

Course Registration System (Using Java)

Report By: Sayush Khadka

Table of Contents

1. Introduction:	1
2. Class Diagram:.....	3
3. Pseudocode	5
4. Method description	52
i. INGCollege.....	52
5. Test:	55
5.1 Test 1: To test that the program can be compiled and run using the command prompt.....	55
5.2 Test 2:	59
5.2.1 Test 2a: To add course for Academic Course.....	59
5.2.2 Test 2b: To add a Course for Non-Academic Course	61
5.2.3 Test 2c: To register Academic Course	63
5.2.4 Test 2d: To register Non-Academic Course	66
5.2.5 Test 2e: To remove Non-Academic Course	68
5.3 Test 3:	69
5.3.1 Test 3a:	69
i. To add duplicate CourseID (In Academic Course)	69
ii. To add duplicate CourseID (In Non-Academic Course).....	71
5.3.2 Test 3b:	74
i. To register already registered course (Academic Course)	74
ii. To register already registered course (Non-Academic Course).....	76
5.3.3 Test 3c	79
6. Error detection and correction	82
i. Syntax error.....	82
ii. Semantic Error	84
iii. Logical Error	86
7. Conclusion:	88
Bibliography	89
Appendix 1:	90
INGCollege.....	90

Appendix2:	145
Course, Academic Course, NonAcademicCourse	145
i. Course class.....	145
ii. AcademicCourse class	148
iii. NonAcademicCourse class	152

Table of tables:

Table 1 To test that the program can be compiled and run using the command prompt	56
Table 2 (Test 2a) To add a Course for Academic Course	59
Table 3 To add a Course for Non-Academic Course	61
Table 4 (Test 2c) To register Academic Course	63
Table 5 (Test 2d) To register Non-Academic Course	66
Table 6 (Test 2e) To remove Non-Academic Course	68
Table 7 (Test 3a (Academic Course)) To add duplicate CourseID in Academic Course	69
Table 8 (Test 3a (Non-Academic Course)) To add duplicate CourseID in Non-Academic Course.....	71
Table 9 (Test 3b (Academic Course)) To register already registered course (Academic Course)	74
Table 10 (Test 3b (Non-Academic Course)) To register already registered course (Non-Academic Course).....	76
Table 11 To remove the Non-Academic Course which is already removed.	79

Table of figures:

Figure 1 Class diagram for INGCollege class	3
Figure 2 Class diagram for INGCollege class (Continued).....	4
Figure 3 (Test 1) The command lines and the executed program	56
Figure 4 (Test 1) Command lines given to execute and run the program.....	57
Figure 5 (Test 1) Frame	58
Figure 6 (Test2a) The values are entered into the text field	60
Figure 7 (Test 2a) The add button for Academic Course is pressed	60
Figure 8 (Test2b) The values are entered into the text field	62
Figure 9 (Test 2b) The add button for Non-Academic Course is pressed	62
Figure 10 (Test 2c) The values are entered into the text field	64
Figure 11 (Test 2c) The register button for Academic Course is pressed	65
Figure 12 (Test 2d) The values are entered into the text field	67
Figure 13 (Test 2d) The register button for Non-Academic Course is pressed	67
Figure 14 (Test 2e) The remove button for Non-Academic Course is pressed (the course is removed and a message dialogue box appears.	68
Figure 15 (Test 3a (Academic Course)) The values are entered into the text field. The add button is pressed	70
Figure 16 (Test 3 a (Academic Course)) Different values are entered in the text field with the same CourseID. The add button is pressed.	70
Figure 17 (Test 3a (Non-Academic Course)) The values are entered into the text field. The add button is pressed.....	72
Figure 18 (Test 3 a (Non-Academic Course)) Different values are entered in the text field with the same CourseID. The add button is pressed.....	73
Figure 19 (Test 3b (Academic Course)) The values are entered in the text field with the same CourseID. (To register Academic Course).....	75
Figure 20 (Test 3b (Academic Course)) The register button is pressed. (To register already registered Academic Course)	76
Figure 21 (Test 3b (Non-Academic Course)) The values are entered in the text field with the same CourseID. (To register Non-Academic Course)	77

Figure 22 (Test 3b (Non-Academic Course)) The register button is pressed. (To register already registered Non-Academic Course)	78
Figure 23 (Test 3b (Non-Academic Course)) The values are entered in the text field with the same CourseID. (To remove the Non-Academic Course)	80
Figure 24 (Test 3b (Non-Academic Course)) The remove button is pressed. (To remove the Non-Academic Course which is already removed)	81
Figure 25 Syntax Error (Detection)	82
Figure 26 Syntax Error (Correction)	83
Figure 27 Semantic Error (Detection)	84
Figure 28 Semantic Error (Correction)	85
Figure 29 Logical Error (Detection)	86
Figure 30 Logical Error (Correction)	88

1. Introduction:

The project has been created in Java programming language using Blue J development environment.

(Java is a programming language which is intended to have as few implementation requirements as possible, is open source, and runs on all platforms. Java is object-oriented, and class-based that allows users to create programs). (Beal, 2021)

The project consists of 4 classes (3 from the previous project and 1 additional in the new project). They are:

- i) Course (The main class or the superclass)
- ii) AcademicCourse (The subclass that extends the main class or the superclass)
- iii) NonAcademicCourse (Another subclass that also extends the main class or the superclass)

New Class:

- iv) INGCollege (The GUI Holding class)

The project is a GUI (Graphical User Interface) based project. It's a graphical user interface with windows, icons, and buttons. The word was coined in the 1970s to distinguish between graphical and text-based interfaces, such as command line interfaces. Today, however, graphical user interfaces (GUIs) make up practically all digital interfaces. (Christensson, 2006)

The GUI based project is user friendly and is also based on real-world issue structure. Basically every digital interfaces is based on GUI at the current time. The program or the project has been developed with the help of Blue J. The project contains previous classes too of the previous project. Each class has its attributes also known as variables. Similarly, conditional statements have also been used for conditional checking. Each class also consists of their different methods and constructors which helps in creating the project.

