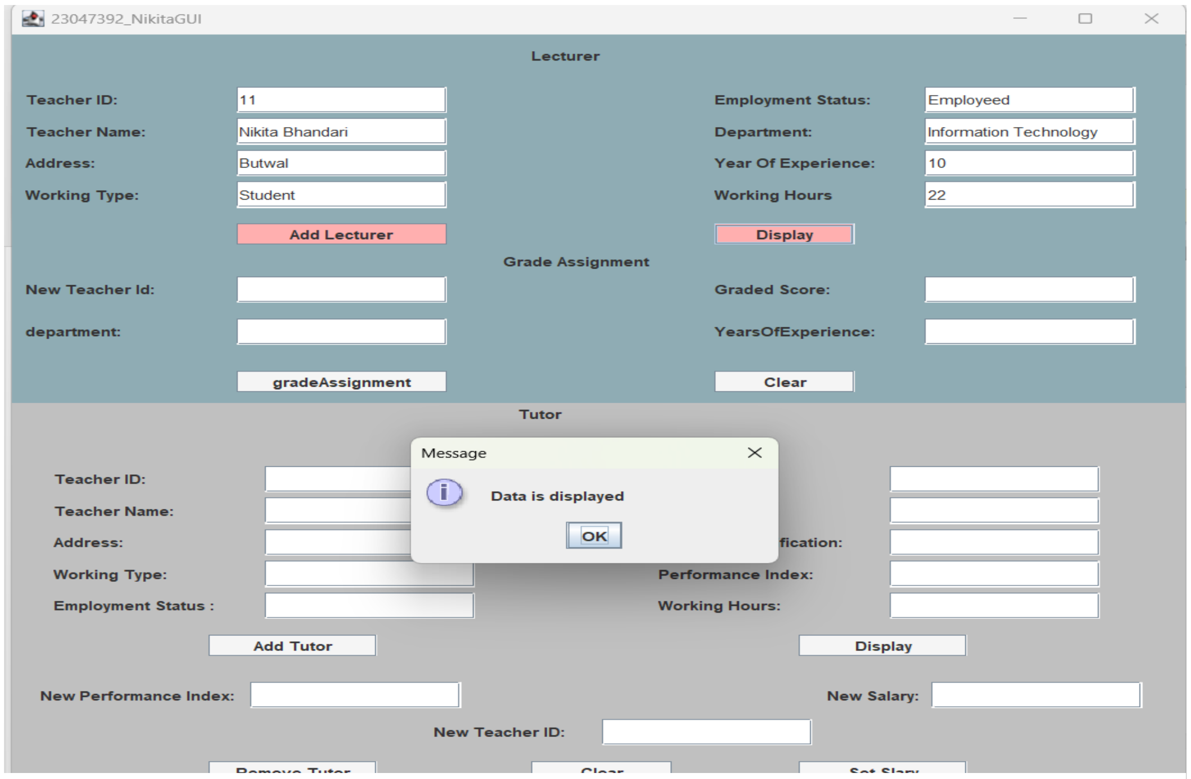
Figure 13: Test of add Lecturer


BlueJ: Terminal Window - 23047392_Nikita Bhandari
Options

```
Teacher Id=11
Teacher name=Nikita Bhandari
Address=Butwal
Working Type=Student
Employment Status=Employeed
Working Hours=22
Department=Information Technology
Years of Experience=10
Graded Score= Not graded yet
```

Can only enter input while your program is running

Figure 15: Display of Lecturer

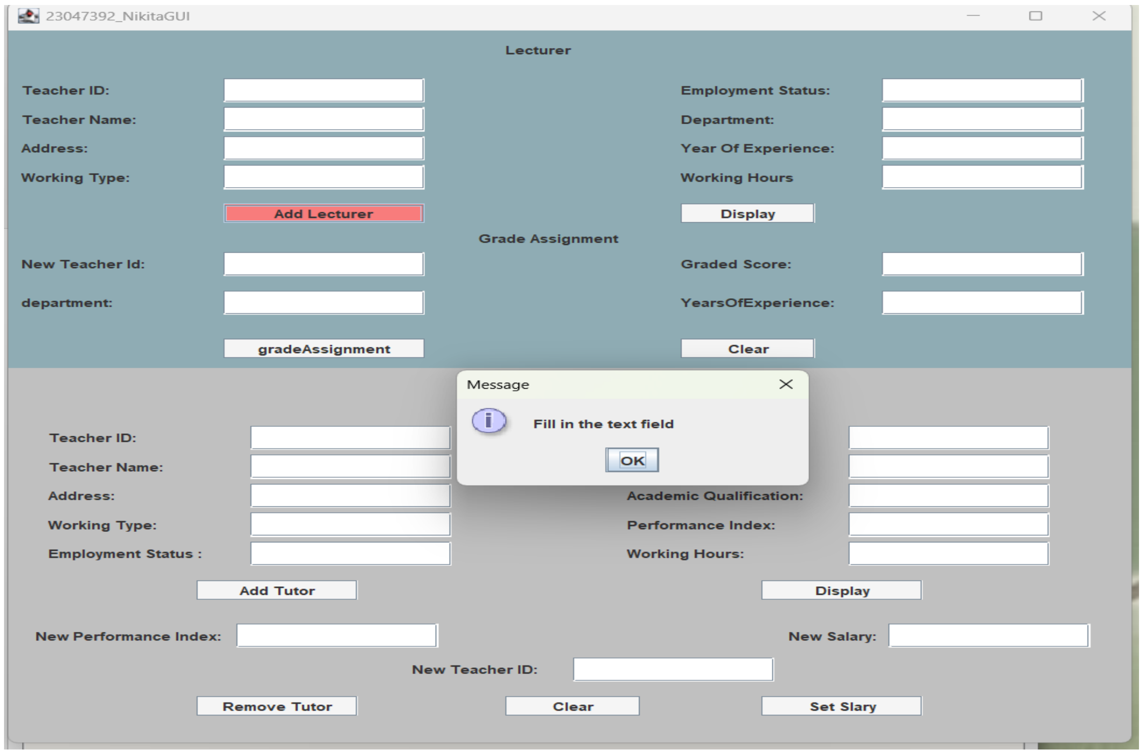


Teacher ID:	11	Employment Status:	Employeed
Teacher Name:	Nikita Bhandari	Department:	Information Technology
Address:	Butwal	Year Of Experience:	10
Working Type:	Student	Working Hours	22
New Teacher Id:		Graded Score:	
department:		YearsOfExperience:	

Data is displayed

Teacher ID:			
Teacher Name:			
Address:		fication:	
Working Type:		Performance Index:	
Employment Status :		Working Hours:	
New Performance Index:		New Salary:	
New Teacher ID:			

Figure 14: Test of Display Button Lecturer

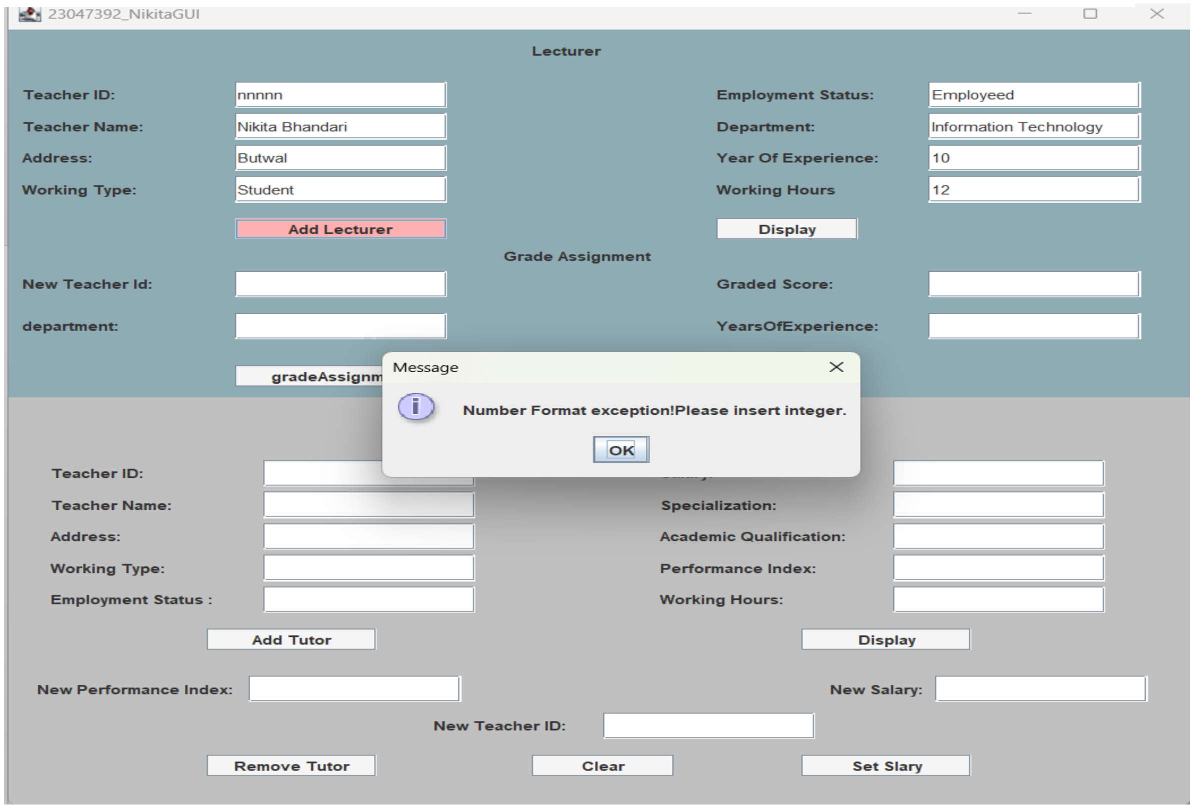


Message

Fill in the text field

OK

Figure 16: Empty Text Field



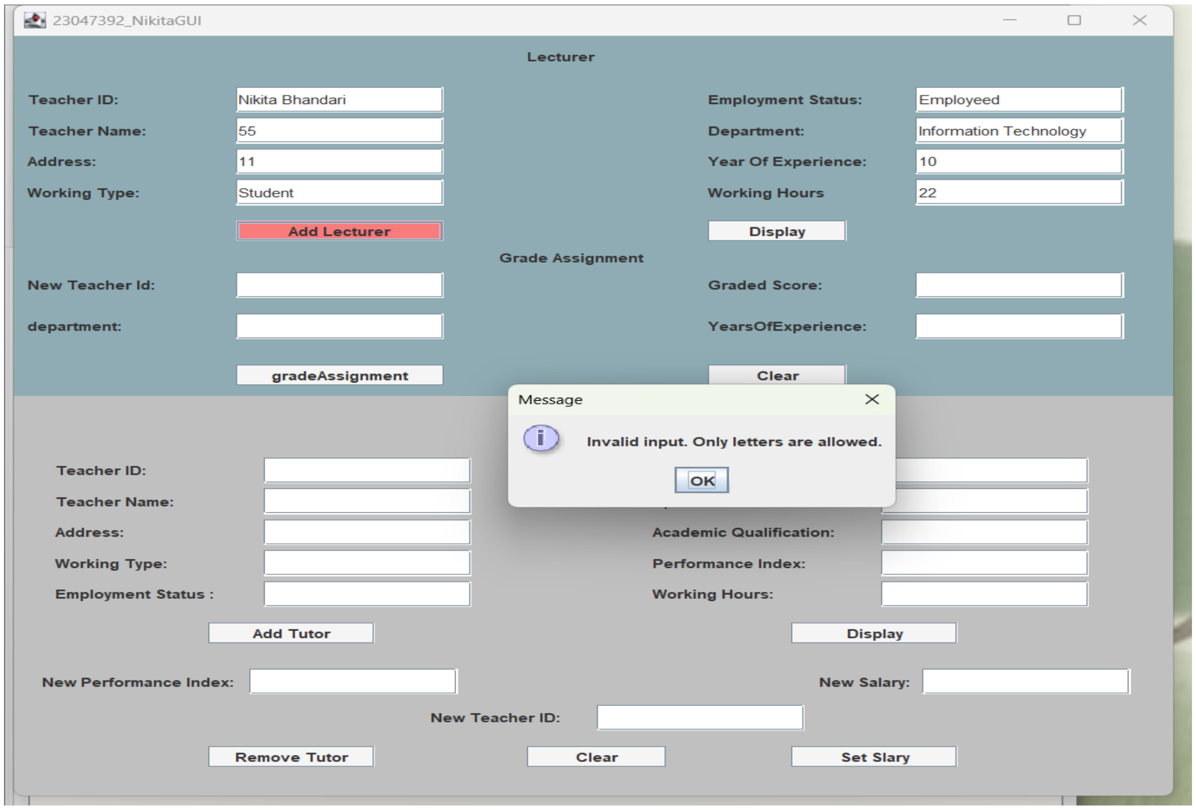
Teacher ID: nnnnn
Teacher Name: Nikita Bhandari
Address: Butwal
Working Type: Student
Employment Status: Employeed
Department: Information Technology
Year Of Experience: 10
Working Hours 12

Message

Number Format exception!Please insert integer.

OK

Figure 17: Number Format Exception


23047392_NikitaGUI

Lecturer

Teacher ID: Nikita Bhandari
Teacher Name: 55
Address: 11
Working Type: Student
Employment Status: Employeed
Department: Information Technology
Year Of Experience: 10
Working Hours 22

Grade Assignment

New Teacher Id:
department:
Graded Score:
YearsOfExperience:

Message

Invalid input. Only letters are allowed.

Teacher ID:
Teacher Name:
Address:
Working Type:
Employment Status :
Academic Qualification:
Performance Index:
Working Hours:
New Performance Index:
New Salary:
New Teacher ID:

Figure 18: String error

5.3 Testing of Graded Score

Objective	To Grade the score
Action	The values are entered in the text field Then the add button is pressed and finally the display. New Teacher Id: 11 Graded Score: 79 Years of Experience: 12 Department: Nursing
Expected Result	All the buttons should be working properly.
Actual Result	The button did work properly.
Conclusion	The test was successful.

