Module Code & Module Title

CS4001NI Programming

COURSEWORK-2

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2021 Spring

Student Name : Punam Thapa Magar

Group : C3

London Met ID:20048968

College ID : NPO1CP4S210273

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 in order for my assignment to be accepted and marked. I am fully aware that late submission will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

This is the second coursework which was appointed to us from the module “Introduction to Programming”. The program is done by utilizing BlueJ. The primary nature of this coursework is to develop the Graphical User Interface (GUI) which is a type of UI that permits client to collaborate with electronic devices through graphical symbols and sound pointer like essential documentation, rather than text based UI, typed command labels or text navigation. Moreover, the main motive of this project is to make a class INGCollege which is directly addressed to the course and two subclasses namely academic and non-academic course class respectively. The program comprise of frame, panel, buttons and textfield alongside the execution of actionlistener, eventhandling and exceptionhandling with the goal that the class of java swings are introduced in a well way.

In all sincerity, In this coursework we were approached to make GUI for a framework which stores the details of Course including academic course and non academic course from our past coursework and to make a new project with another class called INGCollege. So basically, The INGCollege is the subclass orta child class where Course is the parent class and the academic and non academic classes are also a child class. Furthermore, The INGCollege class set the information about class of java swings, java.awt, java.awt.event, java.util.Arraylist and java.util.Iterator and so on. The title given for the Frame was Course Registration and the the gridLayout was set. In this same way, Different components were created along with their setFonts, setBounds and setBackground color.

A actionPerformed(ActionEvent e) method was used to direct event handling and exception handling. Moreover, getSource() object was used to returns the object on which the event was occurring. In this project, the exception was handled using try catch block.

2 Class Diagram

Class diagram is a static diagram, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram. (Anon., n.d.)

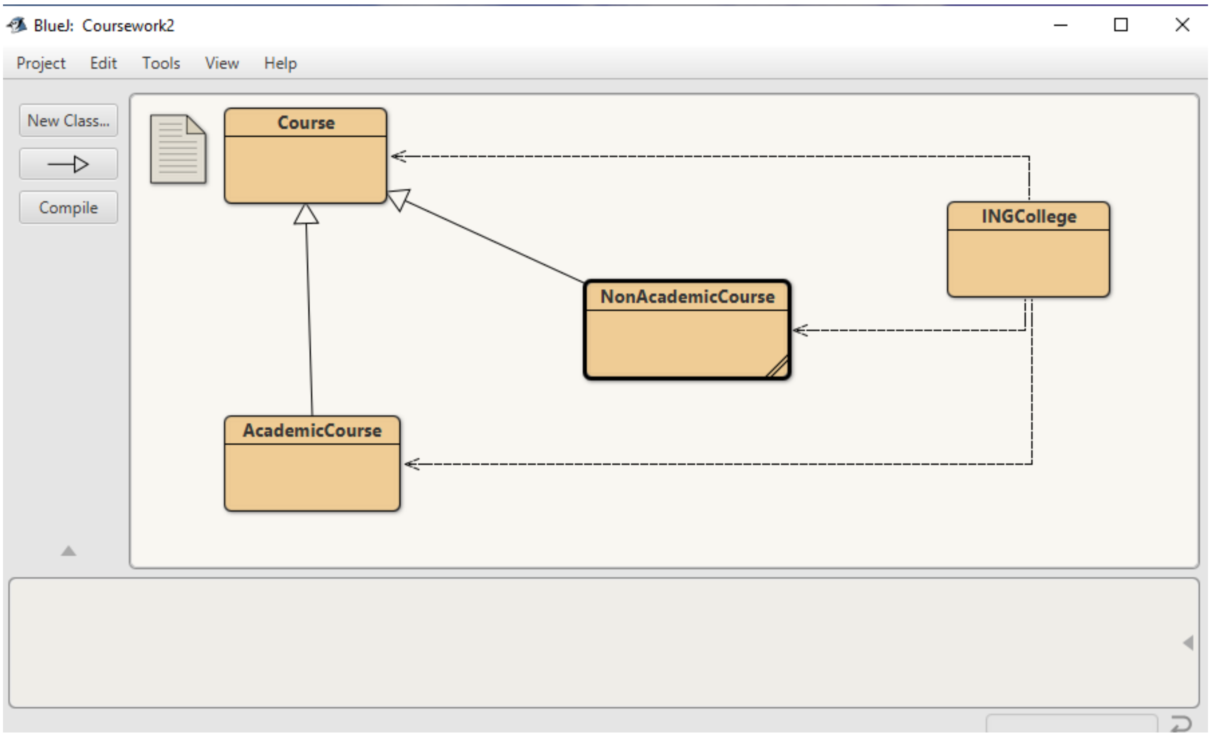


Figure 1 Class Diagram of classes in BlueJ

2.1 Class Diagram of INGCollege Class

INGCollege
-JFrame : frame -JPanel : panel, panel2 JLabel : title, n_title, CourseID, n_CourseID, Coursename, n_Coursename, Courseleader, n_Courseleader, Duration, n_Duration, Lecturername, n_Lecturername, Level, n_Level, Credit, n_Credit, Startdate, n_Startdate, Completiondate, n_Completiondate, Numberofassessments, n_Numberofassessments, Instructorname, n_Instructorname, Examdate, n_Examdate, Prerequisites, n_Prerequisites. -JTextField: Ac_ CourseID, non_CourseID, Ac_Coursename, non_Coursename, Ac_Courseleader, non_Courseleader, Ac_Duration, non_Duration, Ac_Lecturername, non_Lecturername, Ac_Level, non_Level, Ac_Credit, non_Credit, Ac_Startdate, non_Startdate, Ac_Completiondate, non_Completiondate, Ac_Numberofassessments, non_Numberofassessments, Ac_Instructorname, non_Instructorname, Ac_Examdate, non_Examdate, Ac_Prerequisites, non_Prerequisites. -JButton: Ac_Register, non_Register, Ac_Display, non_Display, Ac_Add, non_Add, Ac_Clear, non_Clear, non_Remove.

+ INGCollege()
+ actionPerformed(ActionEvent e)
+ static void main(String[] args)

Table 1: Class Diagram of INGCollege

3 Pseudocode

Pseudo means imitation and code, which implies the code related with the order written in a programming language or script code. On the off chance that interpreted unreservedly, pseudocode implies impersonation or impersonation of the programming language code.

.Essentially, pseudocode is a language that permits programmers to think an issue that should be addressed quickly without contemplating the code or grammar of a specific programming language. In pseudocode there are no principles recorded as a writing the syntax. So pseudocode is utilized to depict the rationale in succession from a program without first contemplating the programming language.

The point is a description of a computer programming algorithm that uses the structural conventions of a programming language, and has the goal of being easily read by humans not by machines. Pseudocode usually does not use elements – details that are not necessary for the needs of human understanding of an algorithm, such as the declaration of variables, code or sub-routine for systems that have specific properties. (Sam, 2020).