In the current project Jawa Swing has been used to create the Course Registration window-based frame or application. The GUI of the INGCollege

project contains 3 different windows or frames. One being the main or the home frame and the other two being the Academic Course Registration and the Non-Academic Course Registration frames or windows respectively. Furthermore, the project has been well described, displayed and documented through class diagrams, description of different methods, pseudocode, running of tests, error detection, and correction below.

2. Class Diagram:



Figure 1 Class diagram for INGCollege class

```

- label_instructorNameNon : JLabel
- field_instructorNameNon : JTextField
- label_startingDateNon : JLabel
- combobox_yearsNon : JComboBox
- combobox_monthsNon : JComboBox
- combobox_daysNon : JComboBox
- label_completionDateNon : JLabel
- combobox_yearsNon1 : JComboBox
- combobox_monthsNon1 : JComboBox
- combobox_daysNon1 : JComboBox
- label_examDateNon : JLabel
- combobox_yearsNon2 : JComboBox
- combobox_monthsNon2 : JComboBox
- combobox_daysNon2 : JComboBox
- button_displayNon : JButton
- button_clearNon : JButton
- button_registerNon : JButton
- button_removeNon : JButton
+ obj_ing : INGCollege
+ obj_Academic : AcademicCourse

+ aList : ArrayList<Course>
- add_academicID : String
- add_academicName : String
- add_academicDuration : String
- add_academicLevel : String
- add_academicCredit : String
- add_numberOfAssessments : String
- register_academicID : String
- register_courseLeader : String
- register_lecturerName : String
- register_startingDate : String
- register_completionDate : String
- academicExist : boolean
- ref_academic : AcademicCourse
- message_academic : JOptionPane
- ref_academicDisplay : AcademicCourse
+ obj_NonAcademic : NonAcademicCourse

- add_nonAcademicCourseID : String
- add_nonAcademicCourseName : String
- add_nonAcademicDuration : String
- add_nonAcademicPrerequisite : String
- message_nonAcademic : JOptionPane
- register_nonAcademicID : String
- register_nonCourseLeader : String
- register_nonInstructorName : String
- register_nonStartingDate : String
- register_nonCompletionDate : String
- register_nonExamDate : String
- ref_nonAcademic : NonAcademicCourse
- nonAcademicExist : boolean
- remove_nonAcademicID : String
- ref_nonAcademicRemove : NonAcademicCourse
- ref_nonAcademicDisplay : NonAcademicCourse

+ mainGUI() : void
+ academicCourseGUI() : void
+ nonAcademicCourseGUI() : void
+ main(String[] args) : void
+ actionPerformed(ActionEvent e) : void

```

Figure 2 Class diagram for INGCollege class (Continued)

Note: The class diagrams have been made using “diagrams.net”. (Anon., n.d.)

3. Pseudocode

START

CREATE a class INGCollege which implements the interface ActionListener

DECLARE the instance attributes or the variables

CREATE a method mainGUI()

DO

INITIALIZE the JFrame to frame_main

SET the title of frame_main to Course Registration

SET the visibility of frame_main to true

SET resizable of frame_main to true

SET the bounds of frame_main to (520,130,517,445)

INITIALIZE the JPanel as panel_main

SET the layout of panel_main to null

SET the background of panel_main to (Color.white)

SET the bounds of panel_main to (0,0,503,430)

INITIALIZE the JPanel as panel_titleMain

SET the layout of panel_titleMain to null

SET the bounds of panel_titleMain to (0,0,503,70)

INITIALIZE the Color as color_titleMain
GIVE (253,223,91) to color_titleMain
SET the background of panel_titlemain to (color_titleMain)

INITIALIZE the JLabel to titleMain
SET the text of titleMain to Course Registration
SET the bounds of titleMain to (150,20,250,30)
SET the foreground to (Color.black)

SET the font as fontMain
GIVE ("Arial",Font.PLAIN,25) to fontMain
SET the font of titleMain to (fontMain)
ADD titleMain to panel_titleMain

INITIALIZE the JLabel to label_info
SET the text of label_info to Select on the button to register the course
SET the bounds of label_info to (60,100,450,25)

SET the font as font_info
GIVE ("Arial",Font.PLAIN,20) to font_info
SET the font of label_info to (font_info)
ADD label_info to panel_main

INITIALIZE the JButton to button_mainAcademic

SET the text of button_mainAcademic to Academic Course

SET the bounds of button_mainAcademic to (30,175,200,30)

SET the font as font_mainAcademic

GIVE ("Arial",Font.BOLD,15) to font_mainAcademic

SET the font of button_mainAcademic to (font_mainAcademic)

INITIALIZE the Color as color_mainAcademic

GIVE (42,136,173) to color_mainAcademic

SET the background of button_mainAcademic to (color_mainAcademic)

SET the foreground of button_mainAcademic to (Color.white)

ADD ActionListener to button_mainAcademic

ADD button_mainAcademic to panel_main

INITIALIZE the JButton to button_mainNonAcademic

SET the text of button_mainNonAcademic to Non Academic Course

SET the bounds of button_mainNonAcademic to (270,175,200,30)

SET the font as font_mainNonAcademic

GIVE ("Arial",Font.BOLD,15) to font_mainNonAcademic

SET the font of button_mainNonAcademic to (font_mainNonAcademic)

INITIALIZE the Color as color_mainNonAcademic
GIVE (42,136,173) to color_mainNonAcademic
SET the background of button_mainNonAcademic to
 (color_mainNonAcademic)
SET the foreground of button_mainNonAcademic to (Color.white)

ADD ActionListener to button_mainNonAcademic

ADD button_mainNonAcademic to panel_main

ADD panel_titleMain to frame_main
ADD panel_main to frame_main
END DO

CREATE a method academicCourseGUI()

DO

INITIALIZE the JFrame to frame_academic

SET the title of frame_academic to Academic Course Registration

SET the visibility of frame_academic to true

SET resizable of frame_academic to true

SET the bounds of frame_academic to (405,165,720,445)

INITIALIZE the JPanel as panel_academic

SET the layout of panel_academic to null

SET the background of panel_academic to (Color.white)

SET the bounds of panel_academic to (0,0,710,430)

INITIALIZE the JPanel as panel_titleAcademic

SET the layout of panel_titleAcademic to null

SET the bounds of panel_titleAcademic to (0,0,710,60)