Table 5: Table for grade assignment

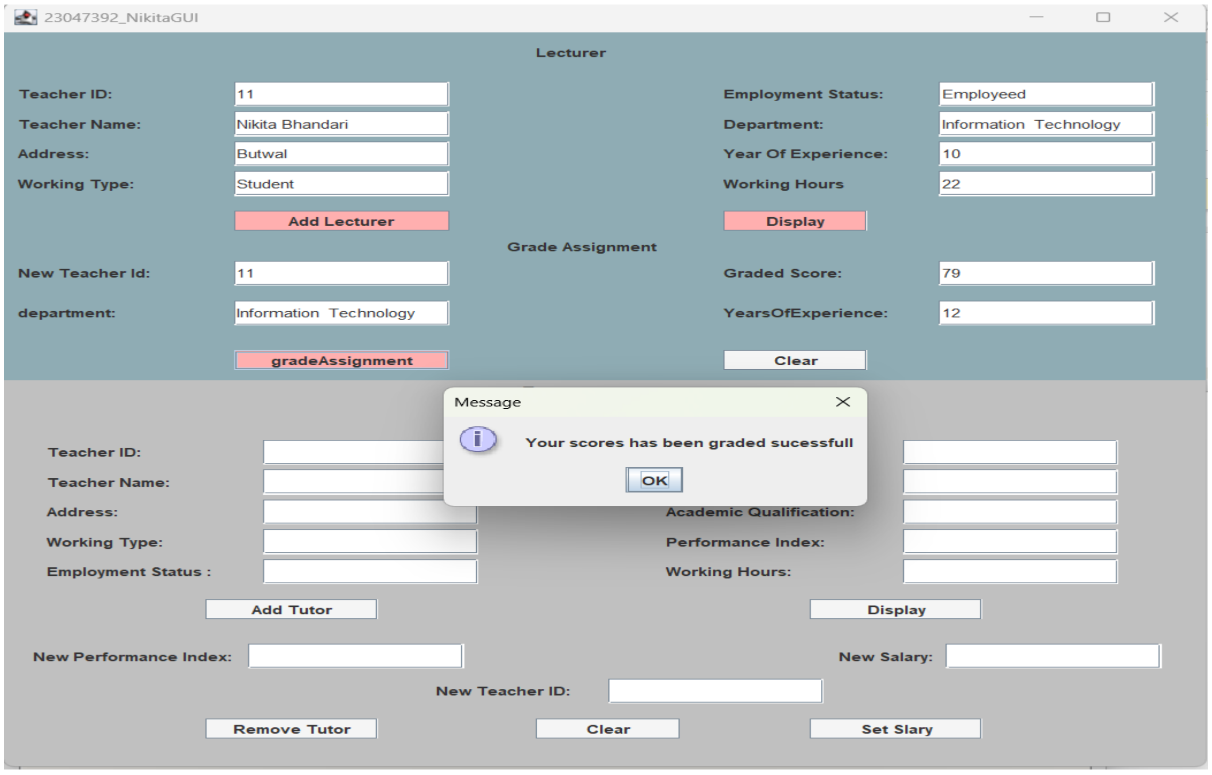
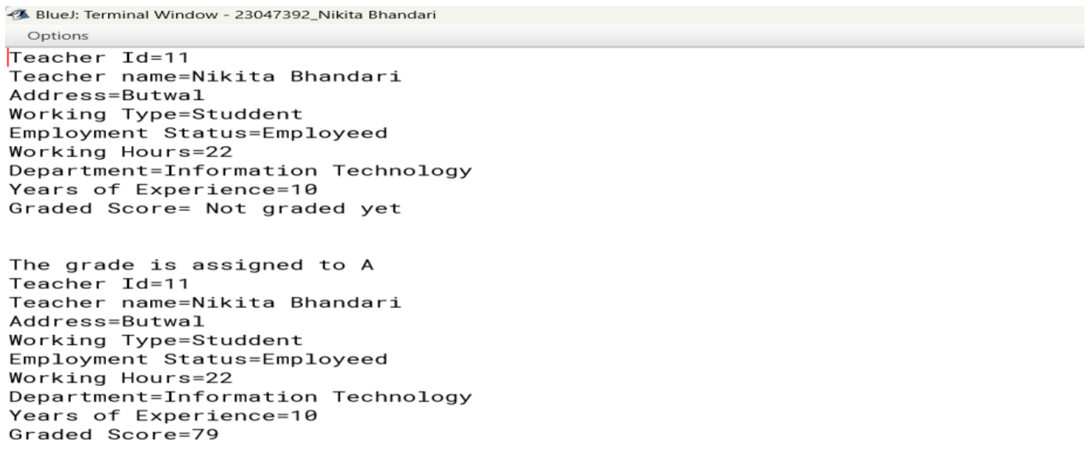
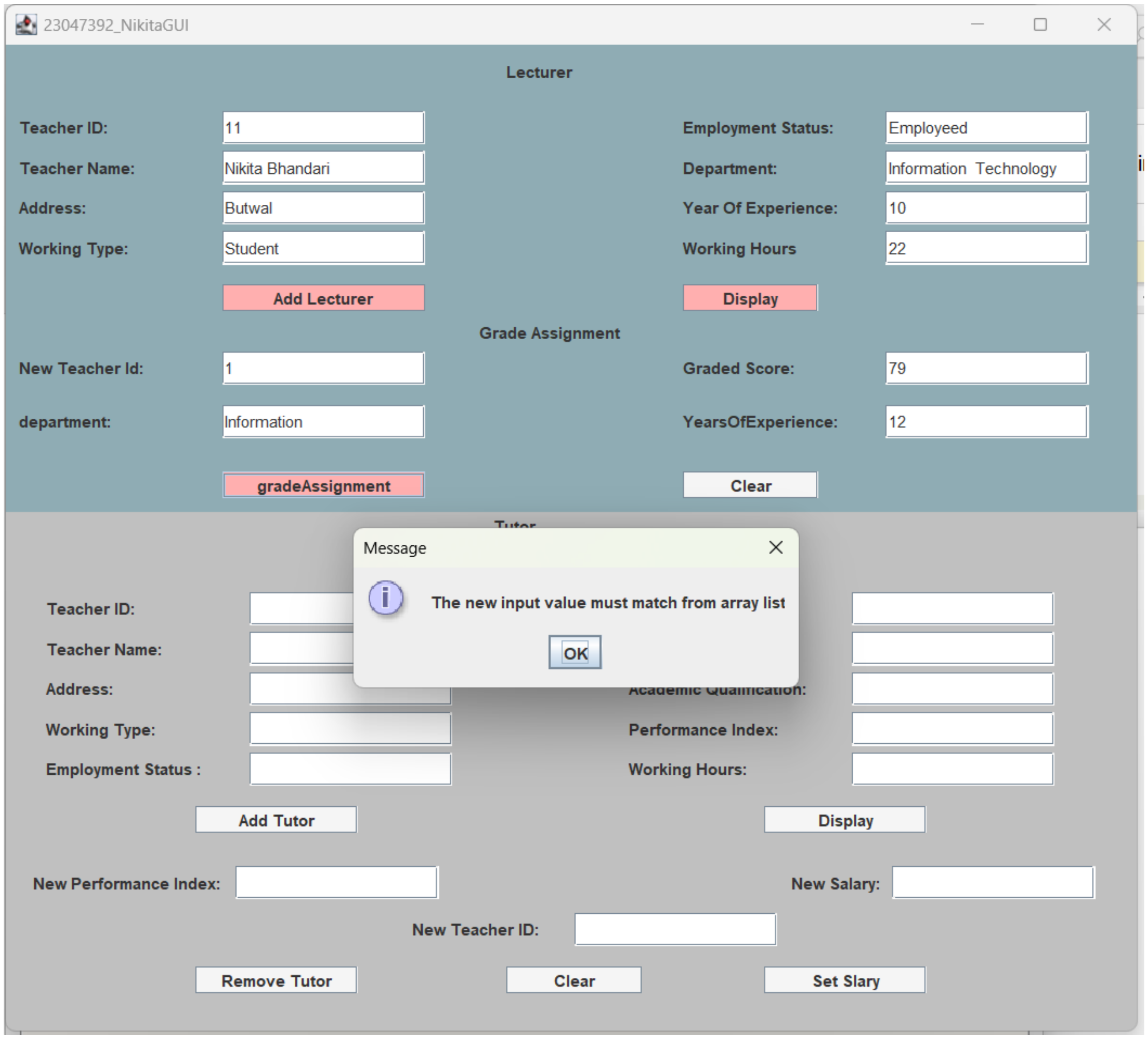


Figure 19: Graded Score of Lecturer



```
Teacher Id=11
Teacher name=Nikita Bhandari
Address=Butwal
Working Type=Studdent
Employment Status=Employeed
Working Hours=22
Department=Information Technology
Years of Experience=10
Graded Score= Not graded yet


The grade is assigned to A
Teacher Id=11
Teacher name=Nikita Bhandari
Address=Butwal
Working Type=Studdent
Employment Status=Employeed
Working Hours=22
Department=Information Technology
Years of Experience=10
Graded Score=79
```

Figure 20: Display of graded Score



23047392_NikitaGUI

Lecturer

Teacher ID: 11

Teacher Name: Nikita Bhandari

Address: Butwal

Working Type: Student

Employment Status: Employeed

Department: Information Technology

Year Of Experience: 10

Working Hours 22

Grade Assignment

New Teacher Id: 1

department: Information

Graded Score: 79

YearsOfExperience: 12

Message

The new input value must match from array list

Teacher ID:

Teacher Name:

Address:

Working Type:

Employment Status :

Academic Qualification:

Performance Index:

Working Hours:

New Performance Index:

New Salary:

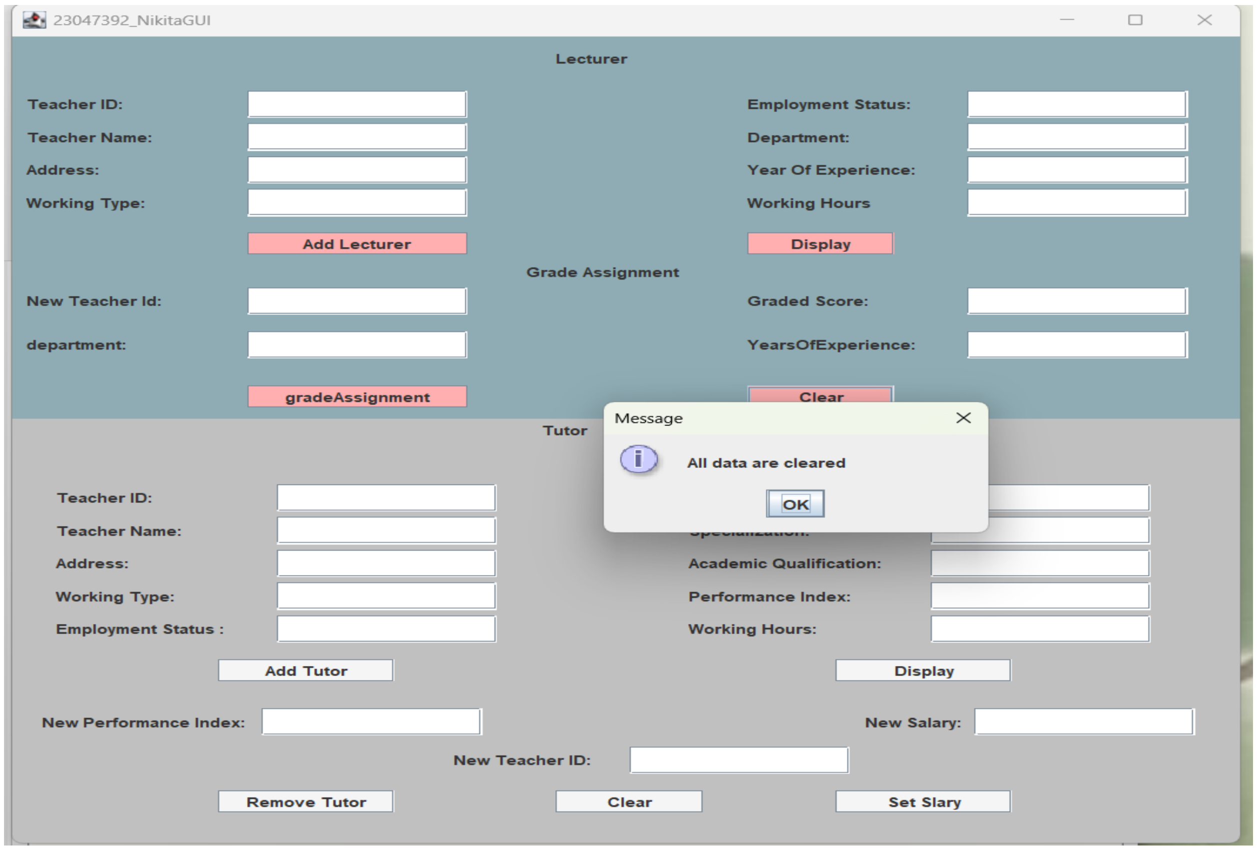
New Teacher ID:

Figure 21: Error in gardeAssignment

5.4 Testing of Clear Button of Lecturer

Objective	To clear
Action	The clear button is pressed
Expected Result	All the buttons should be clear every text field properly.
Actual Result	The button did work properly.
Conclusion	The test was successful.