- ❖ INGCollege

```
CREATE class INGCollege

        Declare frame as Jframe

        Declare panel, panel2 as JPanel

        Declare  title, n_title, CourseID, n_CourseID, Coursename,
        n_Coursename,Courseleader, n_Courseleader, Duration, n_Duration,
        Lecturername, n_Lecturername, Level, n_Level, Credit, n_Credit, Startdate,
        n_Startdate, Completiondate, n_Completiondate, Numberofassessments,
        n_Numberofassessments, Instructorname, n_Instructorname, Examdate,
        n_Examdate, Prerequisites,   n_Prerequisites as JLabel.
```

```
Declare  Ac_CourseID, non_CourseID, Ac_Coursename, non_Coursename, Ac_Courseleader, non_Courseleader, Ac_Duration, non_Duration, Ac_Lecturername, non_Lecturername, Ac_Level, non_Level, Ac_Credit, non_Credit, Ac_Startdate, non_Startdate, Ac_Completiondate,non_Completiondate, Ac_Numberofassessments, non_Numberofassessments, Ac_Instructorname, non_Instructorname, Ac_Examdate,non_Examdate, Ac_Prerequisites, non_Prerequisites as JTextfield.

Declare   Ac_Register, non_Register, Ac_Display, non_Display, Ac_Clear, non_Clear, Ac_Add, non_Add, non_Remove as JButton.

CREATE arraylist named courselist

CREATE constructor INGCollege()

DO

        Assign frame for Course Registration form

        Set int width, int height for frame size

        Set GridLayout

        Add panel in frame

        Add panel2 in frame


        Assign panel for Academic Course

        Set x coordinate, y coordinate, width and height for panel

        Set Backgroundcolor in panel
```

```
Assign panel2 for Non Academic Course
Set x coordinate, y coordinate, width and height for panel2
Set Backgroundcolor in panel2

Assign jLabel title as Academic Course
Set x coordinate, y coordinate, width and height for label title
Add title to panel

Assign jLabel n_title as Non Academic Course
Set x coordinate, y coordinate, width and height for label n_title
Add n_title to panel2

Assign jLabel CourseID as Academic Course
Set x coordinate, y coordinate, width and height for label CourseID
Add CourseID to panel

Assign jLabel n_CourseID as Non Academic Course
Set x coordinate, y coordinate, width and height for label n_CourseID
Add n_CourseID to panel2
```

```
Assign jLabel  Coursename as Academic Course
Set x coordinate, y coordinate, width and height for label Coursename
Add Coursename to panel

Assign jLabel n_Coursename as Non Academic Course
Set x coordinate, y coordinate, width and height for label n_Coursename
Add n_Coursename to panel2

Assign jLabel Courseleader as Academic Course
Set x coordinate, y coordinate, width and height for label Courseleader
Add Courseleader to panel

Assign jLabel n_Courseleader as Academic Course
Set x coordinate, y coordinate, width and height for label n_Courseleader
Add n_Courseleader to panel2

Assign jLabel Duration as Academic Course
Set x coordinate, y coordinate, width and height for label Duration
Add Duration to panel
```

```
Assign jLabel n_Duration as Non Academic Course
Set x coordinate, y coordinate, width and height for label n_Duration
Add n_Duration to panel

Assign jLabel Lecturername as Academic Course
Set x coordinate, y coordinate, width and height for label Lecturername
Add Lecturername to panel

Assign jLabel n_Lecturername as Non Academic Course
Set x coordinate, y coordinate, width and height for label n_Lecturername
Add n_Lecturername to panel

Assign jLabel Level as Academic Course
Set x coordinate, y coordinate, width and height for label Level
Add Level to panel

Assign jLabel n_Level as Non Academic Course
Set x coordinate, y coordinate, width and height for label n_Level
Add n_Level to panel2
```

```
Assign jLabel Credit as Academic Course
Set x coordinate, y coordinate, width and height for label Credit
Add Credit to panel2

Assign jLabel n_Credit as Non Academic Course
Set x coordinate, y coordinate, width and height for label n_Credit
Add n_Credit to panel2

Assign jLabel Startdate as Academic Course
Set x coordinate, y coordinate, width and height for label Startdate
Add Startdate to panel

Assign jLabel n_Startdate as Non Academic Course
Set x coordinate, y coordinate, width and height for label n_Startdate
Add n_Startdate to panel2

Assign jLabel Completiondate as Academic Course
Set x coordinate, y coordinate, width and height for label Completiondate
Add Completiondate to panel
```

```
Assign jLabel n_Completiondate as Non Academic Course

Set x coordinate, y coordinate, width and height for label n_Completiondate

Add n_Completiondate to panel2


Assign jLabel Numberofassessments as Academic Course

Set x coordinate, y coordinate, width and height for label Numberofassessments

Add Numberofassessments to panel


Assign jLabel n_Numberofassessments as Non Academic Course

Set x coordinate, y coordinate, width and height for label n_Numberofassessments

Add n_Numberofassessments to panel2


Assign jLabel Instructorname as Academic Course

Set x coordinate, y coordinate, width and height for label Instructorname

Add Instructorname to panel
```

```
Assign jLabel n_Instructorname as Non Academic Course

Set x coordinate, y coordinate, width and height for label n_Instructorname

Add n_Instructorname to panel2


Assign jLabel Examdate as Academic Course

Set x coordinate, y coordinate, width and height for label Examdate

Add Examdate to panel


Assign jLabel n_Examdate as Non Academic Course

Set x coordinate, y coordinate, width and height for label n_Examdate

Add n_Examdate to panel2


Assign jLabel Prerequisites as Academic Course

Set x coordinate, y coordinate, width and height for label Prerequisites

Add Prerequisites to panel


Assign jLabel n_Prerequisites as Non Academic Course

Set x coordinate, y coordinate, width and height for label n_Prerequisites

Add n_Prerequisites to panel2
```

```
Assign jTextField Ac_CourseID as Academic Course
Set x coordinate, y coordinate, width and height for textfield Ac_CourseID
Add Ac_CourseID to panel

Assign jTextField non_CourseID as Non Academic Course
Set x coordinate, y coordinate, width and height for textfield non_CourseID
Add non_CourseID to panel2

Assign jTextFiled Ac_Coursename as Academic Course
Set x coordinate, y coordinate, width and height for textfield Ac_Coursename
Add Ac_Coursename to panel

Assign jTextField non_Coursename as Non Academic Course
Set x coordinate, y coordinate, width and height for textfield non_Coursename
Add non_CourseID to panel2
```

```
Assign jTextField Ac_Courseleader as Academic Course
Set x coordinate, y coordinate, width and height for textfield Ac_Courseleader
Add Ac_Courseleader to panel

Assign jTextField non_Courseleader as Academic Course
Set x coordinate, y coordinate, width and height for textfield non_Courseleader
Add non_Courseleader to panel2

Assign jTextField Ac_Duration as Academic Course
Set x coordinate, y coordinate, width and height for textfield Ac_Duration
Add Ac_Duration to panel

Assign jTextField non_Duration as Non Academic Course
Set x coordinate, y coordinate, width and height for textfield non_Duration
Add Ac_Duration to panel2
```

```
Assign jTextField Ac_Lecturername as Academic Course
Set x coordinate, y coordinate, width and height for textfield Ac_Lecturername
Add Ac_Lecturername to panel

Assign jTextField non_Lecturername as Non Academic Course
Set x coordinate, y coordinate, width and height for textfield non_Lecturername
Add non_Lecturername to panel2

Assign jTextField Ac_Level as Academic Course
Set x coordinate, y coordinate, width and height for textfield Ac_Level
Add Ac_Level to panel

Assign jTextField non_Level as Non Academic Course
Set x coordinate, y coordinate, width and height for textfield non_Level
Add non_Level to panel2
```

```
Assign jTextField Ac_Credit as Academic Course

Set x coordinate, y coordinate, width and height for textfield Ac_Credit

Add Ac_Credit to panel


Assign jTextField non_Credit as Non Academic Course

Set x coordinate, y coordinate, width and height for textfield non_Credit

Add non_Credit to panel2


Assign jTextField Ac_Startdate as Academic Course

Set x coordinate, y coordinate, width and height for textfield Ac_Startdate

Add Ac_Startdate to panel


Assign jTextField non_Startdate as Non Academic Course

Set x coordinate, y coordinate, width and height for textfield non_Startdate

Add non_Startdate to panel2
```

```
Assign jTextField Ac_Completiondate as Academic Course

Set x coordinate, y coordinate, width and height for textfield Ac_Completiondate

Add Ac_Completiondate to panel


Assign jTextField non_Completiondate as Non Academic Course

Set x coordinate, y coordinate, width and height for textfield non_Completiondate

Add non_Completiondate to panel2


Assign jTextField Ac_Numberofassessments as Academic Course

Set x coordinate, y coordinate, width and height for textfield Ac_Numberofassessments

Add Ac_Numberofassessments to panel


Assign jTextFiled non_Numberofassessments as Non Academic Course

Set x coordinate, y coordinate, width and height for textfield non_Numberofassessments

Add non_Numberofassessments to panel2
```

```
Assign jTextField Ac_Instructorname as Academic Course
Set x coordinate, y coordinate, width and height for textfield Ac_Instructorname
Add Ac_Instructorname to panel

Assign jTextField non_Instructorname as Non Academic Course
Set x coordinate, y coordinate, width and height for textfield non_Instructorname
Add non_Instructorname to panel2

Assign jTextFiled Ac_Examdate as Academic Course
Set x coordinate, y coordinate, width and height for textfield Ac_Examdate
Add Ac_Examdate to panel

Assign jTextField non_Examdate as Non Academic Course
Set x coordinate, y coordinate, width and height for textfield non_Examdate
Add non_Examdate to panel2
```

```
Assign jTextField Ac_Prerequisites as Academic Course
Set x coordinate, y coordinate, width and height for textfield Ac_Prerequisites
Add Ac_Prerequisites to panel

Assign jTextField non_Prerequisites as Non Academic Course
Set x coordinate, y coordinate, width and height for textfield non_Prerequisites
Add non_Prerequisites to panel2

Assign JButton Ac_Register as Academic Course
Set x coordinate, y coordinate, width and height for button Ac_Register
Add Ac_Register to panel

Assign JButton non_Register as Non Academic Course
Set x coordinate, y coordinate, width and height for button non_Register
Add non_Register to panel2
```

```
Assign JButton Ac_Display as Academic Course

Set x coordinate, y coordinate, width and height for button Ac_Display

Add Ac_Display to panel


Assign JButton non_Display as Non Academic Course

Set x coordinate, y coordinate, width and height for button non_Display

Add non_Display to panel2


Assign JButton Ac_Clear as Academic Course

Set x coordinate, y coordinate, width and height for button Ac_Clear

Add Ac_Clear to panel


Assign JButton non_Clear as Non Academic Course

Set x coordinate, y coordinate, width and height for button non_Clear

Add non_Clear to panel2
```

```
Assign JButton Ac_Add as Academic Course
Set x coordinate, y coordinate, width and height for button Ac_Add
Add Ac_Add to panel

Assign JButton non_Add as Non Academic Course
Set x coordinate, y coordinate, width and height for button non_Add
Add non_Add to panel2

Assign JButton non_Remove as Non Academic Course
Set x coordinate, y coordinate, width and height for button non_Remove
Add non_Remove to panel2

Add actionListener for Ac_Register
Add actionListener for non_Register
Add actionListener for Ac_Clear
Add actionListener for non_Clear
Add actionListerner for Ac_Add
Add actionListener for non_Add
```

```
        **Add** actionListener for non_Remove
        **Set** DefaultCloseOperation for frame
**END DO**
**CREATE** method actionPerformed(ActionEvent e)
**DO**
        **Initialize** count = 0
        **Initialize** count1 = 0
        **IF**(e.getSource () == Ac_Add)
        **ENDIF**
        **IF** textfileds is empty
        **ENDIF**
        Display(Please kindly fill the form.)
**ELSE**
        **Try**
        **Convert** duration, credit  and numberofassessments to int datatype
        stores in String variable
        **Create** object of Academic Course AC
        **Pass** parameters Level of String data type, credit of int data type,
        CourseID of String data type, numberofassessments of int data type,
        Coursename of String data type, duration of int data type.
        **Add** the Academic Course object AC to arraylist courselist
        Display(Course added)

        Catch

        Check for the exception

        Display(Please enter a valid number)

ENDDO

CREATE button for non Academic Course

DO

        IF(e.getSource() == non_Add)

        ENDIF

        IF textfied is empty

        Display( Please kindly fill the form)

        ENDIF

ELSE

        Try

        Convert duration to int data type and stores in String variable

        Create object of Non Academic Course NAC

        Pass parameters n_CourseID of String data type, n_Coursename of
        String data type, duration of int data type , n_Prerequisites of String
        data type

        Add the Non Academic Course object NAC to arraylist courselist

        Display(Course added)
        Catch
        Check for exception
        Display( Please enter a valid number)
ENDDO
CREATE Register button for Academic Course
DO
        IF(e.getSource() == Ac_Register)
        ENDIF
        IF textfield is empty
        Display( Please kindly fill the form)
        ENDIF
ELSE
        For(i = 0; i < courselist.size();i++)
        IF((getcourseID() in courselist.get(i)) equal Ac_CourseID.getText())))
        ENDIF
        CREATE object of Academic Course academic_course and store date
        IF(academic_course.isRegistered == false)
        Register academic_course (Ac_Courseleader.getText(),
        Ac_Lecturername.getText(), Ac_Startdate..getText(),
        Ac_Completiondate.getText())
        Display( Course registered successfully)
        ENDIF
ELSE
        Display(Course Exists)
ELSE
        Display(Invalid Input)
ENDDO


CREATE Register button for Non Academic Course
DO
        IF(e.getSourse() == non_Register)
        IF textfiled is empty()
        Display( Please kindly fill the form)
        ENDIF
ELSE
        For( i = 0; i < courselist.size(); i++)
        If((getCourseID() in courselist.get(i)) equals non_CourseID.getText()))
        Create object of Non Academic Course non_academic_course and
        store date
        ENDIF
        IF(non_academic_course.isRegisterd == false)
        Register non_academic_course(non_Courseleader.getText(),
        non_Lecturername.getText(), non_Startdate.getText(),
        non_Completiondate.getText(), non_Examdate.getText())

        Display(Course registered successfully)

        ENDIF

ELSE

        Display(Course Exists)

ELSE

        Display(Invalid input)

ENDDO

CREATE display button for Academic Course

DO

        IF(e.getSource() == Ac_Display)

        ENDIF

        For(Course Z in arraylist courselist)

        IF(Z instanceof AcademicCourse)

        CREATE object of Academic Course AC and store data

        Display AC

        ENDIF

ENDDO
```

```
CREATE display button for Non Academic Course

DO

        IF(e.getSource() == Ac_Display)

        ENDIF

        For(Course Y : courselist)

        IF(Y instanceof AcademicCourse)

        CREATE object of Non Academic Course NAC and store data

        Display NAC

        ENDIF

ENDDO

CREATE clear button for Academic Course and Non Academic Course

DO

        IF(e.getSource() == Ac_Clear) || e.getSource() == non_Clear)

        Set as empty for Ac_CourseID textfield

        Set as empty for non_CourseID textfiled


        Set as empty for Ac_Coursename textfield

        Set as empty for non_Coursename textfiled
        Set as empty for Ac_Duration textfield

        Set as empty for non_Duration textfiled


        Set as empty for Ac_Instructorname textfield

        Set as empty for non_Instructorname textfiled


        Set as empty for Ac_Courseleader textfield

        Set as empty for non_Courseleader textfiled


        Set as empty for Ac_Level textfield

        Set as empty for non_Level textfiled


        Set as empty for Ac_Credit textfield

        Set as empty for non_Credit textfiled


        Set as empty for Ac_Startdate textfield

        Set as empty for non_Startdate textfiled


        Set as empty for Ac_Examdate textfield

        Set as empty for non_Examdate textfiled
```

```
                Set as empty for Ac_Completiondate textfield

                Set as empty for non_Completionddate textfiled



                Set as empty for Ac_Numberofassessments textfield

                Set as empty for non_Numberofassessments textfiled



                Set as empty for Ac_Prerequisites textfield

                Set as empty for non_Prerequisites textfiled



                Set as empty for Ac_Lecturername textfield

                Set as empty for non_Lecturername textfiled

ENDIF

ENDDO

CREATE remove button for Non Academic Course

DO

                IF(e.getSource() == non_Remove)

                ENDIF

                IF textfield is empty

                Display(Please kindly fill the form)

                ENDIF
```

```
ELSE
        Initialize count1 += 1
        For(Course x : courselist)
        IF(non_CourseID.getText().getText() equals(x.getCourseID()) && x
        instanceof NonAcademicCourse)
        Create object of NonAcademicCourse as NAC and store date
        Remove non_CourseID.getText() from courselist
        Remove NAC
        IF(count1 == 1)
        Display(The course has already been removed)
        ENDIF
ELSE
        Break
ENDDO
Do
Create main method()
        Create object c of constructor
ENDDO
```

4 Method Description

A method is a block of code which only runs when it is called. We can pass data, known as parameters, into a method. Methods are used to perform certain actions, and they are also known as functions. (Anon., n.d.)

In this program different methods has been used. We have four classes which have used different methods. Different methods of the parent class have been used in child class.

- INGCollege

The methods used in INGCollege class are given below:

I. actionPerformed(ActionEvent e) :

   The purpose of actionPerformed(actionEvent e) is to called just after the user performs an action. So basically, actionPerformed is a method to direct event handling and exception handling. At the point when the user clicks onscreen button, the button fires an action event. This outcome in the summon of the action listener actionPerformed(the only method in the Action Listener interface). The single argument to the method is an ActionEvent object that gives data about the event and its source.

II. main method( ):

   Java main method is the entry point from where the execution begins which syntax is consistently public static void main (String[] args). This is the main method for the entire class. Here, the constructor of the INGCollege is called to performed and lead the entire program.

5 Testing

Testing is the way toward executing a program or application to discover its blunder and to perform effectively so a product ought to be error-free.

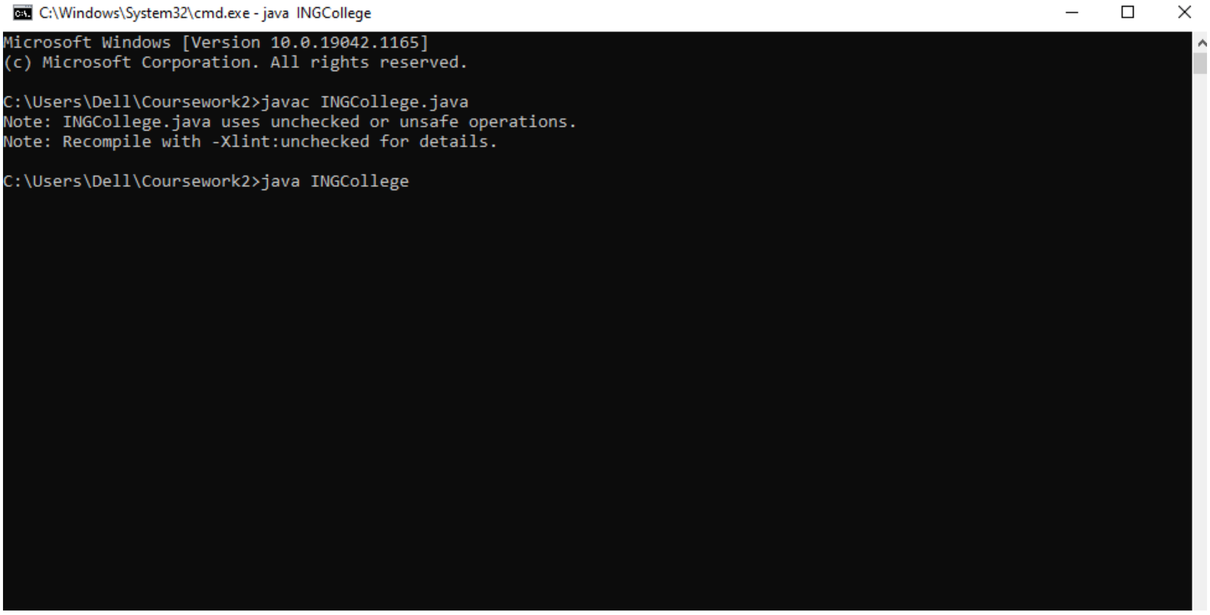
In simple word, testing is a strategy for discovering how well the program functions. In different terms, testing communicates what level of information or skill has been procured. All in all, testing is utilized at significant checkpoints in the general cycle to decide if objectives are being met.

6 Test 1: To test that the program compiled the java files and run the file using the command prompt

Test No:	1
Objective:	To Test that the program compiled the java files and run the file using the command prompt.
Action:	→The GUI is called with following arguements: 1. Go to the location where the file was save. 2. Erase the location and type cmd instead. 3. When the command prompt is open type javac classname.java. 4. Enter. 5. Again type java classname. 6. Enter
Expected result:	GUI should be opened.
Actual result:	GUI was open.
Conclusion:	The Test is successful.