INITIALIZE the Color as color_titleAcademic

GIVE (253,223,91) to color_titleAcademic

SET the background of panel_titleAcademic to (color_titleAcademic)

INITIALIZE the JLabel to title_academic

SET the text of title_academic to Academic Course

SET the bounds of title_academic to (275,15,250,30)

SET the foreground to (Color.black)

SET the font as font_academic

GIVE ("Arial",Font.PLAIN,20) to font

SET the font of title_academic to (font_academic)

ADD title_academic to panel_titleAcademic

INITIALIZE the JLabel to label_courseID

SET the text of label_courseID to CourseID

SET the bounds of label_courseID to (30,85,150,25)

SET the font as font_one

GIVE ("Arial",Font.PLAIN,15) to font_one

SET the font of label_courseID to (font_one)

ADD label_courseID to panel_academic

INITIALIZE the JTextField to field_courseID

SET the bounds of field_courseID to (150,85,120,25)

SET the font of field_courseID to (font_one)

ADD field_courseID to panel_academic

INITIALIZE the JLabel to label_courseName

SET the text of label_courseName to Course name

SET the bounds of label_courseName to (30,125,120,25)

SET the font as font_two

GIVE ("Arial",Font.PLAIN,15) to font_two

SET the font of label_courseName to (font_two)

ADD label_courseName to panel_academic

INITIALIZE the JTextField to field_courseName

SET the bounds of field_courseName to (150,125,120,25)

SET the font of field_courseName to (font_two)

ADD field_courseName to panel_academic

INITIALIZE the JLabel to label_duration

SET the text of label_duration to Duration

SET the bounds of label_duration to (30,165,120,25)

SET the font as font_three

GIVE ("Arial",Font.PLAIN,15) to font_three

SET the font of label_duration to (font_three)

ADD label_duration to panel_academic

INITIALIZE the JTextField to field_duration

SET the bounds of field_duration to (150,165,120,25)

SET the font of field_duration to (font_three)

ADD field_duration to panel_academic

INITIALIZE the JLabel to label_level

SET the text of label_level to Level

SET the bounds of label_level to (30,205,120,25)

SET the font as font_four

GIVE ("Arial",Font.PLAIN,15) to font_four

SET the font of label_level to (font_four)

ADD label_level to panel_academic

INITIALIZE the JTextField to field_level

SET the bounds of field_level to (150,205,120,25)

SET the font of field_level to (font_four)

ADD field_level to panel_academic

INITIALIZE the JLabel to label_credit

SET the text of label_credit to Credit

SET the bounds of label_credit to (30,245,120,25)

SET the font as font_five

GIVE ("Arial",Font.PLAIN,15) to font_five

SET the font of label_credit to (font_five)

ADD label_credit to panel_academic

INITIALIZE the JTextField to field_credit

SET the bounds of field_credit to (150,245,120,25)

SET the font of field_credit to (font_five)

ADD field_credit to panel_academic

INITIALIZE the JLabel to label_numberOfAssessments

SET the text of label_numberOfAssessments to Number of

SET the bounds of label_numberOfAssessments to (30,285,190,25)

SET the font as font_six

GIVE ("Arial",Font.PLAIN,15) to font_six

SET the font of label_numberOfAssessments to (font_six)

ADD label_numberOfAssessments to panel_academic

INITIALIZE the JLabel to label_numberOfAssessments1
SET the text of label_numberOfAssessments1 to assessments
SET the bounds of label_numberOfAssessments1 to (30,300,190,25)

SET the font as font_six
GIVE ("Arial",Font.PLAIN,15) to font_six
SET the font of label_numberOfAssessments1 to (font_six)
ADD label_numberOfAssessments1 to panel_academic

INITIALIZE the JTextField to field_numberOfAssessments
SET the bounds of field_numberOfAssessments to (150,290,120,25)
SET the font of field_numberOfAssessments to (font_six)
ADD field_numberOfAssessments to panel_academic

INITIALIZE the JButton to button_add
SET the text of button_add to Add
SET the bounds of button_add to (105,350,100,30)

SET the font as font_seven
GIVE ("Arial",Font.BOLD,15) to font_seven
SET the font of button_add to (font_seven)

INITIALIZE the Color as color_add

GIVE (42,136,173) to color_add

SET the background of button_add to (color_add)

SET the foreground of button_add to (Color.white)

ADD ActionListener to button_add

ADD button_add to panel_academic

INITIALIZE the JButton to button_changeCourse1

SET the text of button_changeCourse1 to Change Course

SET the bounds of button_changeCourse1 to (350,80,200,30)

SET the font as font_eight

GIVE ("Arial",Font.BOLD,15) to font_eight

SET the font of button_changeCourse1 to (font_eight)

INITIALIZE the Color as color_changeCourse1

GIVE (42,136,173) to color_changeCourse1

SET the background of button_changeCourse1 to (color_changeCourse1)

SET the foreground of button_changeCourse1 to (Color.white)

ADD ActionListener to button_changeCourse1

ADD button_changeCourse1 to panel_academic

INITIALIZE the JLabel to label_courseLeader
SET the text of label_courseLeader to Course Leader
SET the bounds of label_courseLeader to (350,125,150,25)

SET the font as font_nine
GIVE ("Arial",Font.PLAIN,15) to font_nine
SET the font of label_courseLeader to (font_nine)
ADD label_courseLeader to panel_academic

INITIALIZE the JTextField to field_courseLeader
SET the bounds of field_courseLeader to (475,125,120,25)
SET the font of field_courseLeader to (font_nine)
ADD field_courseLeader to panel_academic

INITIALIZE the JLabel to label_lecturerName
SET the text of label_lecturerName to Lecturer Name
SET the bounds of label_lecturerName to (350,165,150,25)

SET the font as font_ten
GIVE ("Arial",Font.PLAIN,15) to font_ten
SET the font of label_lecturerName to (font_ten)
ADD label_lecturerName to panel_academic

INITIALIZE the JTextField to field_lecturerName
SET the bounds of field_lecturerName to (475,165,120,25)

SET the font of field_lecturerName to (font_ten)