Table 6: Table Of clear Lecturer



Message

All data are cleared

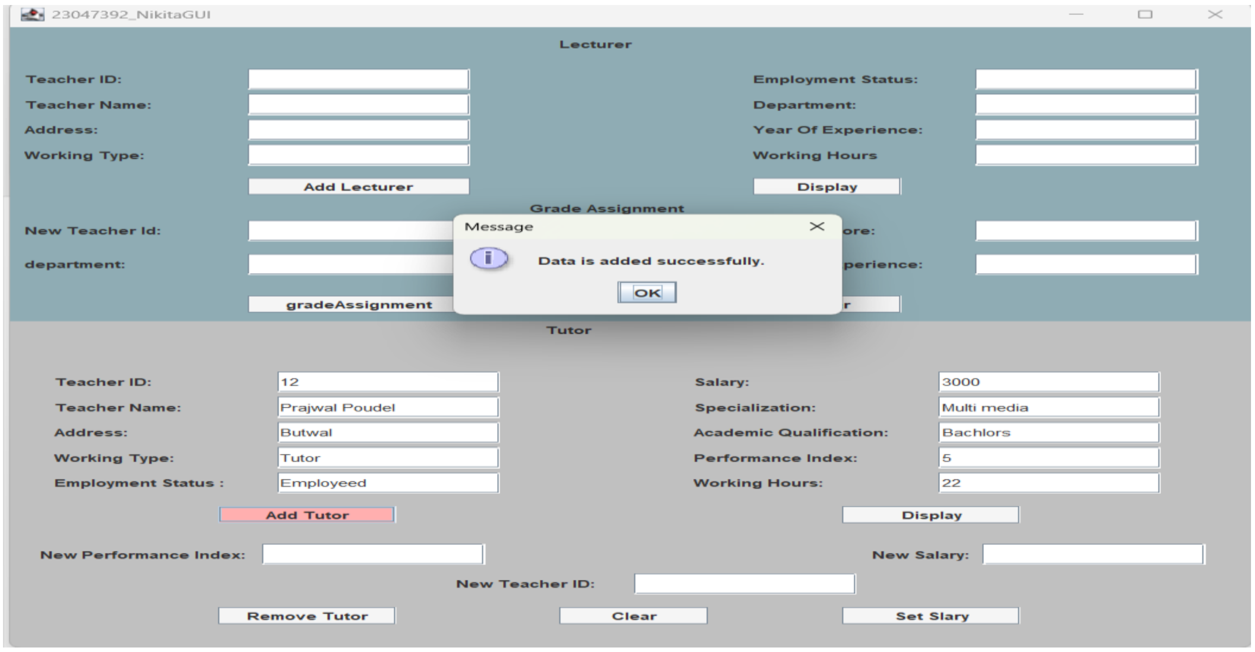
OK

Figure 22: Clear of Lecturer

5.5 Testing of add Tutor and display

Objective	Add Tutor and display it
Action	The values are entered in the text field Teacher ID: 12 Teacher Name: Prajwal Poudel Address: Butwal Working Type: Tutor Employment Status: Employeed Salary : 3000 Specialization: Multi media Academic Qualification: Bachlors Performance Index: 5 Working Hours 22 Then the add button is pressed and finally the display.
Expected Result	All the buttons should be working properly.
Actual Result	The button did work properly.
Conclusion	The test was successful.

Table 7: Table of Add and Display Tutor


23047392_NikitaGUI

Lecturer

Teacher ID:
Teacher Name:
Address:
Working Type:
Employment Status:
Department:
Year Of Experience:
Working Hours

Grade Assignment

New Teacher Id:
department:

Message

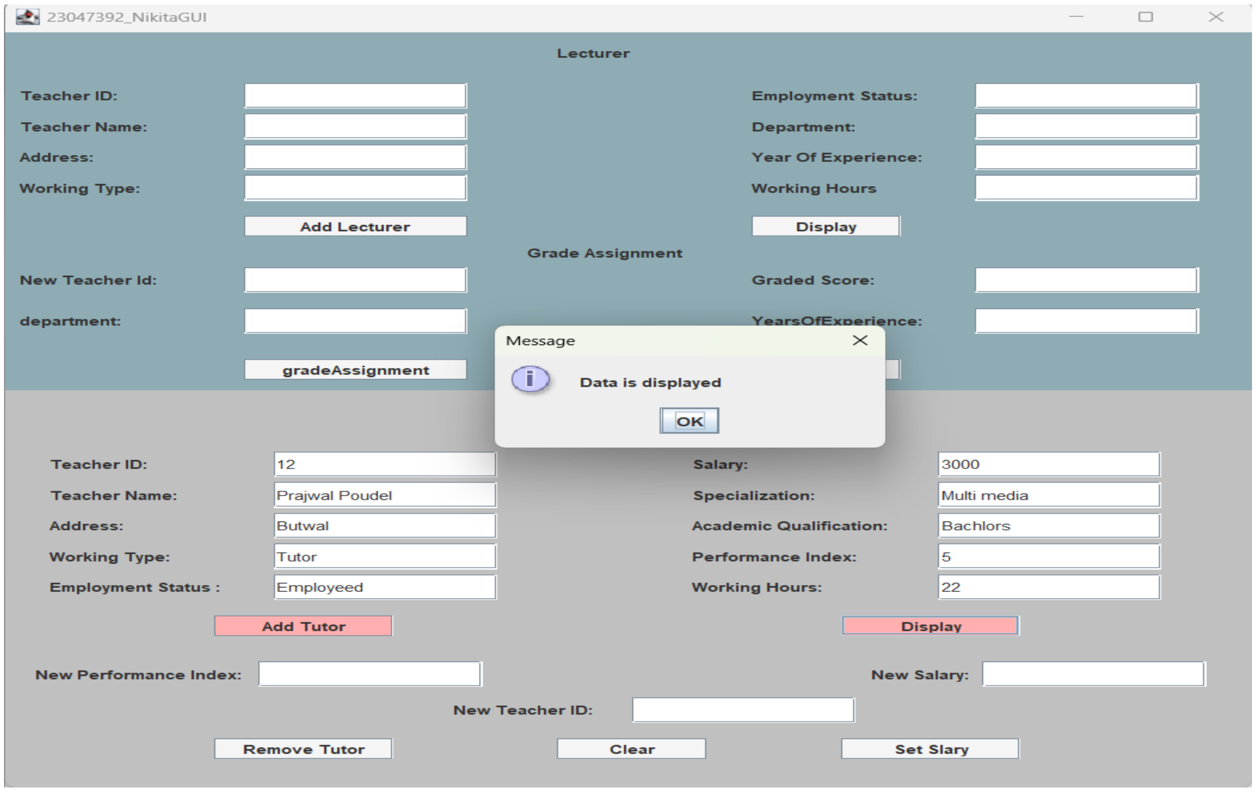
Data is added successfully.

Tutor

Teacher ID: 12
Teacher Name: Prajwal Poudel
Address: Butwal
Working Type: Tutor
Employment Status : Employeed
Salary: 3000
Specialization: Multi media
Academic Qualification: Bachlors
Performance Index: 5
Working Hours: 22

New Performance Index:
New Salary:
New Teacher ID:

Figure 23: Add of Tutor


23047392_NikitaGUI

Lecturer

Teacher ID:
Teacher Name:
Address:
Working Type:
Employment Status:
Department:
Year Of Experience:
Working Hours

Grade Assignment

New Teacher Id:
department:
Graded Score:
YearsOfExperience:

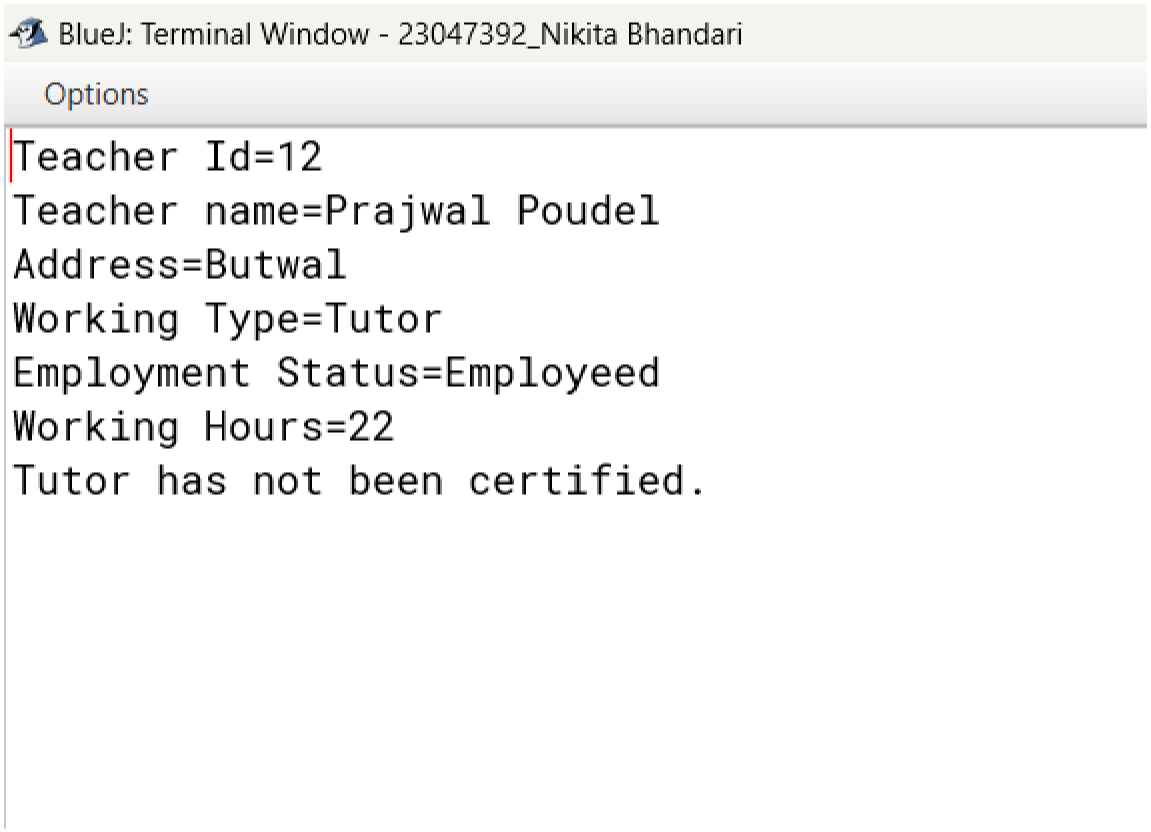
Message

Data is displayed

Teacher ID: 12
Teacher Name: Prajwal Poudel
Address: Butwal
Working Type: Tutor
Employment Status : Employeed
Salary: 3000
Specialization: Multi media
Academic Qualification: Bachlors
Performance Index: 5
Working Hours: 22

New Performance Index:
New Salary:
New Teacher ID:

Figure 24: Displaay of Tutor



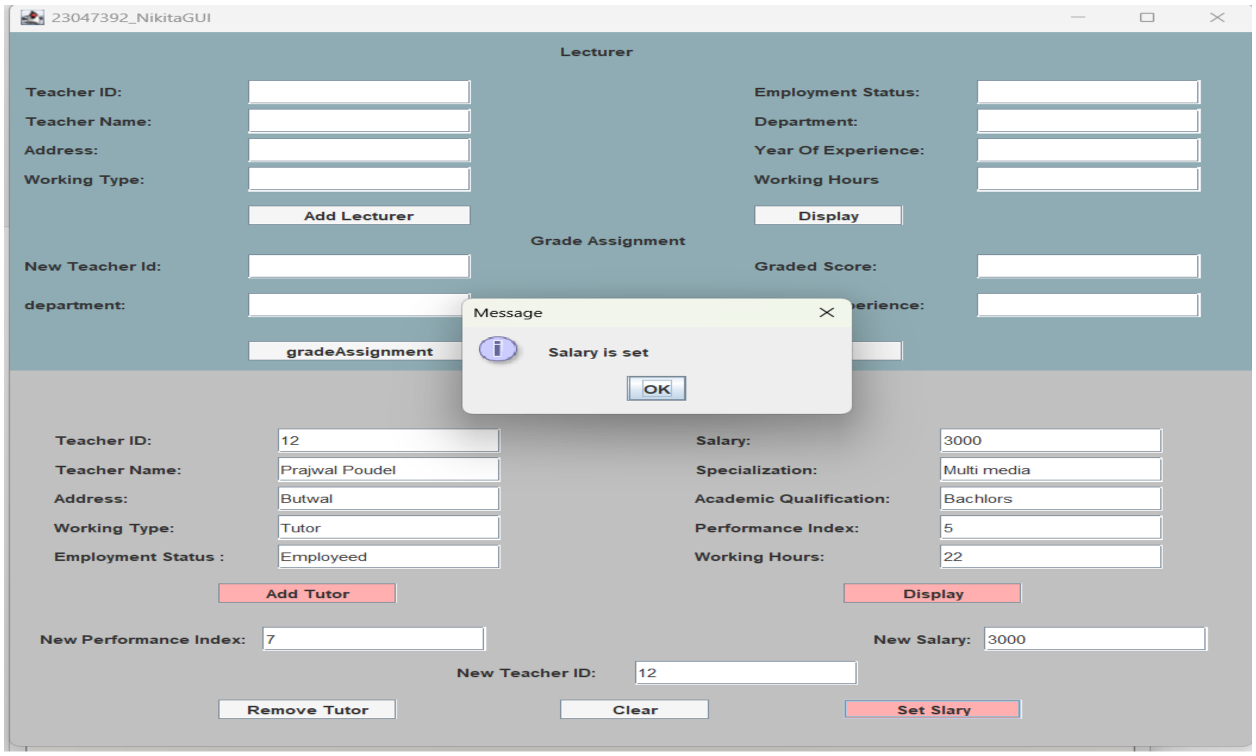
```
Teacher Id=12
Teacher name=Prajwal Poudel
Address=Butwal
Working Type=Tutor
Employment Status=Employeed
Working Hours=22
Tutor has not been certified.
```

Figure 25: Display data of Tutor

5.6 Testing of New Salary

Objective	To set new salary.
Action	The values are entered in the text field Then the add button is pressed and finally the display. New Teacher Id: 12 Salary: 3000 New Performance Index: 7 Then the new salary button is pressed.
Expected Result	All the buttons should be working properly.
Actual Result	The button did work properly.
Conclusion	The test was successful.