Table 2: To test the program can be compiled and run using the command prompt

Output result:



```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dell\Coursework2>javac INGCollege.java
Note: INGCollege.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\Users\Dell\Coursework2>java INGCollege
```

Figure 2: Testing in cmd

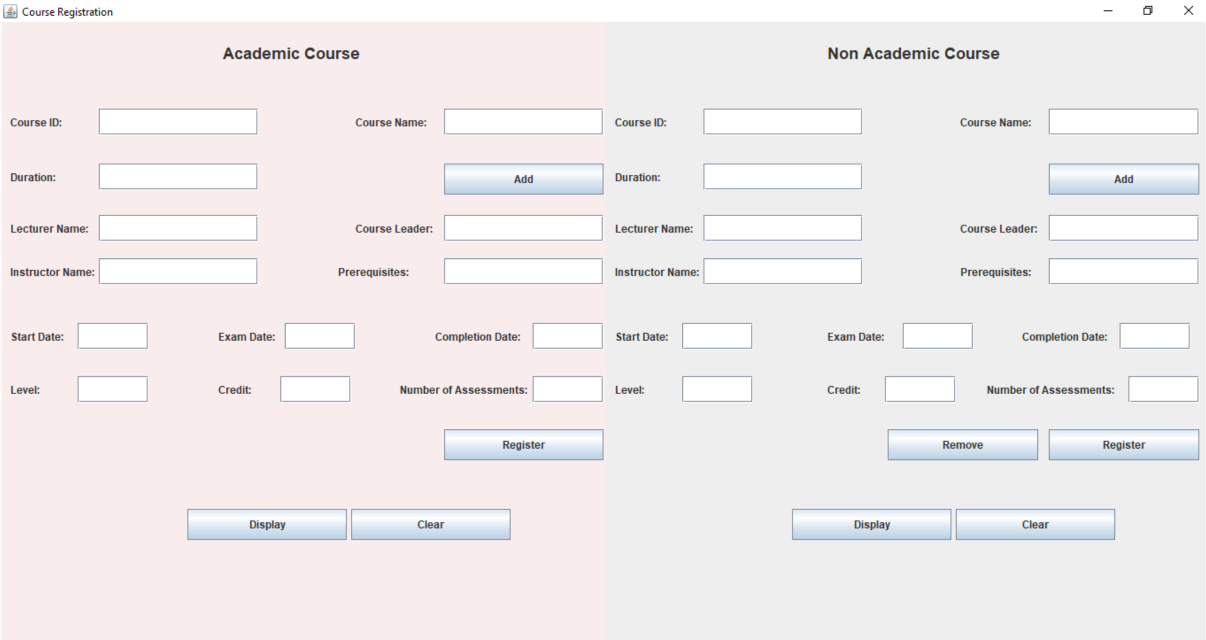


Figure 3: Screenshot of output after compiling and running through cmd

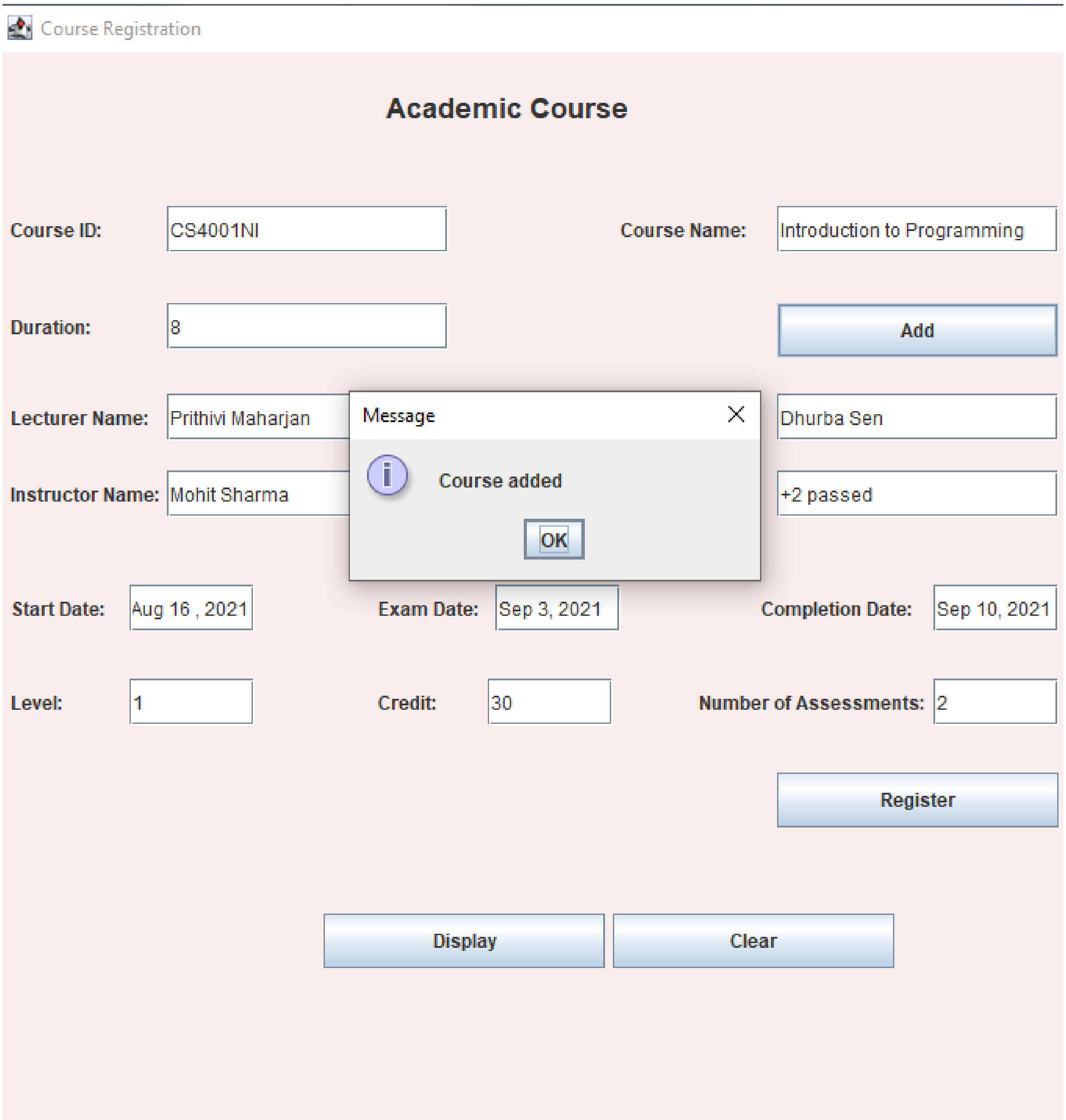
7 Test 2: Test for adding, registering and removing Academic Course and Non Academic Course class

a. Add course for Academic Course

Test No:	2 (a)
Objectives:	Add course for Academic Course.
Action:	→ The required text field of Academic Course class are filled and afterward click on add button.
Expected result:	When the add button is pressed, the course will be added and a message dialog box will appear.
Actual result:	When the add button was pressed, the course is added and a message dialog box is appeared.
Conclusion:	The test is successful.

Table 3 : To Test Add course for Academic course

Output result



Course Registration

Academic Course

Course ID: CS4001NI

Course Name: Introduction to Programming

Duration: 8

Lecturer Name: Prithivi Maharjan

Dhurba Sen

Instructor Name: Mohit Sharma

+2 passed

Message

Course added

Start Date: Aug 16 , 2021

Exam Date: Sep 3, 2021

Completion Date: Sep 10, 2021

Level: 1

Credit: 30

Number of Assessments: 2

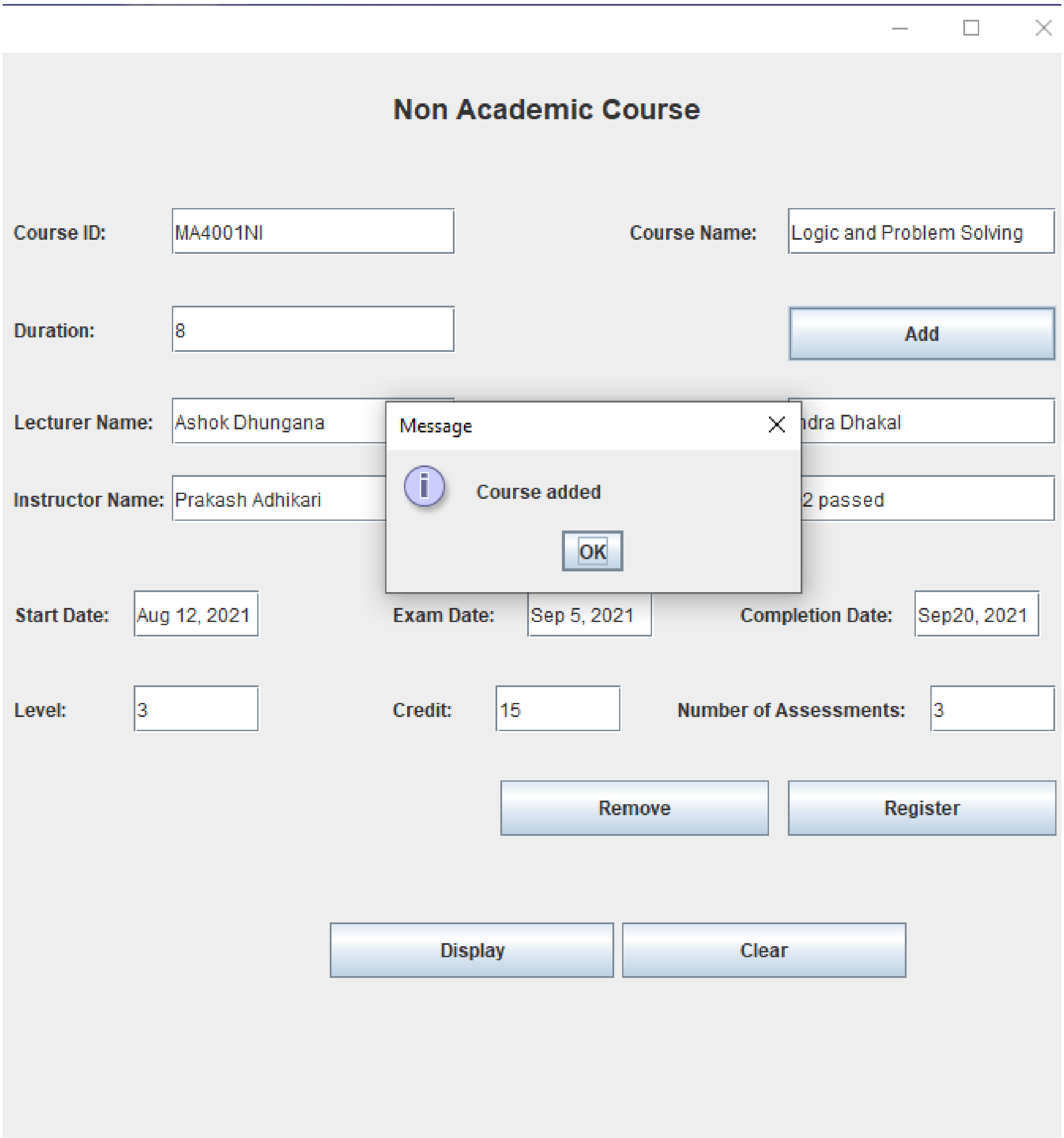
Figure 4: Screenshot of course added in Academic Course

b. Add course for Non Academic Course

Test no :	2 (b)
Objective:	Add course for Non Academic Course.
Action:	→ The required text field of Non Academic Course class are filled and afterward click on add button.
Expected result:	When the add button is pressed, the course will be added and a message dialog box will appear.
Actual result:	When the add button is pressed, the course is added and a message dialog box is appear.
Conclusion:	The text is successful.

Table 4: To Test Add course for Non-Academic Course

Output result



Non Academic Course

Course ID: MA4001NI

Course Name: Logic and Problem Solving

Duration: 8

Lecturer Name: Ashok Dhungana

ndra Dhakal

Instructor Name: Prakash Adhikari

2 passed

Message

Course added

Start Date: Aug 12, 2021

Exam Date: Sep 5, 2021

Completion Date: Sep20, 2021

Level: 3

Credit: 15

Number of Assessments: 3

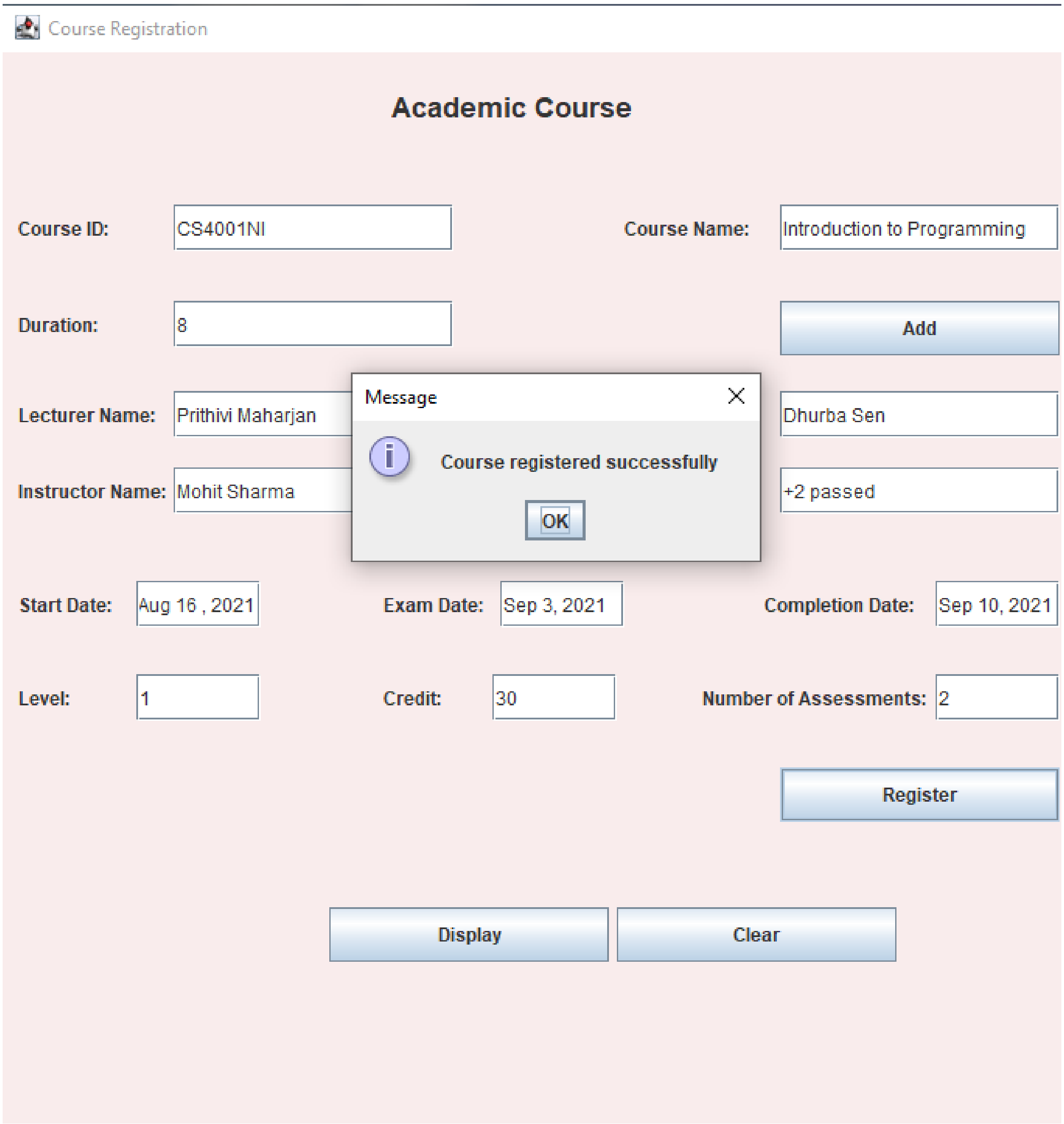
Figure 5: Screenshot of Course added in Non- Academic Course

c. Register Academic Course

Test no:	2 (c)
Objective:	Register Academic Course.
Action	→ The required text field of Academic Course class are filled and afterward click on register button.
Expected result:	When the register button is pressed, the course will be registered and dialog box will appear with an appropriate message.
Actual result:	When the register button is pressed, the course is registered and a dialog box appeared with an appropriate message.
Conclusion:	The test is successful.

