LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS4001NI Programming

COURSEWORK-2

Assessment Weightage & Type

30% Individual Coursework

Semester and Year

Spring 2021

Student Name: Achyut Adhikari

Group: C2

London Met ID: 20048811

College ID: NP01CP4S210055

Assignment Due Date: 20th August 2021

Assignment Submission Date: 20th August 2021

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

Contents

List of figures

List of Tables

1.Introduction

Java is object-oriented, multi-propose and platform independent programming language made by Oracle Corporation. Java helps programmer to create programs from foundation level to high level which leads develops to use java in various sector. Programming language like java has its own syntax which was influence form the syntax of ‘C’ and ‘C++’. I learned about GUI frameworks and how they are made using the Java language. Java Swing is a lightweight Graphical User Interface (GUI) toolbox that integrates rich gadget settings. It incorporates the opportunity to create GUI segments with your Java applications and is self-contained stage.

Figure 1 Java logo

The main purpose of these courses is to design a curriculum management system for an educational institution. The program captures course, academic and non-academic information, such as course name, course leader, credit hour, teacher's name, teacher's name to keep academic record.

2. Tools Used

2.1 Blue J

BlueJ is a development environment that allows you to develop Java programs quickly and easily. (Oracle, 1999) BlueJ was created to help the learning and instructing of item situated programming, and its plan varies from other advancement conditions as a result. Its main features are:

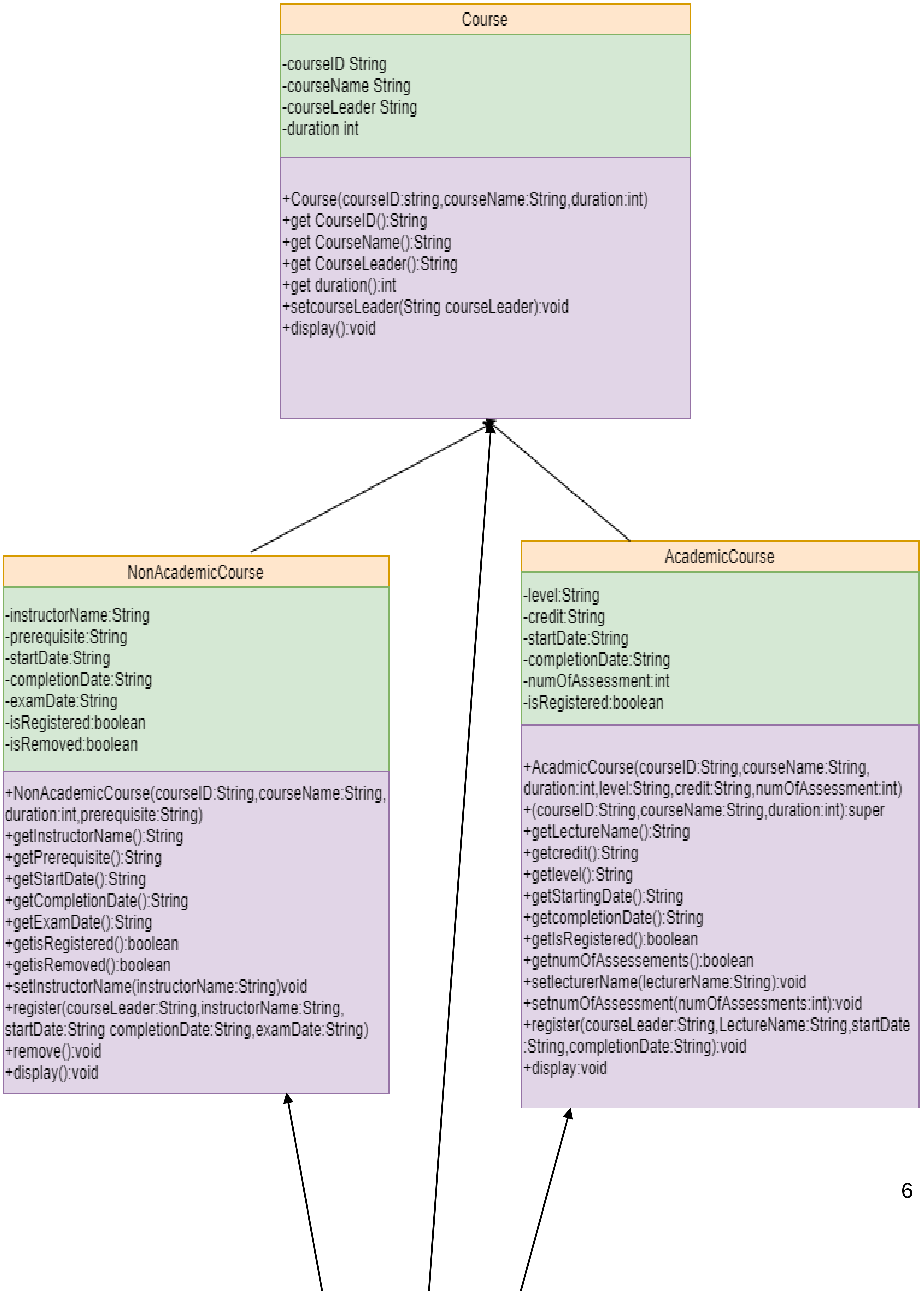
- Simple and allows beginners to get started more quickly and without being overwhelmed.
- BlueJ is a site full of teaching resources.
- BlueJ allows you to interact with objects and directly invoke Java expressions.
- BlueJ runs on windows, Mac, Linux, and other platforms which run java.

2.2 Draw.io

Draw.io is an open-source site for building applications, and the world's most generally utilized program-based end-client outlining applications. (Anon, 2005) It is free online outline making programming for flowcharts, process graphs, organization graphs, and UML, ER, and system charts. Draw.io is an open-source technology stack for building diagramming applications, and the world’s most widely used browser-based end-user diagramming software. Its main features are:

- It’s based on the viral effect of a free, open-source application.
- You can take the existing draw.io base and build for specific vertical markets, add, and improve on features.

3. Class Diagram


Course
-courseID String
-courseName String
-courseLeader String
-duration int
+Course(courseID:string,courseName:String,duration:int)
+get CourseID():String
+get CourseName():String
+get CourseLeader():String
+get duration():int
+setcourseLeader(String courseLeader):void
+display():void
NonAcademicCourse
-instructorName:String
-prerequisite:String
-startDate:String
-completionDate:String
-examDate:String
-isRegistered:boolean
-isRemoved:boolean
+NonAcademicCourse(courseID:String,courseName:String,
duration:int,prerequisite:String)
+getInstructorName():String
+getPrerequisite():String
+getStartDate():String
+getCompletionDate():String
+getExamDate():String
+getisRegistered():boolean
+getisRemoved():boolean
+setInstructorName(instructorName:String)void
+register(courseLeader:String,instructorName:String,
startDate:String completionDate:String,examDate:String)
+remove():void
+display():void
AcademicCourse
-level:String
-credit:String
-startDate:String
-completionDate:String
-numOfAssessment:int
-isRegistered:boolean
+AcadmicCourse(courseID:String,courseName:String,
duration:int,level:String,credit:String,numOfAssessment:int)
+(courseID:String,courseName:String,duration:int):super
+getLectureName():String
+getcredit():String
+getlevel():String
+getStartingDate():String
+getcompletionDate():String
+getIsRegistered():boolean
+getnumOfAssessements():boolean
+setlecturerName(lecturerName:String):void
+setnumOfAssessment(numOfAssessments:int):void
+register(courseLeader:String,LectureName:String,startDate
:String,completionDate:String):void
+display:void

INGCollage
+mainframe: JFrame +Buttonaca: JButton +Buttonnon: JButton +buttom1: JButton + button 2: JButton + button 4: JButton + button 5: JButton + button 6: JButton + button 7: JButton + button 8: JButton + button 9: JButton + button 10: JButton +nonacademic: JPanel +academic: JPanel + JTextField 1: JTextField + JTextField 2: JTextField + JTextField 3: JTextField + JTextField 4: JTextField + JTextField 5: JTextField + JTextField 6: JTextField + JTextField 7: JTextField + JTextField 8: JTextField + JTextField 9: JTextField + JTextField 10: JTextField + JTextField 11: JTextField + JTextField 12: JTextField + JTextField:13 JTextField + JTextField:14 JTextField + JTextField:15 JTextField + JTextField:16 JTextField + JTextField:17 JTextField