Table 8: Table of set salary



Lecturer

Teacher ID:		Employment Status:	
Teacher Name:		Department:	
Address:		Year Of Experience:	
Working Type:		Working Hours	

Grade Assignment

New Teacher Id:		Graded Score:	
department:		erience:	

Message

Salary is set

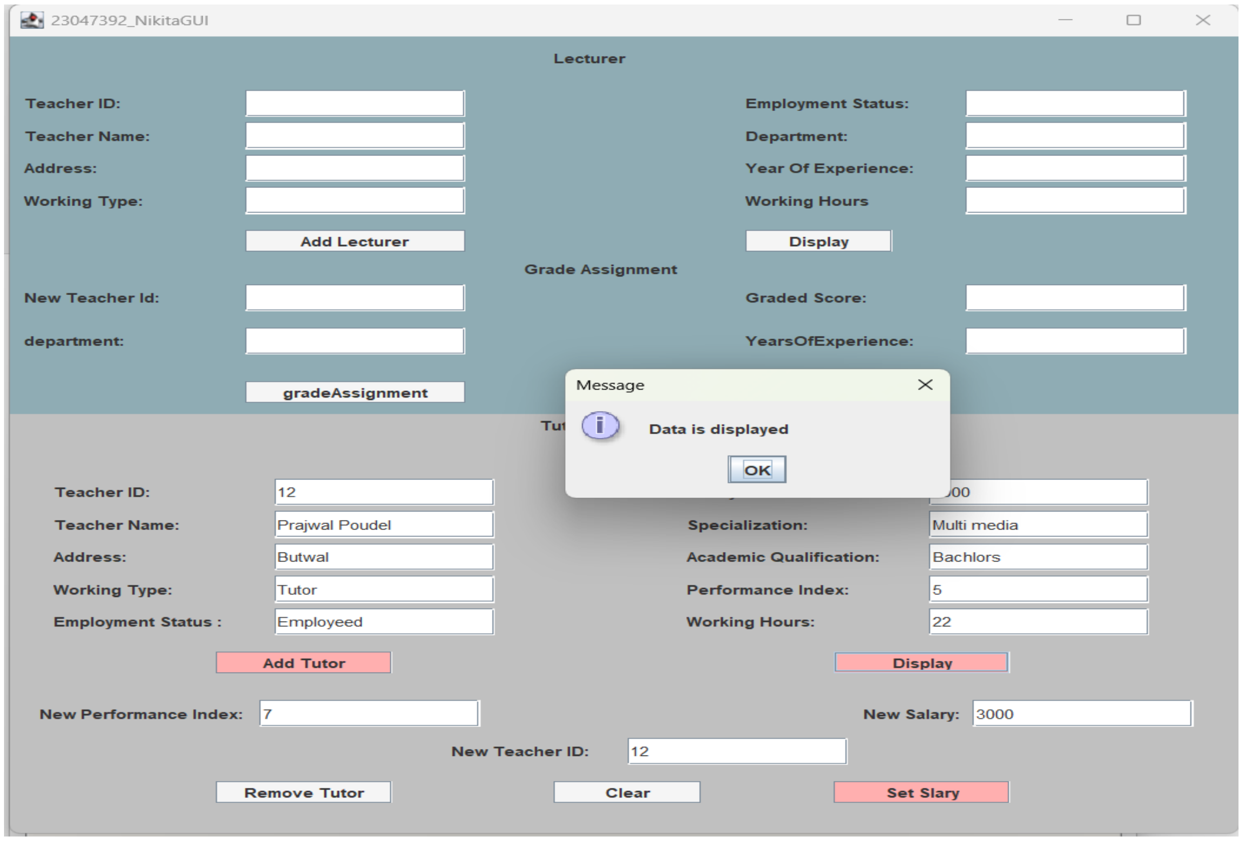
Teacher ID:	12	Salary:	3000
Teacher Name:	Prajwal Poudel	Specialization:	Multi media
Address:	Butwal	Academic Qualification:	Bachlors
Working Type:	Tutor	Performance Index:	5
Employment Status :	Employeed	Working Hours:	22

New Performance Index: 7

New Salary: 3000

New Teacher ID: 12

Figure 26: Set salary of Tutor



23047392_NikitaGUI

Lecturer

Teacher ID:		Employment Status:	
Teacher Name:		Department:	
Address:		Year Of Experience:	
Working Type:		Working Hours	

Add Lecturer   Display

Grade Assignment

New Teacher Id:		Graded Score:	
department:		YearsOfExperience:	

gradeAssignment

Message

Data is displayed

OK

Teacher ID:	12		00
Teacher Name:	Prajwal Poudel	Specialization:	Multi media
Address:	Butwal	Academic Qualification:	Bachlors
Working Type:	Tutor	Performance Index:	5
Employment Status :	Employeed	Working Hours:	22

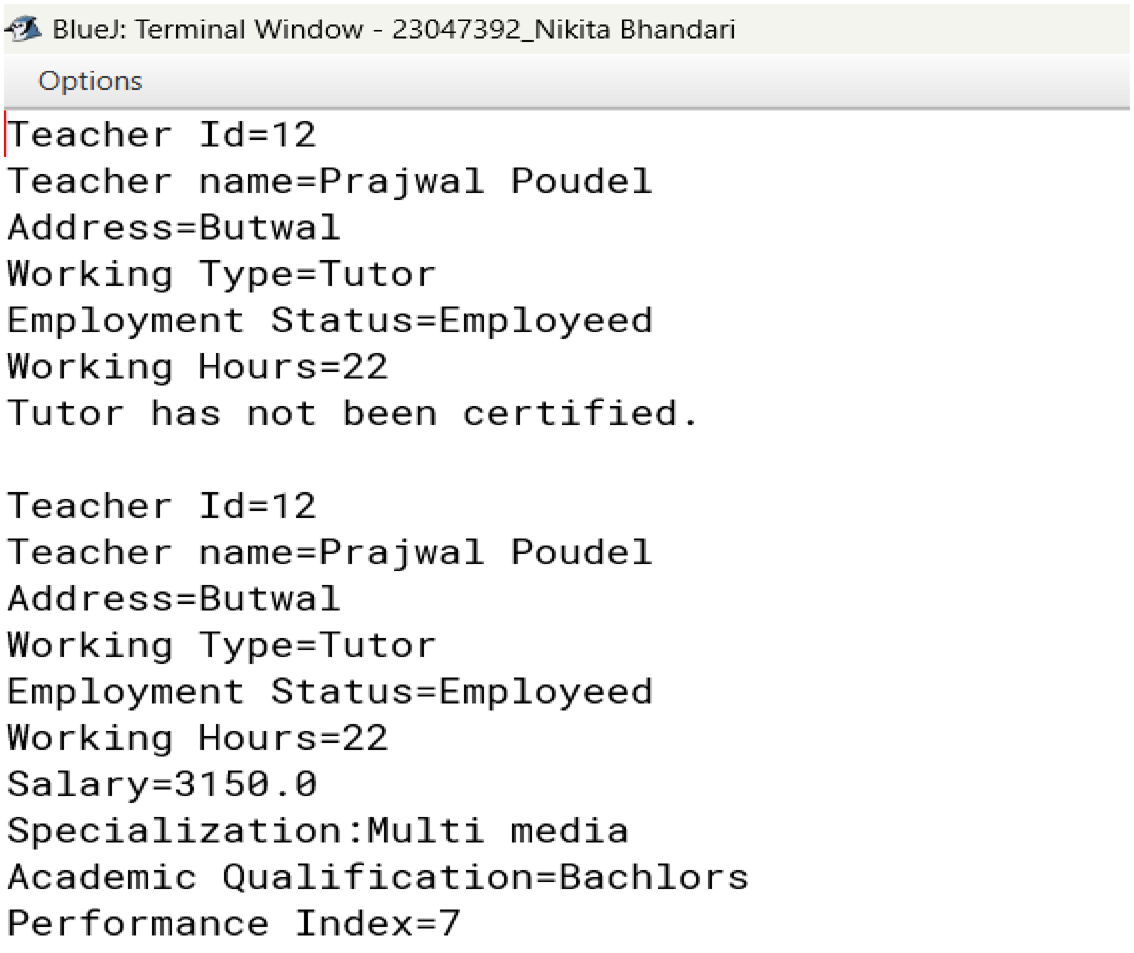
Add Tutor   Display

New Performance Index: 7   New Salary: 3000

New Teacher ID: 12

Remove Tutor   Clear   Set Slary

Figure 27: Display of set salary



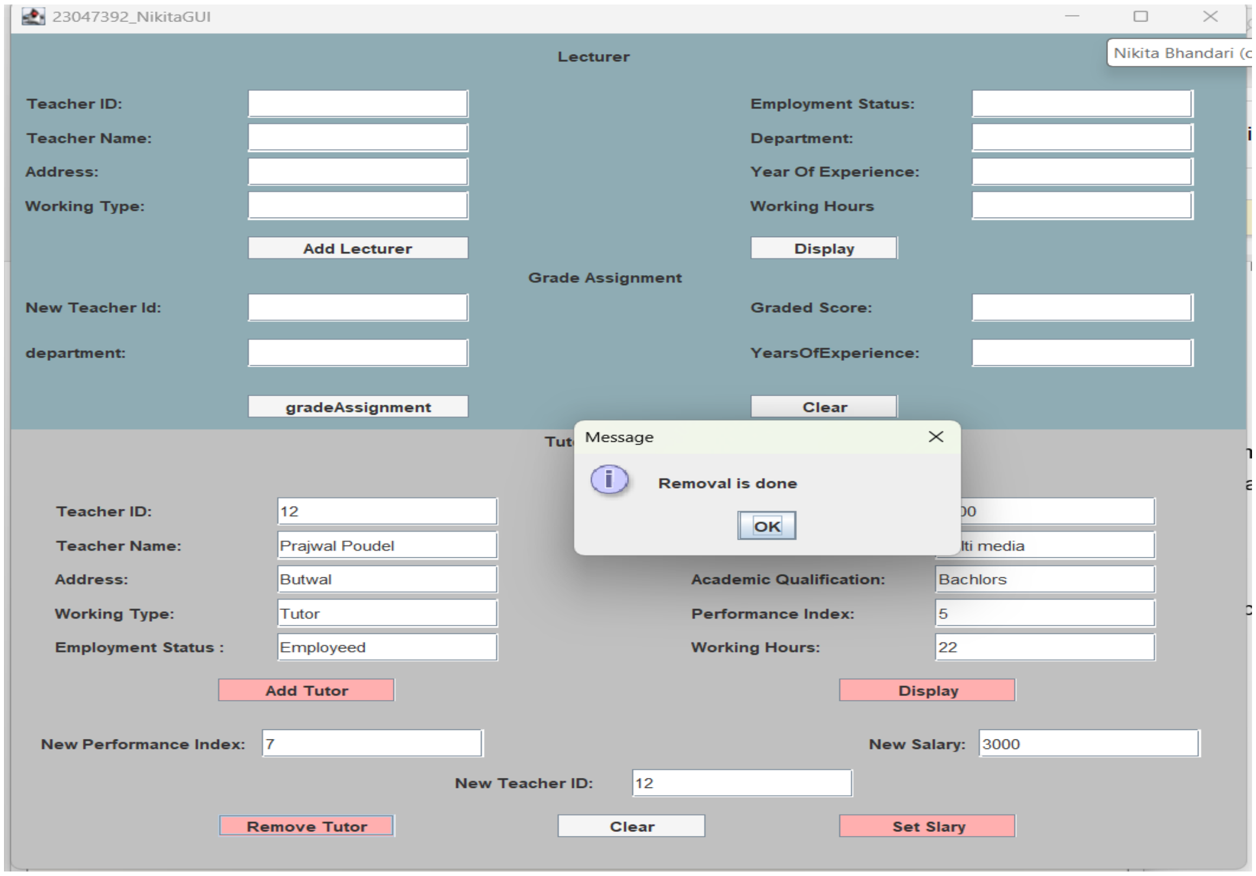
```
Teacher Id=12
Teacher name=Prajwal Poudel
Address=Butwal
Working Type=Tutor
Employment Status=Employeed
Working Hours=22
Tutor has not been certified.

Teacher Id=12
Teacher name=Prajwal Poudel
Address=Butwal
Working Type=Tutor
Employment Status=Employeed
Working Hours=22
Salary=3150.0
Specialization:Multi media
Academic Qualification=Bachlors
Performance Index=7
```

Figure 28: Salary being Displayed

5.7 Testing of Remove Tutor

Objective	To remove tutor.
Action	Then the remove salary button is pressed.
Expected Result	All the buttons should remove tutor properly.
Actual Result	The button did work properly.
Conclusion	The test was successful.

Table 9: Table if remove Tutor


23047392_NikitaGUI

Nikita Bhandari (c

Lecturer

Teacher ID:
Teacher Name:
Address:
Working Type:
Employment Status:
Department:
Year Of Experience:
Working Hours

Grade Assignment

New Teacher Id:
department:
Graded Score:
YearsOfExperience:

Tut

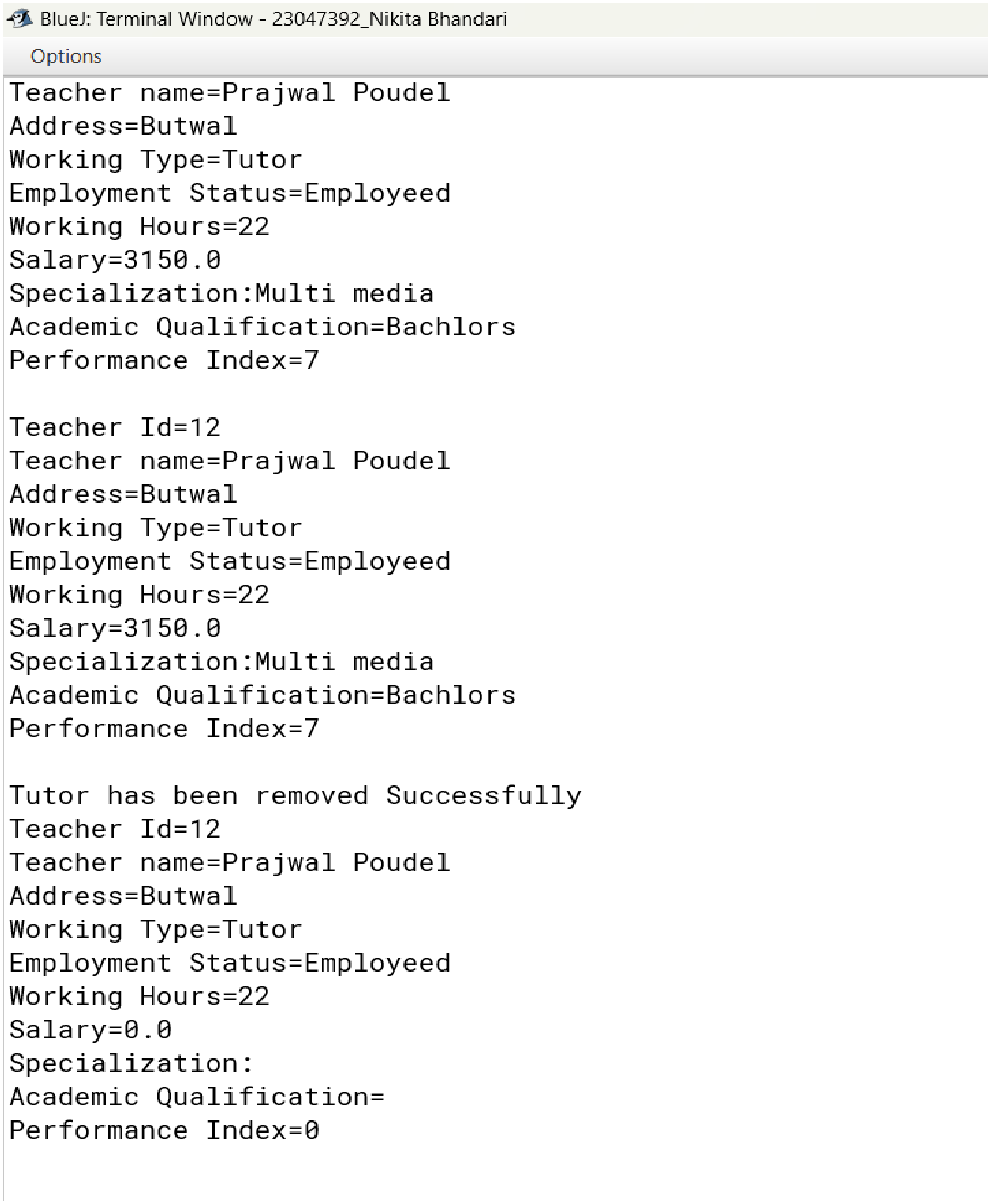
Message

Removal is done

Teacher ID: 12
Teacher Name: Prajwal Poudel
Address: Butwal
Working Type: Tutor
Employment Status : Employeed
00
lti media
Academic Qualification: Bachlors
Performance Index: 5
Working Hours: 22