Table 5 : To Test for registration of Academic Course

Output result



Course Registration

Academic Course

Course ID: CS4001NI

Course Name: Introduction to Programming

Duration: 8

Lecturer Name: Prithivi Maharjan

Dhurba Sen

Instructor Name: Mohit Sharma

+2 passed

Message

Course registered successfully

Start Date: Aug 16 , 2021

Exam Date: Sep 3, 2021

Completion Date: Sep 10, 2021

Level: 1

Credit: 30

Number of Assessments: 2

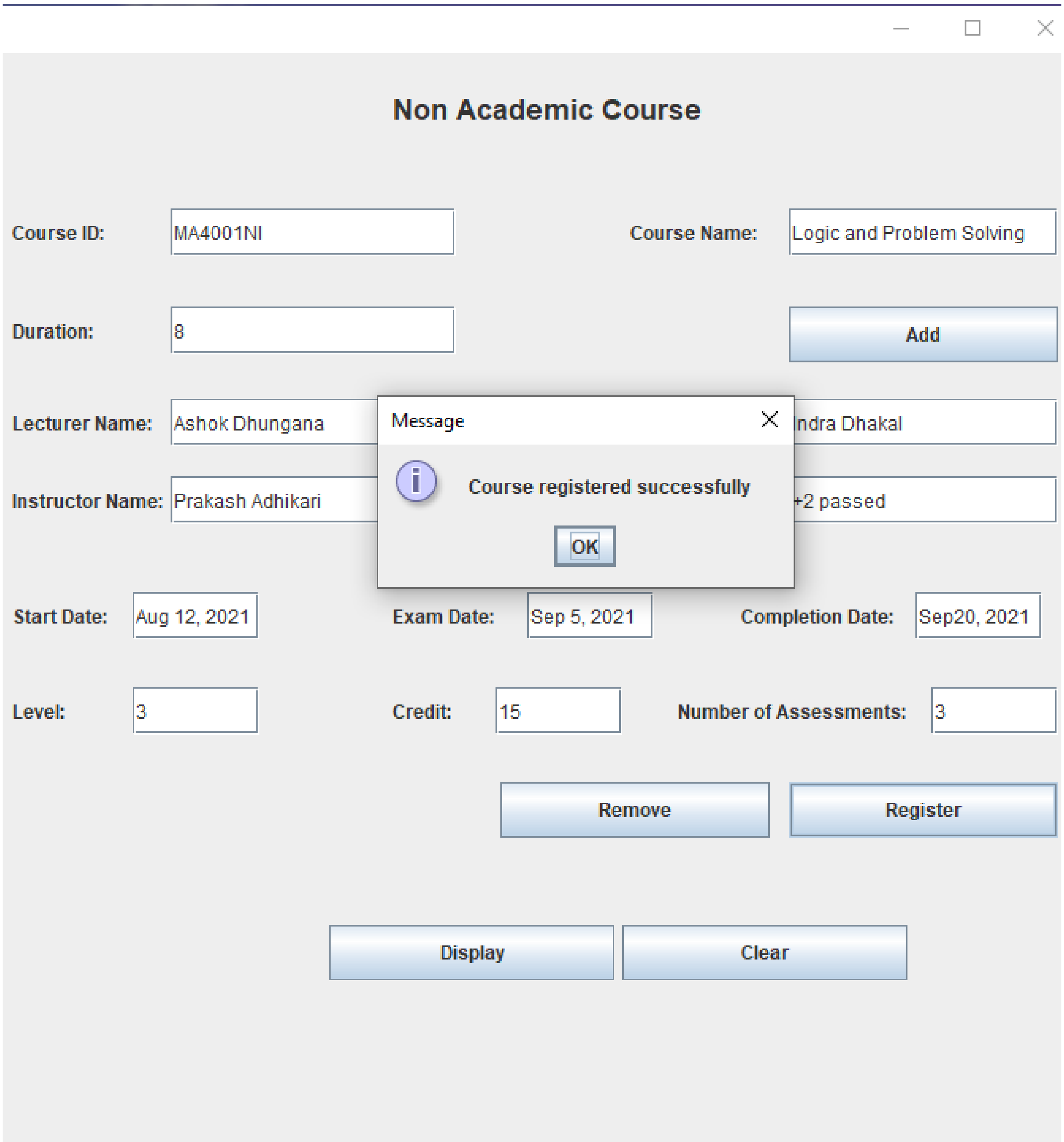
Figure 6 : Screenshot of course being registered successfully in Academic Course

d. Register Non Academic Course

Test no:	2 (c)
Objective:	Register Non Academic Course.
Action	→ The required text field of Non Academic Course class are filled and afterward click on register button.
Expected result:	When the register button is pressed, the course will be registered and dialog box will appear with an appropriate message course registered successfully.
Actual result:	When the register button is pressed, the course is registered and a dialog box appeared with an appropriate message course registerd successfully.
Conclusion:	The test is successful.

Table 6: ToTest for registration of Non Academic Course

Output result



Course ID: MA4001NI

Course Name: Logic and Problem Solving

Duration: 8

Lecturer Name: Ashok Dhungana

ndra Dhakal

Instructor Name: Prakash Adhikari

+2 passed

Start Date: Aug 12, 2021

Exam Date: Sep 5, 2021

Completion Date: Sep20, 2021

Level: 3

Credit: 15

Number of Assessments: 3

Course registered successfully

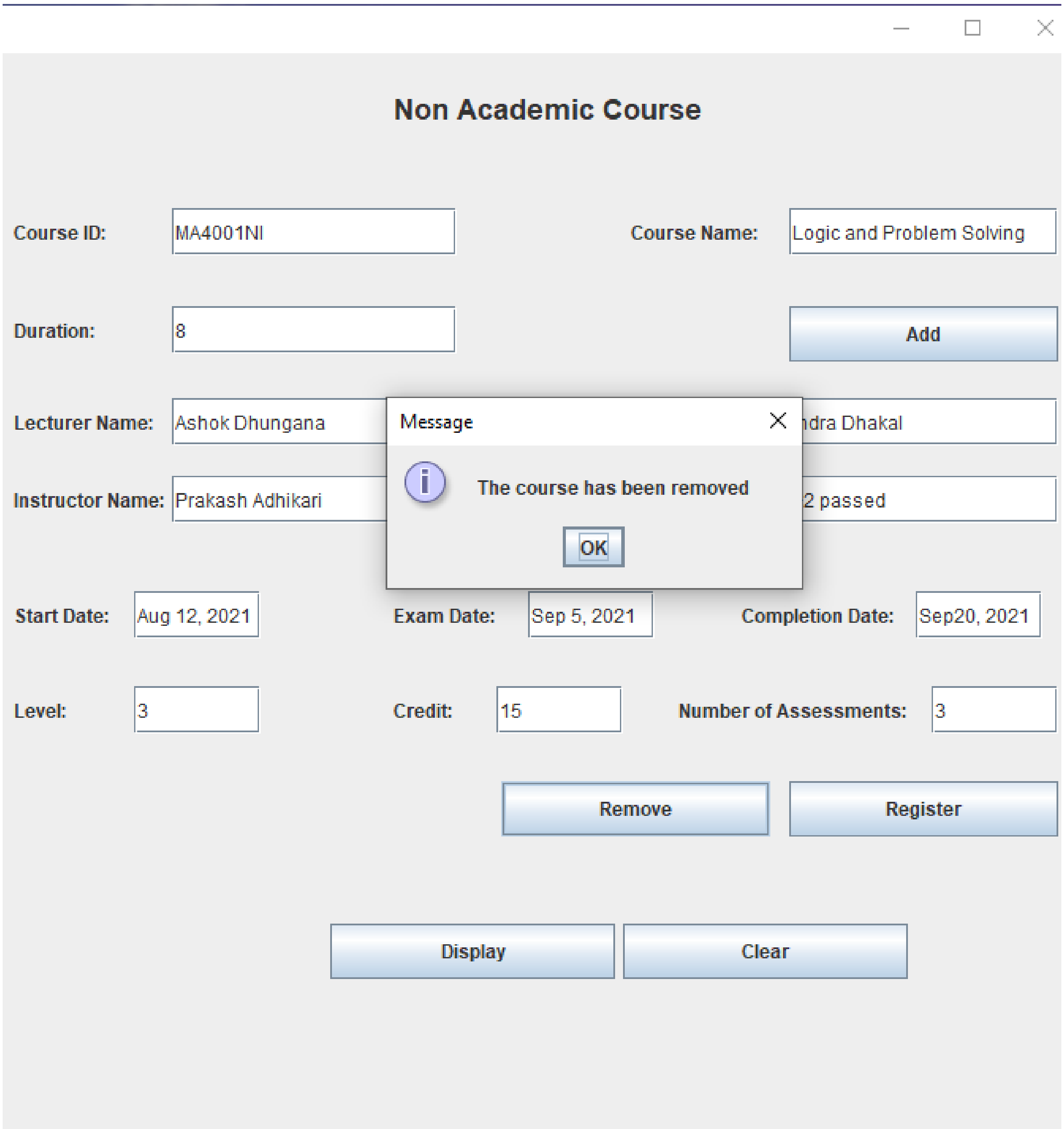
Figure 7: Screenshot of course being registered successfully in Non Academic Course

e. Remove Non Academic Course

Test no:	2 (e)
Objective:	Remove course from Non Academic Course.
Action:	→ The required text field of Non Academic Course class are filled and afterward click on Remove button.
Expected result:	When the remove button is pressed, the course will be removed and dialog box will appear with an appropriate message the course has been removed.
Actual result:	When the remove button is pressed, the course is removed and dialog box appear with an appropriate message the course has been removed.
Conclusion:	The test is successful.

Table 7: To Test for removing Non Academic Course

Output result



Non Academic Course

Course ID: MA4001NI

Course Name: Logic and Problem Solving

Duration: 8

Lecturer Name: Ashok Dhungana

dra Dhakal

Instructor Name: Prakash Adhikari

2 passed

Start Date: Aug 12, 2021

Exam Date: Sep 5, 2021

Completion Date: Sep20, 2021

Level: 3

Credit: 15

Number of Assessments: 3

Message

The course has been removed

Figure 8 : Screenshot of Non Academic Course where the course has been removed

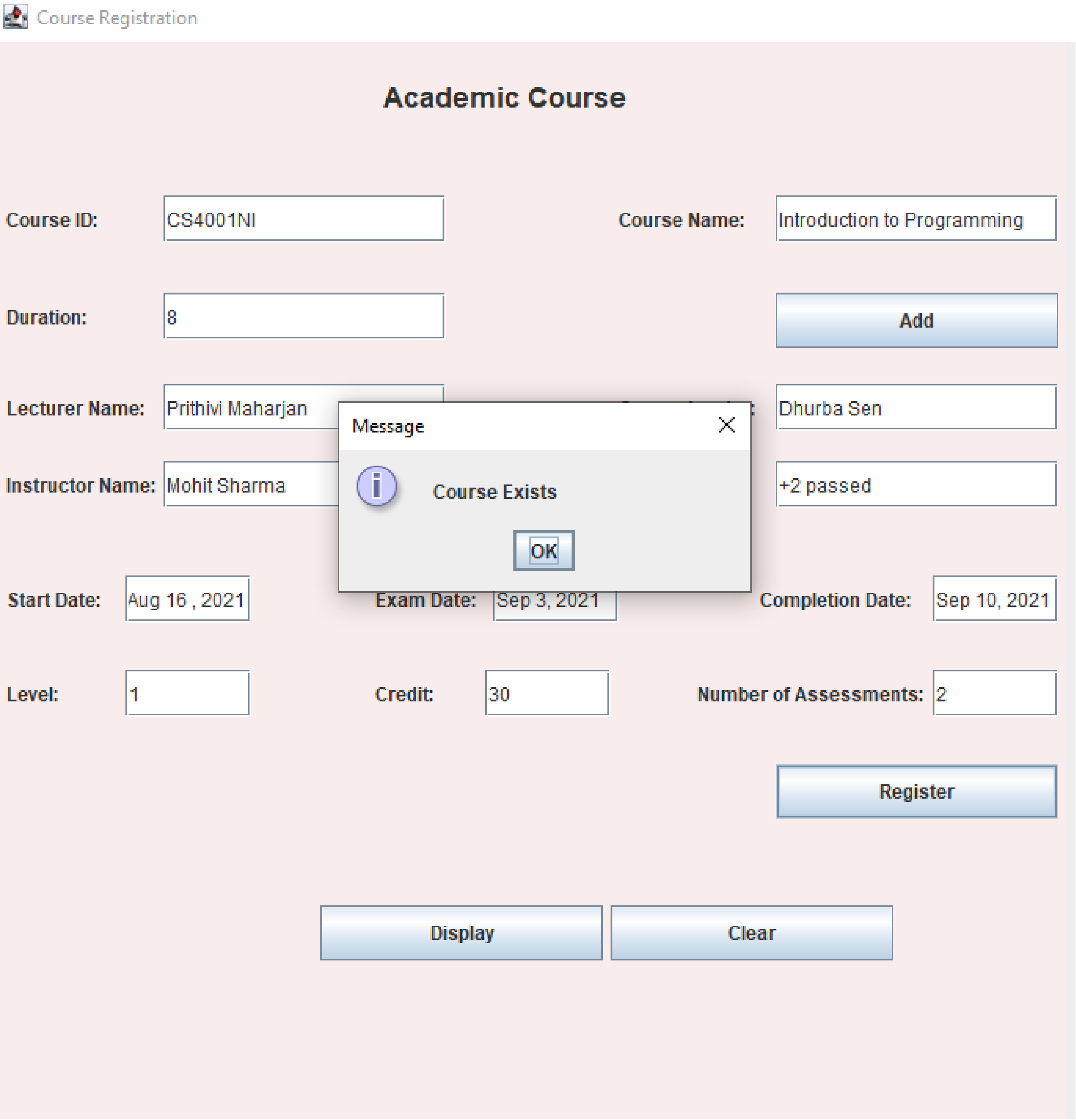
8 Test 3: Test that appropriate dialog boxes appear when:

8.1 Trying to register already registered course

Test no:	3 (b)
Objective:	Try to register already registered course.
Action:	→ The required text field of Academic Course and Non Academic Course class are filled and afterward click on registered button.
Expected result:	When the register button is pressed twice, a message saying course exits will appear.
Actual result:	When the remove button is pressed twice, a message saying course exits appears.
Conclusion:	The test is successful.