+ JTextField:18 JTextField

+ JTextField:19 JTextField

+ JTextField:20 JTextField

+ JTextField:21 JTextField

+filemenu: JMenu

+helpmenu: JMenu

+fileItem1: JMenuItem

+fileItem2: JMenuItem

~ CourseArraylist: ArrayList <course>

+actionPerformed(ActionEvent e): void

+main(String[] args) : void

4. Pseudo Code

ING Collage:

```
CREATE class INGCollage
    DECLARE JFrame mainframe
    DECLARE Jpanel acapanel1 nonpanel2
    DECLARE JLabel label0, label1, label2, label3, label4, label5, label6, label7, label8, label9, label10,
    label11, label12, label13, label23, label14, label15, label16, label17, label18, label19, label20, label21, label22.
    DECLARE JTextField textField1, textField2, textField3, textField4, textField5, textField6, textField7, textField8, textField9, textField10, textField11, textField12, textField13, textField14, textField15, textField16, textField17, textField18, textField19, textField20, textField21.
    DECLARE JButton button1, button2, button3, button4, button5, button6, button7, button8, button9.
    DECLARE JMenu filemenu, editmenu, helpmenu
    DECLARE JMenuItem helpItem1, fileItem2, fileItem3, fileItem4, fileItem5.
    CREATE Arraylist listjava of type course
    CREATE constructor INGCollage ()
DO
    CREATE JFrame with name mainframe
    CREATE JPanel with name Focus
    CREATE JPanel with name Academic
    CREATE JPanel with name Nonacademic
    CREATE JButton with name Buttonaca
    CREATE JButton with name Buttonnon

    ADD ActionListener to Buttonaca
    ADD ActionListener to Buttonnon
    SET focusable to Button1 to false
    SET focusable to Button2 to false
    SET Font emphasis and size to Buttonaca
    SET Font emphasis and size to Buttonnon
    SET Enabled to buttonaca to false

    CREATE JMenubar of name M1
    CREATE JMenu of name filemenu
    CREATE JMenu of name helpmenu
    CREATE JMenuItem of name FileItem1
    CREATE JMenuItem of name FileItem2
    ADD FileItem1 to filemenu
    ADD FileItem2 to filemenu
    ADD filemenu to M1
    ADD helpmenu to M1
    SET JMenuBar M1 to mainframe
    ADD ActionListener to FileItem1
    ADD ActionListener to helpmenu
    CREATE JLabel of name Label1
    SET Horizontal Alignment of Label1 to Center
    SET Vertical Alignment of Label 1 to Center

    CREATE JLabel of name Label2
    CREATE JTextField of name textField1
    CREATE JLabel of name Label3
    CREATE JTextField of name textField2
    CREATE JLabel of name Label4
    CREATE JTextField of name textField3
    CREATE JLabel of name Label5
    CREATE JTextField of name textField4
    CREATE JLabel of name Label6
    CREATE JTextField of name textField5
    CREATE JLabel of name Label7
    CREATE JTextField of name textField6
    CREATE JButton of name button1
    CREATE JLabel of name Label8
    CREATE JTextField of name textField7
    CREATE JLabel of name Label9
    CREATE JTextField of name textField8
    CREATE JLabel of name Label10
    CREATE JTextField of name textField9
    CREATE JLabel of name Label11
    CREATE JTextField of name textField10
    CREATE JLabel of name Label12
    CREATE JTextField of name textField11
    CREATE JButton of name button2

    CREATE JButton of name button3
    CREATE JButton of name button4
    CREATE JButton of name button5

    SET font, emphasis, and size of button1
    SET font, emphasis, and size of button2
    SET font, emphasis, and size of button3
    SET font, emphasis, and size of button4
    SET font, emphasis, and size of button5

    SET Background of button1
    SET Background of button2
    SET Background of button3
    SET Background of button4
    SET Background of button5

    SET Background of Buttonaca
    SET Background of Buttonnon

    ADD ActionListener to button1
    ADD ActionListener to button2
    ADD ActionListener to button3
```

```
    ADD ActionListener to button4
    ADD ActionListener to button5

    SET Bounds of Label1
    SET Bounds of Label2
    SET Bounds of Label3
    SET Bounds of Label4
    SET Bounds of Label5
    SET Bounds of Label6
    SET Bounds of Label7
    SET Bounds of Label8
    SET Bounds of Label9
    SET Bounds of Label10
    SET Bounds of Label11
    SET Bounds of Label12

    SET Bounds of textField1
    SET Bounds of textField2
    SET Bounds of textField3
    SET Bounds of textField4
    SET Bounds of textField5
    SET Bounds of textField6
    SET Bounds of textField7
    SET Bounds of textField8
    SET Bounds of textField9
    SET Bounds of textField10
    SET Bounds of textField11

    SET Bounds of button1
    SET Bounds of button2
    SET Bounds of button3
    SET Bounds of button4
    SET Bounds of button5
    ADD Label1 to academic
    ADD Label2 to academic
    ADD Label3 to academic
    ADD Label4 to academic
    ADD Label5 to academic
    ADD Label6 to academic
    ADD Label7 to academic
    ADD Label8 to academic
    ADD Label9 to academic
    ADD Label10 to academic
    ADD Label11 to academic
    ADD Label12 to academic

    ADD textField1 to academic
    ADD textField2 to academic
    ADD textField3 to academic
    ADD textField4 to academic
    ADD textField5 to academic
    ADD textField6 to academic
    ADD textField7 to academic
    ADD textField8 to academic
    ADD textField9 to academic
    ADD textField10 to academic
    ADD textField11 to academic

    ADD button1 to academic
    ADD button2 to academic
    ADD button3 to academic
    ADD button4 to academic
    ADD button5 to academic
    ADD Label13 to nonacademic
    ADD Label14 to nonacademic
    ADD Label15 to nonacademic
    ADD Label16 to nonacademic
    ADD Label17 to nonacademic
    ADD Label18 to nonacademic
    ADD Label19 to nonacademic
    ADD Label20 to nonacademic
    ADD Label21 to nonacademic
    ADD Label22 to nonacademic

    ADD textField12 to nonacademic
    ADD textField13 to nonacademic
    ADD textField14 to nonacademic
    ADD textField15 to nonacademic
    ADD textField16 to nonacademic
    ADD textField17 to nonacademic
    ADD textField18 to nonacademic
    ADD textField19 to nonacademic
    ADD textField20 to nonacademic
    ADD textField21 to nonacademic


    ADD ActionListener to button1
    ADD ActionListener to button2
    ADD ActionListener to button3
    ADD ActionListener to button4
    ADD ActionListener to button5

    SET Background of academic
    SET Background of nonacademic
    SET Background of Focus
```

```
SET Layout of academic to null
ADD academic to Mainframe
SET Bounds of academic

SET Layout of nonacademic to null
ADD nonacademic to mainframe
SET Bounds of nonacademic

SET Visible of academic to true
SET Visible of nonacademic to false

SET Font, emphasis, and size of Label1
SET Horizontal Alignment of Label1 to Center
ADD Buttonaca to Focus
ADD Buttonnon to Focus
SET Layout of Focus to null
ADD window to mainframe
SET Bounds of Focus
SET Size of mainframe
SET Layout of mainframe to null
SET Visible of mainframe to true
SET DefaultCloseOperation of frame to Exit on Close
CALL void actionPerformed (ActionEvent e)

IF (e. getSource () ==Buttonnon)
 DO
SET Visible of academic to false
SET Visible of nonacademic to true
SET Enabled of Buttonaca to true
SET Enabled of Buttonnon to false
END DO
ELSE IF (e. getSource () ==Buttonnon
DO
```

```
SET Visible of nonacademic to false
SET Visible of academic to true
SET Enabled of Buttonnon to true
SET Enabled of Buttonaca to false
END DO
CALL void actionPerformed (ActionEvent e)
    DO
    IF (e. getSource () ==Buttonnon) DO
    SET Visible of academic to false
    SET Visible of nonacademic to true
    SET Enabled of Buttonaca to true SET Enabled of Buttonnon to false

ELSE IF (e. getSource () ==Buttonnon)
    DO
    SET Visible of nonacademic to false
    SET Visible of academic to true
    SET Enabled of Buttonnon
    to true
    SET Enabled of Buttonaca to false
 ELSE IF (e. getSource () == button5 || e. getSource () == button5)
    DO
    SET textField1 to empty
    SET textField2 to empty
    SET textField3 to empty
    SET textField4 to empty
    SET textField5 to empty
    SET textField6 to empty
    SET textField7 to empty
    SET textField8 to empty
    SET textField9 to empty
    SET textField10 to empty
    SET textField11 to empty
    SET textField12 to empty
```

```
                SET textField13 to empty
                SET textField14 to empty
                SET textField15 to empty
                SET textField16 to empty
                SET textField17 to empty
                SET textField18 to empty
                SET textField19 to empty
                SET textField20 to empty
                SET textField21 to empty

        ELSE IF (e. getSource () == btn1)
                DO TRY
                GET text from textField1 and ASSIGN it to courseid
                GET text from textField2 and ASSIGN it to courseName
                GET text from textField3and ASSIGN it to courseduration
                GET text from textField4 and ASSIGN it to level
                GET text from textField5 and ASSIGN it to credit
                GET text from textField6 and ASSIGN it to numOfAssesment

        IF (courseiD.isEmpty() || courseName.isEmpty() || courseduration.isEmpty() ||
level.isEmpty() || credit. isEmpty() || numOfAssesment.isEmpty())
        DO
                THROW new exception
         ELSE
                DO
        IF (listJava.isEmpty())
                DO
                CONVERT credit to Integer and ASSIGN it to Credit
                CONVERT courseduration to Integer and ASSIGN it to duration
                CONVERT numOfAssessment to Integer and ASSIGN it to numOfAssessment
        CREATE an object of AcademicCourse(PASS parameter courseid, courseName,
duration, level, Credit, numOfAssessment) of name academicA
        ADD academicA to listJava
```

```
SHOW Message Dialouge with message  “The course is added”

SET textField1 to empty
SET textField2 to empty
SET textField3 to empty
SET textField4 to empty
SET textField5 to empty
SET textField6 to empty

ELSE
        DO
                DECLARE variable valid with Boolean data type and assign it to false
FOR(Course co : listJava)
        DO
                IF(co instanceof AcademicCourse && courseID.equals(co.getcourseID()))
        DO
                SHOW Message Dialogue with message “Course has already been
        registered.”
                ASSIGN true to variable valid
                END IF

        END DO

IF (!valid)
        DO
        DECLARE a variable Credit with data type int, change credit into Integer and
assign it to Credit
        DECLARE a variable duration with data type int, change duration into Integer and
assign it to duration
        DECLARE a variable numOfAssessment with data type int, change
numOfAssesment into Integer and assign it to numOfAssessment
        SHOW Message Dialouge with message “Thank you! The course is added."
        CREATE an object of AcademicCourse(PASS parameter courseid, courseName,
    duration, level, Credit, numOfAssessment) of name academicA
        ADD academicA to listJava

        SET textField1to empty
        SET textField2 to empty
        SET textField3 to empty
        SET textField4 to empty
        SET textField5 to empty
        SET textField6 to empty

            END IF
        END DO
    END DO
    END DO

        CATCH (NumberFormatException es)
        DO
            SHOW Message Dialogue with message "Credit, Duration and Number of
        Assessment can be whole number only!!"
        END DO
    CATCH (Exception es)
    DO
    SHOW Message Dialogue with message "Fields cannot be empty!”
    END DO

     ELSE IF(e.getSource == button2)
        DO TRY
        GET text from textField6 and ASSIGN it to courseid
        GET text from textField7 and ASSIGN it to courseleader
        GET text from textField8 and ASSIGN it to lecturer
        GET text from textField9 and ASSIGN it to startdate
        GET text from textField10 and ASSIGN it to competiondate
        IF courseid. IsEmpty() || courseleader. IsEmpty() || lecturername.isEmpty() ||
    startdate.isEmpty() || completiondate.isEmpty()(
    DO
        THROW new Exception
    ELSE
            DO
            IF (listJava.isEmpty())
        DO
        SHOW Message Dialogue with message “Add a course first.”
    ELSE
        DO
            DECLARE variable Valid Check with Boolean data type and assign it to
        false
        FOR(Course co : listJava)
           DO
            IF(co instanceof AcademicCourse && courseID.equals(co.getcourseID()))
            DO
                DOWNCAST Course to AcademicCourse with name ac
        IF (ac.getisRegistered()==false)
            DO
                CALL register(PASS parameters courseleader, lecturer, startdate,
            completiondate, examdate) method of AcademicCourse Class
                SHOW Message Dialogue with message “ The course has been
            registered.”
            ASSIGN true to variable Valid Check

            SET textField7to empty
            SET textField8 to empty
            SET textField9 to empty
            SET textField10 to empty

        ELSE
```

```
          DO
                    SHOW Message Dialogue with message “course is already registered.”
                    ASSIGN true to Valid Check
                              END IF
                    END IF
          END DO
         IF (!ValidCheck)
         DO
         SHOW Message Dialogue with message “No course with such id is added
                              END IF
                    END IF
         END IF
END TRY

CATCH (Exception es)
         DO
         SHOW Message Dialogue with message “Fields cannot be empty.”
         END DO
ELSE IF(e.getSource() == button4 )
         IF (listJava. Size()<=0)
         SHOW Message Dialogue with message "Course is not yet added!”
ELSE
         FOR (Course co: listJava)
         DO
                    IF (co instanceof AcademicCourse)
                    DOWNCAST Course to AcademicCourse with name ac
                    CALL display() method
                    END IF
         END DO
END IF
 ELSE IF (e.getSource() == button1)
         TRY
         GET text from textField1 and ASSIGN it to courseid
```

```
    GET text from textField2 and ASSIGN it to courseName
    GET text from textField3and ASSIGN it to courseduration
    GET text from textField4 and ASSIGN it to prerequisite
    IF(courseID.isEmpty() || courseName.isEmpty() || courseduration.isEmpty() || prerequisite.isEmpty())
    THROW new exception
    ELSE
    IF (listJava.isEmpty())
    CONVERT courseduration to Integer and ASSIGN it to duration
    CREATE an object of NonAcademicCourse(PASS parameter courseid, courseName, duration, prerequisite) of name academicA
    ADD academicA to listJava
     SHOW Message Dialouge with message " The course is added."

    SET textField12to empty
    SET textField13 to empty
    SET textField14 to empty
    SET textField15 to empty

ELSE
    DECLARE variable valid with Boolean data type and assign it too false
     FOR(Course co : listJava)
    DO
    IF(co instanceof NonAcademicCourse && courseID.equals(co.getcourseID()))
    SHOW Message Dialogue with message "Course with the give Course ID is already added."
    ASSIGN true to variable valid
    END IF
END DO
     IF (!valid)
    DECLARE a variable duration with data type int, change courseduration into Integer and assign it to duration
    SHOW Message Dialouge with message "course is added."
```

```
CREATE an object of NonAcademicCourse(PASS parameter courseid, courseName, duration, prerequisite) of name nonacademic
    ADD nonacademicA to listJava

    SET textField12to empty
    SET textField13 to empty
    SET textField14 to empty
    SET textField15 to empty
        END IF
    END IF
END DO
 END TRY

CATCH (NumberFormatException es)
    DO
    SHOW Message Dialogue with message "Credit, Duration and Number of Assessment can be whole number only!!"
    END DO
        CATCH (Exception es)
    DO
    SHOW Message Dialogue with message "Fields cannot be empty!”
END DO

ELSE IF(e.getSource == button2)
TRY

    GET text from textField16 and ASSIGN it to courseleader
    GET text from textField17 and ASSIGN it to lecturer
    GET text from textField18 and ASSIGN it to startdate
    GET text from textField19 and ASSIGN it to competiondate
    GET text from textField20 and ASSIGN it to examdate
    IF (CourseID.isEmpty() || courseleader. IsEmpty() || instructor.isEmpty() || startdate.isEmpty() || completiondate.isEmpty() || examdate.isEmpty())
    THROW new Exception
ELSE
    IF (listJava.isEmpty())
    SHOW Message Dialogue with message “Add a course first to register.”
ELSE
    DECLARE variable ValidCheck with Boolean data type and assign it to false
    FOR(Course co : listJava)
    DO
        IF(co instanceof NonAcademicCourse && courseID.equals(co.getcourseID()))
        DOWNCAST Course to NonAcademicCourse with name nac
        IF (nac.getisRegistered()==false)
        CALL register(PASS parameters courseleader, lecturer, startdate, completiondate) method of NonAcademicCourse Class
        SHOW Message Dialogue with message “ The course has been registered.”

        ASSIGN true to variable ValidCheck
        SET textField16to empty
        SET textField17 to empty
        SET textField18 to empty
        SET textField19 to empty
        SET textField20 to empty
        SET textField21 to empty
ELSE
    SHOW Message Dialogue with message “The course with given ID is already registered.”
        ASSIGN true to variable ValidCheck
        SET textField16to empty
        SET textField17 to empty
        SET textField18 to empty
```

```
                SET textField19 to empty
                SET textField20 to empty
                SET textField21 to empty
                END IF
            END IF

        END DO
            IF (!ValidCheck)
                SHOW Message Dialogue with message “No course with such id is added.”
                END IF
            END IF
        END IF
        END TRY

        CATCH (Exception es)
            DO
                SHOW Message Dialogue with message “Fields cannot be empty.
        END DO
                ELSE IF(e.getSource() == button4 )
            IF (listJava.size()<=0)
        SHOW Message Dialogue with message "Course is not yet added!”
        ELSE
            FOR (Course co: listJava)
            DO
                IF (co instanceof AcademicCourse)
                    DOWNCAST Course to AcademicCourse with name ac
                        CALL display() method
                    END IF
                END DO
            END IF
        ELSE IF (e.getSource()==button3)
            GET text from textField16 and ASSIGN it to courseid
            IF (CourseID.isEmpty())
            SHOW Message Dialog with message “Enter the Courseid to remove a course.”
        ELSE
            IF (listJava.isEmpty)
            SHOW Message Dialog with message "Register a course before removing."
         ELSE
            DECLARE variable ValidCheck with Boolean data type and assign it to false
            FOR (Course co : listJava)
            DO
                IF (co instanceof NonAcademicCourse &&
            courseID.equals(co.getcourseID()))
                DOWNCAST Course to NonAcademicCourse with name nac
        IF (nac.getisRemoved()==false)
            CALL remove(PASS parameters courseid) method of NonAcademicCourse Class
            SHOW Message Dialog with message “The course has been removed.”
                    ASSIGN true to the variable ValidCheck
                SET textField16to empty
                END IF
            END IF
        END DO

        IF (!ValidCheck)
            SHOW Message Dialog with message “No course with such ID has been added.”
                END IF
            END IF
        END IF
        ELSE IF(e.getSourcs()==fileItem1

        SET textField1to empty
        SET textField2 to empty
        SET textField3 to empty
        SET textField4 to empty
        SET textField5 to empty
        SET textField6 to empty
```

```
SET textField7to empty
SET textField8 to empty
SET textField9 to empty
SET textField10 to empty
SET textField11to empty
SET textField12 to empty
SET textField13 to empty
SET textField14 to empty
SET textField15to empty
SET textField16 to empty
SET textField17 to empty
SET textField18 to empty
SET textField19 to empty
SET textField20  to empty
SET textField21 to empty

ELSE IF (e.getSource() == fileItem2)
    System.exit(0)
ELSE IF(e.getSource() == help)
    SHOW Message Dialog with message "This form is to register the academic and nonacademic courses."
    END IF
END DO
CREATE main method
    DO
        CREATE object INGCollege()
END DO
```

5. Method Description

5.1 INGCollage ()

This method INGCollage is a constructor where the user interface has been built.

5.2 showMessageDialog ()

This method creates a message dialog along with appropriate message type.

5.3 equals ()

This method compares the given strings depending on content of string. Returns true if the contents of strings are identical and returns false if none of the characters are identical

5.4 actionPerformed()

This method is used to call just after the user performs an action.

5.5 getText()

This method returns or receives texts of a component.

5.6 setText()

This method sets the current text that this StringCharacterIterator will read.

5.7 getSource()

This method receives or figures out which button is active as pressed by the user

5.8 isEmpty()

This method determines whether a string is empty.

5.9 display()

This method displayed information entered the GUI in the textfields .

5.10 remove()

This method sets isRemoved to true and removes nonacademic course from the array Coursearraylist.

5.11 add()

This method adds data to a specified list.

5.12 parseInt()

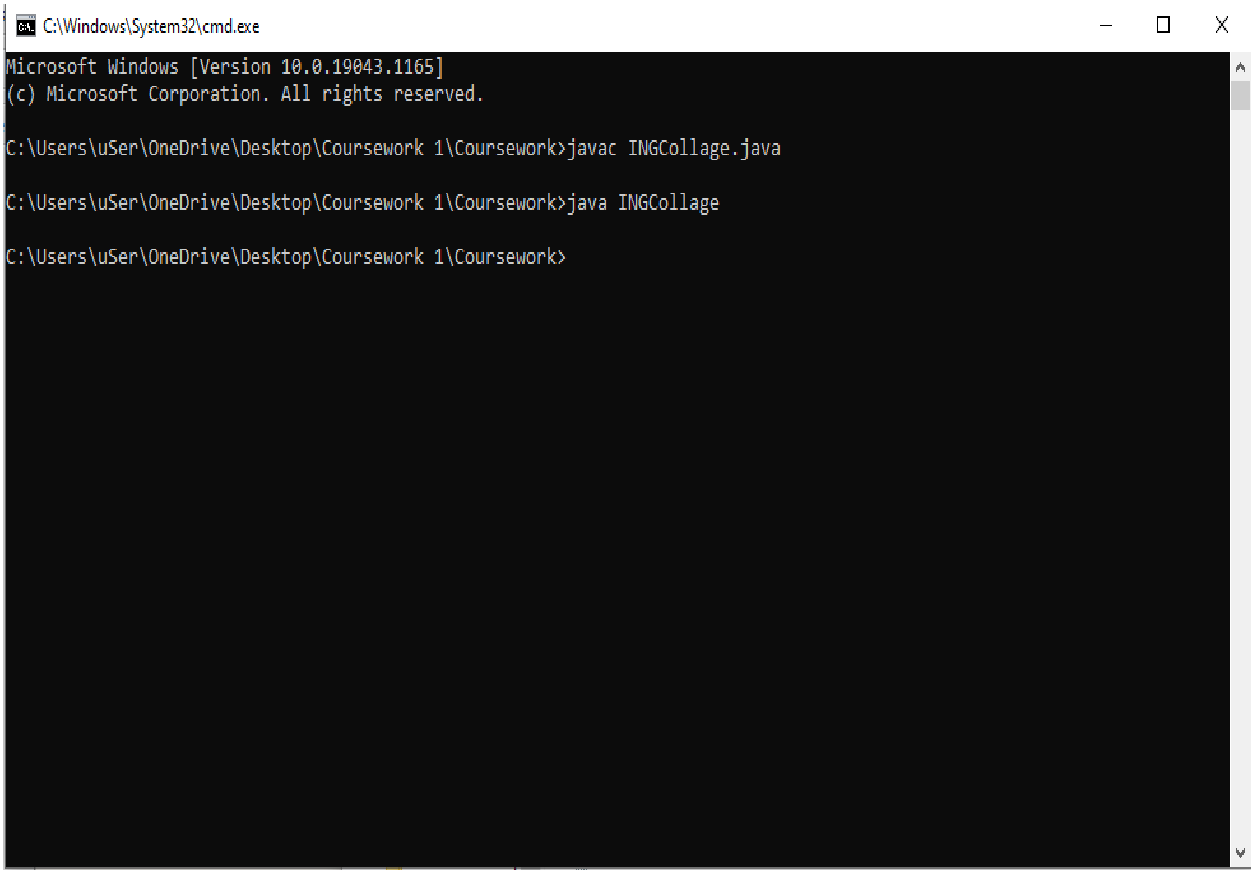
This method takes primitive data type String and coverts to integer passing parameter

6.Testing

6.1 Test1

Objective	This test to check if the program can be compiled and run using the command prompt
Action	Compile and run INGCollege.java file using command prompt
Expected Result	The file would run, and a GUI frame would start
Actual Result	The file was run, and a GUI frame started
Conclusion	The test was successful