New Performance Index: 7
New Salary: 3000
New Teacher ID: 12

Figure 29: Remove of Tutor


BlueJ: Terminal Window - 23047392_Nikita Bhandari

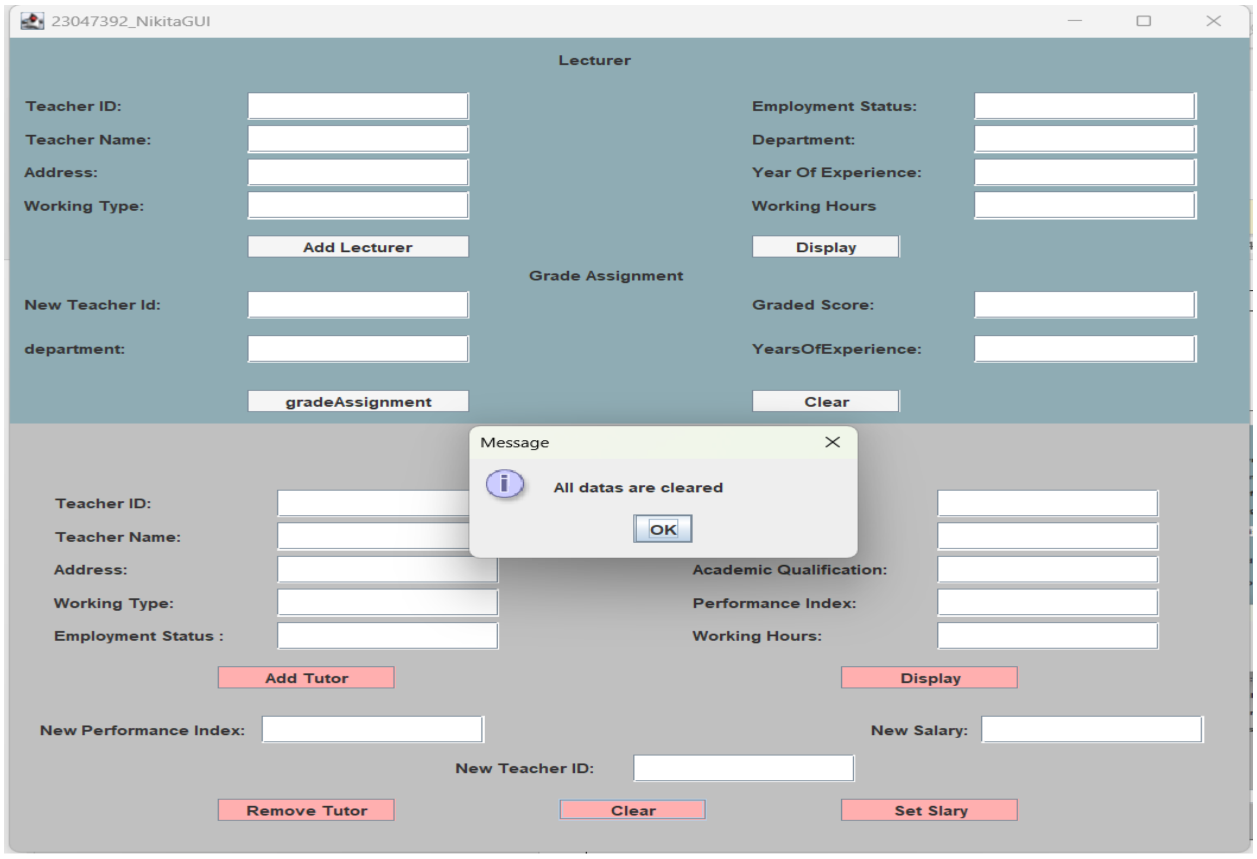
Options

```
Teacher name=Prajwal Poudel
Address=Butwal
Working Type=Tutor
Employment Status=Employeed
Working Hours=22
Salary=3150.0
Specialization:Multi media
Academic Qualification=Bachlors
Performance Index=7

Teacher Id=12
Teacher name=Prajwal Poudel
Address=Butwal
Working Type=Tutor
Employment Status=Employeed
Working Hours=22
Salary=3150.0
Specialization:Multi media
Academic Qualification=Bachlors
Performance Index=7

Tutor has been removed Successfully
Teacher Id=12
Teacher name=Prajwal Poudel
Address=Butwal
Working Type=Tutor
Employment Status=Employeed
Working Hours=22
Salary=0.0
Specialization:
Academic Qualification=
Performance Index=0
```

Figure 30: Remove Tutor Display

5.8 Test to clear tutor

Objective	To clear
Action	The clear button is pressed
Expected Result	All the buttons should be clear every text field properly.
Actual Result	The button did work properly.
Conclusion	The test was successful.



All datas are cleared

Figure 31: Clear Tutor

Table 10: Table of Clear Tutor

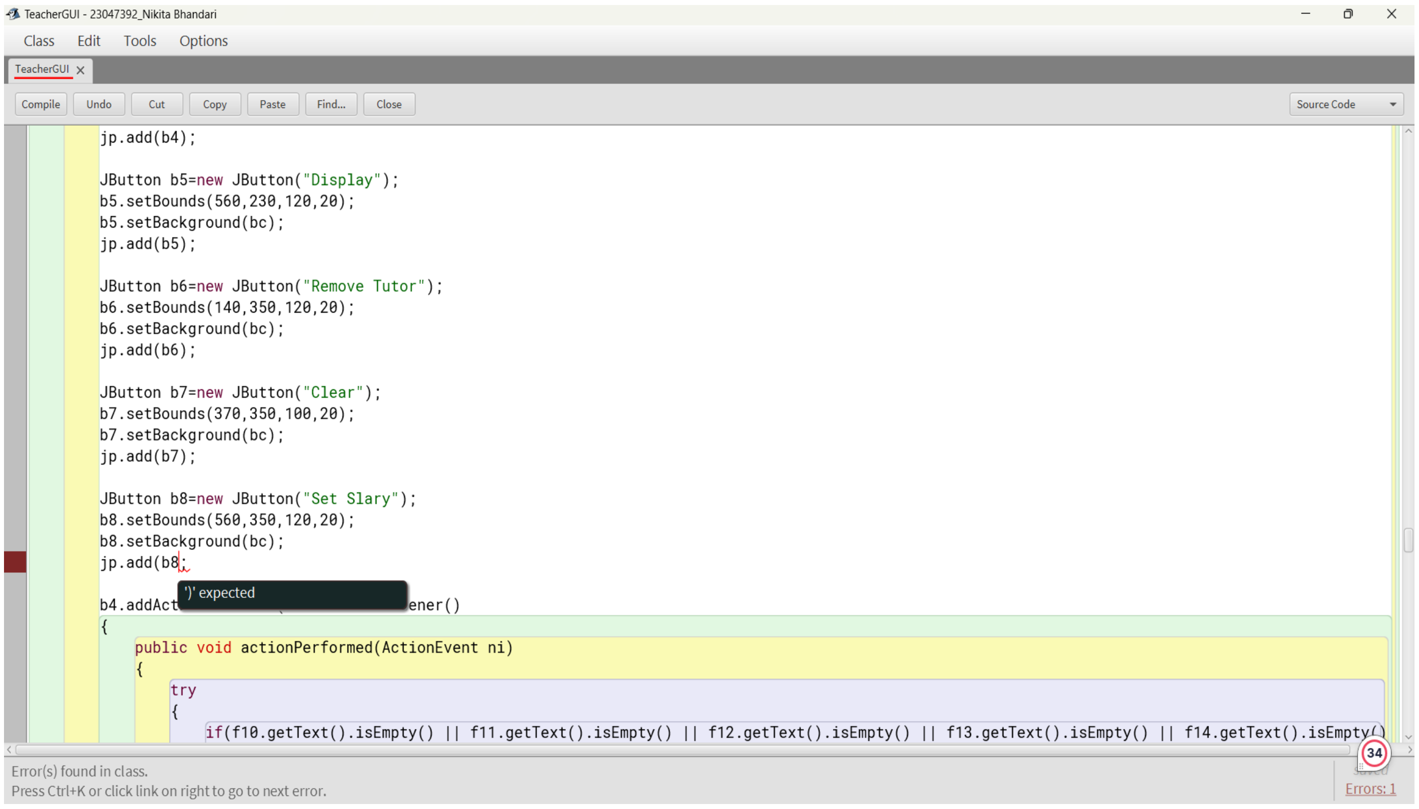
6 Error Detection

Errors happen unknowingly or sometimes due to the malfunction of the program.

6.1 Syntax Error

Syntax errors occur when there is a mistake in the programming language syntax.

Error detection: A bracket was detected to be missed i.e)



TeacherGUI - 23047392_Nikita Bhandari
Class Edit Tools Options
TeacherGUI
Compile Undo Cut Copy Paste Find... Close
Source Code

```
jp.add(b4);

JButton b5=new JButton("Display");
b5.setBounds(560,230,120,20);
b5.setBackground(bc);
jp.add(b5);

JButton b6=new JButton("Remove Tutor");
b6.setBounds(140,350,120,20);
b6.setBackground(bc);
jp.add(b6);

JButton b7=new JButton("Clear");
b7.setBounds(370,350,100,20);
b7.setBackground(bc);
jp.add(b7);

JButton b8=new JButton("Set Slary");
b8.setBounds(560,350,120,20);
b8.setBackground(bc);
jp.add(b8;

b4.addAct                          ener()
{
    public void actionPerformed(ActionEvent ni)
    {
        try
        {
            if(f10.getText().isEmpty() || f11.getText().isEmpty() || f12.getText().isEmpty() || f13.getText().isEmpty() || f14.getText().isEmpty()
```

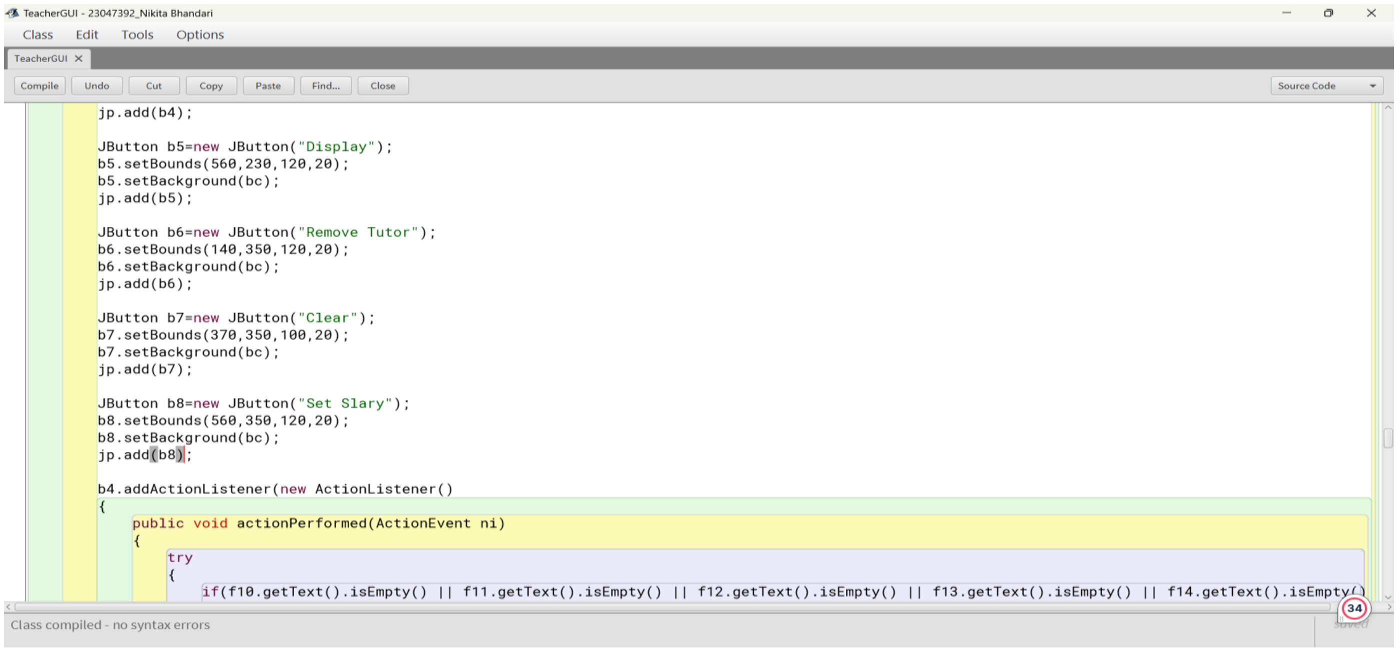
')' expected

Error(s) found in class.
Press Ctrl+K or click link on right to go to next error.

Errors: 1

Figure 32: Error detection of syntax

Error correction: A bracket) was added to correct the error.