Table 8 : To Test for registering already registered course in Academic Course and Non Academic Course

Output Result



Course Registration

Academic Course

Course ID: CS4001NI

Course Name: Introduction to Programming

Duration: 8

Lecturer Name: Prithivi Maharjan

Dhurba Sen

Instructor Name: Mohit Sharma

+2 passed

Start Date: Aug 16 , 2021

Exam Date: Sep 3, 2021

Completion Date: Sep 10, 2021

Level: 1

Credit: 30

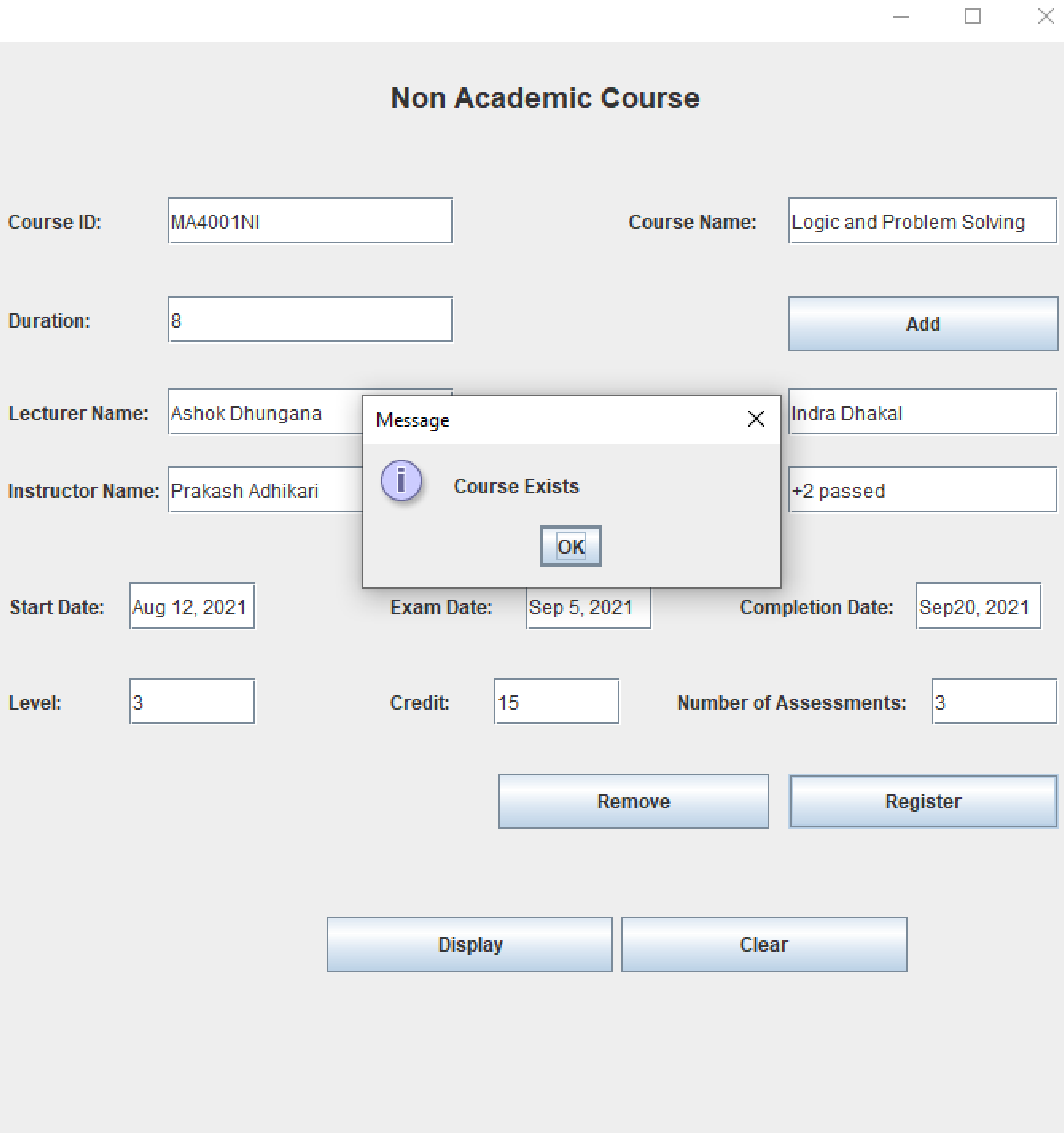
Number of Assessments: 2

Message

Course Exists

Figure 9: Screenshot of registering already registered course in Academic Course

Output



Non Academic Course

Course ID: MA4001NI

Course Name: Logic and Problem Solving

Duration: 8

Lecturer Name: Ashok Dhungana

Instructor Name: Prakash Adhikari

Indra Dhakal

+2 passed

Start Date: Aug 12, 2021

Exam Date: Sep 5, 2021

Completion Date: Sep20, 2021

Level: 3

Credit: 15

Number of Assessments: 3

Message

Course Exists

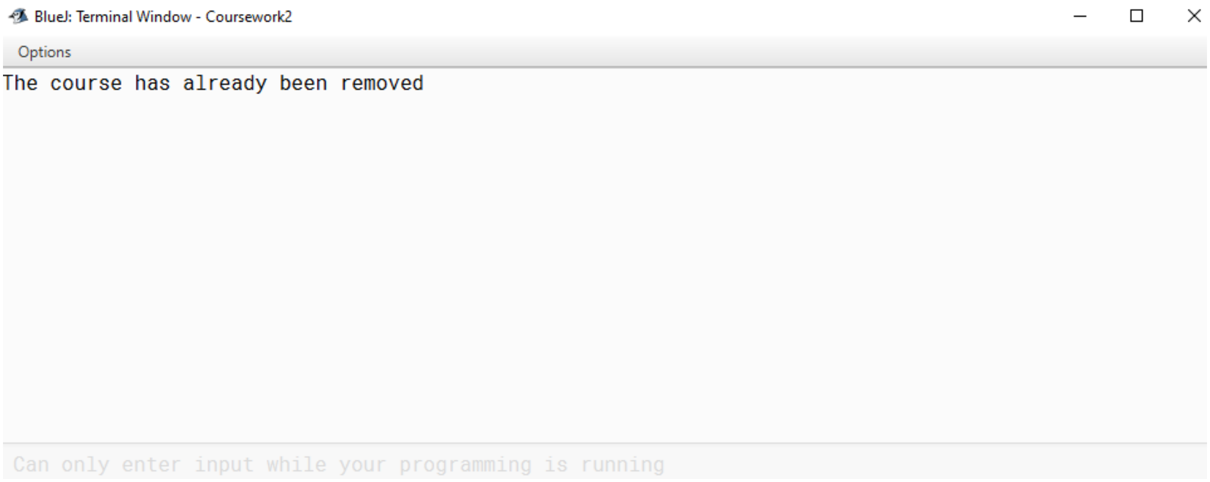
Figure 10 : Screenshot of registering already registered course in Non Academic Course

8.2 Trying to remove the non-academic course which is already removed.

Test no:	3(c)
Objectives:	Try to remove the non-academic course which is already removed.
Action:	→ The required text field of Non Academic Course class are filled and afterward click on remove button.
Expected result:	When the removed button is pressed twice, a message saying course has already been removed will display.
Actual result:	When the removed button is pressed twice, a message saying course has already been removed displays.
Conclusion:	The test is successful.

Table 9: To test for removing the Non Academic course which is already removed

Output Result:



BlueJ: Terminal Window - Coursework2
Options

```
The course has already been removed
```

Can only enter input while your programming is running

Figure 11: Screenshot of displaying the course has already been removed

9 Error Detection and Correction

The errors were seen during the program. All those errors were solved by observing those errors closely.

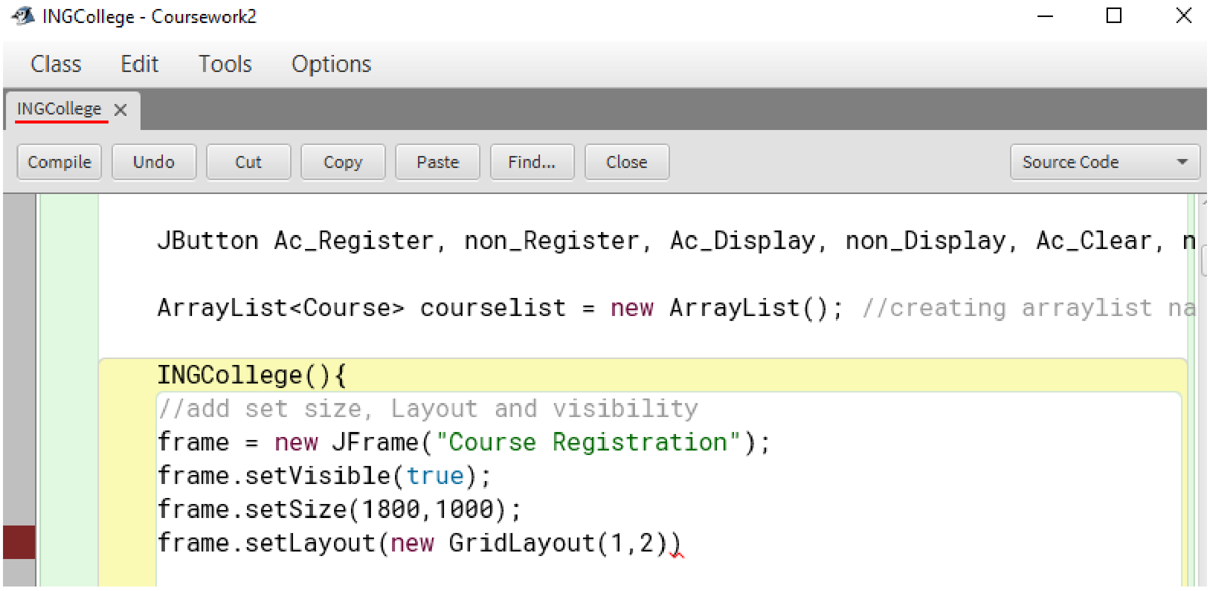
9.1 Syntax Error

Everything in a computer is designed in a concrete syntax form. If our input does not match that set of syntax, there are high chances of us facing a syntax error. By definition, we can say, that syntax error is a mistake in the input by the user therefore, the computer cannot answer the input question.

A syntax error is usually caused when the user does not add the correct punctuation or the correct code as it should be accurate in programming language. Even if one alphabet from the code is missing, you will be redirected to a syntax error.

It is very important for users to focus on the spelling, as well as the punctuation when adding a command. (Rehman, n.d.)

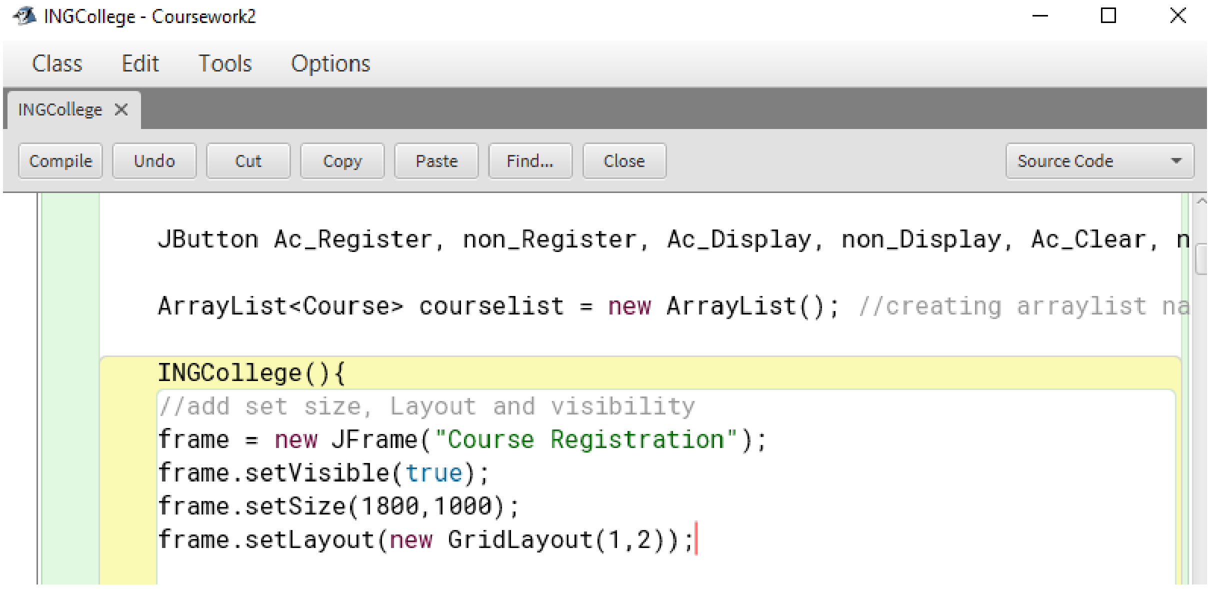
The errors found in a program are given below:



```
    JButton Ac_Register, non_Register, Ac_Display, non_Display, Ac_Clear, n
    ArrayList<Course> courselist = new ArrayList(); //creating arraylist na
    INGCollege(){
    //add set size, Layout and visibility
    frame = new JFrame("Course Registration");
    frame.setVisible(true);
    frame.setSize(1800,1000);
    frame.setLayout(new GridLayout(1,2))
```

Figure 12: Error due to missing semi-colon

The error was solved by closing the following by semi-colon.



```
    JButton Ac_Register, non_Register, Ac_Display, non_Display, Ac_Clear, n
    ArrayList<Course> courselist = new ArrayList(); //creating arraylist na
    INGCollege(){
    //add set size, Layout and visibility
    frame = new JFrame("Course Registration");
    frame.setVisible(true);
    frame.setSize(1800,1000);
    frame.setLayout(new GridLayout(1,2));
```

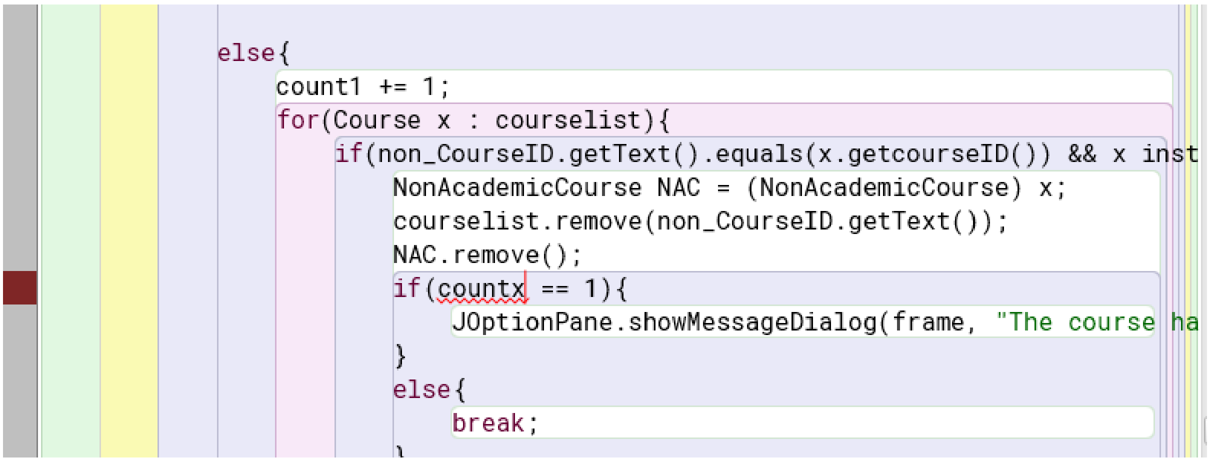
Figure 13: Error solved

9.2 Semantic Error

A semantic error is a violation of the principles of significance of a characteristics language or a programming language. When there are semantic blunders in a C++ program, the compiler makes and interpretation of the program into executable code. More often than not, semantic errors do not produce compiler warnings.