ADD field_lecturerName to panel_academic

INITIALIZE the JLabel to label_startingDate

SET the text of label_startingDate to Starting Date

SET the bounds of label_startingDate to (350,205,150,25)

SET the font as font_eleven

GIVE ("Arial",Font.PLAIN,15) to font_eleven

SET the font of label_startingDate to (font_eleven)

ADD label_startingDate to panel_academic

INITIALIZE the JComboBox to combobox_years

SET the bounds of combobox_years to (475,205,60,25)

ADD combobox_years to panel_academic

INITIALIZE the JComboBox to combobox_months

SET the bounds of combobox_months to (540,205,90,25)

ADD combobox_months to panel_academic

INITIALIZE the JComboBox to combobox_days

SET the bounds of combobox_days to (635,205,40,25)

ADD combobox_days to panel_academic

INITIALIZE the JLabel to label_completionDate

SET the text of label_completionDate to Completion Date

SET the bounds of label_completionDate to (350,205,150,25)

SET the font as font_twelve

GIVE ("Arial",Font.PLAIN,15) to font_twelve

SET the font of label_completionDate to (font_twelve)

ADD label_completionDate to panel_academic

INITIALIZE the JComboBox to combobox_years1

SET the bounds of combobox_years1 to (475,245,60,25)

ADD combobox_years1 to panel_academic

INITIALIZE the JComboBox to combobox_months1

SET the bounds of combobox_months1 to

ADD combobox_months1 to panel_academic (540,245,90,25)

INITIALIZE the JComboBox to combobox_days1

SET the bounds of combobox_days1 to (635,245,40,25)

ADD combobox_days1 to panel_academic

INITIALIZE the JButton to button_register

SET the text of button_register to Register

SET the bounds of button_register to (430,300,175,30)

SET the font as font_thirteen

GIVE ("Arial",Font.BOLD,15) to font_thirteen

SET the font of button_register to (font_thirteen)

INITIALIZE the Color as color_register1

GIVE (42,136,173) to color_register1

SET the background of button_register to (color_register1)

SET the foreground of button_register to (Color.white)

ADD ActionListener to button_register

ADD button_register to panel_academic

INITIALIZE the JButton to button_display

SET the text of button_display to Display

SET the bounds of button_display to (350,350,140,30)

SET the font as font_fourteen

GIVE ("Arial",Font.BOLD,15) to font_fourteen

SET the font of button_display to (font_fourteen)

INITIALIZE the Color as color_display1

GIVE (42,136,173) to color_display1

SET the background of button_display to (color_display1)

SET the foreground of button_display to (Color.white)

ADD ActionListener to button_display

ADD button_display to panel_academic

INITIALIZE the JButton to button_clear

SET the text of button_clear to Clear

SET the bounds of button_clear to (545,350,130,30)

SET the font as font_fifteen

GIVE ("Arial",Font.BOLD,15) to font_fifteen

SET the font of button_clear to (font_fifteen)

INITIALIZE the Color as color_clear1

GIVE (255,102,102) to color_clear1

SET the background of button_clear to (color_clear1)

SET the foreground of button_clear to (Color.white)

ADD ActionListener to button_clear

ADD button_clear to panel_academic

ADD panel_titleAcademic to frame_academic

ADD panel_academic to frame_academic

END DO

CREATE a method nonAcademicCourseGUI()

DO

INITIALIZE the JFrame to frame_non

SET the title of frame_non to Non Academic Course Registration

SET the visibility of frame_non to true

SET resizable of frame_non to true

SET the bounds of frame_non to (450,200,720,485)

INITIALIZE the JPanel as panel_non
SET the layout of panel_non to null
SET the background of panel_non to (Color.white)
SET the bounds of panel_non to (0,0,710,400)

INITIALIZE the JPanel as panel_titleNon
SET the layout of panel_titleNon to null
SET the bounds of panel_titleNon to (0,0,710,60)

INITIALIZE the Color as color_titleNon
GIVE (253,223,91) to color_titleNon
SET the background of panel_titleNon to (color_titleNon)

INITIALIZE the JLabel to title_non
SET the text of title_non to Academic Course
SET the bounds of title_non to (250,15,250,30)
SET the foreground to (Color.black)

SET the font as font_1
GIVE ("Arial",Font.PLAIN,20) to
SET the font of title_non to (font_1)

ADD title_non to panel_titleNon

INITIALIZE the JLabel to label_courseIDNon

SET the text of label_courseIDNon to CourseID

SET the bounds of label_courseIDNon to (30,85,150,25)

SET the font as font_2

GIVE ("Arial",Font.PLAIN,15) to font_2

SET the font of label_courseIDNon to (font_2)

ADD label_courseIDNon to panel_non

INITIALIZE the JTextField to field_courseIDNon

SET the bounds of field_courseIDNon to (150,85,120,25)

SET the font of field_courseIDNon to (font_2)

ADD field_courseIDNon to panel_non

INITIALIZE the JLabel to label_courseNameNon

SET the text of label_courseNameNon to Course name

SET the bounds of label_courseNameNon to (30,125,120,25)

SET the font as font_3

GIVE ("Arial",Font.PLAIN,15) to font_3

SET the font of label_courseNameNon to (font_3)

ADD label_courseNameNon to panel_non

INITIALIZE the JTextField to field_courseNameNon
SET the bounds of field_courseNameNon to (150,125,120,25)
SET the font of field_courseNameNon to (font_3)
ADD field_courseNameNon to panel_non

INITIALIZE the JLabel to label_durationNon
SET the text of label_durationNon to Duration
SET the bounds of label_durationNon to (30,165,120,25)

SET the font as font_4
GIVE ("Arial",Font.PLAIN,15) to font_4
SET the font of label_durationNon to (font_4)
ADD label_durationNon to panel_non

INITIALIZE the JTextField to field_durationNon
SET the bounds of field_durationNon to (150,165,120,25)
SET the font of field_durationNon to (font_4)
ADD field_durationNon to panel_non

INITIALIZE the JLabel to label_prerequisiteNon
SET the text of label_prerequisiteNon to Prerequisite
SET the bounds of label_prerequisiteNon to (30,205,120,25)