Table 1 Test 1



```
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\uSer\OneDrive\Desktop\Coursework 1\Coursework>javac INGCollage.java

C:\Users\uSer\OneDrive\Desktop\Coursework 1\Coursework>java INGCollage

C:\Users\uSer\OneDrive\Desktop\Coursework 1\Coursework>
```

Figure 2 test 1(i)

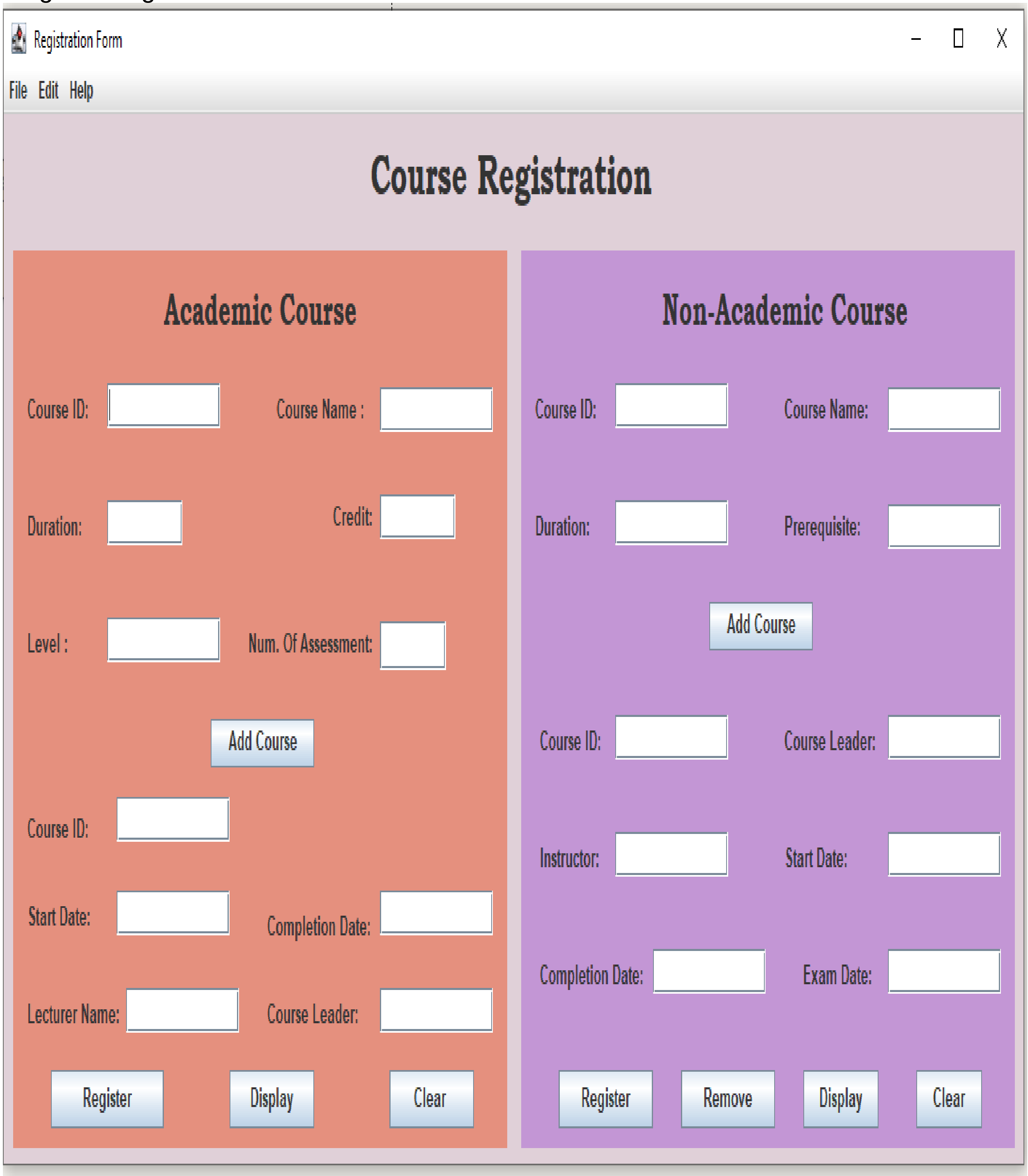


Figure 3

Test (ii)

6.2 Test 2

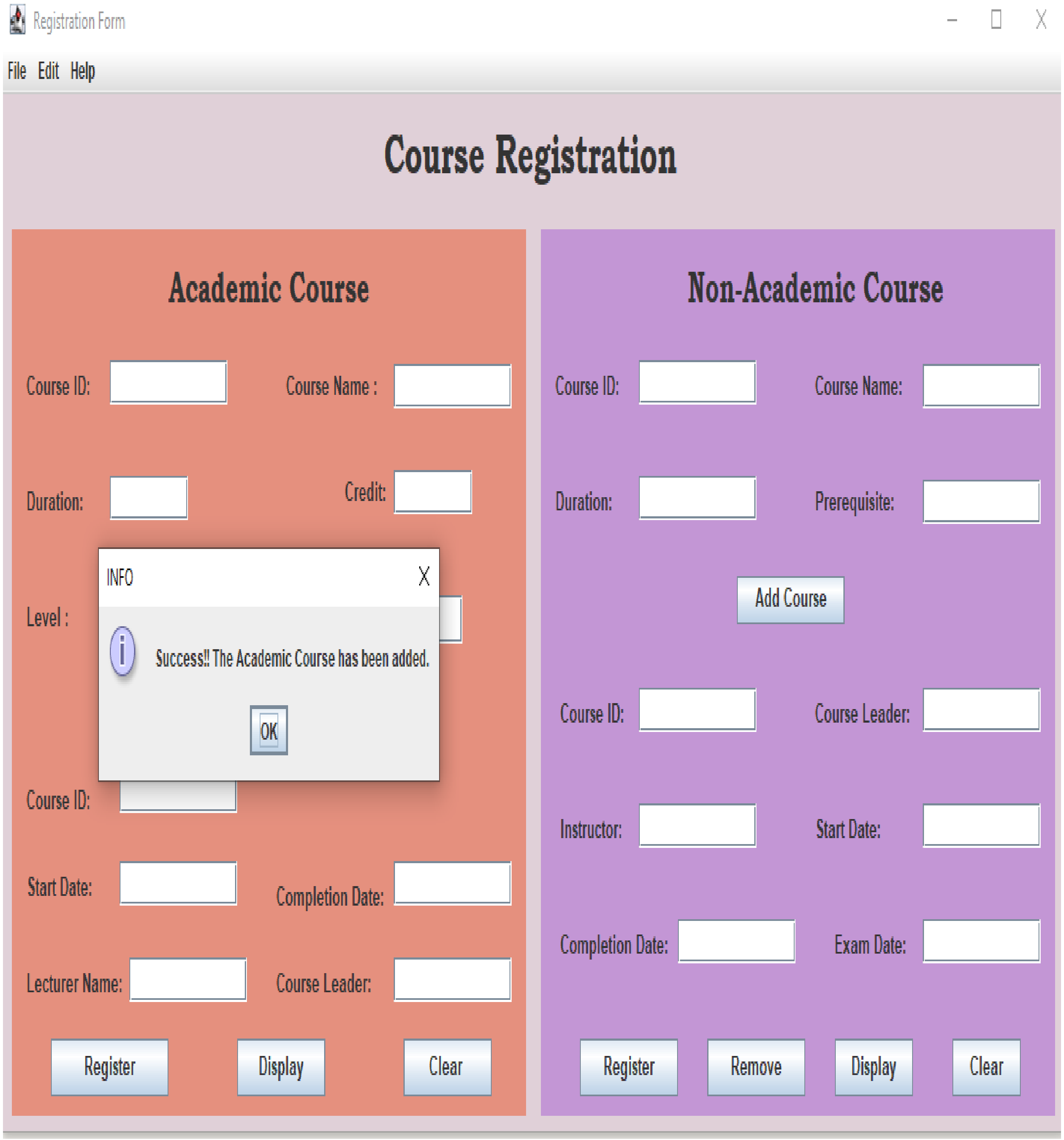
Objective	Test to Add and Register Academic and NonAcademic course and to Remove NonAcademic Course
Action	• Academic Course is Added using following input Course ID = CC4001NI Course Name = Programming Duration = 24 Level = L1 Credit = 30 Number of Assessments = 2 • Academic Course is Registered using following input Course ID = CC4001NI Course Leader = Prithivi Maharjan Lecturer Name = Saroj Kumar Yadav Start Date = 2021-05-3 Completion Date = 2021-05-29 • Non-Academic Course is Added using following input: Course ID = CNA0023 Course Name =Speaker duration = 20 prerequisite = no prerequisite • Non-Academic Course is Added using following input: Course ID = CNA0023

	Course Leader =Lalita Bhattarai Lecturer Name = Ram Baral Start Date = 2021-02-14 Completion Date = 2021-04-23 Exam Date = 2021-06-23 • Non-Academic Course is Removed using following input: Course ID = CNA0023

Expected Result	• The Academic Course would be added, and an appropriate Message Dialog box would appear. • The Academic Course would be Registered, and an appropriate Message Dialog box would appear. • The Non-Academic Course would be added, and an appropriate Message Dialog box would appear. • The Non-Academic Course would be Registered, and an appropriate Message Dialog box would appear. • The Non-Academic Course would be removed, and an appropriate Message Dialog box would appear.
Actual Result	• The Academic Course was added, and an appropriate Message Dialog box appeared. • The Academic Course was Registered, and an appropriate Message Dialog box appeared. • The Non-Academic Course added, and an appropriate Message Dialog box was appeared. • The Non-Academic Course was Registered, and an appropriate Message Dialog box appeared. • The Non-Academic was removed and an appropriate Message Dialog box appeared.

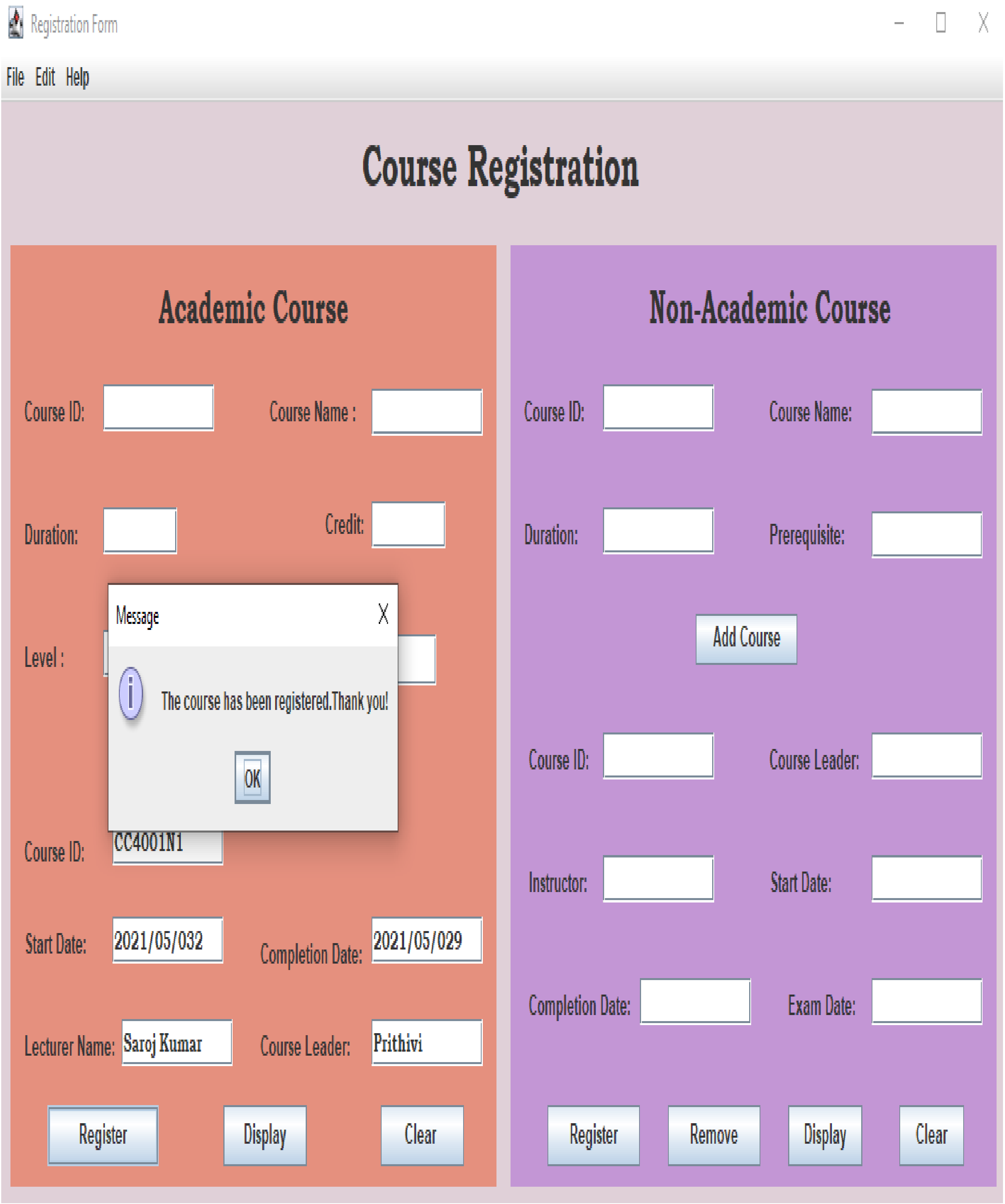
Conclusion	The test was successful

Table 2 Test 2



Success!! The Academic Course has been added.

Figure 4 Test 2i



The course has been registered.Thank you!