Most of the compile time error are scope and declaration error. For example: undeclared or multiple declared identifiers. Type mismatched is another compile time error. The semantic error can arises using the wrong variable or using wrong operator or doing operation in wrong order. (Anon., n.d.)

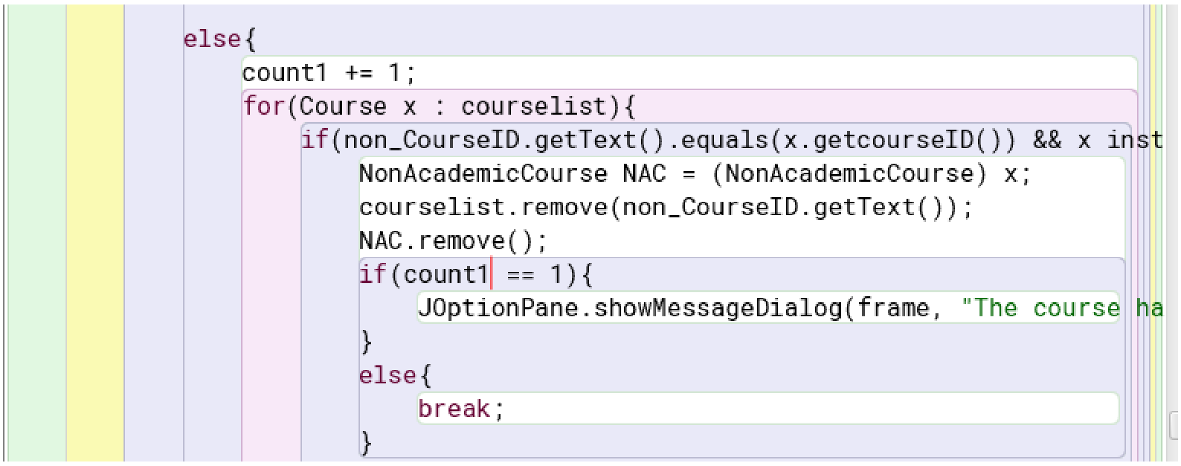
The errors found in a program is given below:



```
else{
    count1 += 1;
    for(Course x : courselist){
        if(non_CourseID.getText().equals(x.getcourseID()) && x inst
            NonAcademicCourse NAC = (NonAcademicCourse) x;
            courselist.remove(non_CourseID.getText());
            NAC.remove();
            if(countx == 1){
                JOptionPane.showMessageDialog(frame, "The course ha
            }
            else{
                break;
            }
```

Figure 14 : Error due undeclared variable x

The error was solved by giving declared variable 1.



```
else{
    count1 += 1;
    for(Course x : courselist){
        if(non_CourseID.getText().equals(x.getcourseID()) && x inst
            NonAcademicCourse NAC = (NonAcademicCourse) x;
            courselist.remove(non_CourseID.getText());
            NAC.remove();
            if(count1 == 1){
                JOptionPane.showMessageDialog(frame, "The course ha
            }
            else{
                break;
            }
```

Figure 15 : Error solved

9.3 Logical Error

The logical error is a program error made by the programmer while writing the program source code. In general, logical error executes unexpected results for logic value. Both interpreter and compiler and even scripting language has logical errors. Unlike syntax error and runtime error, the logical error cannot be detected by the compiler and interpreter.

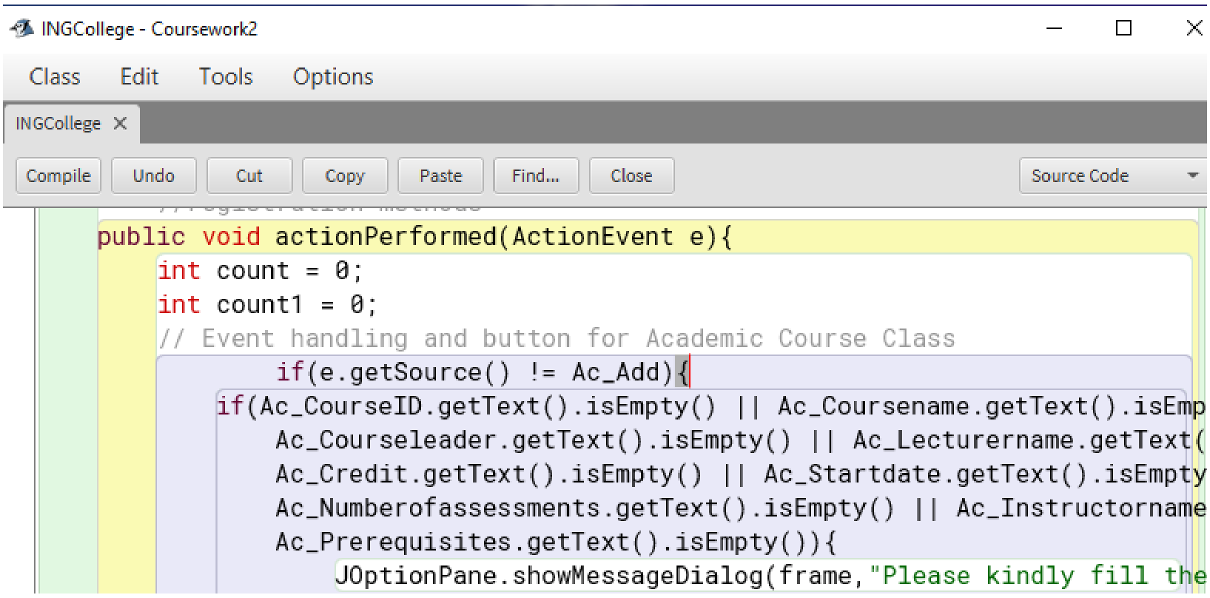
A logical error consistently does the following:

i) It permits the program to run (or possibly to begin).

ii) It leads surprising outcomes. The program may appear to work fine almost 100% of the time, yet during that other 1 % of the time, something odd happens.

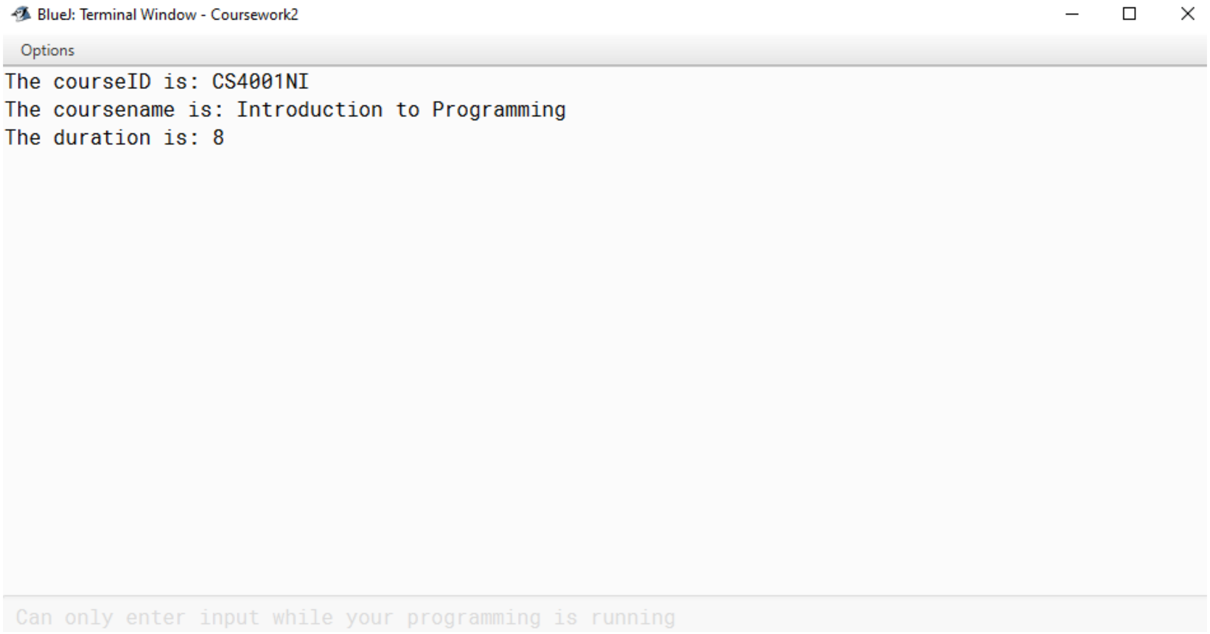
Basically, logical errors occur while writing the variable logic, for example, calculation something. A logical error does not give any error popup with an error code but it does not execute the desired output as intented. (Anon., n.d.)

The logical error seen in the program is given below:



```
public void actionPerformed(ActionEvent e){
    int count = 0;
    int count1 = 0;
    // Event handling and button for Academic Course Class
            if(e.getSource() != Ac_Add){
        if(Ac_CourseID.getText().isEmpty() || Ac_Coursename.getText().isEmp
            Ac_Courseleader.getText().isEmpty() || Ac_Lecturername.getText(
            Ac_Credit.getText().isEmpty() || Ac_Startdate.getText().isEmpty
            Ac_Numberofassessments.getText().isEmpty() || Ac_Instructorname
            Ac_Prerequisites.getText().isEmpty()){
                JOptionPane.showMessageDialog(frame,"Please kindly fill the
```

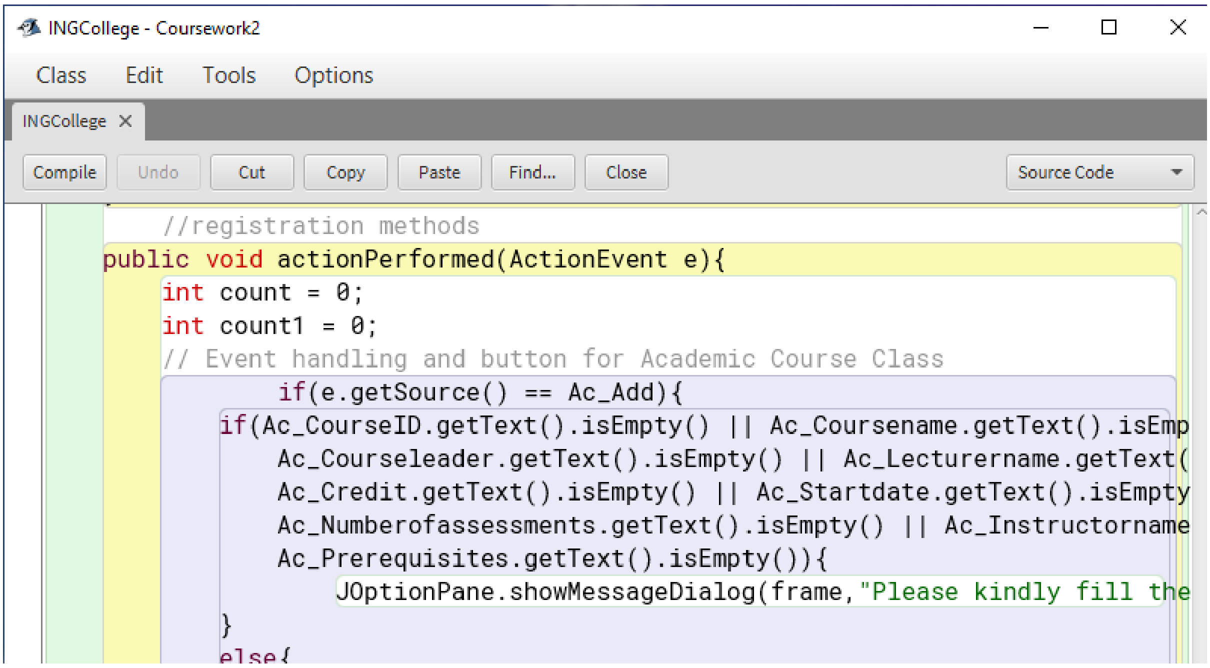
Figure 16: Logical error of the program


BlueJ: Terminal Window - Coursework2
Options
Can only enter input while your programming is running

```
The courseID is: CS4001NI
The coursename is: Introduction to Programming
The duration is: 8
```

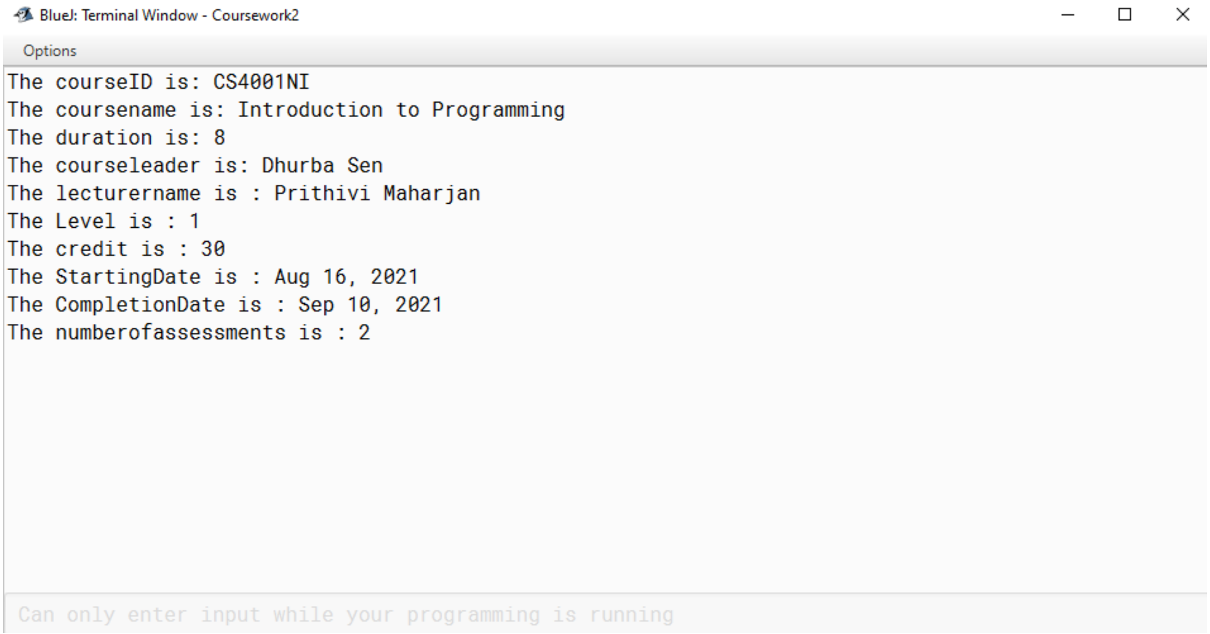
Figure 17: Displaying Academic course detail due to error

The error was seen during the displaying of the details. So, to remove the error we corrected the statement in the code which was suitable for the program.



```
    //registration methods
public void actionPerformed(ActionEvent e){
    int count = 0;
    int count1 = 0;
    // Event handling and button for Academic Course Class
            if(e.getSource() == Ac_Add){
        if(Ac_CourseID.getText().isEmpty() || Ac_Coursename.getText().isEmp
            Ac_Courseleader.getText().isEmpty() || Ac_Lecturername.getText(
            Ac_Credit.getText().isEmpty() || Ac_Startdate.getText().isEmpty
            Ac_Numberofassessments.getText().isEmpty() || Ac_Instructorname
            Ac_Prerequisites.getText().isEmpty()){
                JOptionPane.showMessageDialog(frame,"Please kindly fill the
        }
        else{
```

Figure 18: Error fix


BlueJ: Terminal Window - Coursework2
Options

```
The courseID is: CS4001NI
The coursename is: Introduction to Programming
The duration is: 8
The courseleader is: Dhurba Sen
The lecturername is : Prithivi Maharjan
The Level is : 1
The credit is : 30
The StartingDate is : Aug 16, 2021
The CompletionDate is : Sep 10, 2021
The numberofassessments is : 2
```

Can only enter input while your programming is running

Figure 19: Displaying Academic course detail with no error

10 Conclusion

To whole up, this coursework was relegated to us for making GUI for a framework which stores the details of Course including academic course and non academic course from our past coursework and to make a new project with another class called INGCollege. Four distinct classes were create in this protect exceptionally. The course class is the parent class and the academic, non academic and INGCollege classes are the sub classes. , The INGCollege class set the information about class of java swings, java.awt, java.awt.event, java.util.Arraylist and java.util.Iterator and so on. So this project helps me a lot to acquire information about the diverse data types, event handling and exception handling such as try and catch block.