TeacherGUI - 23047392_Nikita Bhandari
Class Edit Tools Options
TeacherGUI
Compile Undo Cut Copy Paste Find... Close
Source Code

```
jp.add(b4);

JButton b5=new JButton("Display");
b5.setBounds(560,230,120,20);
b5.setBackground(bc);
jp.add(b5);

JButton b6=new JButton("Remove Tutor");
b6.setBounds(140,350,120,20);
b6.setBackground(bc);
jp.add(b6);

JButton b7=new JButton("Clear");
b7.setBounds(370,350,100,20);
b7.setBackground(bc);
jp.add(b7);

JButton b8=new JButton("Set Slary");
b8.setBounds(560,350,120,20);
b8.setBackground(bc);
jp.add(b8);

b4.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ni)
    {
        try
        {
            if(f10.getText().isEmpty() || f11.getText().isEmpty() || f12.getText().isEmpty() || f13.getText().isEmpty() || f14.getText().isEmpty()
```

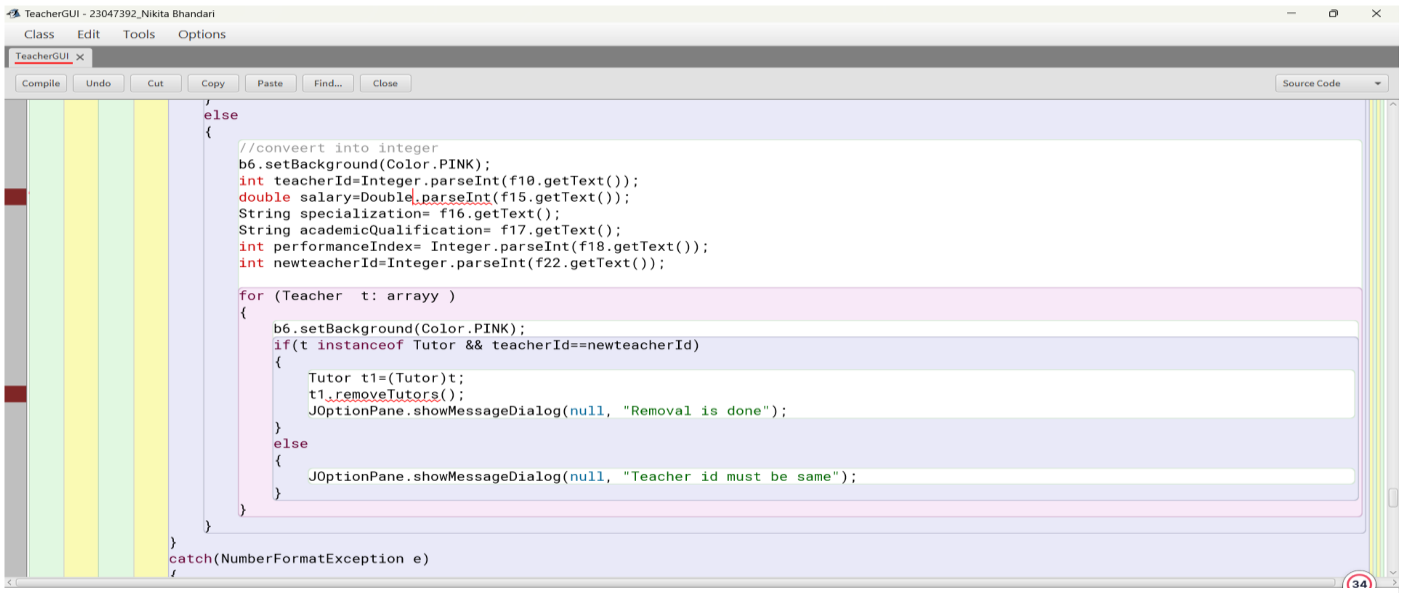
Class compiled - no syntax errors

Figure 33: Error correction of syntax error

6.2 Semantic error

These errors are found when compiled

Error detection: There were two undeclared methods i.e. Double instead of Integer and removeTutors() instead of removeTutor().

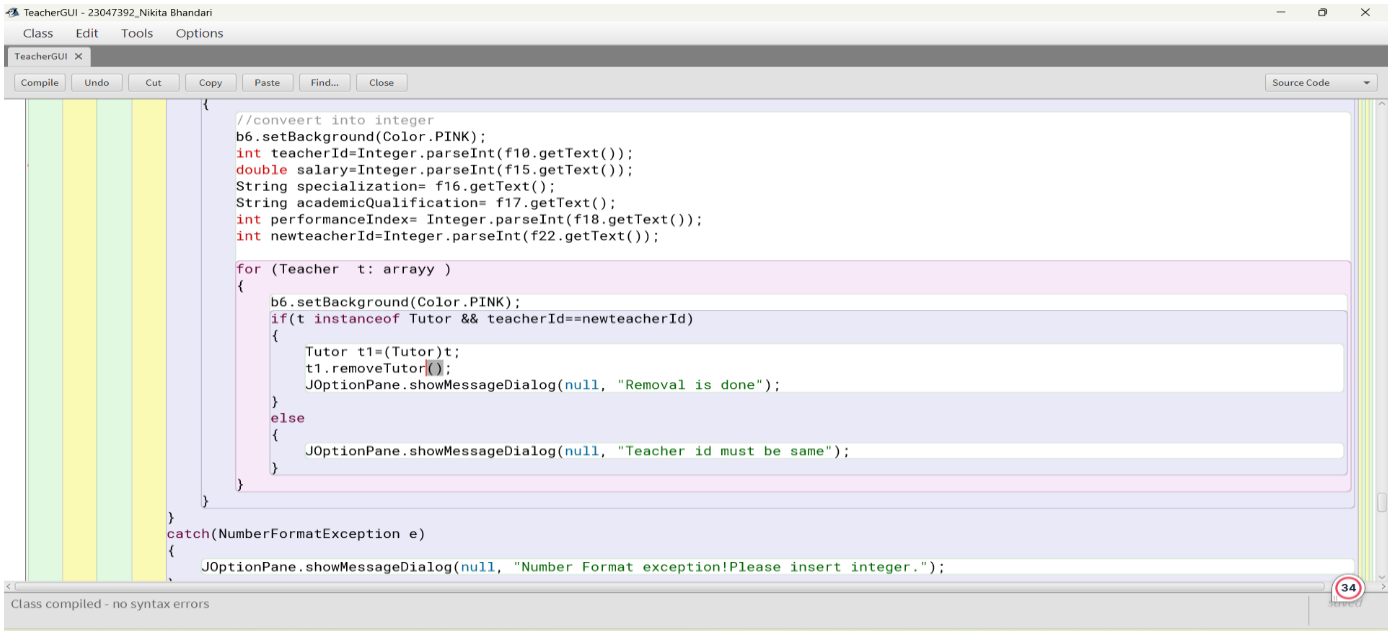


TeacherGUI - 23047392_Nikita Bhandari
Class Edit Tools Options
TeacherGUI
Compile Undo Cut Copy Paste Find... Close
Source Code

```
    else
    {
        //conveert into integer
        b6.setBackground(Color.PINK);
        int teacherId=Integer.parseInt(f10.getText());
        double salary=Double.parseInt(f15.getText());
        String specialization= f16.getText();
        String academicQualification= f17.getText();
        int performanceIndex= Integer.parseInt(f18.getText());
        int newteacherId=Integer.parseInt(f22.getText());

        for (Teacher  t: arrayy )
        {
            b6.setBackground(Color.PINK);
            if(t instanceof Tutor && teacherId==newteacherId)
            {
                Tutor t1=(Tutor)t;
                t1.removeTutors();
                JOptionPane.showMessageDialog(null, "Removal is done");
            }
            else
            {
                JOptionPane.showMessageDialog(null, "Teacher id must be same");
            }
        }
    }
}
catch(NumberFormatException e)
```

Figure 34: Error Detection of semantic error

Error Correction: The correct datatype i.e. Integer has been used and the correct method removeTeacher has been used.


TeacherGUI - 23047392_Nikita Bhandari
Class Edit Tools Options
TeacherGUI
Compile Undo Cut Copy Paste Find... Close
Source Code

```
{
    //conveert into integer
    b6.setBackground(Color.PINK);
    int teacherId=Integer.parseInt(f10.getText());
    double salary=Integer.parseInt(f15.getText());
    String specialization= f16.getText();
    String academicQualification= f17.getText();
    int performanceIndex= Integer.parseInt(f18.getText());
    int newteacherId=Integer.parseInt(f22.getText());

    for (Teacher  t: arrayy )
    {
        b6.setBackground(Color.PINK);
        if(t instanceof Tutor && teacherId==newteacherId)
        {
            Tutor t1=(Tutor)t;
            t1.removeTutor();
            JOptionPane.showMessageDialog(null, "Removal is done");
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Teacher id must be same");
        }
    }
}
}
catch(NumberFormatException e)
{
    JOptionPane.showMessageDialog(null, "Number Format exception!Please insert integer.");
```

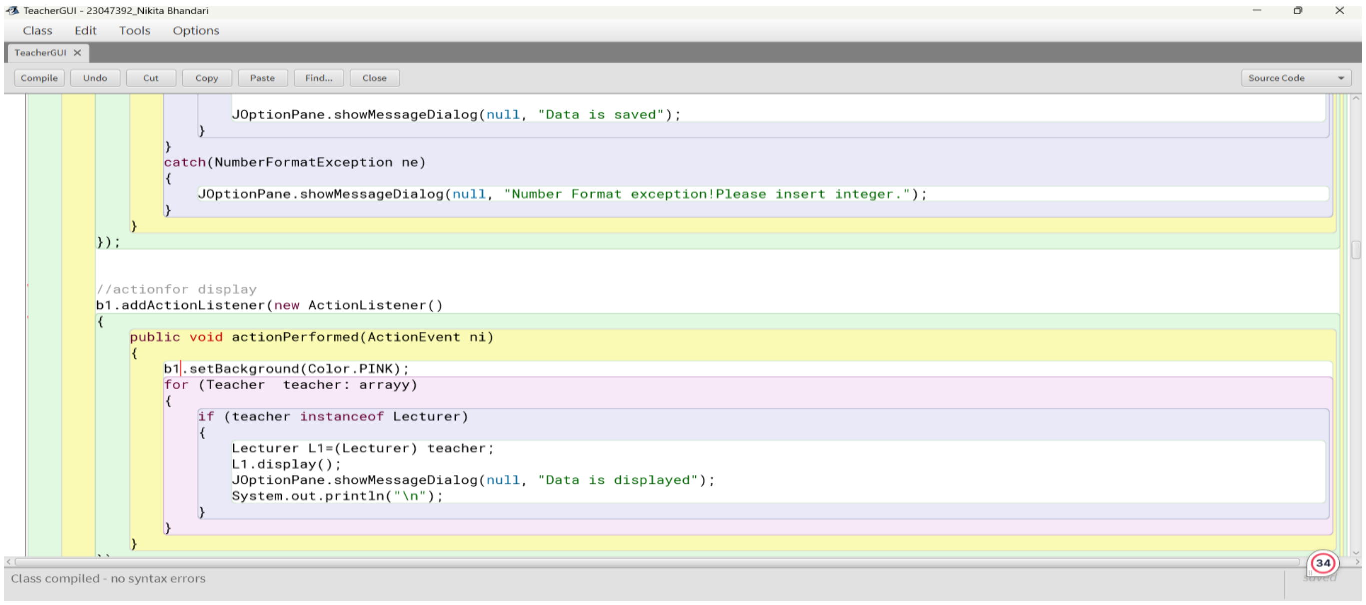
Class compiled - no syntax errors

Figure 35: Error Correction OF sematic error

6.3 Logical error

These types of errors are correct syntactically but wrong during showing output.

Error Detection: In the button b1 b2's color has been changed which means when the button is clicked there will be change in b2.

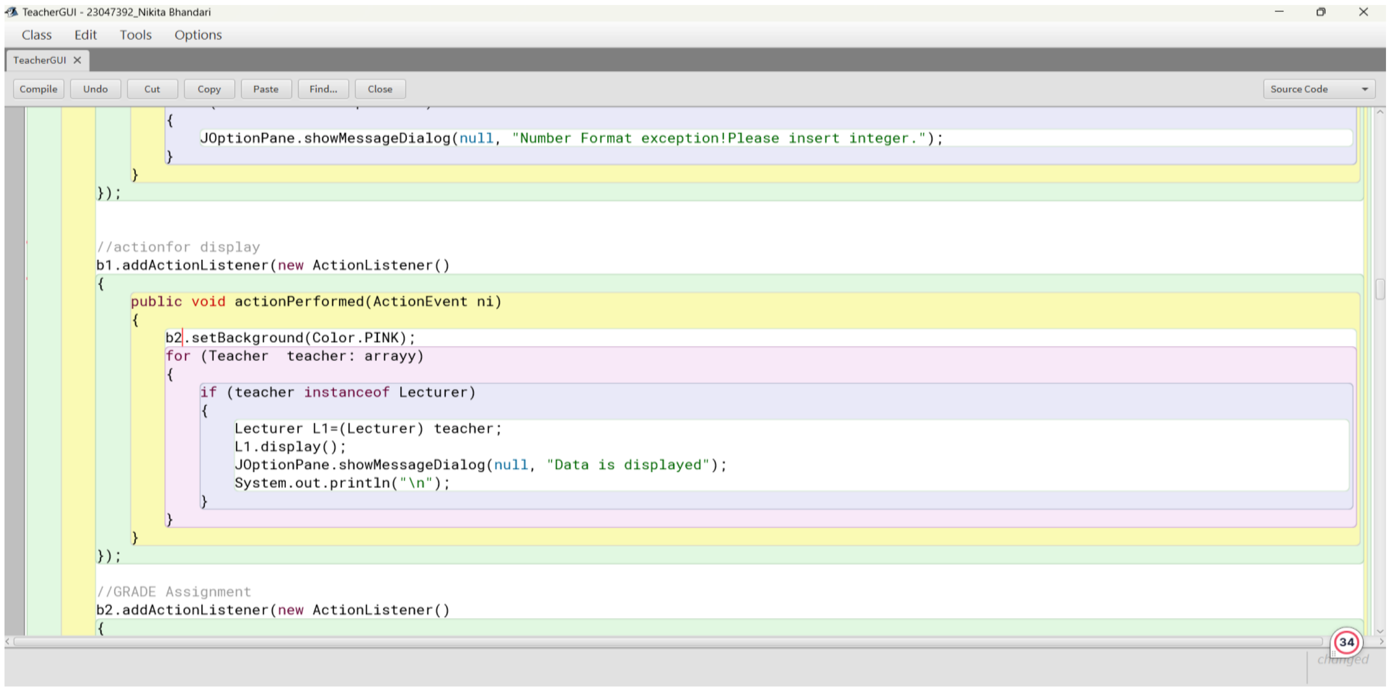

TeacherGUI - 23047392_Nikita Bhandari

```
                JOptionPane.showMessageDialog(null, "Data is saved");
            }
        }
        catch(NumberFormatException ne)
        {
            JOptionPane.showMessageDialog(null, "Number Format exception!Please insert integer.");
        }
    }
});

//actionfor display
b1.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ni)
    {
        b1.setBackground(Color.PINK);
        for (Teacher  teacher: arrayy)
        {
            if (teacher instanceof Lecturer)
            {
                Lecturer L1=(Lecturer) teacher;
                L1.display();
                JOptionPane.showMessageDialog(null, "Data is displayed");
                System.out.println("\n");
            }
        }
    }
```

Class compiled - no syntax errors

Figure 36: Error Detection of Logical Error

Error Correction: The b1 color will be change when b1 is clicked.