Course ID: CC4001N1

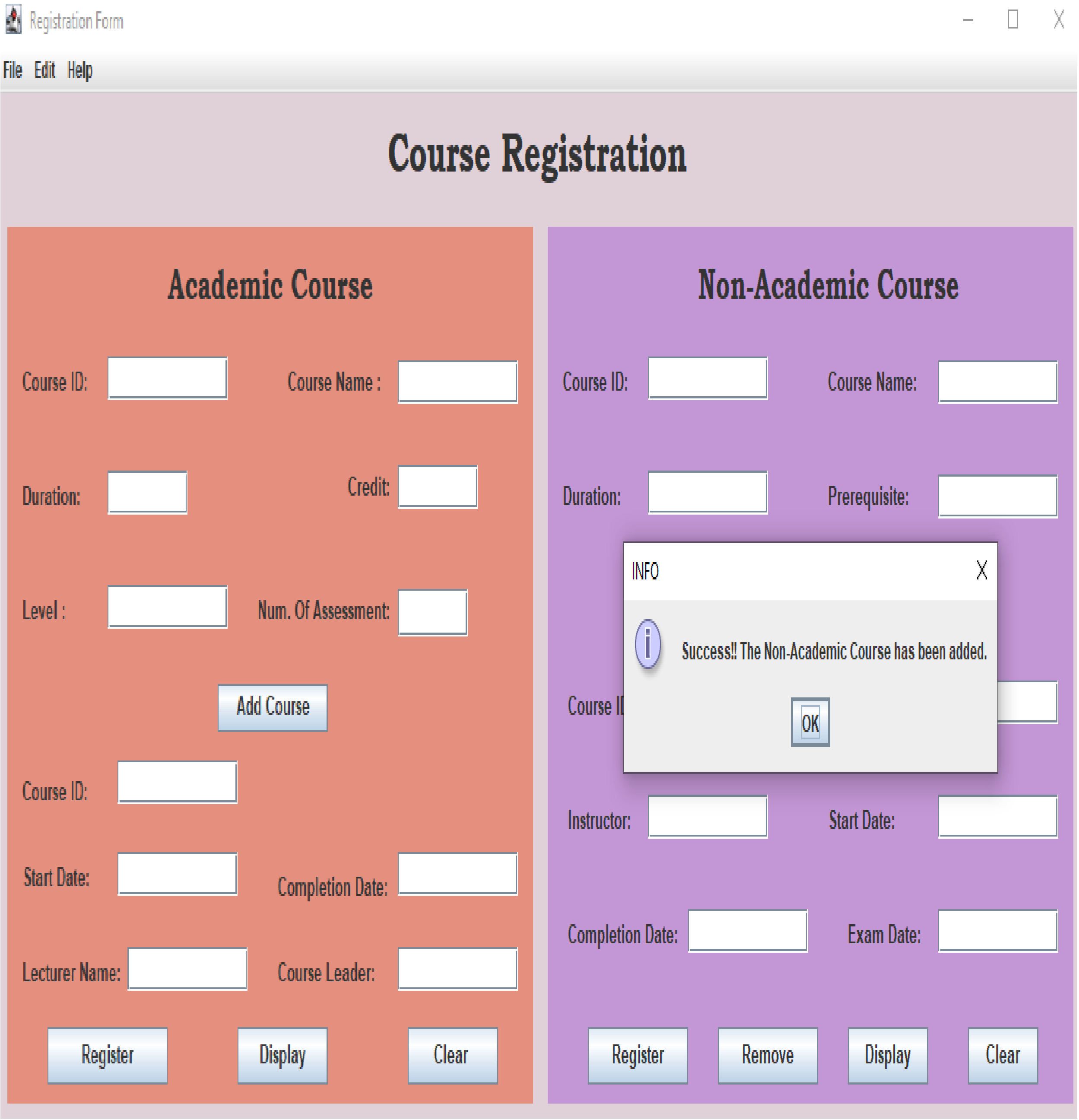
Start Date: 2021/05/032

Completion Date: 2021/05/029

Lecturer Name: Saroj Kumar

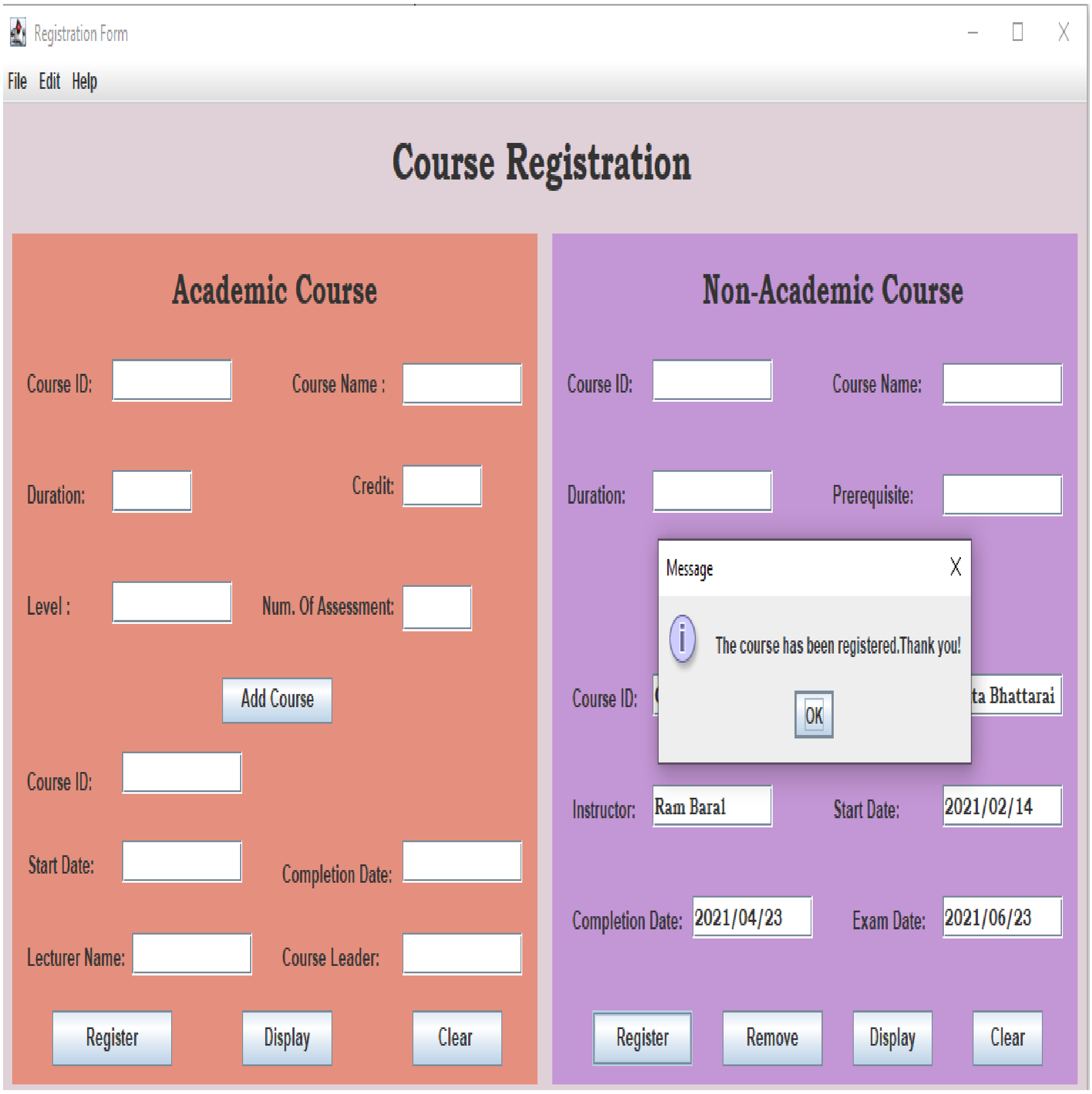
Course Leader: Prithivi

Figure 5 Test 2ii



Success!! The Non-Academic Course has been added.

Figure 6 Test 2 iii



The course has been registered.Thank you!

ta Bhattarai

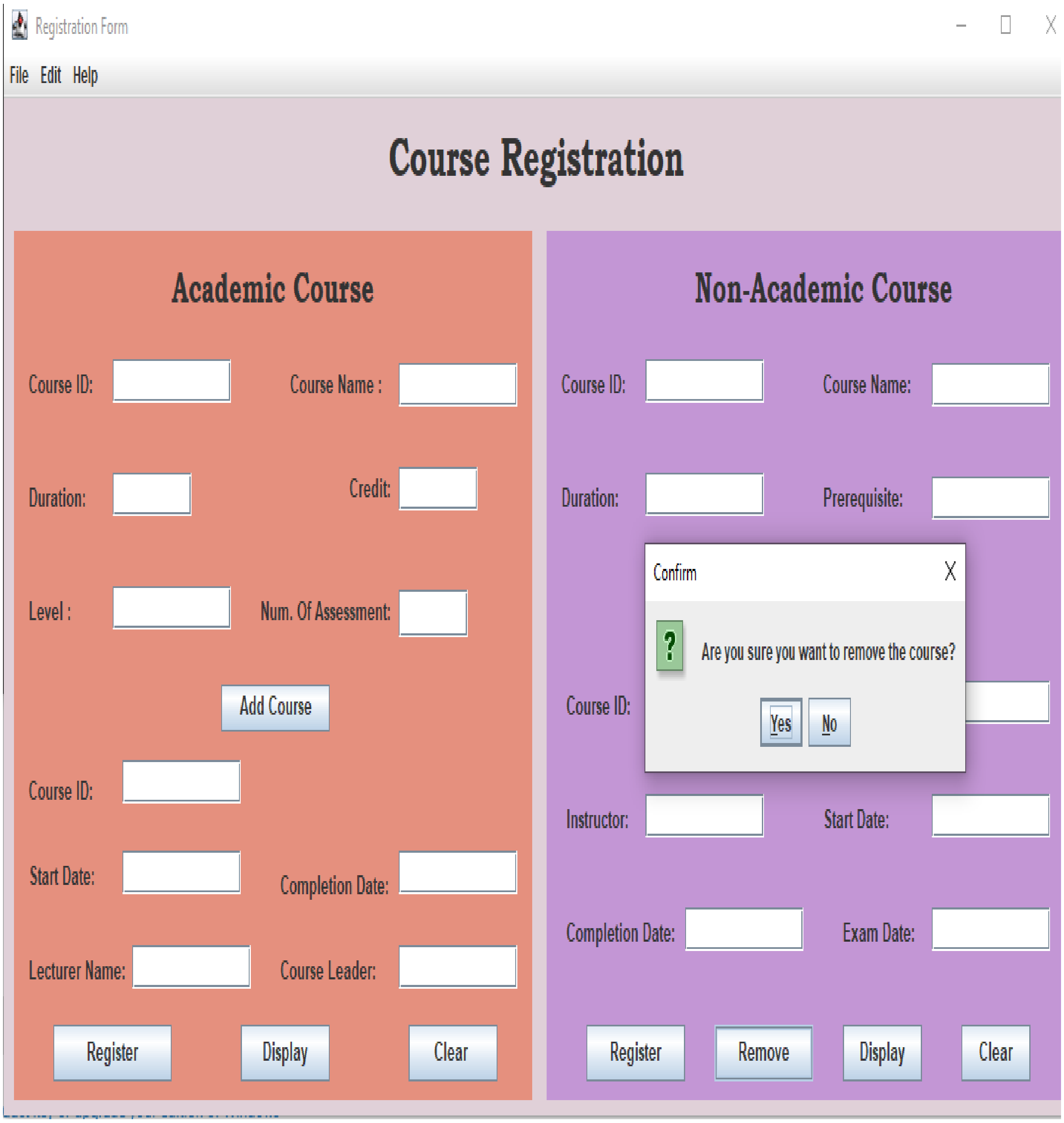
Instructor: Ram Baral

Start Date: 2021/02/14

Completion Date: 2021/04/23

Exam Date: 2021/06/23

Figure 7 Test 2 iv



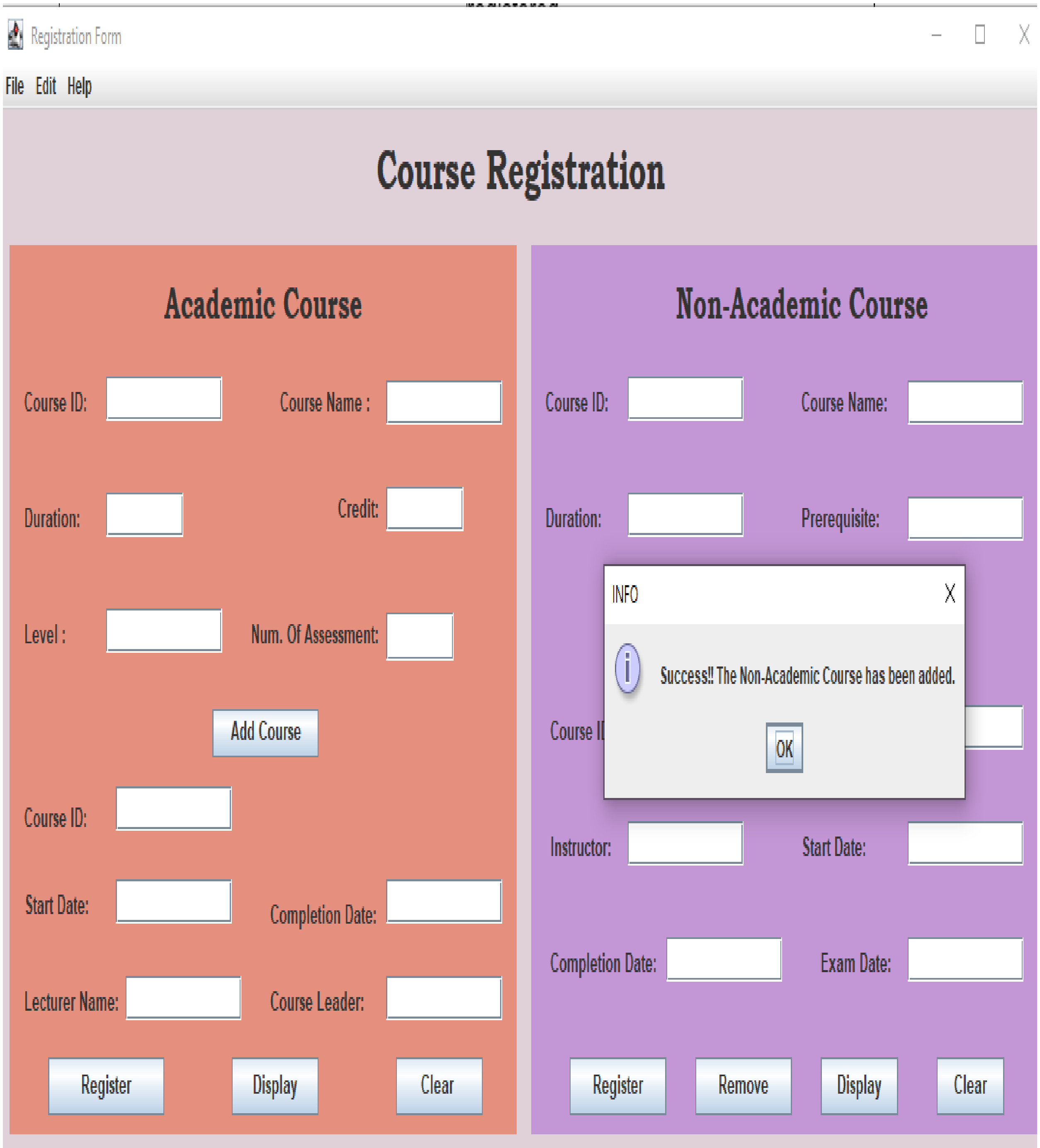
Are you sure you want to remove the course?

Figure 8 Test 2

6.3 Test 3

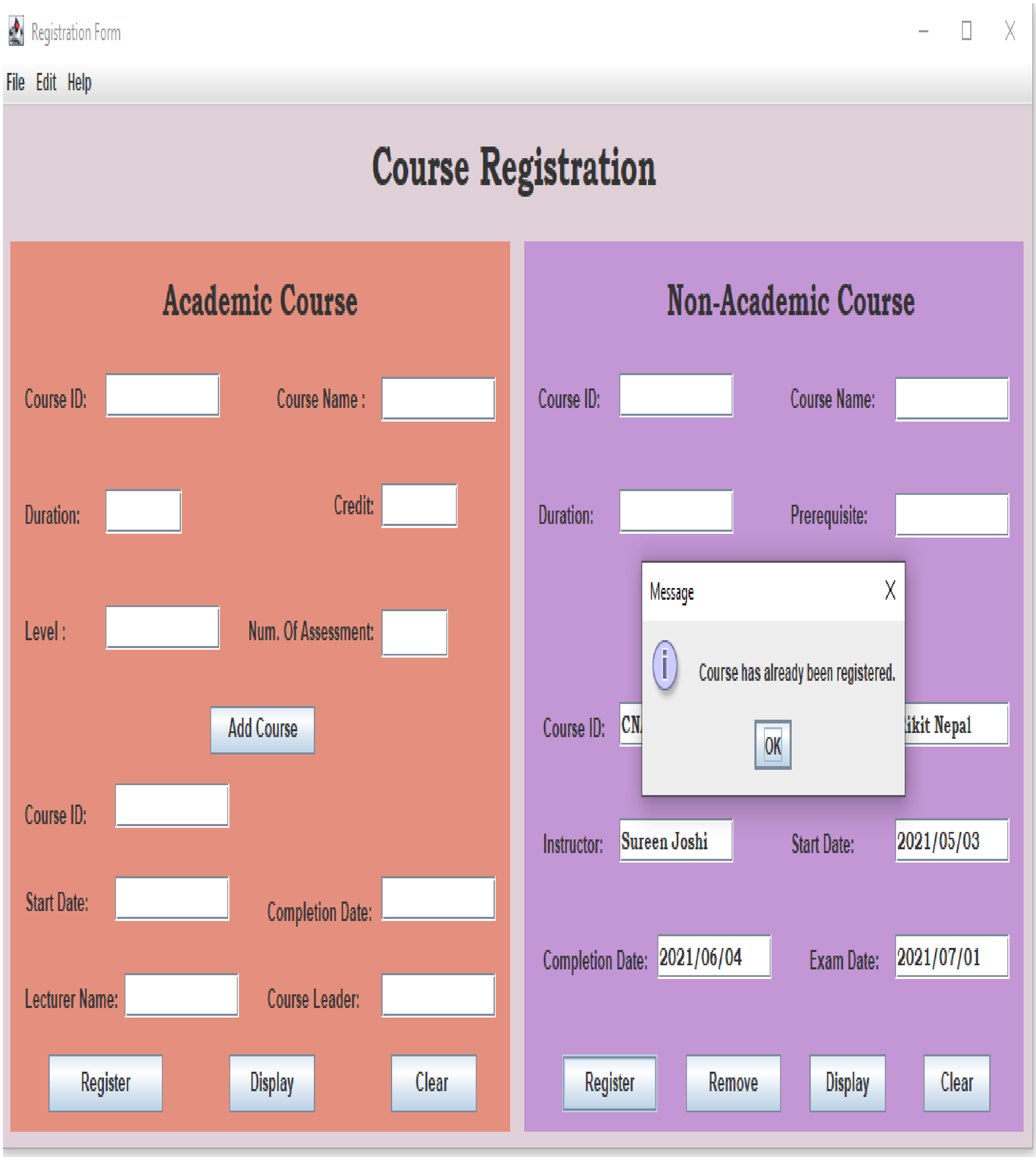
Objective	Test to check if appropriate dialog boxes appear when: • Trying to add duplicate Course ID • Trying to register already registered course • Trying to remove the non-academic course which is already removed.
Action Performed	• A Non-Academic Course is added with Course ID: CNA0023 which is already added • A Non-Academic Course is registered with Course ID : CNA0023 which is already registered • The registered course with Course ID : CNA0023 is removed first and tried to remove again without registering
Expected Result	Appropriate Dialog boxes would appear
Actual Result	Appropriate Dialog boxes appeared
Conclusion	The test was successful

Table 3 test 3



Success!! The Non-Academic Course has been added.