While doing this coursework I needed to confront the challenged particularly in the coding segment as numerous mistakes occurred and its was kind of tough too I was exceptionally confounded. I was likewise somewhat confused in regards to the exception handling and pseudocode. Yet in order to conquer the confusion and troubles, a lot of explores were finished with respect to the applicable topics. Cent percent exertion was given to finish the undertaking allocated in this topic. Research was done in regards to java programming from the beginning and it was executed in this coursework which helped a lot to achieve the tasks. Regular interaction with instructors, consistent exertion and a lot of explores, going through the lecture slides and surfing in the web assisted with acquiring sound information about java and its purpose. I likewise, became more acquainted with about the various methods which are utilized in the program, I knew about various terms alongside their capabilities.

Despite the fact that it was challenging and troublesome from the outset, the coursework was finished on schedule and submitted on time as well. I had the chance to learn numerous new things and themes which I knew about. It was decent encounter to foster a program utilizing java and it was enjoyable to chip away at this project as well.

11 Appendix1 : codes of INGCollege

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

import java.util.ArrayList;

import java.util.Iterator;


public class INGCollege implements ActionListener{


    private JFrame frame;


    private JPanel panel, panel2;


    private JLabel title, n_title, CourseID, n_CourseID, Coursename,
n_Coursename,Courseleader, n_Courseleader, Duration, n_Duration,
Lecturername, n_Lecturername, Level, n_Level, Credit, n_Credit, Startdate,
n_Startdate, Completiondate, n_Completiondate, Numberofassessments,
n_Numberofassessments, Instructorname, n_Instructorname, Examdate,
n_Examdate, Prerequisites,  n_Prerequisites;


    private JTextField Ac_CourseID, non_CourseID, Ac_Coursename,
non_Coursename, Ac_Courseleader, non_Courseleader, Ac_Duration,
non_Duration, Ac_Lecturername, non_Lecturername, Ac_Level, non_Level,
Ac_Credit, non_Credit, Ac_Startdate, non_Startdate,
Ac_Completiondate,non_Completiondate, Ac_Numberofassessments,
non_Numberofassessments, Ac_Instructorname,  non_Instructorname,
Ac_Examdate,non_Examdate, Ac_Prerequisites, non_Prerequisites;
```

```
    private JButton Ac_Register, non_Register, Ac_Display, non_Display, Ac_Clear,
non_Clear, Ac_Add, non_Add, non_Remove;


    ArrayList<Course> courselist = new ArrayList();


        INGCollege(){
        frame = new JFrame("Course Registration");
        frame.setVisible(true);
        frame.setSize(1800,1000);
        frame.setLayout(new GridLayout(1,2));

        title = new JLabel("Academic Course");
        title.setBounds(250,00,250,70);
        title.setFont(new Font("NewTimesRoman",Font.BOLD,18));

        CourseID = new JLabel("Course ID:");
        Ac_CourseID= new JTextField();
        CourseID.setBounds(10,98,90,30);
        Ac_CourseID.setBounds(110,98,180,30);

        Coursename = new JLabel("Course Name:");
        Ac_Coursename= new JTextField();
        Coursename.setBounds(400,98,90,30);
        Ac_Coursename.setBounds(500,98,180,30);
```

```
Duration = new JLabel("Duration:");

Ac_Duration= new JTextField();

Duration.setBounds(10,160,90,30);

Ac_Duration.setBounds(110,160,180,30);


Lecturername = new JLabel("Lecturer Name:");

Ac_Lecturername= new JTextField();

Lecturername.setBounds(10,218,90,30);

Ac_Lecturername.setBounds(110,218,180,30);



Instructorname = new JLabel("Instructor Name:");

Ac_Instructorname = new JTextField();

Instructorname.setBounds(10,267,100,30);

Ac_Instructorname.setBounds(110,267,180,30);



Level = new JLabel("Level:");

Ac_Level= new JTextField();

Level.setBounds(10,400,50,30);

Ac_Level.setBounds(86,400,80,30);
```

```
Numberofassessments = new JLabel("Number of Assessments:");
Ac_Numberofassessments = new JTextField();
Numberofassessments.setBounds(450,400,200,30);
Ac_Numberofassessments.setBounds(600,400,80,30);

Startdate = new JLabel("Start Date:");
Ac_Startdate= new JTextField();
Startdate.setBounds(10,340,80,30);
Ac_Startdate.setBounds(86,340,80,30);

Examdate = new JLabel("Exam Date:");
Ac_Examdate= new JTextField();
Examdate.setBounds(245,340,80,30);
Ac_Examdate.setBounds(320,340,80,30);

Completiondate = new JLabel("Completion Date:");
Ac_Completiondate= new JTextField();
Completiondate.setBounds(490,340,180,30);
Ac_Completiondate.setBounds(600,340,80,30);

Prerequisites = new JLabel("Prerequisites:");
Ac_Prerequisites= new JTextField();
Prerequisites.setBounds(380,267,100,30);
Ac_Prerequisites.setBounds(500,267,180,30);
```

```
Courseleader = new JLabel("Course Leader:");

Ac_Courseleader= new JTextField();

Courseleader.setBounds(400,218,90,30);

Ac_Courseleader.setBounds(500,218,180,30);


Credit = new JLabel("Credit:");

Ac_Credit= new JTextField();

Credit.setBounds(245,400,100,30);

Ac_Credit.setBounds(315,400,80,30);



Ac_Register= new JButton("Register");

Ac_Register.setBounds(500,460,180,35);


Ac_Display= new JButton("Display");

Ac_Display.setBounds(210,550,180,35);


Ac_Clear= new JButton("Clear");

Ac_Clear.setBounds(395,550,180,35);


Ac_Add= new JButton("Add");

Ac_Add.setBounds(500,160,180,35);
```

```
n_title = new JLabel("Non Academic Course");

n_title.setBounds(250,00,250,70);

n_title.setFont(new Font("NewTimesRoman",Font.BOLD,18));


n_CourseID = new JLabel("Course ID:");

non_CourseID= new JTextField();

n_CourseID.setBounds(10,98,90,30);

non_CourseID.setBounds(110,98,180,30);


n_Coursename = new JLabel("Course Name:");

non_Coursename= new JTextField();

n_Coursename.setBounds(400,98,90,30);

non_Coursename.setBounds(500,98,170,30);


n_Duration = new JLabel("Duration:");

non_Duration= new JTextField();

n_Duration.setBounds(10,160,90,30);

non_Duration.setBounds(110,160,180,30);


n_Lecturername = new JLabel("Lecturer Name:");

non_Lecturername= new JTextField();

n_Lecturername.setBounds(10,218,90,30);

non_Lecturername.setBounds(110,218,180,30);
```

```
n_Instructorname = new JLabel("Instructor Name:");

non_Instructorname= new JTextField();

n_Instructorname.setBounds(10,267,100,30);

non_Instructorname.setBounds(110,267,180,30);



n_Level = new JLabel("Level:");

non_Level= new JTextField();

n_Level.setBounds(10,400,50,30);

non_Level.setBounds(86,400,80,30);


n_Numberofassessments= new JLabel("Number of Assessments:");

non_Numberofassessments= new JTextField();

n_Numberofassessments.setBounds(430,400,200,30);

non_Numberofassessments.setBounds(590,400,80,30);


n_Startdate = new JLabel("Start Date:");

non_Startdate= new JTextField();

n_Startdate.setBounds(10,340,80,30);

non_Startdate.setBounds(86,340,80,30);
```

```
n_Examdate = new JLabel("Exam Date:");

non_Examdate = new JTextField();

n_Examdate.setBounds(250,340,80,30);

non_Examdate.setBounds(335,340,80,30);


n_Completiondate = new JLabel("Completion Date:");

non_Completiondate= new JTextField();

n_Completiondate.setBounds(470,340,180,30);

non_Completiondate.setBounds(580,340,80,30);


n_Prerequisites = new JLabel("Prerequisites:");

non_Prerequisites= new JTextField();

n_Prerequisites.setBounds(400,267,100,30);

non_Prerequisites.setBounds(500,267,170,30);


n_Courseleader = new JLabel("Course Leader:");

non_Courseleader= new JTextField();

n_Courseleader.setBounds(400,218,90,30);

non_Courseleader.setBounds(500,218,170,30);


n_Credit = new JLabel("Credit:");

non_Credit= new JTextField();

n_Credit.setBounds(250,400,100,30);

non_Credit.setBounds(315,400,80,30);
```

```
non_Register= new JButton("Register");

non_Register.setBounds(500,460,170,35);


non_Display = new JButton("Display");

non_Display.setBounds(210,550,180,35);


non_Clear= new JButton("Clear");

non_Clear.setBounds(395,550,180,35);


non_Add= new JButton("Add");

non_Add.setBounds(500,160,170,35);


non_Remove= new JButton("Remove");

non_Remove.setBounds(318,460,170,35);



panel = new JPanel();

panel.setLayout(null);

panel.setBounds(0,0,850,1000);

Color C1 = new Color(229,255,249);

panel.setBackground(C1);
```

```
panel2 = new JPanel();

panel2.setLayout(null);

panel.setBounds(900,10,900,1000);

Color C2 = new Color(249,236,236);

panel.setBackground(C2);


frame.add(panel);

frame.add(panel2);

panel.add(title);

panel.add(CourseID);

panel.add(Ac_CourseID);

panel.add(Coursename);

panel.add(Ac_Coursename);

panel.add(Duration);

panel.add(Ac_Duration);

panel.add(Lecturername);

panel.add(Ac_Lecturername);

panel.add(Instructorname);

panel.add(Ac_Instructorname);

panel.add(Level);

panel.add(Ac_Level);

panel.add(Numberofassessments);

panel.add(Ac_Numberofassessments);

panel.add(Startdate);
panel.add(Ac_Startdate);

panel.add(Examdate);

panel.add(Ac_Examdate);

panel.add(Completiondate);

panel.add(Ac_Completiondate);

panel.add(Prerequisites);

panel.add(Ac_Prerequisites);

panel.add(Courseleader);

panel.add(Ac_Courseleader);

panel.add(Credit);

panel.add(Ac_Credit);

panel.add(Ac_Register);

panel.add(Ac_Display);

panel.add(Ac_Clear);

panel.add(Ac_Add);


panel2.add(n_title);

panel2.add(n_CourseID);

panel2.add(non_CourseID);

panel2.add(n_Coursename);

panel2.add(non_Coursename);

panel2.add(n_Duration);

panel2.add(non_Duration);

panel2.add(n_Lecturername);
```

```
panel2.add(non_Lecturername);

panel2.add(n_Instructorname);

panel2.add(non_Instructorname);

panel2.add(n_Level);

panel2.add(non_Level);

panel2.add(n_Numberofassessments);

panel2.add(non_Numberofassessments);

panel2.add(n_Startdate);

panel2.add(non_Startdate);

panel2.add(n_Examdate);

panel2.add(non_Examdate);

panel2.add(n_Completiondate);

panel2.add(non_Completiondate);

panel2.add(n_Prerequisites);

panel2.add(non_Prerequisites);

panel2.add(n_Courseleader);

panel2.add(non_Courseleader);

panel2.add(n_Credit);

panel2.add(non_Credit);

panel2.add(non_Register);

panel2.add(non_Display);

panel2.add(non_Clear);

panel2.add(non_Add);

panel2.add(non_Remove);
```

```
        Ac_Register.addActionListener(this);

        non_Register.addActionListener(this);


        Ac_Display.addActionListener(this);

        non_Display.addActionListener(this);


        Ac_Clear.addActionListener(this);

        non_Clear.addActionListener(this);


        Ac_Add.addActionListener(this);

        non_Add.addActionListener(this);


        non_Remove.addActionListener(this);




        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }


    public void actionPerformed(ActionEvent e){

        int count = 0;

        int count1 = 0;

                if(e.getSource() == Ac_Add){

            if(Ac_CourseID.getText().isEmpty() || Ac_Coursename.getText().isEmpty() ||
Ac_Duration.getText().isEmpty() ||
                Ac_Courseleader.getText().isEmpty() ||
Ac_Lecturername.getText().isEmpty() || Ac_Level.getText().isEmpty() ||
                Ac_Credit.getText().isEmpty() || Ac_Startdate.getText().isEmpty() ||
Ac_Completiondate.getText().isEmpty() ||
                Ac_Numberofassessments.getText().isEmpty() ||
Ac_Instructorname.getText().isEmpty() || Ac_Examdate.getText().isEmpty() ||
                Ac_Prerequisites.getText().isEmpty()){
                    JOptionPane.showMessageDialog(frame,"Please kindly fill the required
textfields: Course ID,Coursename, Duration, Courseleader,Lecturer Name, Level,
Credit, Startdate, Completiondate, Numberofassessments, Instructorname,
Examdate,Prerequisites");
            }
            else{
                try{
                    String CourseID = Ac_CourseID.getText();
                    String Coursename = Ac_Coursename.getText();
                    int duration = Integer.parseInt(Ac_Duration.getText());
                    String Level = Ac_Level.getText();
                    int credit = Integer.parseInt(Ac_Credit.getText());
                    int numberofassessments =
Integer.parseInt(Ac_Numberofassessments.getText());