SET the font as font_5

GIVE ("Arial",Font.PLAIN,15) to font_5

SET the font of label_prerequisiteNon to (font_5)

ADD label_prerequisiteNon to panel_non

INITIALIZE the JTextField to field_prerequisiteNon

SET the bounds of field_prerequisiteNon to (150,205,120,25)

SET the font of field_prerequisiteNon to (font_5)

ADD field_prerequisiteNon to panel_non

INITIALIZE the JButton to button_addNon

SET the text of button_addNon to Add

SET the bounds of button_addNon to (105,265,100,30)

SET the font as font_6

GIVE ("Arial",Font.BOLD,15) to font_6

SET the font of button_addNon to (font_6)

INITIALIZE the Color as color_addNon

GIVE (255,102,102) to color_addNon

SET the background of button_addNon to (color_addNon)

SET the foreground of button_addNon to (Color.white)

ADD ActionListener to button_addNon

ADD button_addNon to panel_non

INITIALIZE the JButton to button_changeCourse2
SET the text of button_changeCourse2 to Change Course
SET the bounds of button_changeCourse2 to (350,80,200,30)

SET the font as font_7
GIVE ("Arial",Font.BOLD,15) to font_7
SET the font of button_changeCourse2 to (font_7)

INITIALIZE the Color as color_changeCourse2
GIVE (255,102,102) to color_changeCourse2
SET the background of button_changeCourse2 to (color_changeCourse2)
SET the foreground of button_changeCourse2 to (Color.white)

ADD ActionListener to button_changeCourse2

ADD button_changeCourse2 to panel_non

INITIALIZE the JLabel to label_courseLeaderNon
SET the text of label_courseLeaderNon to Course Leader
SET the bounds of label_courseLeaderNon to (350,125,150,25)

SET the font as font_8
GIVE ("Arial",Font.PLAIN,15) to font_8
SET the font of label_courseLeaderNon to (font_8)
ADD label_courseLeaderNon to panel_non

INITIALIZE the JTextField to field_courseLeaderNon
SET the bounds of field_courseLeaderNon to (475,125,120,25)
SET the font of field_courseLeaderNon to (font_8)
ADD field_courseLeaderNon to panel_non

INITIALIZE the JLabel to label_instructorNameNon
SET the text of label_instructorNameNon to Instructor Name
SET the bounds of label_instructorNameNon to (350,165,150,25)

SET the font as font_9
GIVE ("Arial",Font.PLAIN,15) to font_9
SET the font of label_instructorNameNon to (font_9)
ADD label_instructorNameNon to panel_non

INITIALIZE the JTextField to field_instructorNameNon
SET the bounds of field_instructorNameNon to (475,165,120,25)
SET the font of field_instructorNameNon to (font_9)
ADD field_instructorNameNon to panel_non

INITIALIZE the JLabel to label_startingDateNon
SET the text of label_startingDateNon to Starting Date
SET the bounds of label_startingDateNon to (350,205,150,25)

SET the font as font_10

GIVE ("Arial",Font.PLAIN,15) to font_10
SET the font of label_startingDate to (font_10)
ADD label_startingDateNon to panel_non

INITIALIZE the JComboBox to combobox_yearsNon
SET the bounds of combobox_yearsNon to (475,205,60,25)
ADD combobox_yearsNon to panel_non

INITIALIZE the JComboBox to combobox_monthsNon
SET the bounds of combobox_monthsNon to (540,205,90,25)
ADD combobox_monthsNon to panel_non

INITIALIZE the JComboBox to combobox_daysNon
SET the bounds of combobox_daysNon to (635,205,40,25)
ADD combobox_daysNon to panel_non

INITIALIZE the JLabel to label_completionDateNon
SET the text of label_completionDateNon to Completion Date
SET the bounds of label_completionDateNon to (350,205,150,25)

SET the font as font_11
GIVE ("Arial",Font.PLAIN,15) to font_11
SET the font of label_completionDateNon to (font_11)
ADD label_completionDateNon to panel_non

INITIALIZE the JComboBox to combobox_yearsNon1

SET the bounds of combobox_yearsNon1 to (475,205,60,25)
ADD combobox_yearsNon1 to panel_non

INITIALIZE the JComboBox to combobox_monthsNon1
SET the bounds of combobox_monthsNon1 to (540,205,90,25)
ADD combobox_monthsNon1 to panel_non

INITIALIZE the JComboBox to combobox_daysNon1
SET the bounds of combobox_daysNon1 to (635,205,40,25)
ADD combobox_daysNon1 to panel_non

INITIALIZE the JLabel to label_examDateNon
SET the text of label_examDateNon to Exam Date
SET the bounds of label_examDateNon to (350,205,150,25)

SET the font as font_12
GIVE ("Arial",Font.PLAIN,15) to font_12
SET the font of label_examDateNon to (font_12)
ADD label_examDateNon to panel_non

INITIALIZE the JComboBox to combobox_yearsNon2
SET the bounds of combobox_yearsNon2 to (475,285,60,25)
ADD combobox_yearsNon2 to panel_non

INITIALIZE the JComboBox to combobox_monthsNon2
SET the bounds of combobox_monthsNon2 to (540,285,90,25)

ADD combobox_monthsNon2 to panel_non

INITIALIZE the JComboBox to combobox_daysNon2

SET the bounds of combobox_daysNon2 to (635,285,40,25)

ADD combobox_daysNon2 to panel_non

INITIALIZE the JButton to button_registerNon

SET the text of button_registerNon to Register

SET the bounds of button_registerNon to (350,340,140,30)

SET the font as font_13

GIVE ("Arial",Font.BOLD,15) to font_13

SET the font of button_registerNon to (font_13)

INITIALIZE the Color as color_register2

GIVE (255,102,102) to color_register2

SET the background of button_registerNon to (color_register2)

SET the foreground of button_registerNon to (Color.white)

ADD ActionListener to button_registerNon

ADD button_registerNon to panel_non

INITIALIZE the JButton to button_removeNon

SET the text of button_removeNon to Remove

SET the bounds of button_removeNon to (545,340,130,30)

SET the font as font_14

GIVE ("Arial",Font.BOLD,15) to font_14

SET the font of button_removeNon to (font_14)