Figure 9 test 3i



Course ID: CN

ikit Nepal

Instructor: Sureen Joshi

Start Date: 2021/05/03

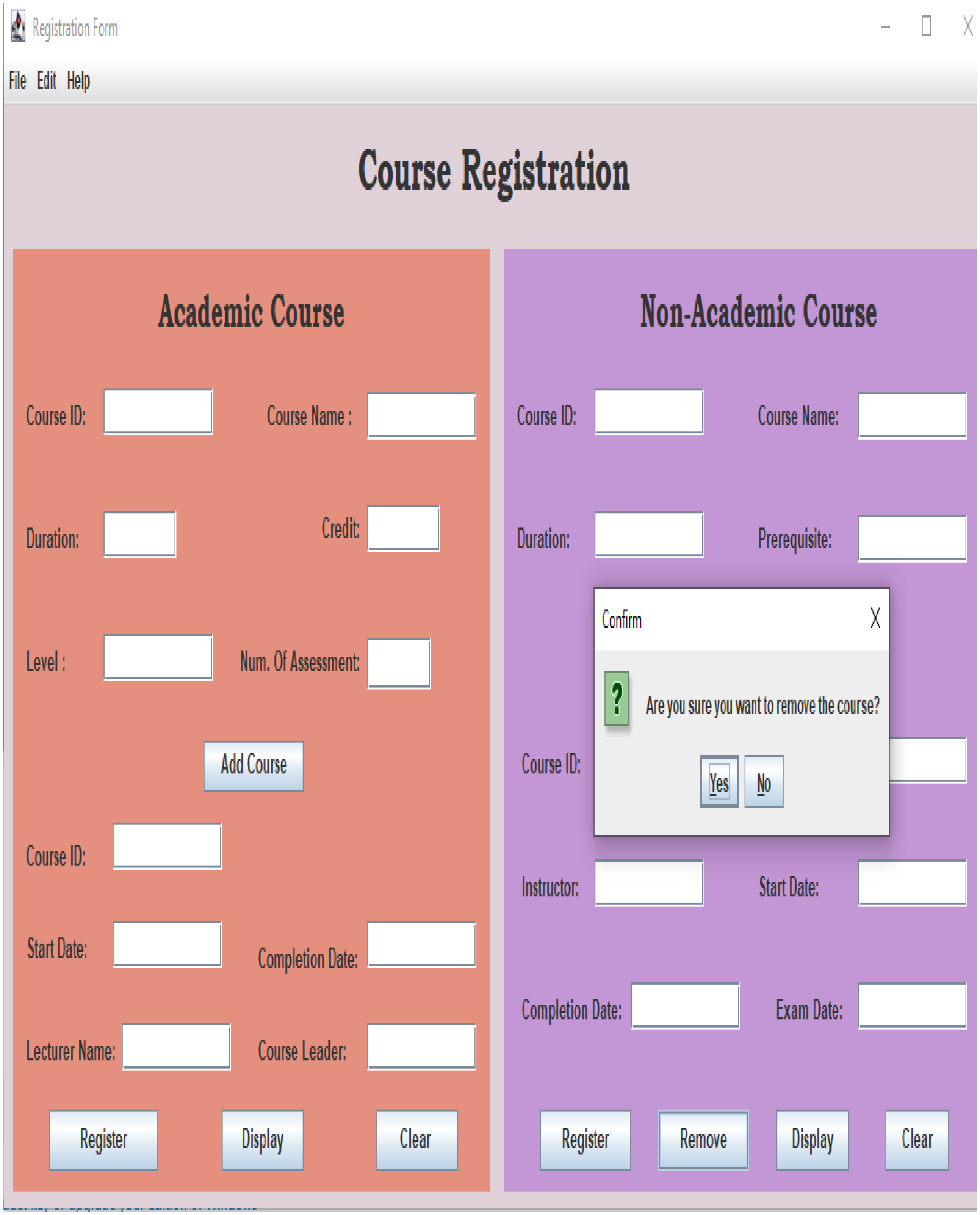
Completion Date: 2021/06/04

Exam Date: 2021/07/01

Message

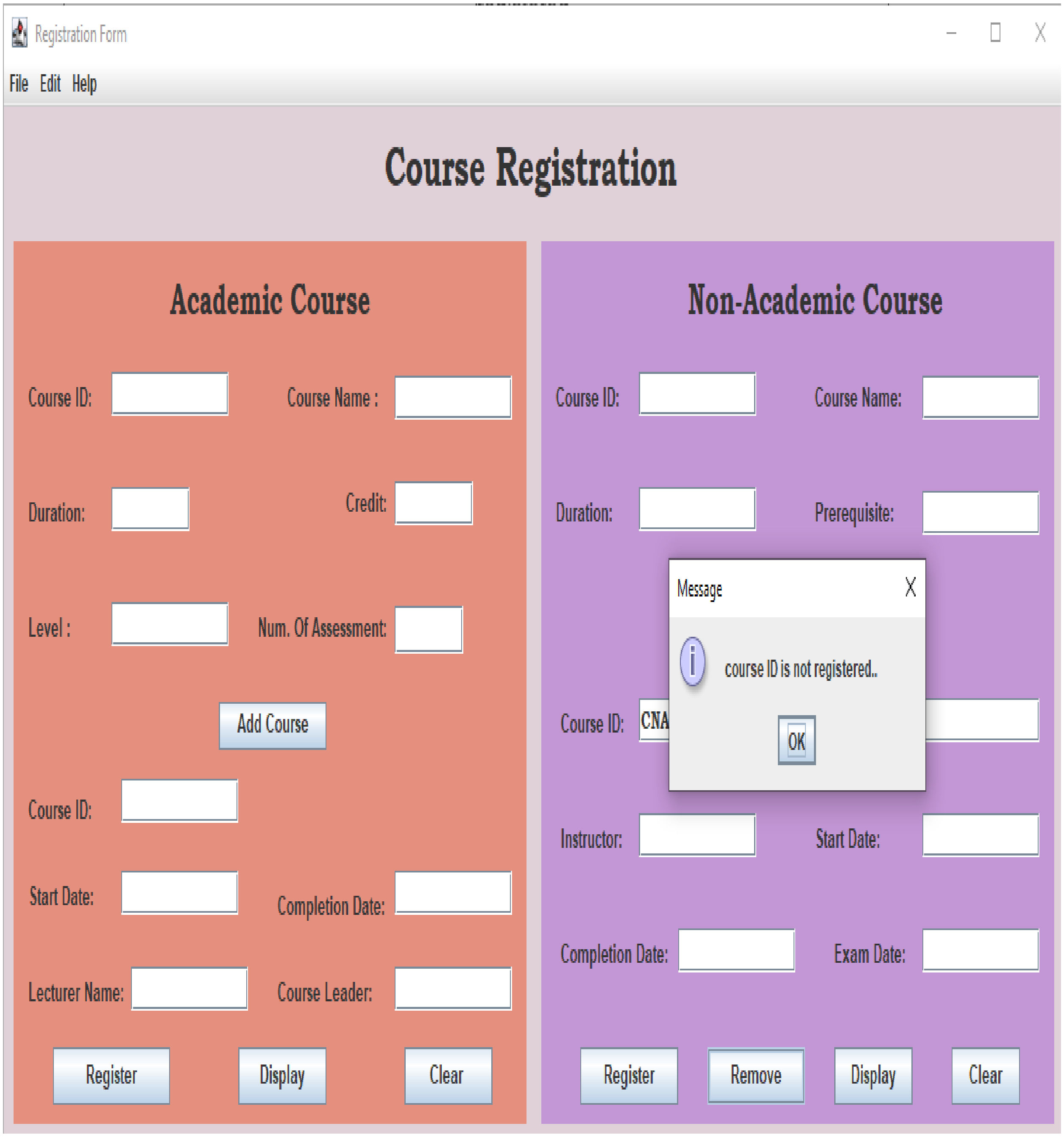
Course has already been registered.

Figure 10 test 3ii



Are you sure you want to remove the course?

Figure 11 test 3iii



Course ID: CNA

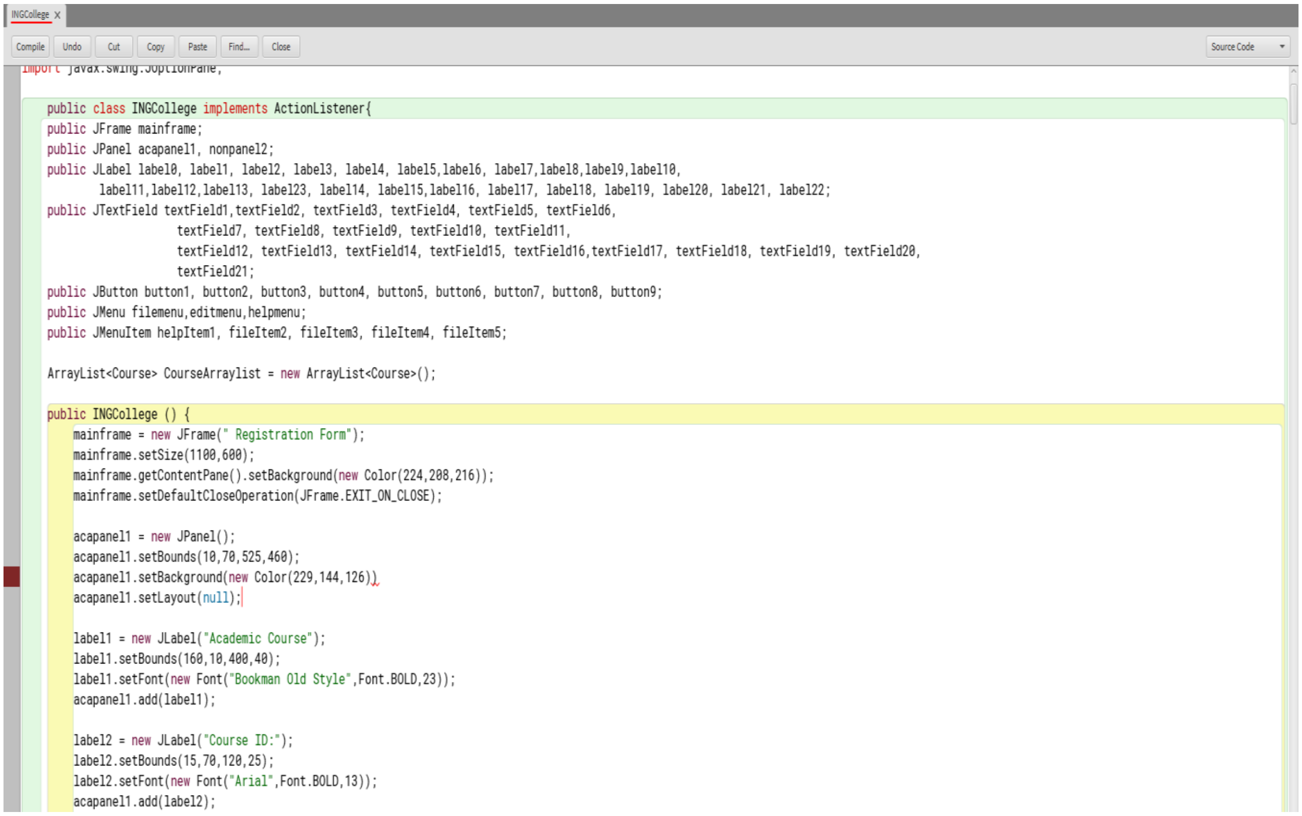
course ID is not registered..

Figure 12 test 3 i

7. Error

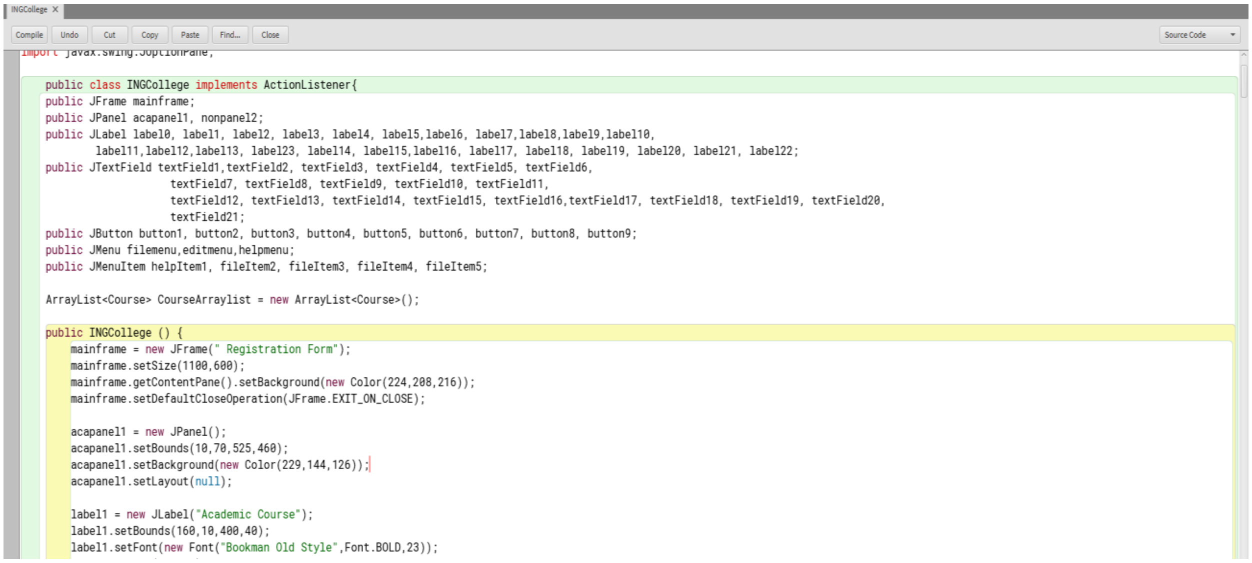
7.1 Syntax Error

Syntax error in a java code is one in which the language you use to create your code is incorrect. It’s a mistake of in source code of the program. While doing this program I found a syntax error as I had missed the semicolon in my source code.



```
import javax.swing.JOptionPane;

public class INGCollege implements ActionListener{
public JFrame mainframe;
public JPanel acapanel1, nonpanel2;
public JLabel label0, label1, label2, label3, label4, label5,label6, label7,label8,label9,label10,
       label11,label12,label13, label23, label14, label15,label16, label17, label18, label19, label20, label21, label22;
public JTextField textField1,textField2, textField3, textField4, textField5, textField6,
                 textField7, textField8, textField9, textField10, textField11,
                 textField12, textField13, textField14, textField15, textField16,textField17, textField18, textField19, textField20,
                 textField21;
public JButton button1, button2, button3, button4, button5, button6, button7, button8, button9;
public JMenu filemenu,editmenu,helpmenu;
public JMenuItem helpItem1, fileItem2, fileItem3, fileItem4, fileItem5;

ArrayList<Course> CourseArraylist = new ArrayList<Course>();

public INGCollege () {
    mainframe = new JFrame(" Registration Form");
    mainframe.setSize(1100,600);
    mainframe.getContentPane().setBackground(new Color(224,208,216));
    mainframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    acapanel1 = new JPanel();
    acapanel1.setBounds(10,70,525,460);
    acapanel1.setBackground(new Color(229,144,126))
    acapanel1.setLayout(null);

    label1 = new JLabel("Academic Course");
    label1.setBounds(160,10,400,40);
    label1.setFont(new Font("Bookman Old Style",Font.BOLD,23));
    acapanel1.add(label1);

    label2 = new JLabel("Course ID:");
    label2.setBounds(15,70,120,25);
    label2.setFont(new Font("Arial",Font.BOLD,13));
    acapanel1.add(label2);
```