TeacherGUI - 23047392_Nikita Bhandari

```
        {
            JOptionPane.showMessageDialog(null, "Number Format exception!Please insert integer.");
        }
    }
});

//actionfor display
b1.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ni)
    {
        b2.setBackground(Color.PINK);
        for (Teacher  teacher: arrayy)
        {
            if (teacher instanceof Lecturer)
            {
                Lecturer L1=(Lecturer) teacher;
                L1.display();
                JOptionPane.showMessageDialog(null, "Data is displayed");
                System.out.println("\n");
            }
        }
    }
});

//GRADE Assignment
b2.addActionListener(new ActionListener()
{
```

Figure 37: Error Correction Of Logical Error

7 .Conclusion

The end of the coursework has provided me with the knowledge of making a Graphical User Interface (GUI). Different software tools including Blue J, Draw.io, and MS Word were utilized to complete the coursework, enhancing my proficiency in using them. The assignment was to add a new class to the previous coursework so that a GUI could be created that will store the details of teachers and also details in an ArrayList. The GUI can allow users to input values in the text.

After this coursework, I was confident enough to develop GUI. I learned how to design and implement the GUI in Java practically. I have experience integrating them with pre-existing classes and using them to make a fully functional application. Now I am, familiar with various Swing components, like JTextFields, JButtons, and JLabels, and how to handle the user input and interactions by using ActionListeners for the respective buttons. Apart from this, I have also learned on how to handle exceptions such as NumberFormatExceptions using try-catch blocks while converting text to numeric data types.

For me, the difficult part was to make the GUI, more aesthetic and user-friendly. It was hard to align them properly, adjusting size and positions. It was also difficult for me to implement an action listener for buttons and error handling. Integrating the current GUI with the previous classes i.e. Teacher, Tutor, and Lecturer was also difficult. Taking multiple screenshot of the testing part and error detection was very chaotic and full of confusion. Even the pseudocode was hard to write.

Overall, As a result, this coursework was fun to do. After some research and guidance from the teacher, I was able to complete my work in time. This coursework is the future roadmap for my GUI journey.

8. Appendix

8.1 Teacher

```
//teacher class is created
public class Teacher
{
    //attribute is created
    private int teacherId;
    private String teacherName;
    private String address;
    private String workingType;
    private String employmentStatus;
    private int workingHours;
    //Parameterized constructor is created
    public Teacher( int teacherId,String teacherName,String address,String
workingType,String employmentStatus,int workingHours)
    {
        this.teacherId = teacherId;
        this.teacherName = teacherName;
        this.address = address;
        this.workingType = workingType;
        this.employmentStatus = employmentStatus;
        this.workingHours= workingHours;
    }
    //gettter or accessor method for attributes
    public int getTeacherId()
    {
        return teacherId;
    }
    public String getTeacherName()
    {
        return teacherName;
    }
    public String getAddress()
    {
        return address;
    }
    public String getWorkingType()
    {
        return workingType;
    }
    public String getEmploymentStatus()
    {
        return employmentStatus;
    }
    public int getWorkingHours()
```

```
{
    return workingHours;
}
//mutator or setter method to set workinghours
public void setWorkingHours(int newWorkingHours)
{
    this.workingHours = newWorkingHours;
}
//method to display teachers details
public void display()
{
    System.out.println("Teacher Id="+this.teacherId);
    System.out.println("Teacher name="+this.teacherName);
    System.out.println("Address="+this.address);
    System.out.println("Working Type="+this.workingType);
    System.out.println("Employment Status="+this.employmentStatus);
    if(workingHours == 0)
    {
        System.out.println("Working Hours = WorkingHours is not assigned!");
    }
    else
    {
        System.out.println("Working Hours="+workingHours);
        }
    }
}
```

8.2 Lecturer

```
//lecturer is a sub class of teacher that inherits Teacher
public class Lecturer extends Teacher
{
    // additional attributes for Lecture class
    private String department;
    private int yearsOfExperience;
    private int gradedScore;
    private boolean hasGraded;
    //constructor with seven parameters is created
    public Lecturer( int teacherId,String teacherName,String address,String workingType,
    String employmentStatus,String department,int yearsOfExperience, int workingHours)
    {
        //a call is made to superclass constructor having five parameter

super(teacherId,teacherName,address,workingType,employmentStatus,workingHours);
        this.department = department;
        this.yearsOfExperience = yearsOfExperience;
        this.gradedScore = gradedScore;
        this.hasGraded = false;
    }
    //accessor or getter method for lectures class attributes
```

```
public String getDepartment()
{
    return department;
}
public int getYearsOfExperience()
{
    return yearsOfExperience;
}
public int getGradedScore()
{
    return gradedScore;
}
public boolean getHasGraded()
{
    return hasGraded;
}
//mutator or setter method for gradedScore attribute
public void setGradedScore(int newGradedScore)
{
    this.gradedScore = newGradedScore;
}
//method is created to grade assignment
public void gradeAssignment(int score,String Department,int YearsOfExperience)
```

```
{
    if (!hasGraded && yearsOfExperience >= 5 && department.equals(Department))
    {
        setGradedScore(score);
        if (score >= 70)
        {

            System.out.println("The grade is assigned to A");
        }
        else if
        (score >= 60)
        {

            System.out.println("The grade is assigned to B");
        }
        else if (score >= 50)
        {

            System.out.println("The grade is assigned to C");
        }
        else if (score >= 40)
        {
            System.out.println("The garde is assigned to D");
        }
        else
        {

            System.out.println("The grade is assigned to E");
        }
        //mark the assignment has been graded
        hasGraded = true;
    }
    else
    {
        //print message if the lecturer can't grade assignment
        System.out.println("Assignment is not graded!");
    }
}
//method to display details of Lecturer
public void display()
{
    //calling display method from superclass Teacher
    super.display();
    System.out.println("Department="+this.department);
    System.out.println("Years of Experience="+this.yearsOfExperience);
    if (gradedScore != 0
    )
    {
        System.out.println("Graded Score="+ gradedScore);
    }
    else
    {
        System.out.println("Graded Score= Not graded yet");
    }
}
}
```

8.3 Tutor

```
//tutor is subclass of lecturer
public class Tutor extends Teacher
{
    //Additional attribute is created
    private double salary;
    private String specialization;
    private String academicQualification;
    private int performanceIndex;
    private boolean isCertified;
    //parameterized constructor is created
    public Tutor(int teacherId,String teacherName,String address,String workingType,String employmentStatus
    ,int workingHours,double salary,String specialization,String academicQualification,int performanceIndex)
    {

super(teacherId,teacherName,address,workingType,employmentStatus,workingHours);
        this.salary = salary;
        this.specialization = specialization;
        this.academicQualification = academicQualification;
        this.performanceIndex = performanceIndex;
        this.isCertified = false;
    }
    //accessor method
    public double getSalary()
    {
        return salary;
    }
    public String getSpecialization()
    {
        return specialization;
    }
    public String getAcademicQualification()
    {
        return academicQualification;
    }
    public int getPerformanceIndex()
    {
        return performanceIndex;
    }
    public boolean isCertified()
    {
        return isCertified;
    }
    //setter method to set new salary
```

```
public void setSalaryAndCertification(double newsalary, int newPerformanceIndex)
{

    if (newPerformanceIndex > 5 && getWorkingHours() > 20)
    {
        double appraisalPercentage;
        performanceIndex= newPerformanceIndex;
        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 is not approved. Tutor is not certified yet.");
        }
    }
    //method to remove tutor
    public void removeTutor()
    {
        if(isCertified)
        {
            //Setting attributes to zero
            salary=0;
            specialization= "";
            academicQualification="";
            performanceIndex=0;
            //isCertified=false;
            System.out.println("Tutor has been removed Successfully");
        }
        else
        {
            System.out.println("Tutor has been certified.Removal is not allowed");
        }
    }
    //method to display details of the  tutor

    public void display()

    {

        super.display();

        if(isCertified)


        {

            System.out.println("Salary=" +salary);

            System.out.println("Specialization:"+specialization);

            System.out.println("Academic Qualification="+academicQualification);

            System.out.println("Performance Index="+performanceIndex);

        }

        else

        {

            System.out.println("Tutor has not been certified.");


        }

    }

}
```

6.4 Teacher GUI

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

import java.awt.Color;

import java.util.ArrayList;




public class TeacherGUI

{

    private JFrame jf;

    private JPanel jp1, jp;

    private JLabel jl1, jl2, jl3, j1, j2, j3, j4, j5, j6, j7, j8, j9,

    j10,j11, j12,j13, j14, j15, j16, j17, j18, j19, j20, j21, j22;

    private JTextField  f1, f2, f3, f4, f5, f6, f7, f8, f9, f10,

    f11, f12,f13, f14, f15, f16, f17, f18, f19, f20, f21, f22;

    private JButton b, b1, b2, b3, b4, b5, b6, b7;


    ArrayList<Teacher> arrayy= new ArrayList<Teacher>();


    public void GUI()
```

```
{
    JFrame jf=new JFrame("23047392_NikitaGUI");//object is created
    jf.setLayout(null);//for positioning
    jf.setSize(850,775);//seting size breadth then height



    Color c= new Color(143, 172, 180);
    Color bc=new Color(245, 245, 245);
    Color no=new Color(248,124,124);

    JPanel jp1=new JPanel();
    jp1.setBounds(0,0,850,350);
    jp1.setBackground(c);
    jp1.setLayout(null);
    jf.add(jp1);

    JLabel Jl1=new JLabel("Lecturer");//label JL1 is made
    Jl1.setBounds(370,5,50,30);//right, down, length and height
    jp1.add(Jl1);// add in l1

    JLabel l=new JLabel("Teacher ID:");
    l.setBounds(10,50,100,25);
    jp1.add(l);
```

```
JTextField f=new JTextField();//textfield f is created

f.setBounds(160,50,150,25);//right, down, length and height

jp1.add(f);//added f in jf


JLabel l1=new JLabel("Teacher Name:");

l1.setBounds(10,80,100,25);

jp1.add(l1);


JTextField f1=new JTextField();

f1.setBounds(160,80,150,25);

jp1.add(f1);


JLabel l2=new JLabel("Address:");

l2.setBounds(10,110,100,25);

jp1.add(l2);


JTextField f2=new JTextField();

f2.setBounds(160,110,150,25);

jp1.add(f2);


JLabel l3=new JLabel("Working Type:");

l3.setBounds(10,140,100,25);
```

```
jp1.add(l3);


JTextField f3=new JTextField();

f3.setBounds(160,140,150,25);

jp1.add(f3);


JLabel l4=new JLabel("Working Hours");

l4.setBounds(500,140,150,25);

jp1.add(l4);


JTextField f4=new JTextField();

f4.setBounds(650,140,150,25);

jp1.add(f4);




JLabel l5=new JLabel("Employment Status:");

l5.setBounds(500,50,150,25);

jp1.add(l5);


JTextField f5=new JTextField();

f5.setBounds(650,50,150,25);

jp1.add(f5);
```

```
JLabel l6=new JLabel("Department:");
l6.setBounds(500,80,100,25);
jp1.add(l6);

JTextField f6=new JTextField();
f6.setBounds(650,80,150,25);
jp1.add(f6);

JLabel l7=new JLabel("Year Of Experience:");
l7.setBounds(500,110,150,25);
jp1.add(l7);

JTextField f7=new JTextField();
f7.setBounds(650,110,150,25);
jp1.add(f7);

JLabel Jl3=new JLabel("Grade Assignment");//label JL1 is made
Jl3.setBounds(350,200,150,30);//right, down, length and height
jp1.add(Jl3);// add in l1

JLabel l8=new JLabel("Graded Score:");
l8.setBounds(500,230,150,25);
jp1.add(l8);
```

```
JTextField f8=new JTextField();

f8.setBounds(650,230,150,25);

jp1.add(f8);


JLabel l9=new JLabel("New Teacher Id:");

l9.setBounds(10,230,150,25);

jp1.add(l9);


JTextField f9=new JTextField();

f9.setBounds(160,230,150,25);

jp1.add(f9);


JLabel l23=new JLabel("YearsOfExperience:");

l23.setBounds(500,270,150,25);

jp1.add(l23);


JTextField f23=new JTextField();

f23.setBounds(650,270,150,25);

jp1.add(f23);


JLabel l24=new JLabel("department:");

l24.setBounds(10,270,150,25);
```

```
jp1.add(l24);


JTextField f24=new JTextField();

f24.setBounds(160,270,150,25);

jp1.add(f24);


JButton b=new JButton("Add Lecturer");

b.setBounds(160,180,150,20);

b.setBackground(bc);

jp1.add(b);


JButton b2=new JButton("gradeAssignment");

b2.setBounds(160,320,150,20);

b2.setBackground(bc);

jp1.add(b2);


JButton b1=new JButton("Display");//Button b is created

b1.setBounds(500,180,100,20);//right, down, left, right

b1.setBackground(bc);//backgroud color is set

jp1.add(b1);