INITIALIZE the Color as color_remove2

GIVE (255,102,102) to color_remove2

SET the background of button_removeNon to (color_remove2)

SET the foreground of button_removeNon to (Color.white)

ADD ActionListener to button_removeNon

ADD button_removeNon to panel_non

INITIALIZE the JButton to button_displayNon

SET the text of button_displayNon to Display

SET the bounds of button_displayNon to (350,390,140,30)

SET the font as font_15

GIVE ("Arial",Font.BOLD,15) to font_15

SET the font of button_displayNon to (font_15)

INITIALIZE the Color as color_display2

GIVE (255,102,102) to color_display2

SET the background of button_displayNon to (color_display2)

SET the foreground of button_displayNon to (Color.white)

ADD ActionListener to button_displayNon

ADD button_displayNon to panel_non

INITIALIZE the JButton to button_clearNon

SET the text of button_clearNon to Clear

SET the bounds of button_clearNon to (545,390,130,30)

SET the font as font_16

GIVE ("Arial",Font.BOLD,15) to font_16

SET the font of button_clearNon to (font_16)

INITIALIZE the Color as color_clear2

GIVE (255,102,102) to color_clear2

SET the background of button_clearNon to (color_clear2)

SET the foreground of button_clearNon to (Color.white)

ADD ActionListener to button_clearNon

ADD button_clearNon to panel_non

ADD panel_titleNon to frame_non

ADD panel_non to frame_non

END DO

CREATE a method actionPerformed(ActionEvent e)

DO

if(e.getSource() == button_mainAcademic)

DO

CALL academicCourseGUI()

END DO

else if(e.getSource() == button_mainNonAcademic)

DO

CALL nonAcademicCourseGUI()

END DO

if(e.getSource() == button_changeCourse1)

DO

CALL nonAcademicCourseGUI()

DISPOSE frame_academic

END DO

else if(e.getSource() == button_changeCourse2)

DO

CALL academicCourseGUI()

DISPOSE frame_non

END DO

if(e.getSource() == button_add)

DO

try

DO

add_academicID = field_courseID.getText();

add_academicName=field_courseName.getText();

add_academicDuration=field_duration.getText();

int

duration=Integer.parseInt(add_academicDuration());


```

add_academicLevel = field_level.getText();
add_academicCredit = field_credit.getText();
add_numberOfAssessments=
field_numberOfAssessments.getText();
int numberOfAssessments_A=
Integer.parseInt(add_numberOfAssessments);
academicExist = false;

for(Course var_add : aList)
    DO
        if(var_add.getCourseID().equals(add_academicID))
            DO
                academicExist = true;
                break;
            END DO
        END DO
    END DO

if(academicExist == false)
    DO
        obj_Academic=new
        AcademicCourse(add_academicID,add_acade
        micName,duration_A,add_academicLevel,add
        _academicCredit, numberOfAssessments_A);

        aList.add(obj_Academic);

        message_academic = new JOptionPane();
        String final_msgAcademic = "Course ID: " +
        add_academicID + "\n Course Name: " +
        add_academicName + "\n Duration: " +
        add_academicDuration + "\n Level: " +

```

```
add_academicLevel + "\n Credit: " +  
add_academicCredit + "\n Number of  
assessments: " + add_numberOfAssessments  
+ "\n \n has been successfully added \n to  
Academic Course";
```

```
message_academic.showMessageDialog(frame  
e_academic, final_msgAcademic);
```

END DO

```
else if(academicExist == true)
```

DO

```
JOptionPane.showMessageDialog(frame_acad  
emic, "Course ID " + add_academicID + " is  
already added.");
```

END DO

```
catch (NumberFormatException numberFormat_one)
```

DO

```
try
```

DO

```
if(add_academicID.equals("") &&  
add_academicName.equals("")  
&&  
add_academicDuration.equals("")  
&&  
add_academicLevel.equals("")  
&&  
add_academicCredit.equals("")  
&&  
add_numberOfAssessments.equ  
als("") )
```

DO

```
JOptionPane.showMessageDialog  
g(frame_academic,"Please don  
not leave the text field empty or  
fill the text field properly.",  
"ERROR",  
JOptionPane.ERROR_MESSAG  
E);
```

```
END DO
```

```
END DO
```

```
END DO
```

```
catch(NullPointerException nullPointer_one)
```

```
DO
```

```
JOptionPane.showMessageDialog  
g(frame_academic,"Please don  
not leave the text field empty or  
fill the text field properly.",  
"ERROR",  
JOptionPane.ERROR_MESSAG  
E);
```

```
END DO
```

```
END DO
```

```
END DO
```

```
if(e.getSource() == button_register)
```

```
DO
```

```
register_academicID = field_courseID.getText();
```

```
register_courseLeader = field_courseLeader.getText();
```

```
register_lecturerName = field_lecturerName.getText();
```

```

String
years=(combobox_years.getSelectedItemAt()).toString();
String
months= (combobox_months.getSelectedItemAt()).toString();
String
days= (combobox_days.getSelectedItemAt()).toString();

register_startingDate = years + months + days;

```

```

String
years1= (combobox_years1.getSelectedItemAt()).toString();
String
months1=
(combobox_months1.getSelectedItemAt()).toString();
String
days1= (combobox_days1.getSelectedItemAt()).toString();

register_completionDate = years1 + months1 + days1;

```

```

for(Course var : aList)
    DO
        If(var.getCourseID().equals(register_academicID))

            DO
                if(var instanceof AcademicCourse)
                    DO
                        ref_academic = (AcademicCourse)var;