Figure 13 Syntax Error

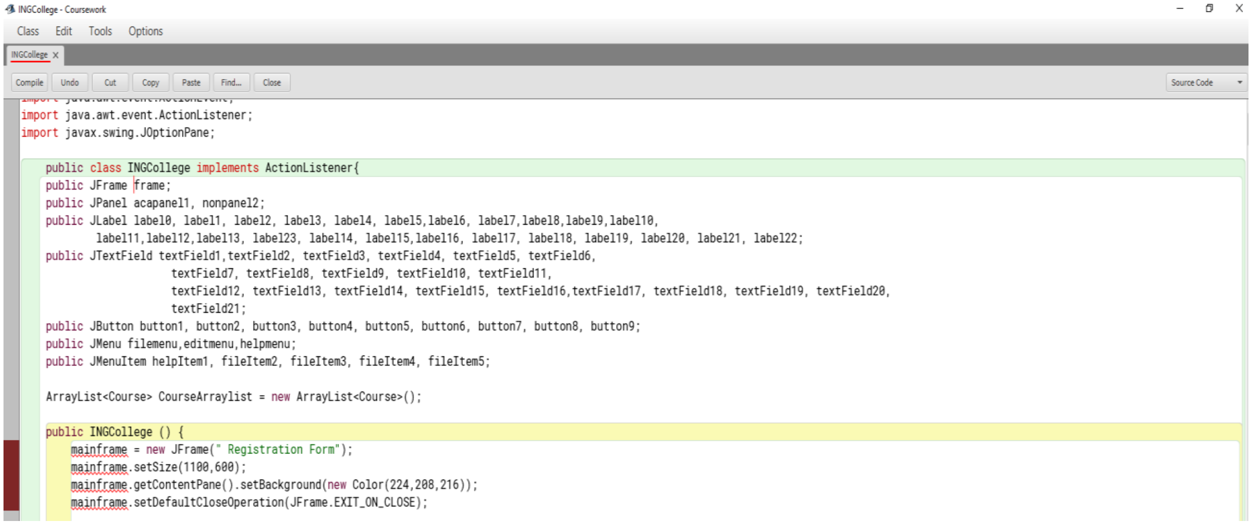


```
import javax.swing.JOptionPane;

    public class INGCollege implements ActionListener{
    public JFrame mainframe;
    public JPanel acapanel1, nonpanel2;
    public JLabel label0, label1, label2, label3, label4, label5,label6, label7,label8,label9,label10,
            label11,label12,label13, label23, label14, label15,label16, label17, label18, label19, label20, label21, label22;
    public JTextField textField1,textField2, textField3, textField4, textField5, textField6,
                      textField7, textField8, textField9, textField10, textField11,
                      textField12, textField13, textField14, textField15, textField16,textField17, textField18, textField19, textField20,
                      textField21;
    public JButton button1, button2, button3, button4, button5, button6, button7, button8, button9;
    public JMenu filemenu,editmenu,helpmenu;
    public JMenuItem helpItem1, fileItem2, fileItem3, fileItem4, fileItem5;

    ArrayList<Course> CourseArraylist = new ArrayList<Course>();

    public INGCollege () {
        mainframe = new JFrame(" Registration Form");
        mainframe.setSize(1100,600);
        mainframe.getContentPane().setBackground(new Color(224,208,216));
        mainframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        acapanel1 = new JPanel();
        acapanel1.setBounds(10,70,525,460);
        acapanel1.setBackground(new Color(229,144,126));
        acapanel1.setLayout(null);

        label1 = new JLabel("Academic Course");
        label1.setBounds(160,10,400,40);
        label1.setFont(new Font("Bookman Old Style",Font.BOLD,23));
```

Figure 14 Correction of Syntax Error

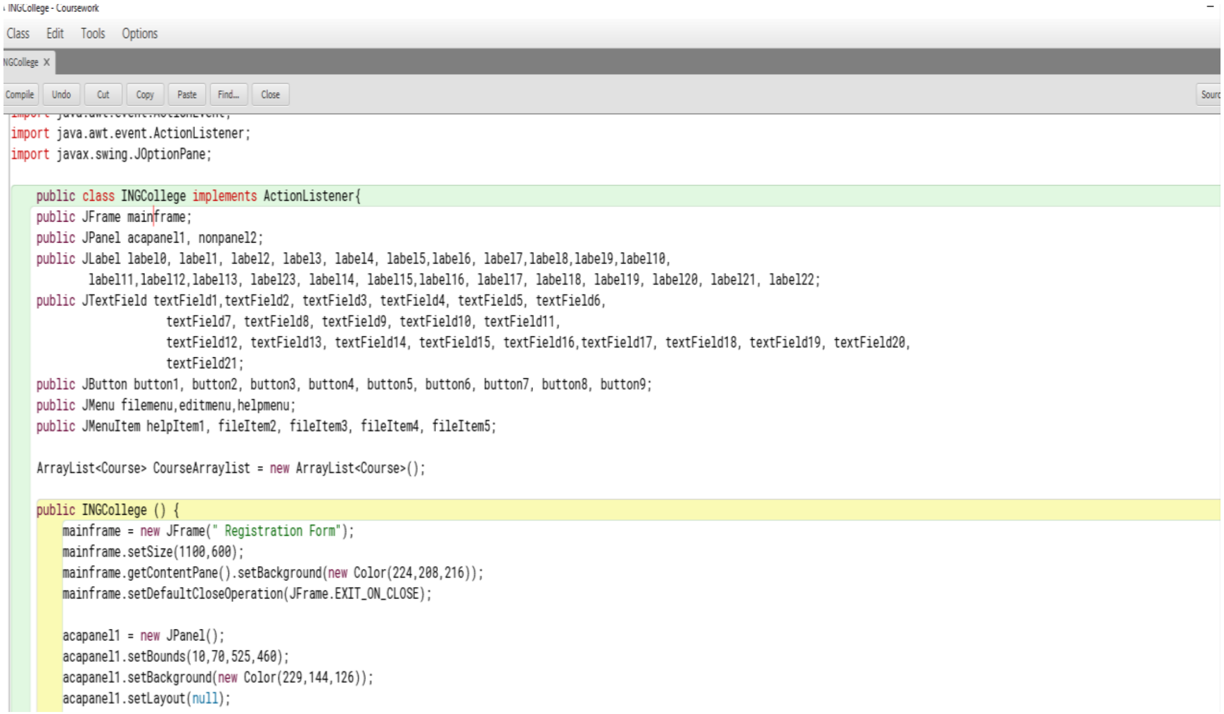
7.2 Semantic Error

The most common semantic error is one in which the code uses a variable that isn’t initialized properly. Error was made in the program by missing declaration of a JFrame in the program. The variable was declared, and the program was resolved.



```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JOptionPane;

    public class INGCollege implements ActionListener{
    public JFrame frame;
    public JPanel acapanel1, nonpanel2;
    public JLabel label0, label1, label2, label3, label4, label5,label6, label7,label8,label9,label10,
            label11,label12,label13, label23, label14, label15,label16, label17, label18, label19, label20, label21, label22;
    public JTextField textField1,textField2, textField3, textField4, textField5, textField6,
                      textField7, textField8, textField9, textField10, textField11,
                      textField12, textField13, textField14, textField15, textField16,textField17, textField18, textField19, textField20,
                      textField21;
    public JButton button1, button2, button3, button4, button5, button6, button7, button8, button9;
    public JMenu filemenu,editmenu,helpmenu;
    public JMenuItem helpItem1, fileItem2, fileItem3, fileItem4, fileItem5;

    ArrayList<Course> CourseArraylist = new ArrayList<Course>();

    public INGCollege () {
        mainframe = new JFrame(" Registration Form");
        mainframe.setSize(1100,600);
        mainframe.getContentPane().setBackground(new Color(224,208,216));
        mainframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Figure 15 Semantic Error

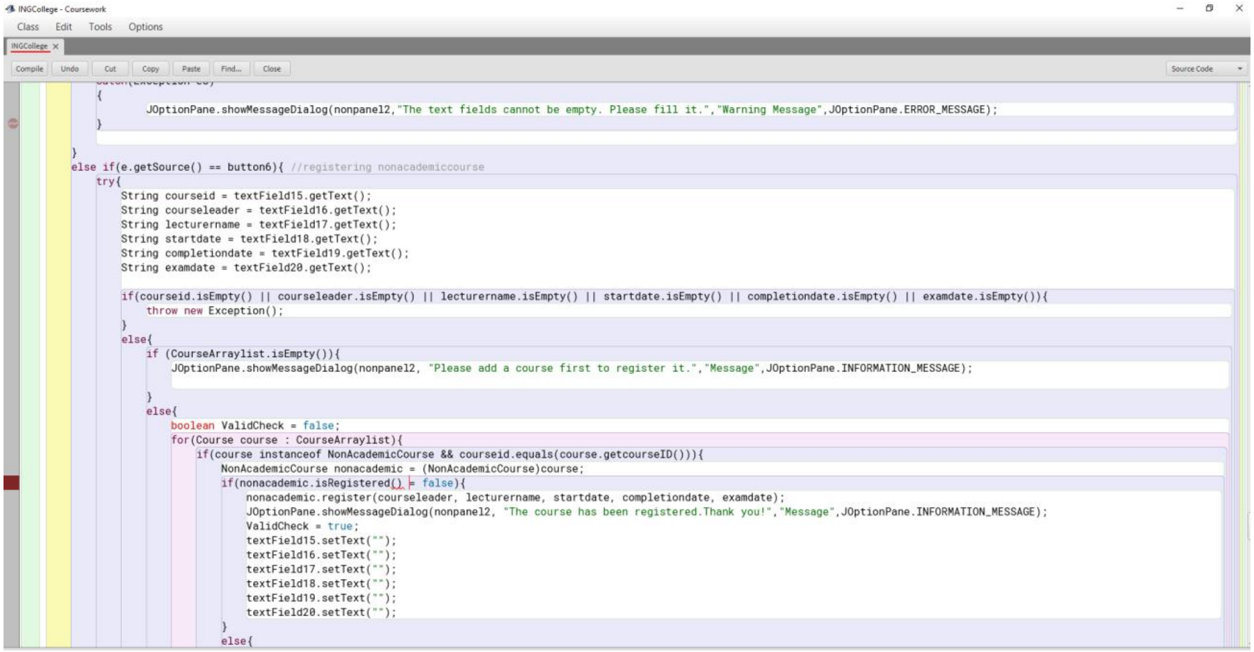


```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JOptionPane;

public class INGCollege implements ActionListener{
public JFrame mainframe;
public JPanel acapanel1, nonpanel2;
public JLabel label0, label1, label2, label3, label4, label5,label6, label7,label8,label9,label10,
       label11,label12,label13, label23, label14, label15,label16, label17, label18, label19, label20, label21, label22;
public JTextField textField1,textField2, textField3, textField4, textField5, textField6,
                 textField7, textField8, textField9, textField10, textField11,
                 textField12, textField13, textField14, textField15, textField16,textField17, textField18, textField19, textField20,
                 textField21;
public JButton button1, button2, button3, button4, button5, button6, button7, button8, button9;
public JMenu filemenu,editmenu,helpmenu;
public JMenuItem helpItem1, fileItem2, fileItem3, fileItem4, fileItem5;

ArrayList<Course> CourseArraylist = new ArrayList<Course>();

public INGCollege () {
    mainframe = new JFrame(" Registration Form");
    mainframe.setSize(1100,600);
    mainframe.getContentPane().setBackground(new Color(224,208,216));
    mainframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    acapanel1 = new JPanel();
    acapanel1.setBounds(10,70,525,460);
    acapanel1.setBackground(new Color(229,144,126));
    acapanel1.setLayout(null);
```

Figure 16 Correction of semantic error

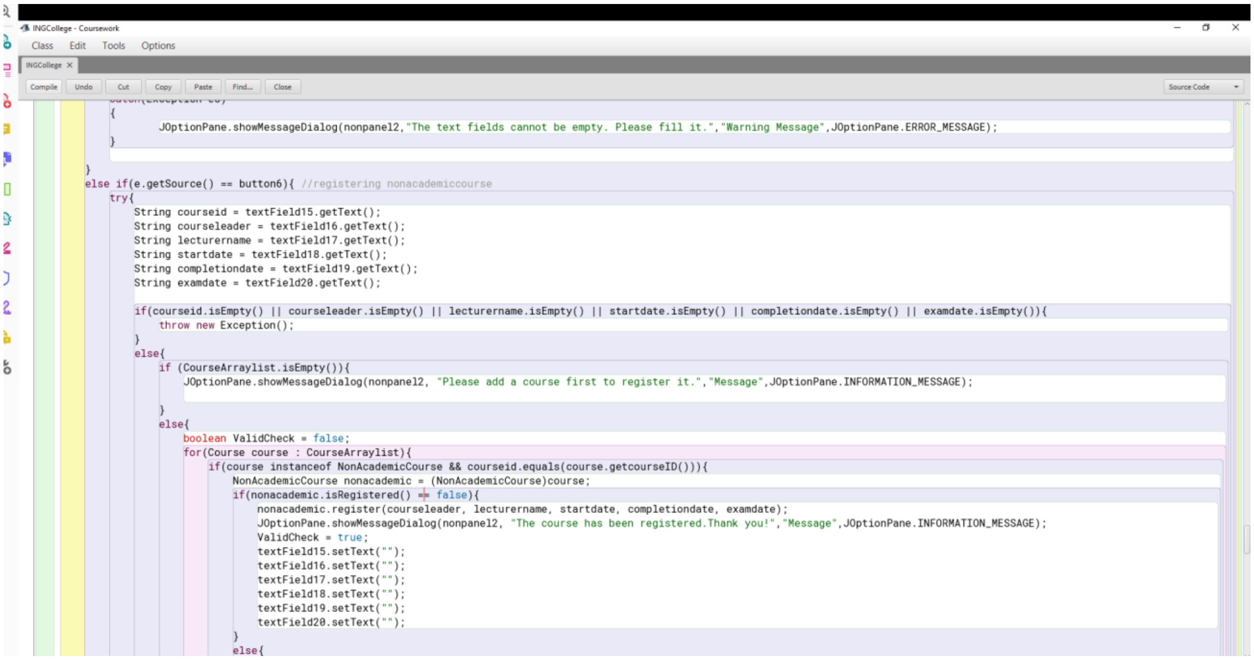
7.3 Logical Error

Logical errors in Java programming can be extremely difficult to find because they don’t reflect any sort of coding problem or an error in the use of Java language elements. While doing this problem, I discovered a logical problem that did not arise during data compilation or data. The error was with the Java operator which was invalid. The Java operator used was "=" while the actual operator to be used was "==". User = assign value while == is used to compare values.