JButton b3=new JButton("Clear");

b3.setBounds(500,320,100,20);
```

```
        b3.setBackground(bc);

        jp1.add(b3);



        b.addActionListener(new ActionListener()

        {

            public void actionPerformed(ActionEvent ni)

            {

                try

                {

                    if(f.getText().isEmpty()  ||  f1.getText().isEmpty()  ||  f2.getText().isEmpty()  ||
f3.getText().isEmpty()

                    ||  f4.getText().isEmpty()||  f5.getText().isEmpty()  ||f6.getText().isEmpty()  ||
f7.getText().isEmpty())

                    {

                        b.setBackground(no);

                        JOptionPane.showMessageDialog(null, "Fill in the text field");

                    }

                    else  if  (!f1.getText().matches("[a-zA-Z  ]+")  ||!f2.getText().matches("[a-zA-Z
]+") ||!f3.getText().matches("[a-zA-Z ]+")

                    ||!f5.getText().matches("[a-zA-Z ]+") ||!f6.getText().matches("[a-zA-Z ]+"))

                    {

                        b.setBackground(no);
                        JOptionPane.showMessageDialog(null, "Invalid input. Only letters are allowed.");
                    }
                    else
                    {
                        //integer ma convert handinxw-step 1
                        b.setBackground(Color.PINK);
                        int teacherId=Integer.parseInt(f.getText());
                        String teacherName= f1.getText();
                        String address= f2.getText();
                        String workingType= f3.getText();
                        String employmentStatus= f5.getText();
                        String department= f6.getText();
                        int yearsOfExperience=Integer.parseInt(f7.getText());
                        int workingHours= Integer.parseInt(f4.getText());

                        Lecturer lobj = new Lecturer(teacherId,teacherName,address,workingType,
employmentStatus,department,yearsOfExperience,workingHours);

                        arrayy.add(lobj);

                        JOptionPane.showMessageDialog(null, "Data is added successfully");
                }
            }
            catch(NumberFormatException ne)
            {
                JOptionPane.showMessageDialog(null, "Number Format exception!Please
insert integer.");
            }
        }
    });



    //actionfor display
    b1.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent ni)
        {
            b1.setBackground(Color.PINK);
            for (Teacher  teacher: arrayy)
            {
                if (teacher instanceof Lecturer)
                {
                    Lecturer L1=(Lecturer) teacher;
                    L1.display();
                    JOptionPane.showMessageDialog(null, "Data is displayed");

                    System.out.println("\n");

                }

            }

        }

    });


    //GRADE Assignment

    b2.addActionListener(new ActionListener()

    {

        public void actionPerformed(ActionEvent ni)

        {

            try

            {

                if(f.getText().isEmpty()    ||    f9.getText().isEmpty()    ||f8.getText().isEmpty()
||f23.getText().isEmpty()

                || f24.getText().isEmpty())

                {

                    b.setBackground(no);

                    JOptionPane.showMessageDialog(null, "Fill in the text field");

                }

                else

                {
```

```
//Converts into integer

int newteacherId=Integer.parseInt(f9.getText());

int score = Integer.parseInt(f8.getText());

int YearsOfExperience = Integer.parseInt(f23.getText());

String Department=f24.getText();

for (Teacher teacher: arrayy)

{

   if(teacher instanceof Lecturer && teacher.getTeacherId()==newteacherId

   && ((Lecturer)teacher).getDepartment().equals(Department))

   {

      b2.setBackground(Color.PINK);

      //downcasting

      Lecturer L1=(Lecturer) teacher;

      L1.gradeAssignment(score,Department,YearsOfExperience);

      JOptionPane.showMessageDialog(null, "Your scores has been graded sucessfull");

   }

   else

   {

      JOptionPane.showMessageDialog(null, "The new input value must match from array list");

   }
```

```
                    }
                }
            }
            catch(Exception ne)
            {
                f24.setText("");
                f8.setText("");
                f23.setText("");
                JOptionPane.showMessageDialog(null,"Number Format exception!Please
insert integer.");
            }
        }
    });

    //actionfor clear button
    b3.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent ni)
        {
            b3.setBackground(Color.PINK);
            f.setText("");
            f1.setText("");
            f2.setText("");
```

```
          f3.setText("");
          f4.setText("");
          f5.setText("");
          f6.setText("");
          f7.setText("");
          f8.setText("");
          f9.setText("");
          f23.setText("");
          f24.setText("");

          JOptionPane.showMessageDialog(null, "All data are cleared");
     }
   });

   //JPanel for Tutor class is created

   JPanel jp=new JPanel();
   jp.setBounds(0,340,850,400);
   jp.setBackground(Color.LIGHT_GRAY);
   jp.setLayout(null);
   jf.add(jp);

   JLabel Jl2=new JLabel("Tutor");
```

```
Jl2.setBounds(360,5,100,30);

jp.add(Jl2);


JLabel l10=new JLabel("Teacher ID:");

l10.setBounds(30,70,100,25);

jp.add(l10);


JTextField f10=new JTextField();

f10.setBounds(180,70,150,25);

jp.add(f10);


JLabel l11=new JLabel("Teacher Name:");

l11.setBounds(30,100,100,25);

jp.add(l11);


JTextField f11=new JTextField();

f11.setBounds(180,100,150,25);

jp.add(f11);


JLabel l12=new JLabel("Address:");

l12.setBounds(30,130,100,25);

jp.add(l12);
```

```
JTextField f12=new JTextField();

f12.setBounds(180,130,150,25);

jp.add(f12);


JLabel l13=new JLabel("Working Type:");

l13.setBounds(30,160,100,25);

jp.add(l13);


JTextField f13=new JTextField();

f13.setBounds(180,160,150,25);

jp.add(f13);


JLabel l14=new JLabel("Employment Status :");

l14.setBounds(30,190,150,25);

jp.add(l14);


JTextField f14=new JTextField();

f14.setBounds(180,190,150,25);

jp.add(f14);



JLabel l15=new JLabel("Salary:");

l15.setBounds(460,70,100,25);
```

```
jp.add(l15);

JTextField f15=new JTextField();
f15.setBounds(625,70,150,25);
jp.add(f15);

JLabel l16=new JLabel("Specialization:");
l16.setBounds(460,100,100,25);
jp.add(l16);

JTextField f16=new JTextField();
f16.setBounds(625,100,150,25);
jp.add(f16);

JLabel l17=new JLabel("Academic Qualification:");
l17.setBounds(460,130,150,25);
jp.add(l17);

JTextField f17=new JTextField();
f17.setBounds(625,130,150,25);
jp.add(f17);

JLabel l18=new JLabel("Performance Index:");
```

```
l18.setBounds(460,160,150,25);

jp.add(l18);


JTextField f18=new JTextField();

f18.setBounds(625,160,150,25);

jp.add(f18);


JLabel l19=new JLabel("Working Hours:");

l19.setBounds(460,190,150,25);

jp.add(l19);


JTextField f19=new JTextField();

f19.setBounds(625,190,150,25);

jp.add(f19);


JLabel l20=new JLabel("New Salary:");

l20.setBounds(580,275,150,25);

jp.add(l20);


JTextField f20=new JTextField();

f20.setBounds(655,275,150,25);

jp.add(f20);
```

```
JLabel l21=new JLabel("New Performance Index:");
l21.setBounds(20,275,150,25);
jp.add(l21);

JTextField f21=new JTextField();
f21.setBounds(170,275,150,25);
jp.add(f21);

JLabel l22=new JLabel("New Teacher ID:");
l22.setBounds(300,310,150,25);
jp.add(l22);

JTextField f22=new JTextField();
f22.setBounds(420,310,150,25);
jp.add(f22);

JButton b4=new JButton("Add Tutor");
b4.setBounds(140,230,120,20);
b4.setBackground(bc);
jp.add(b4);

JButton b5=new JButton("Display");
b5.setBounds(560,230,120,20);
```

```
b5.setBackground(bc);
jp.add(b5);

JButton b6=new JButton("Remove Tutor");
b6.setBounds(140,350,120,20);
b6.setBackground(bc);
jp.add(b6);

JButton b7=new JButton("Clear");
b7.setBounds(370,350,100,20);
b7.setBackground(bc);
jp.add(b7);

JButton b8=new JButton("Set Slary");
b8.setBounds(560,350,120,20);
b8.setBackground(bc);
jp.add(b8);

b4.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ni)
    {
        try
        {
            if(f10.getText().isEmpty() || f11.getText().isEmpty() || f12.getText().isEmpty()
|| f13.getText().isEmpty() || f14.getText().isEmpty()
            || f15.getText().isEmpty() ||f16.getText().isEmpty() || f17.getText().isEmpty()
||f18.getText().isEmpty() ||f19.getText().isEmpty() )
            {
                b4.setBackground(no);
                JOptionPane.showMessageDialog(null, "Please fill in the text field");
            }
            if (!f11.getText().matches("[a-zA-Z ]+") || !f12.getText().matches("[a-zA-Z ]+")
||!f13.getText().matches("[a-zA-Z ]+")
            || !f14.getText().matches("[a-zA-Z ]+") || !f16.getText().matches("[a-zA-Z ]+")
||!f17.getText().matches("[a-zA-Z ]+")  )
            {
                b4.setBackground(no);
                JOptionPane.showMessageDialog(null, "Invalid input. Only letters are
allowed.");
            }
            else
            {
                //integer ma convert handinxw-step 1
                b4.setBackground(Color.PINK);
                int teacherId=Integer.parseInt(f10.getText());
                String teacherName= f11.getText();
```

```
                    String address= f12.getText();
                    String workingType= f13.getText();
                    String employmentStatus= f14.getText();
                    double salary=Integer.parseInt(f15.getText());
                    String specialization= f16.getText();
                    String academicQualification= f17.getText();
                    int performanceIndex= Integer.parseInt(f18.getText());
                    int workingHours= Integer.parseInt(f19.getText());

                    Tutor tobj = new Tutor(teacherId,teacherName,address,

workingType,employmentStatus,workingHours,salary,specialization,academicQualificati
on,performanceIndex);

                    arrayy.add(tobj);

                    JOptionPane.showMessageDialog(null, "Data is added successfully.");
                  }
                }
                catch(NumberFormatException e)
                {
                  JOptionPane.showMessageDialog(null, "Number Format exception!Please
insert integer.");
```

```
        }
    }
});




//for display button
b5.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ni)
    {
        for (Teacher  t: arrayy)
        {
            b5.setBackground(Color.PINK);
            if(t instanceof Tutor)
            {
                Tutor t1=(Tutor)t;
                t1.display();
                JOptionPane.showMessageDialog(null, "Data is displayed");
                System.out.print("\n");
            }
        }
    }
```

```
        });
        //button to remove tutor
        b6.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent ni)
            {
                try
                {
                    if(f10.getText().isEmpty() || f15.getText().isEmpty() ||f16.getText().isEmpty()
|| f17.getText().isEmpty()
                    ||f18.getText().isEmpty() ||f22.getText().isEmpty() )
                    {
                        b6.setBackground(no);
                        JOptionPane.showMessageDialog(null, "Please fill in the text field");
                    }
                    if ( !f16.getText().matches("[a-zA-Z ]+") ||!f17.getText().matches("[a-zA-Z ]+")
)
                    {
                        b6.setBackground(no);
                        JOptionPane.showMessageDialog(null, "Invalid input. Only letters are
allowed.");
                    }
                    else
```

```
{
    //conveert into integer
    b6.setBackground(Color.PINK);
    int teacherId=Integer.parseInt(f10.getText());
    double salary=Integer.parseInt(f15.getText());
    String specialization= f16.getText();
    String academicQualification= f17.getText();
    int performanceIndex= Integer.parseInt(f18.getText());
    int newteacherId=Integer.parseInt(f22.getText());

    for (Teacher  t: arrayy )
    {
        b6.setBackground(Color.PINK);
        if(t instanceof Tutor && teacherId==newteacherId)
        {
            Tutor t1=(Tutor)t;
            t1.removeTutor();
            JOptionPane.showMessageDialog(null, "Removal is done");
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Teacher id must be same");
        }
```

```
                    }
                }
            }
            catch(NumberFormatException e)
            {
                JOptionPane.showMessageDialog(null, "Number Format exception!Please
insert integer.");
            }
        }
    });



    //kjggdfggfiu
    b8.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent ni)
        {
            try
             {
                if(f10.getText().isEmpty() || f20.getText().isEmpty()
                ||f21.getText().isEmpty() ||f22.getText().isEmpty() )
                {
                    b8.setBackground(no);
                    JOptionPane.showMessageDialog(null, "Please fill in the text field");
                }
                if ( !f16.getText().matches("[a-zA-Z ]+") ||!f17.getText().matches("[a-zA-Z ]+")
)
                {
                    b8.setBackground(no);
                    JOptionPane.showMessageDialog(null, "Invalid input. Only letters are
allowed.");
                }
                else
                {
                    //conveert into integer
                    b8.setBackground(Color.PINK);
                    double newsalary=Integer.parseInt(f20.getText());
                    int newPerformanceIndex= Integer.parseInt(f21.getText());
                    int newteacherId=Integer.parseInt(f22.getText());
                    for (Teacher  t: arrayy )
                    {
                        if(t instanceof Tutor && t.getTeacherId()==newteacherId)
                        {
                            Tutor t1=(Tutor)t;
                            t1.setSalaryAndCertification(newsalary,newPerformanceIndex);
                            JOptionPane.showMessageDialog(null, "Salary is set");
                        }
                        else
                        {
                            JOptionPane.showMessageDialog(null, "Teacher id must be same");
                        }
                    }
                }
            }
            catch(NumberFormatException e)
            {
                JOptionPane.showMessageDialog(null, "Number Format exception!Please
insert integer.");
            }
        }
    });

    //actionfor clear button
    b7.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent ni)
        {
            b7.setBackground(Color.PINK);
            f10.setText("");
    }

            f11.setText("");

            f12.setText("");

            f13.setText("");

            f14.setText("");

            f15.setText("");

            f16.setText("");

            f17.setText("");

            f18.setText("");

            f19.setText("");

            f20.setText("");

            f21.setText("");

            f22.setText("");

            JOptionPane.showMessageDialog(null, "All datas are cleared");

          }

        });

        jf.setVisible(true);//invisible is made visible

    }

    public static void main (String [] args)

    {


        TeacherGUI obj = new TeacherGUI();

        obj.GUI();

    }
```

```
}
```