                    AcademicCourse AC = new AcademicCourse(Level, credit, CourseID,
numberofassessments, Coursename, duration);
                        courselist.add(AC);
                    JOptionPane.showMessageDialog(null, "Course added");
            }
            catch(Exception ex){
               JOptionPane.showMessageDialog(null, "Please enter a valid number");
            }
         }
      }

      if(e.getSource() == non_Add){
         if(non_CourseID.getText().isEmpty() || non_Coursename.getText().isEmpty()
|| non_Duration.getText().isEmpty() ||
            non_Courseleader.getText().isEmpty() ||
non_Lecturername.getText().isEmpty() || non_Level.getText().isEmpty() ||
            non_Credit.getText().isEmpty() || non_Startdate.getText().isEmpty() ||
non_Completiondate.getText().isEmpty() ||
            non_Numberofassessments.getText().isEmpty() ||
non_Instructorname.getText().isEmpty() || non_Examdate.getText().isEmpty() ||
            non_Prerequisites.getText().isEmpty()){

            JOptionPane.showMessageDialog(frame,"Please kindly fill the required
textfields: Course ID,Coursename, Duration, Courseleader,Lecturer Name, Level,
Credit, Startdate, Completiondate, Numberofassessments, Instructorname,
Examdate,Prerequisites");
         }
         else{
            try{
            String n_CourseID = non_CourseID.getText();

            String n_Coursename = non_Coursename.getText();

            int duration = Integer.parseInt(non_Duration.getText());

            String n_Prerequisites = non_Prerequisites.getText();


            NonAcademicCourse NAC = new NonAcademicCourse(n_CourseID,
n_Coursename, duration , n_Prerequisites); //order of NonAcademicCourse class

               courselist.add(NAC);

            JOptionPane.showMessageDialog(null, "Course added");


            }

            catch(Exception ex){

            JOptionPane.showMessageDialog(null, "Please enter a valid number");

            }

         }

      }


       if(e.getSource() == Ac_Register){

         if(Ac_CourseID.getText().isEmpty() || Ac_Coursename.getText().isEmpty() ||
Ac_Duration.getText().isEmpty() ||

            Ac_Courseleader.getText().isEmpty() ||
Ac_Lecturername.getText().isEmpty() || Ac_Level.getText().isEmpty() ||

            Ac_Credit.getText().isEmpty() || Ac_Startdate.getText().isEmpty() ||
Ac_Completiondate.getText().isEmpty() ||

            Ac_Numberofassessments.getText().isEmpty() ||
Ac_Instructorname.getText().isEmpty() || Ac_Examdate.getText().isEmpty() ||
        Ac_Prerequisites.getText().isEmpty()){
          JOptionPane.showMessageDialog(frame,"Please kindly fill the required
textfields: Course ID,Coursename, Duration, Courseleader,Lecturer Name, Level,
Credit, Startdate, Completiondate, Numberofassessments, Instructorname,
Examdate,Prerequisites");
        }
      else{
        for(int i = 0;i< courselist.size();i++){
          if((courselist.get(i).getcourseID()).equals(Ac_CourseID.getText())){
            AcademicCourse academic_course =
(AcademicCourse)(courselist.get(i));
            if(academic_course.isRegistered == false){
              academic_course.register(Ac_Courseleader.getText(),
Ac_Lecturername.getText(), Ac_Startdate.getText(),
              Ac_Completiondate.getText());


              JOptionPane.showMessageDialog(null, "Course registered
successfully");


            }
            else{
              JOptionPane.showMessageDialog(null, " Course Exists");
            }
          }
          else{
            JOptionPane.showMessageDialog(null, "Invalid Input");
```

```
                    }
                }
            }
        }

        if(e.getSource() == non_Register){
            if(non_CourseID.getText().isEmpty() || non_Coursename.getText().isEmpty()
|| non_Duration.getText().isEmpty() ||
                non_Courseleader.getText().isEmpty() ||
non_Lecturername.getText().isEmpty() || non_Level.getText().isEmpty() ||
                non_Credit.getText().isEmpty() || non_Startdate.getText().isEmpty() ||
non_Completiondate.getText().isEmpty() ||
                non_Numberofassessments.getText().isEmpty() ||
non_Instructorname.getText().isEmpty() || non_Examdate.getText().isEmpty() ||
                non_Prerequisites.getText().isEmpty()){

                JOptionPane.showMessageDialog(frame,"Please kindly fill the required
textfields:Course ID,Coursename, Duration, Courseleader,Lecturer Name, Level,
Credit, Startdate, Completiondate, Numberofassessments, Instructorname,
Examdate,Prerequisites");
            }
            else{
                for(int i = 0; i<courselist.size();i++){
                    if((courselist.get(i).getcourseID()).equals(non_CourseID.getText())){
                        NonAcademicCourse non_academic_course =
(NonAcademicCourse)(courselist.get(i));
                        if(non_academic_course.isRegistered == false){
                            non_academic_course.register(non_Courseleader.getText(),
non_Lecturername.getText(), non_Startdate.getText(),
                            non_Completiondate.getText(), non_Examdate.getText());


                            JOptionPane.showMessageDialog(null, "Course registered
successfully");
                        }
                        else{
                            JOptionPane.showMessageDialog(null, "Course Exists");
                        }
                    }
                    else{
                        JOptionPane.showMessageDialog(null, "Invalid input");
                    }
                }
            }
        }


        if(e.getSource() == Ac_Display){
            for(Course Z : courselist){
                if (Z instanceof AcademicCourse){
                    AcademicCourse AC = (AcademicCourse)Z;
                    AC.display();
                }
            }
```

```
}


if(e.getSource() == non_Display){
    for(Course Y : courselist){
        if(Y instanceof  NonAcademicCourse){
            NonAcademicCourse NAC = (NonAcademicCourse)Y;
            NAC.display();
        }
    }
}


if(e.getSource() == Ac_Clear || e.getSource() == non_Clear){
    Ac_CourseID.setText("");
    non_CourseID.setText("");


    Ac_Coursename.setText("");
    non_Coursename.setText("");


    Ac_Duration.setText("");
    non_Duration.setText("");


    Ac_Instructorname.setText("");
    non_Instructorname.setText("");
```

```
Ac_Courseleader.setText("");

non_Courseleader.setText("");


Ac_Level.setText("");

non_Level.setText("");


Ac_Credit.setText("");

non_Credit.setText("");


Ac_Startdate.setText("");

non_Startdate.setText("");




Ac_Examdate.setText("");

non_Examdate.setText("");


Ac_Completiondate.setText("");

non_Completiondate.setText("");


Ac_Numberofassessments.setText("");

non_Numberofassessments.setText("");


Ac_Prerequisites.setText("");

non_Prerequisites.setText("");
```

```
            Ac_Lecturername.setText("");
            non_Lecturername.setText("");
        }

        if(e.getSource() == non_Remove){
            if(non_CourseID.getText().isEmpty() || non_Coursename.getText().isEmpty()
|| non_Duration.getText().isEmpty() ||
                non_Courseleader.getText().isEmpty() ||
non_Lecturername.getText().isEmpty() || non_Level.getText().isEmpty() ||
                non_Credit.getText().isEmpty() || non_Startdate.getText().isEmpty() ||
non_Completiondate.getText().isEmpty() ||
                non_Numberofassessments.getText().isEmpty() ||
non_Instructorname.getText().isEmpty() || non_Examdate.getText().isEmpty() ||
                non_Prerequisites.getText().isEmpty()){

                JOptionPane.showMessageDialog(frame,"Please kindly fill the required
textfields: Course ID,Coursename, Duration, Courseleader,Lecturer Name, Level,
Credit, Startdate, Completiondate, Numberofassessments, Instructorname,
Examdate,Prerequisites");
                }

            else{
                count1 += 1;
                for(Course x : courselist){
                    if(non_CourseID.getText().equals(x.getcourseID()) && x instanceof
NonAcademicCourse){
```

```
                        NonAcademicCourse NAC = (NonAcademicCourse) x;
                        courselist.remove(non_CourseID.getText());
                        NAC.remove();
                        if(count1 == 1){
                            JOptionPane.showMessageDialog(frame, "The course has already been removed");
                        }
                        else{
                            break;
                        }
                    }
                }
            }
        }
    }
    public static void main (String[] args){
        INGCollege c = new INGCollege();
    }
}
```

12 Appendix2: Codes of Course, Academic Course and Non Academic Course

Codes of Course Class

```
public class Course{

    String courseID,coursename, courseleader;

    int duration;

    Course(String courseID,String coursename,int duration){

        this.courseID=courseID;

        this.coursename=coursename;

        courseleader="";

        this.duration=duration;


        }

        public String getcourseID(){

             return courseID;

        }


        public String getcoursename(){

            return coursename;

        }

        public String getcourseleader(){

            return courseleader;

        }

        public int getduration(){
```

```
        return duration;

    }

    public void setcourseleader(String newname){

        this.courseleader=newname;

    }

    public void display(){

        System.out.println("The courseID is: "+courseID);

        System.out.println("The coursename is: "+coursename);

        System.out.println("The duration is: "+duration);

        if(courseleader!=""){

            System.out.println("The courseleader is: "+courseleader);

        }


    }


}
```

Codes of Academic Course Class

```
public class AcademicCourse extends Course{
    int numberofassessments;
    int credit;
    String lecturername, Level, StartingDate, CompletionDate;
    boolean isRegistered;

    public AcademicCourse(String Level,  int credit, String courseID, int numberofassessments ,String coursename,int Duration){
        super(courseID,coursename,Duration);
        this.Level=Level;
        this.credit=credit;
        this.numberofassessments=numberofassessments;
        lecturername="";
        StartingDate="";
        CompletionDate="";
        isRegistered=false;
        }
        //accessor/getter method
        public String getlecturername(){
            return lecturername;
        }
        public String getLevel(){
            return Level;
        }
```

```
public int getcredit(){
    return credit;
}
public String getStartingDate(){
    return StartingDate;
}
public String getCompletionDate(){
    return CompletionDate;
}
public int getnumberofassessments(){
    return numberofassessments;
}
public boolean getisRegistered(){
    return isRegistered;
}
public void setlecturername(String lecturername){
    this.lecturername=lecturername;
}
public void setnumberofassessments(int numberofassessments){
    this.numberofassessments=numberofassessments;
}
public void register(String courseleader,String lecturername,String StartingDate,String CompletionDate){
    if (isRegistered==true){
        System.out.println("The course is already registered");
```

```
            System.out.println("The courseleader is : "+courseleader);
            System.out.println("The StartingDate is : "+StartingDate);
            System.out.println("The CompletionDate is : "+CompletionDate);
        }else{
            super.setcourseleader(courseleader);
            this.lecturername=lecturername;
            this.StartingDate=StartingDate;
            this.CompletionDate=CompletionDate;
            this.isRegistered=true;
        }
    }
    public void display(){
        super.display();
        if (isRegistered==true){
            System.out.println("The lecturername is : "+lecturername);
            System.out.println("The Level is : "+Level);
            System.out.println("The credit is : "+credit);
            System.out.println("The StartingDate is : "+StartingDate);
            System.out.println("The CompletionDate is : "+CompletionDate);
            System.out.println("The numberofassessments is : 
"+numberofassessments);
        }
    }
}
```

Codes of Non Academic Course Class

```
public class NonAcademicCourse extends Course{
   String instructorname, startingdate,completiondate,examdate,prerequisite;
   boolean isRegistered,isRemoved;
   public NonAcademicCourse(String courseID, String coursename, int duration, 
String prerequisite){
      super(courseID,coursename,duration);
      this.prerequisite=prerequisite;
      startingdate="";
      completiondate="";
      examdate="";
      isRegistered=false;
      isRemoved=false;
   }
   public String getinstructorname(){
      return instructorname;
   }
   public String getstartingdate(){
      return startingdate;
   }
   public String getcompletiondate(){
      return completiondate;
   }
   public String getexamdate(){
      return examdate;
   }
   public String getprerequisite(){
      return prerequisite;
   }
   boolean getisRegistered(){
      return isRegistered;
   }
   boolean getisRemoved(){
      return isRemoved;
   }
   public void setinstructorname(String newinstructor){
      if(isRegistered==false){
          instructorname=newinstructor;
      }
      else{
          System.out.println("The instructorname cannot be changed");
      }
   }
   public void register(String courseleader,String instructorname,String
startingdate,String completiondate,String examdate){
      if(isRegistered==false){
          this.setinstructorname(instructorname);
          this.isRegistered=true;
          this.courseleader=courseleader;
          this.startingdate=startingdate;
          this.completiondate=completiondate;
          this.examdate=examdate;
      }else{
          System.out.println("The course is already registered");
      }
   }
   public void remove(){
      if(isRemoved==true){
          System.out.println("The course has already been removed");
      }else{
          super.setcourseleader("");
          this.instructorname="";
          this.startingdate="";
          this.completiondate="";
          this.examdate="";
          this.isRegistered=false;
          this.isRemoved=true;
          }
   }
   public void display(){
      super.display();
      if(isRegistered==true){
      System.out.println("instructor name is : " + instructorname);
      System.out.println("starting date is : " +startingdate);

      System.out.println("completiondate is : " +completiondate);

      System.out.println("examdate is : " +examdate);


      }

   }


}
```

13 Bibliography

Anon., n.d. *findanyanswer.* [Online]
Available at: https://findanyanswer.com/what-is-semantic-error-in-c
[Accessed 17 08 2021].

Anon., n.d. *progracoding.* [Online]
Available at: https://www.progracoding.com/logical-error/
[Accessed 17 08 2021].

Anon., n.d. *www.tutorialspoint.com.* [Online]
Available at: https://www.tutorialspoint.com/uml/uml_class_diagram.htm
[Accessed 10 08 2021].

Anon., n.d. *www.w3schools.com.* [Online]
Available at:
https://www.w3schools.com/java/java_methods.asp#:~:text=Java%20Methods.%20Java.%20Methods.%20A%20method%20is%20a,are%20also%20known%20as%20functions.%20Why%20use%20methods%3F
[Accessed 15 08 2021].

Rehman, H., n.d. *appuals.com.* [Online]
Available at: https://appuals.com/what-is-a-syntax-error/
[Accessed 17 08 2021].

Sam, A., 2020. *notesread.com.* [Online]
Available at: https://notesread.com/definition-pseudocode-functions-structure-examples/
[Accessed 11 08 2021].