```
    {
        JOptionPane.showMessageDialog(nonpanel2,"The text fields cannot be empty. Please fill it.","Warning Message",JOptionPane.ERROR_MESSAGE);
    }

}
else if(e.getSource() == button6){ //registering nonacademiccourse
    try{
        String courseid = textField15.getText();
        String courseleader = textField16.getText();
        String lecturername = textField17.getText();
        String startdate = textField18.getText();
        String completiondate = textField19.getText();
        String examdate = textField20.getText();

        if(courseid.isEmpty() || courseleader.isEmpty() || lecturername.isEmpty() || startdate.isEmpty() || completiondate.isEmpty() || examdate.isEmpty()){
            throw new Exception();
        }
        else{
            if (CourseArraylist.isEmpty()){
                JOptionPane.showMessageDialog(nonpanel2, "Please add a course first to register it.","Message",JOptionPane.INFORMATION_MESSAGE);

            }
            else{
                boolean ValidCheck = false;
                for(Course course : CourseArraylist){
                    if(course instanceof NonAcademicCourse && courseid.equals(course.getcourseID())){
                        NonAcademicCourse nonacademic = (NonAcademicCourse)course;
                        if(nonacademic.isRegistered() = false){
                            nonacademic.register(courseleader, lecturername, startdate, completiondate, examdate);
                            JOptionPane.showMessageDialog(nonpanel2, "The course has been registered.Thank you!","Message",JOptionPane.INFORMATION_MESSAGE);
                            ValidCheck = true;
                            textField15.setText("");
                            textField16.setText("");
                            textField17.setText("");
                            textField18.setText("");
                            textField19.setText("");
                            textField20.setText("");
                        }
                        else{
```

Figure 17 Logical error



```
    {
        JOptionPane.showMessageDialog(nonpanel2,"The text fields cannot be empty. Please fill it.","Warning Message",JOptionPane.ERROR_MESSAGE);
    }

}
else if(e.getSource() == button6){ //registering nonacademiccourse
    try{
        String courseid = textField15.getText();
        String courseleader = textField16.getText();
        String lecturername = textField17.getText();
        String startdate = textField18.getText();
        String completiondate = textField19.getText();
        String examdate = textField20.getText();

        if(courseid.isEmpty() || courseleader.isEmpty() || lecturername.isEmpty() || startdate.isEmpty() || completiondate.isEmpty() || examdate.isEmpty()){
            throw new Exception();
        }
        else{
            if (CourseArraylist.isEmpty()){
                JOptionPane.showMessageDialog(nonpanel2, "Please add a course first to register it.","Message",JOptionPane.INFORMATION_MESSAGE);

            }
            else{
                boolean ValidCheck = false;
                for(Course course : CourseArraylist){
                    if(course instanceof NonAcademicCourse && courseid.equals(course.getcourseID())){
                        NonAcademicCourse nonacademic = (NonAcademicCourse)course;
                        if(nonacademic.isRegistered() == false){
                            nonacademic.register(courseleader, lecturername, startdate, completiondate, examdate);
                            JOptionPane.showMessageDialog(nonpanel2, "The course has been registered.Thank you!","Message",JOptionPane.INFORMATION_MESSAGE);
                            ValidCheck = true;
                            textField15.setText("");
                            textField16.setText("");
                            textField17.setText("");
                            textField18.setText("");
                            textField19.setText("");
                            textField20.setText("");
                        }
                        else{
```

Figure 18 Correction of logical error

8. Conclusion

Java is an integrated development platform for Java programming language, designed specifically for educational purposes, but also suitable for small software development.

Java was developed to support the learning and teaching of an object-oriented program, and its structure differs from other areas of development as a result. The large screen clearly shows the layout of the application phase under development and objects can be built collaboratively and tested.

In this coursework , we were asked to create a GUI program form that stores Course data which includes both subject and non-academic subjects using java swing and awt. The INGCollage section was supposed to keep a list of course types to hold the Academic Course and Nonacademic Course.

During this duration , I learned how to use Java swing and awt to create GUI forms in Java. The implementation of the GUI did not cause much difficulty. The real hard part would add functionality to the Buttons form using the action listener. Adding exactly the item to the list of the same members and the external manager was very confusing. However, with the help of my instructor, I was able to pass on lectures and video recordings, and I was able to do my coursework.

Appendix

For ING Collage

```
           import javax.swing.*;
import java.awt.*;
import java.util.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JOptionPane;

    public class INGCollege implements ActionListener{
    public JFrame mainframe;
    public JPanel acapanel1, nonpanel2;
    public JLabel label0, label1, label2, label3, label4, label5,label6,
label7,label8,label9,label10,
            label11,label12,label13, label23, label14, label15,label16, label17, label18,
label19, label20, label21, label22;
    public JTextField textField1,textField2, textField3, textField4, textField5, textField6,
                        textField7, textField8, textField9, textField10, textField11,
                        textField12, textField13, textField14, textField15, textField16,textField17,
textField18, textField19, textField20,
                        textField21;
    public JButton button1, button2, button3, button4, button5, button6, button7, button8,
button9;
    public JMenu filemenu,editmenu,helpmenu;
    public JMenuItem helpItem1, fileItem2, fileItem3, fileItem4, fileItem5;

    ArrayList<Course> CourseArraylist = new ArrayList<Course>();

    public INGCollege () {
        mainframe = new JFrame(" Registration Form");
```

```
mainframe.setSize(1100,600);
mainframe.getContentPane().setBackground(new Color(224,208,216));
mainframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

acapanel1 = new JPanel();
acapanel1.setBounds(10,70,525,460);
acapanel1.setBackground(new Color(229,144,126));
acapanel1.setLayout(null);

label1 = new JLabel("Academic Course");
label1.setBounds(160,10,400,40);
label1.setFont(new Font("Bookman Old Style",Font.BOLD,23));
acapanel1.add(label1);

label2 = new JLabel("Course ID:");
label2.setBounds(15,70,120,25);
label2.setFont(new Font("Arial",Font.BOLD,13));
acapanel1.add(label2);

textField1 = new JTextField();
textField1.setBounds(100,68,120,23);
textField1.setFont(new Font("Bookman Old Style",Font.BOLD,13));
acapanel1.add(textField1);

label3 = new JLabel("Course Name :");
label3.setBounds(280,65,120,35);
label3.setFont(new Font("Arial",Font.BOLD,13));
acapanel1.add(label3);

textField2 = new JTextField();
textField2.setBounds(390,70,120,23);
```

```
textField2.setFont(new Font("Bookman Old Style",Font.BOLD,13));
acapanel1.add(textField2);

label4 = new JLabel("Duration: ");
label4.setBounds(15,130,120,25);
label4.setFont(new Font("Arial",Font.BOLD,13));
acapanel1.add(label4);

textField3 = new JTextField();
textField3.setBounds(100,128,80,23);
textField3.setFont(new Font("Bookman Old Style",Font.BOLD,13));
acapanel1.add(textField3);

label5 = new JLabel("Credit: ");
label5.setBounds(340,120,190,35);
label5.setFont(new Font("Arial",Font.BOLD,13));
acapanel1.add(label5);

textField4 = new JTextField();
textField4.setBounds(390,125,80,23);
textField4.setFont(new Font("Bookman Old Style",Font.BOLD,13));
acapanel1.add(textField4);

label6 = new JLabel("Level :");
label6.setBounds(15,190,120,25);
label6.setFont(new Font("Arial",Font.BOLD,13));
acapanel1.add(label6);

textField5 = new JTextField();
textField5.setBounds(100,188,120,23);
textField5.setFont(new Font("Bookman Old Style",Font.BOLD,13));
```

```
acapanel1.add(textField5);

label7 = new JLabel("Num. Of Assessment: ");
label7.setBounds(250,185,180,35);
label7.setFont(new Font("Arial",Font.BOLD,13));
acapanel1.add(label7);

textField6 = new JTextField();
textField6.setBounds(390,190,70,25);
textField6.setFont(new Font("Bookman Old Style",Font.BOLD,13));
acapanel1.add(textField6);


label23 = new JLabel("Course ID: ");
label23.setBounds(15,280,180,35);
label23.setFont(new Font("Arial",Font.BOLD,13));
acapanel1.add(label23);

textField21 = new JTextField();
textField21.setBounds(110,280,120,23);
textField21.setFont(new Font("Bookman Old Style",Font.BOLD,13));
acapanel1.add(textField21);

label8 = new JLabel("Start Date: ");
label8.setBounds(15,330,120,25);
label8.setFont(new Font("Arial",Font.BOLD,13));
acapanel1.add(label8);

textField7 = new JTextField();
textField7.setBounds(110,328,120,23);
textField7.setFont(new Font("Bookman Old Style",Font.BOLD,13));
```

```
acapanel1.add(textField7);

label9 = new JLabel("Completion Date: ");
label9.setBounds(270,330,150,35);
label9.setFont(new Font("Arial",Font.BOLD,13));
acapanel1.add(label9);

textField8 = new JTextField();
textField8.setBounds(390,328,120,23);
textField8.setFont(new Font("Bookman Old Style",Font.BOLD,13));
acapanel1.add(textField8);

label10 = new JLabel("Lecturer Name: ");
label10.setBounds(15,380,130,25);
label10.setFont(new Font("Arial",Font.BOLD,13));
acapanel1.add(label10);

textField9 = new JTextField();
textField9.setBounds(120,378,120,23);
textField9.setFont(new Font("Bookman Old Style",Font.BOLD,13));
acapanel1.add(textField9);

label11 = new JLabel("Course Leader: ");
label11.setBounds(270,380,130,25);
label11.setFont(new Font("Arial",Font.BOLD,13));
acapanel1.add(label11);


textField10 = new JTextField();
textField10.setBounds(390,378,120,23);
textField10.setFont(new Font("Bookman Old Style",Font.BOLD,13));
```

```
acapanel1.add(textField10);

button1 = new JButton("Add Course");
button1.setFont(new Font("Arial",Font.BOLD,13));
button1.setBounds(210,240,110,25);
acapanel1.add(button1);
button1.addActionListener(this);

button2 = new JButton("Register");
button2.setFont(new Font("Arial",Font.BOLD,13));
button2.setBounds(40,420,120,30);
acapanel1.add(button2);
button2.addActionListener(this);

button3 = new JButton(" Display ");
button3.setFont(new Font("Arial",Font.BOLD,13));
button3.setBounds(230,420,90,30);
button3.addActionListener(this);

acapanel1.add(button3);

button4 = new JButton("Clear ");
button4.setFont(new Font("Arial",Font.BOLD,13));
button4.setBounds(400,420,90,30);
button4.addActionListener(this);
acapanel1.add(button4);


//code for non academic course in panel 2

nonpanel2 = new JPanel();
```

```
nonpanel2.setBounds(550,70,525,460);
nonpanel2.setBackground(new Color(195,150,213));
nonpanel2.setLayout(null);

label12 = new JLabel("Non-Academic Course");
label12.setBounds(150,10,400,40);
label12.setFont(new Font("Bookman Old Style",Font.BOLD,23));
nonpanel2.add(label12);

label13 = new JLabel("Course ID:");
label13.setBounds(15,70,120,25);
label13.setFont(new Font("Arial",Font.BOLD,13));
nonpanel2.add(label13);

textField11 = new JTextField();
textField11.setBounds(100,68,120,23);
textField11.setFont(new Font("Bookman Old Style",Font.BOLD,13));
nonpanel2.add(textField11);

label14 = new JLabel("Course Name:");
label14.setBounds(280,65,120,35);
label14.setFont(new Font("Arial",Font.BOLD,13));
nonpanel2.add(label14);

textField12 = new JTextField();
textField12.setBounds(390,70,120,23);
textField12.setFont(new Font("Bookman Old Style",Font.BOLD,13));
nonpanel2.add(textField12);

label15 = new JLabel("Duration:");
label15.setBounds(15,130,120,25);
```

```
label15.setFont(new Font("Arial",Font.BOLD,13));
nonpanel2.add(label15);

textField13 = new JTextField();
textField13.setBounds(100,128,120,23);
textField13.setFont(new Font("Bookman Old Style",Font.BOLD,13));
nonpanel2.add(textField13);

label16 = new JLabel("Prerequisite: ");
label16.setBounds(280,125,190,35);
label16.setFont(new Font("Arial",Font.BOLD,13));
nonpanel2.add(label16);

textField14 = new JTextField();
textField14.setBounds(390,130,120,23);
textField14.setFont(new Font("Bookman Old Style",Font.BOLD,13));
nonpanel2.add(textField14);

button5 = new JButton("Add Course");
button5.setFont(new Font("Arial",Font.BOLD,13));
button5.setBounds(200,180,110,25);
nonpanel2.add(button5);
button5.addActionListener(this);

label17 = new JLabel("Course ID: ");
label17.setBounds(20,240,120,25);
label17.setFont(new Font("Arial",Font.BOLD,13));
nonpanel2.add(label17);

textField15= new JTextField();
textField15.setBounds(100,238,120,23);
```

```
textField15.setFont(new Font("Bookman Old Style",Font.BOLD,13));
nonpanel2.add(textField15);

label18 = new JLabel("Course Leader: ");
label18.setBounds(280,235,150,35);
label18.setFont(new Font("Arial",Font.BOLD,13));
nonpanel2.add(label18);

textField16 = new JTextField();
textField16.setBounds(390,238,120,23);
textField16.setFont(new Font("Bookman Old Style",Font.BOLD,13));
nonpanel2.add(textField16);

label19 = new JLabel("Instructor: ");
label19.setBounds(20,300,120,25);
label19.setFont(new Font("Arial",Font.BOLD,13));
nonpanel2.add(label19);

textField17 = new JTextField();
textField17.setBounds(100,298,120,23);
textField17.setFont(new Font("Bookman Old Style",Font.BOLD,13));
nonpanel2.add(textField17);


label20 = new JLabel("Start Date: ");
label20.setBounds(280,295,150,35);
label20 .setFont(new Font("Arial",Font.BOLD,13));
nonpanel2.add(label20 );

textField18 = new JTextField();
textField18.setBounds(390,298,120,23);
```

```
textField18.setFont(new Font("Bookman Old Style",Font.BOLD,13));
nonpanel2.add(textField18);

label21 = new JLabel("Completion Date: ");
label21.setBounds(20,360,120,25);
label21.setFont(new Font("Arial",Font.BOLD,13));
nonpanel2.add(label21);

textField19 = new JTextField();
textField19.setBounds(140,358,120,23);
textField19.setFont(new Font("Bookman Old Style",Font.BOLD,13));
nonpanel2.add(textField19);


label22 = new JLabel("Exam Date: ");
label22.setBounds(300,355,150,35);
label22 .setFont(new Font("Arial",Font.BOLD,13));
nonpanel2.add(label22);

textField20 = new JTextField();
textField20.setBounds(390,358,120,23);
textField20.setFont(new Font("Bookman Old Style",Font.BOLD,13));
nonpanel2.add(textField20);

button6 = new JButton("Register");
button6.setFont(new Font("Arial",Font.BOLD,13));
button6.setBounds(40,420,100,30);
nonpanel2.add(button6);
button6.addActionListener(this);

button7 = new JButton(" Remove ");
```

```
button7.setFont(new Font("Arial",Font.BOLD,13));
button7.setBounds(170,420,100,30);
nonpanel2.add(button7);
button7.addActionListener(this);


button8 = new JButton("Display");
button8.setFont(new Font("Arial",Font.BOLD,13));
button8.setBounds(300,420,80,30);
nonpanel2.add(button8);
button8.addActionListener(this);

button9 = new JButton("Clear");
button9.setFont(new Font("Arial",Font.BOLD,13));
button9.setBounds(420,420,70,30);
button9.addActionListener(this);
nonpanel2.add(button9);

JMenuBar menubar = new JMenuBar();
filemenu = new JMenu("File");
filemenu.add(new JSeparator());

editmenu = new JMenu("Edit");
editmenu.add(new JSeparator());

helpmenu = new JMenu("Help");
helpItem1 = new JMenuItem("About");

helpItem1.addActionListener(this);
helpmenu.add(new JSeparator());
```

```
fileItem2 = new JMenuItem("New1");

fileItem2.addActionListener(this);

fileItem3 = new JMenuItem("Save");
fileItem3.addActionListener(this);


fileItem4 = new JMenuItem("Exit");
fileItem4.addActionListener(this);



filemenu.add(fileItem2);
filemenu.add(fileItem3);
filemenu.add(fileItem4);
helpmenu.add(helpItem1);
menubar.add(filemenu);
menubar.add(editmenu);
menubar.add(helpmenu);
mainframe.setJMenuBar(menubar);



mainframe.setLocationRelativeTo(null);
mainframe.setVisible(true);
mainframe.setLayout(null);
mainframe.setResizable(true);
mainframe.add(acapanel1);
```

```
        mainframe.add(nonpanel2);



        JLabel label0 = new JLabel ("Course Registration");
        label0.setBounds(390,10,400,40);
        label0.setFont(new Font("Bookman Old Style",Font.BOLD,29));
        mainframe.add(label0);

    }

        public void actionPerformed(ActionEvent e){

            if(e.getSource()==button1){ /*adding academic course*/
              try{
              String courseid = textField1.getText();
              String coursename = textField2.getText();
              String duration = textField3.getText();
              String level = textField4.getText();
              String Credit = textField5.getText();
              String numOfAssessments = textField6.getText();
              if(courseid.isEmpty() || coursename.isEmpty() || duration.isEmpty() ||
level.isEmpty() || Credit.isEmpty() ||
              numOfAssessments.isEmpty()){
                  throw new Exception();
              }
              else{
                  int Duration = Integer.parseInt(duration);
                  int NumOfAssessments = Integer.parseInt(numOfAssessments);