```

```
if(ref_academic.getisRegistered() ==  
true)
```

```
DO
```

```
    JOptionPane.showMessageDialog(  
frame_academic,"Sorry, the  
course is already registered.");
```

```
    break;
```

```
END DO
```

```
else if(ref_academic.getisRegistered()  
== false)
```

```
DO
```

```
    ref_academic.register(register_co  
urseLeader,  
register_lecturerName,  
register_startingDate,  
register_completionDate);
```

```
    JOptionPane.showMessageDialog(  
frame_academic,"The course is  
registered successfully.");
```

```
    break;
```

```
END DO
```

```
END DO
```

```
END DO
```

```
else
```

```
DO
```

```
    JOptionPane.showMessageDialog(fram  
e_academic,"The Course ID entered to  
add the course does not matches \n with
```

```

        the Course ID entered to register the
        course.", "ERROR",
        JOptionPane.ERROR_MESSAGE);

        break;

    END DO

END DO

END DO

for(Course var1 : aList)
    DO
        if(e.getSource() == button_display)
            DO
                if(var1 instanceof AcademicCourse)
                    DO
                        ref_academicDisplay=
                        (AcademicCourse)var1;
                        ref_academicDisplay.display();
                    END DO
                END DO
            END DO
        END DO
    END DO

    if(e.getSource() == button_clear)
        DO
            field_courseID.setText("");
            field_courseName.setText("");
            field_duration.setText("");
            field_level.setText("");
            field_credit.setText("");
            field_numberOfAssessments.setText("");

```

```

field_courseID.setText("");
field_courseLeader.setText("");
field_lecturerName.setText("");
combobox_years.setSelectedIndex(0);
combobox_months.setSelectedIndex(0);
combobox_days.setSelectedIndex(0);
combobox_years1.setSelectedIndex(0);
combobox_months1.setSelectedIndex(0);
combobox_days1.setSelectedIndex(0);

```

END DO

```

if(e.getSource() == button_addNon)

```

DO

```

    try

```

DO

```

        add_nonAcademicCourseID =
        field_courseIDNon.getText();

```

```

        add_nonAcademicCourseName =
        field_courseNameNon.getText();

```

```

        add_nonAcademicDuration =
        field_durationNon.getText();

```

```

        int duration_NA =
        Integer.parseInt(add_nonAcademicDuration);

```

```

        add_nonAcademicPrerequisite =
        field_prerequisiteNon.getText();

```

```
nonAcademicExist = false;
```

```
for(Course var_addNon : aList)
```

```
    DO
```

```
        if(var_addNon.getCourseID().equals  
            (add_nonAcademicCourseID))
```

```
            DO
```

```
                nonAcademicExist = true;
```

```
                break;
```

```
            END DO
```

```
    END DO
```

```
if(nonAcademicExist == false)
```

```
DO
```

```
    obj_NonAcademic = new  
    NonAcademicCourse(add_nonAcademicCourse  
        eID , add_nonAcademicCourseName,  
        duration_NA, add_nonAcademicPrerequisite);
```

```
    aList.add(obj_NonAcademic);
```

```
message_nonAcademic = new JOptionPane();
```

```
String final_messageNon = "Course ID: " +  
    add_nonAcademicCourseID + "\nCourse  
    Name: " + add_nonAcademicCourseName +  
    "\nDuration: " + add_nonAcademicDuration +  
    "\nPrerequisite: " +
```



```
add_nonAcademicPrerequisite + "\n\n has  
been successfully added \n to Non Academic  
Course";
```

```
message_nonAcademic.showMessageDialog(f  
rame_non, final_messageNon
```

```
END DO
```

```
else if(nonAcademicExist == true)
```

```
DO
```

```
JOptionPane.showMessageDialog(fram  
e_non, "Course ID " +  
add_nonAcademicCourseID+ " is  
already added.");
```

```
END DO
```

```
END DO
```

```
catch(NumberFormatException numberFormat_two)
```

```
DO
```

```
try
```

```
DO
```

```
if(add_nonAcademicCourseID.equals(""))  
&&  
add_nonAcademicCourseName.equals(  
"") &&  
add_nonAcademicDuration.equals(""))  
&&  
add_nonAcademicPrerequisite.equals("  
) )
```

```
DO
```

```
JOptionPane.showMessag  
eDialog(frame_academic,"  
Please don not leave the  
text field empty or fill the  
text field properly.",
```

```
"ERROR",  
JOptionPane.ERROR_MESSAGE);
```

```
END DO
```

```
END DO
```

```
catch(NullPointerException nullPointer_two)
```

```
DO
```

```
JOptionPane.showMessageDialog(frame_academic,"Please don not leave the  
text field empty or fill the text field  
properly.", "ERROR",  
JOptionPane.ERROR_MESSAGE);
```

```
END DO
```

```
END DO
```

```
END DO
```

```
if(e.getSource() == button_registerNon)
```

```
DO
```

```
register_nonAcademicID = field_courseIDNon.getText();
```

```
register_nonCourseLeader =  
field_courseLeaderNon.getText();
```

```
register_nonInstructorName =  
field_instructorNameNon.getText();
```

```
String
```

```
years= (combobox_yearsNon.getSelectedItem()).toString();
```

```
String
months=
(combobox_monthsNon.getSelectedItem()).toString();
String
days= (combobox_daysNon.getSelectedItem()).toString();

register_nonStartingDate = years + months + days;
```

```
String
years1=
(combobox_yearsNon1.getSelectedItem()).toString();
String
months1=
(combobox_monthsNon1.getSelectedItem()).toString();
String
days1= (combobox_daysNon1.getSelectedItem()).toString();

register_nonCompletionDate = years1 + months1 + days1;
```

```
String
years2=
(combobox_yearsNon2.getSelectedItem()).toString();
String
months2=
(combobox_monthsNon2.getSelectedItem()).toString();
String
```

```
days2= (combobox_daysNon2.getSelectedItemId()).toString();
```

```
register_nonExamDate = years2 + months2 + days2;
```

```
for(Course var_non : aList)
```

```
    DO
```

```
        if(var_non.getCourseID().equals  
            (register_nonAcademicID))
```

```
            DO
```

```
                if(var_non instanceof  
                    NonAcademicCourse)
```

```
                    DO
```

```
                        ref_nonAcademic =  
                        (NonAcademicCour  
                        se)var_non;
```

```
                        if(ref_nonAcademic.  
                            getisRegistered()  
                            == true)
```

```
                            DO
```

```
                                JOptionPane  
                                .showMessa  
                                geDialog(fra  
                                me_non,"Sor  
                                ry, the  
                                course is  
                                already  
                                registered.");
```

```
                                break;
```

```
                            END DO
```

```
else  
if(ref_nonAcademic.  
getisRegistered()  
== false)
```

```
DO
```

```
    ref_nonAcad  
    emic.register  
    (register_non  
    CourseLead  
    er,  
    register_nonI  
    nstructorNa  
    me,  
    register_non  
    StartingDate,  
    register_non  
    CompletionD  
    ate,  
    register_non  
    ExamDate);
```

```
    JOptionPane  
    .showMessa  
    geDialog(fra  
    me_non,"The  
    course is  
    registered  
    successfully.  
    ");
```

```
    break;
```

```
END DO
```

```
END DO
```

```
END DO
```

```
else
```

```
DO
```

```

OptionPane.showMessageDialog(
    frame_non, "The Course ID
    entered to add the course does
    not matches \n with the Course
    ID entered to register the
    course.", "ERROR",
    JOptionPane.ERROR_MESSAG
    E);

    break;