                  AcademicCourse academic = new AcademicCourse(courseid,
coursename,Duration , level, Credit,NumOfAssessments);
```

```
          textField1.setText("");
          textField2.setText("");
          textField3.setText("");
          textField4.setText("");
          textField5.setText("");
          textField6.setText("");


          CourseArraylist.add(academic);
          JOptionPane.showMessageDialog(acapanel1, "Success!! The Academic
Course has been added.", "INFO", JOptionPane.INFORMATION_MESSAGE);



        }
      }

      catch(NumberFormatException es)
          {
        JOptionPane.showMessageDialog(acapanel1, "Please enter a whole number
in duration and no. of assessment.","WARNING MESSAGE",
JOptionPane.ERROR_MESSAGE);
      }

      catch(Exception es)
      {
          JOptionPane.showMessageDialog(acapanel1,"The text fields cannot be
empty. Please fill it.","Warning Message",JOptionPane.ERROR_MESSAGE);
      }
    }
    else if(e.getSource()==button2) { // for registering academic course
      try{
        String courseid = textField21.getText();
        String courseleader = textField10.getText();
        String lecturername = textField9.getText();
        String startdate = textField7.getText();
        String completiondate = textField8.getText();


        if(courseid.isEmpty() || courseleader.isEmpty() || lecturername.isEmpty() ||
startdate.isEmpty() || completiondate.isEmpty()){
          throw new Exception();
        }
        else{
          if (CourseArraylist.isEmpty()){
            JOptionPane.showMessageDialog(acapanel1, "Please add a course first
to register it.","Message",JOptionPane.INFORMATION_MESSAGE);
          }
          else{
            boolean ValidCheck = false;
            //int courseid = Integer.parseInt(courseID);
            for (Course co: CourseArraylist){
              if(co instanceof AcademicCourse &&
courseid.equals(co.getcourseID())){
                AcademicCourse ac = (AcademicCourse)co;
                if(ac.getisRegistered()==false){
                  ac.register(courseleader, lecturername, startdate, completiondate
);
                  JOptionPane.showMessageDialog(acapanel1, "The course has
been registered.Thank you!","Message",JOptionPane.INFORMATION_MESSAGE);
```

```
                                    ValidCheck = true;
                                    textField21.setText("");
                                    textField10.setText("");
                                    textField9.setText("");
                                    textField7.setText("");
                                    textField8.setText("");
                                }
                                else{
                                    JOptionPane.showMessageDialog(acapanel1, "Ops! The Course has already been registered.","Message",JOptionPane.INFORMATION_MESSAGE);
                                    ValidCheck = true;
                                }

                            }



                        }

                        if(!ValidCheck){
                            JOptionPane.showMessageDialog(acapanel1, "Sorry! No course with such ID exist.","Message",JOptionPane.INFORMATION_MESSAGE);

                            textField21.setText("");
                            textField10.setText("");
                            textField9.setText("");
                            textField7.setText("");
                            textField8.setText("");
                        }

                    }
                }
                }
            catch(Exception es)
            {
                JOptionPane.showMessageDialog(acapanel1, "Fields cannot be empty.
Please fill it.","Warning Message",JOptionPane.ERROR_MESSAGE);
            }
        }
        else if(e.getSource() == button4 ){
                textField1.setText(""); textField2.setText(""); textField3.setText("");
textField4.setText("");
                textField5.setText(""); textField6.setText(""); textField7.setText("");
textField8.setText("");
                textField9.setText(""); textField10.setText(""); textField21.setText("");
        }
        else if(e.getSource() == button4){
            if(CourseArraylist.size()<=0){
                JOptionPane.showMessageDialog(acapanel1,"Course is not yet
registered!","INFORMATION",JOptionPane.INFORMATION_MESSAGE);
            }


        else if (e.getSource() == button3) { //displaying academic course
                    for (Course course: CourseArraylist){
                    if(course instanceof AcademicCourse)
                    {
                        AcademicCourse academic = (AcademicCourse)course;
                        academic.display();
                    }
                    else{
                        NonAcademicCourse nonacademic = (NonAcademicCourse)course;
                        nonacademic.display();
                    }


                }

            }
        }
        else if(e.getSource() == button5){ //addingnonacdemic
            try{
                String courseid = textField11.getText();
                String coursename = textField12.getText();
                String Duration = textField13.getText();
                String Prerequisite = textField14.getText();

                if(courseid.isEmpty() || coursename.isEmpty() || Duration.isEmpty() || Prerequisite.isEmpty()){
                    throw new Exception();
                }
                else{
                    int duration = Integer.parseInt(Duration);
                    NonAcademicCourse nonacademic = new NonAcademicCourse(courseid, coursename, duration, Prerequisite);

                    textField11.setText("");
                    textField12.setText("");
                    textField13.setText("");
                    textField14.setText("");


                    CourseArraylist.add(nonacademic);
```

```
          JOptionPane.showMessageDialog(nonpanel2, "Success!! The Non-
Academic Course has been added.", "INFO",
JOptionPane.INFORMATION_MESSAGE);




        }
      }
       catch(NumberFormatException es)
          {
        JOptionPane.showMessageDialog(nonpanel2, "Please enter a whole number
in duration textfield.","WARNING MESSAGE", JOptionPane.ERROR_MESSAGE);
      }

      catch(Exception es)
      {
          JOptionPane.showMessageDialog(nonpanel2,"The text fields cannot be
empty. Please fill it.","Warning Message",JOptionPane.ERROR_MESSAGE);
      }

    }
    else if(e.getSource() == button6){ //registering nonacademiccourse
      try{
        String courseid = textField15.getText();
        String courseleader = textField16.getText();
        String lecturername = textField17.getText();
        String startdate = textField18.getText();
        String completiondate = textField19.getText();
        String examdate = textField20.getText();
        if(courseid.isEmpty() || courseleader.isEmpty() || lecturername.isEmpty() || startdate.isEmpty() || completiondate.isEmpty() || examdate.isEmpty()){
            throw new Exception();
        }
        else{
            if (CourseArraylist.isEmpty()){
                JOptionPane.showMessageDialog(nonpanel2, "Please add a course first to register it.","Message",JOptionPane.INFORMATION_MESSAGE);

            }
            else{
                boolean ValidCheck = false;
                for(Course course : CourseArraylist){
                    if(course instanceof NonAcademicCourse && courseid.equals(course.getcourseID())){
                        NonAcademicCourse nonacademic = (NonAcademicCourse)course;
                        if(nonacademic.isRegistered() == false){
                            nonacademic.register(courseleader, lecturername, startdate, completiondate, examdate);
                            JOptionPane.showMessageDialog(nonpanel2, "The course has been registered.Thank you!","Message",JOptionPane.INFORMATION_MESSAGE);
                            ValidCheck = true;
                            textField15.setText("");
                            textField16.setText("");
                            textField17.setText("");
                            textField18.setText("");
                            textField19.setText("");
                            textField20.setText("");
                        }
                        else{
```

```
                                    JOptionPane.showMessageDialog(nonpanel2, " Course has
already been registered.","Message",JOptionPane.INFORMATION_MESSAGE);
                                    ValidCheck = true;
                                    textField15.setText("");
                                    textField16.setText("");
                                    textField17.setText("");
                                    textField18.setText("");
                                    textField19.setText("");
                                    textField20.setText("");
                                }

                            }

                        }
                        if(!ValidCheck){
                            JOptionPane.showMessageDialog(nonpanel2, "Sorry! No any course
with such ID is added.","Message",JOptionPane.INFORMATION_MESSAGE);
                            textField15.setText("");
                        }
                    }
                }
            }
            catch(Exception es){
                JOptionPane.showMessageDialog(nonpanel2, "Fields cannot be
empty.Please fill it.","Warning Message",JOptionPane.ERROR_MESSAGE);
            }
        }

        else if(e.getSource() == button9){
                textField11.setText(""); textField12.setText(""); textField13.setText("");
textField14.setText("");
```

```
        textField15.setText(""); textField16.setText(""); textField17.setText("");
textField18.setText("");
        textField19.setText(""); textField20.setText("");
    }
    else if(e.getSource() == button7){

      String courseid = textField15.getText();
      String courseleader = textField16.getText();
      String instructorname = textField17.getText();
      String startdate = textField18.getText();
      String completiondate = textField19.getText();
      String examdate = textField20.getText();


      boolean ValidCheck = false;
          for(Course course : CourseArraylist){
            if(course instanceof NonAcademicCourse &&
courseid.equals(course.getcourseID())){
              NonAcademicCourse nonacademic = (NonAcademicCourse)course;
              if(nonacademic.isRemoved() == false)
              {
                nonacademic.remove();

          int response= JOptionPane.showConfirmDialog(nonpanel2, "Are you sure
you want to remove the course?","Confirm",
          JOptionPane.YES_NO_OPTION,JOptionPane.QUESTION_MESSAGE);
                if (response == JOptionPane.YES_OPTION){
            ValidCheck = true;
            textField11.setText("");
            textField12.setText("");
            textField13.setText("");
```

```
                    textField14.setText("");
                    textField15.setText("");
                    textField16.setText("");
                    textField17.setText("");
                    textField18.setText("");
                    textField19.setText("");
                    textField20.setText("");
                    System.out.println("The Course has been removed.");
                }

                }
            }
        }

                if(!ValidCheck){
                    JOptionPane.showMessageDialog(nonpanel2, "course ID is not
registered..","Message",JOptionPane.INFORMATION_MESSAGE);
                }




    }
    else if(e.getSource() == button8 || e.getSource()== button3){



            for (Course course: CourseArraylist){
                if(course instanceof AcademicCourse)
                {
                    AcademicCourse academic = (AcademicCourse)course;
                    academic.display();
                }
```

```
                    else{
                        NonAcademicCourse nonacademic = (NonAcademicCourse)course;
                        nonacademic.display();

                    }



                }




        }
        else if(e.getSource() == fileItem2)
            {
                textField1.setText(""); textField2.setText(""); textField3.setText("");
textField4.setText(""); textField5.setText("");
                textField6.setText(""); textField7.setText(""); textField8.setText("");
textField9.setText(""); textField10.setText("");
                textField11.setText("");textField12.setText("");textField13.setText("");
textField14.setText(""); textField15.setText("");
                textField16.setText(""); textField17.setText(""); textField18.setText("");
textField19.setText(""); textField20.setText("");
                textField21.setText("");
                }

        else if(e.getSource() == fileItem4){
                System.exit(0);

            }

        else if(e.getSource() == helpItem1)
```

```
            {

            JOptionPane.showMessageDialog(mainframe, "This GUI is a registration form
which consists of information related to academic and non academic course in Islington
College.",
                                                         "INFO", JOptionPane.INFORMATION_MESSAGE);

            }

    }



   public static void main(String[]arg)
       {
          new INGCollege();
       }
}
```

FOR course

```
public class Course {
    private String courseID;
    private String courseName;
    private String courseLeader;
    private int duration;
    public Course(String courseID, String courseName, int duration){
        this.courseID = courseID;
        this.courseName = courseName;
        this.duration = duration;
        this.courseLeader = "";
    }
    public String getcourseID(){
        return courseID;
    }
    public String getcourseName(){
        return courseName;
    }
    public int getduration(){
        return duration;
    }
    public void setcourseLeader(String courseLeader){
        this.courseLeader = courseLeader;
    }
    public String getcourseLeader(){
        return courseLeader;
    }
    public void display(){
        System.out.println("The course ID is: " + courseID);
        System.out.println("The course name is: " + courseName);
        System.out.println("The course duration is: " + duration);
        if(courseLeader!=""){
        System.out.println("The course lead is: " + courseLeader);
        }
    }
}
```

FOR academic Course

```
public class AcademicCourse extends Course{
    private String lecturerName;
    private String level;
    private String credit;
    private String startDate;
    private String completionDate;
    private int numOfAssessments;
    Boolean isRegistered;
    public AcademicCourse(String courseID, String courseName, int duration, String
level, String credit, int numOfAssessment){
        super(courseID, courseName, duration);
        this.level = level;
        this.credit = credit;
        this.numOfAssessments = numOfAssessment;
        this.lecturerName = "";
        this.startDate = "";
        this.completionDate = "";
        this.isRegistered = false;
    }
    String getlevel(){
        return level;
    }
    String getcredit(){
        return credit;
    }
    int getnumOfAssessments(){
        return numOfAssessments;
    }
    String getstartDate(){
        return startDate;
    }
    String getcompletionDate(){
        return completionDate;
    }
    Boolean getisRegistered(){
        return isRegistered;
    }
    void setlecturerName(String lecturerName){
        this.lecturerName = lecturerName;
    }
    String getlecturerName(){
        return lecturerName;
    }
    void setnumOfAssessment(int newnumOfAssessments){
        this.numOfAssessments = newnumOfAssessments;
    }
    void register(String courseLeader, String lecturerName, String startDate, String
completionDate){
        if(this.isRegistered == true){
            System.out.println("This course is already registered.");
            System.out.println("The name of the lecturer is " + lecturerName);
            System.out.println("The start date of the course is " + startDate);
            System.out.println("The completion date of the course is  " + completionDate);
        }
        else{
            super.setcourseLeader(courseLeader);
            this.lecturerName = lecturerName;
            this.startDate= startDate;
            this.completionDate = completionDate;
            this.isRegistered = true;
        }
    }
    public void display(){
        super.display();
        if(this.isRegistered = true) {
            System.out.println("The name of the lecturer is " + lecturerName);
            System.out.println("The level of the course is " + level);
            System.out.println("The total credit of the course is " + credit);
            System.out.println("The start date of the course is " + startDate);
            System.out.println("The completion date of the course is  " + completionDate);
            System.out.println("The total number of assessments of this course is " +
numOfAssessments);
        }
        else{
            System.out.println("This course is not registered.");
        }
    }
}
```

For Nonacademic course

```
public class NonAcademicCourse extends Course{
    private String instructorName;
    private int duration;
    private String startDate;
    private String completionDate;
    private String examDate;
    private String prerequisite;
    private Boolean isRegistered;
    private Boolean isRemoved;
    public NonAcademicCourse(String courseID, String courseName, int duration, String
prerequisite){
        super(courseID, courseName, duration);
        this.prerequisite = prerequisite;
        this.startDate = "";
        this.completionDate = "";
        this.examDate = "";
        this.isRegistered = false;
        this.isRemoved = false;
    }
    int duration(){
        return duration;
    }
    String startDate(){
        return startDate;
    }
    String completionDate(){
        return completionDate;
    }
    String examDate(){
        return examDate;
    }
    String prerequisite(){
        return prerequisite;
    }
    Boolean isRegistered(){
        return isRegistered;
    }
    Boolean isRemoved(){
        return isRemoved;
    }
    void setinstructorName(String instructorName){
        if(this.isRegistered == true){
            System.out.println("This course is already registered. So it is not possible to
change the instructor name");
        }
        else{
            this.instructorName = instructorName;
        }
    }
    void register(String courseLeader, String instructorName, String startDate, String completionDate, String examDate){
        if(this.isRegistered == true){
            System.out.println("This course is already registered.");
        }
        else{
            super.setcourseLeader(courseLeader);
            this.instructorName = instructorName;
            this.startDate= startDate;
            this.completionDate = completionDate;
            this.examDate = examDate;
            this.isRegistered = true;
        }
    }
    void remove(){
        if(this.isRemoved == true){
            System.out.println("The course is already removed");
        }
        else{
            System.out.println("The course is removed");
            super.setcourseLeader("");
            this.instructorName = "";
            this.startDate= "";
            this.completionDate = "";
            this.examDate = "";
            this.isRemoved = true;
        }
    }
    public void display(){
        super.display();
        if(this.isRegistered = true){
            System.out.println("The name of the instructor is " + this.instructorName);
            System.out.println("The course will start from " + this.startDate);
            System.out.println("The course will be completed in " + this.completionDate);
            System.out.println("The exam of this course will be held in " + this.examDate);
        }
        else{
            System.out.println("The course has not yet been registered");
        }
    }
}
```

Bibliography

Anon. (2005). *Diagrams*. Retrieved from Anon: https://www.diagrams.net/

Oracle. (1999). *BlueJ*. Retrieved from BlueJ: bluej.org