```

END DO

END DO

END DO

```

if(e.getSource() == button_removeNon)

```

DO

```

    remove_nonAcademicID = field_courseIDNon.getText();

```

```

    for(Course var_non1 : aList)

```

DO

```

        if(var_non1.getCourseID().equals
        (remove_nonAcademicID))

```

DO

```

            if(var_non1 instanceof
            NonAcademicCourse)

```

DO

```

                ref_nonAcademicRemove
                =
                (NonAcademicCourse)var
                _non1;

```

```

                if(ref_nonAcademicRemov

```

```
e.getisRemoved() ==  
false)
```

```
DO
```

```
ref_nonAcademicR  
emove.remove();
```

```
JOptionPane.show  
MessageDialog(fra  
me_non,"The  
course is removed  
successfully.");
```

```
break;
```

```
END DO
```

```
else  
if(ref_nonAcademicRemov  
e.getisRemoved() == true)
```

```
DO
```

```
JOptionPane.show  
MessageDialog(fra  
me_non,"Sorry, the  
course is already  
removed.");
```

```
break;
```

```
END DO
```

```
END DO
```

```
END DO
```

```
else
```

```
DO
```

```
JOptionPane.showMessageDialog(fram  
e_non,"The Course ID entered to add  
the course does not matches \n with the  
Course ID entered to remove the
```

```

        course.", "ERROR",
        JOptionPane.ERROR_MESSAGE);

        break;

    END DO

END DO

END DO

for(Course var_non2 : aList)
    DO
        if(e.getSource() == button_displayNon)
            DO
                if(var_non2 instanceof NonAcademicCourse)
                    DO
                        ref_nonAcademicDisplay =
                        (NonAcademicCourse)var_non2;
                        ref_nonAcademicDisplay.display();
                    END DO
                END DO
            END DO
        END DO
    END DO

    if(e.getSource() == button_clearNon)
        DO
            field_courseIDNon.setText("");
            field_courseNameNon.setText("");
            field_durationNon.setText("");
            field_prerequisiteNon.setText("");

            field_courseIDNon.setText("");
            field_courseLeaderNon.setText("");

```



```
field_instructorNameNon.setText("");
```

```
combobox_yearsNon.setSelectedIndex(0);
```

```
combobox_monthsNon.setSelectedIndex(0);
```

```
combobox_daysNon.setSelectedIndex(0);
```

```
combobox_yearsNon1.setSelectedIndex(0);
```

```
combobox_monthsNon1.setSelectedIndex(0);
```

```
combobox_daysNon1.setSelectedIndex(0);
```

```
combobox_yearsNon2.setSelectedIndex(0);
```

```
combobox_monthsNon2.setSelectedIndex(0);
```

```
combobox_daysNon2.setSelectedIndex(0);
```

```
END DO
```

```
END DO
```

```
END
```

4. Method description

i. INGCollege

a. public void mainGUI() :

This method holds the home or the main GUI in the sense that this method is called from the main method and the other two methods, which are `public void academicCourseGUI()` and `public void nonAcademicCourseGUI()` can be accessed from the main GUI by clicking on their respective buttons.

This method consists of a single `JFrame` and 2 `JPanel`. Other GUI components such as `JLabel` and `JButton` are also included in this method. This method has 2 `JButton` to help users to choose the wanted course registration form by clicking on the respective button. For example: if an individual clicks on a button named Academic Course then a new frame opens up which holds Course Registration for Academic Course.

b. public void academicCourseGUI()

This particular method holds Academic Course Registration. The method basically contains a single frame which has its own title set using the `setTitle` method (pre-defined method). This method also consists of different GUI components like `JLabel`, `TextField`, `JComboBox` and `JButton`.

Different buttons have been included in this GUI or the method using JButton. There are 5 buttons included in this method. The add button basically adds the components filled by the user according. The change course button changes the course to Non Academic Course Registration or opens another frame consisting of Non Academic Course. The register button registers the course. The display buttons display the output and the clear button clears the text fields. The functionalities of the buttons are directed in the actionPerformed method. ActionListener has been added to all the buttons to determine their functions.

c. `public void nonAcademicCourseGUI()`

This particular method holds Non Academic Course Registration. This method also consists of a single frame and two panels. The major GUI components used in this method are JFrame, JPanel, JLabel, JTextField, JComboBox and JButton.

This method also consists of a number of buttons having their own functions respectively. The button structure in this method acts kind of similar to the buttons in the academicCourseGUI(). The major difference is that there is an extra remove button in this method. The add button is used to add the non academic course when clicked. The change button changes to the other course. The register button registers the course. Display button displays the output and the clear button clears the textfields. The remove button removes the non academic course. ActionListener has been added to all the buttons which helps to determine the functioning of the button body in the actionPerformed method.

d. `public void actionPerformed(ActionEvent e)`

The actionPerformed method basically takes action event as a parameter. This method is present in ActionListener interface which conducts event handling. In this particular project, this method is responsible for the functioning of the add buttons, change course buttons, register buttons, display buttons, clear buttons and the remove button.

For Academic Course

i. Add course for Academic Course

The courseId, course name, duration, level, credit, and number of assessments input values are utilized to generate a new AcademicCourse

object, which is then added to an array list of Course class. This is performed when the add button of the AcademicCourse is pressed. Try and catch blocks have also been used for exception handling. Generally, in this project, NumberFormatException and NullPointerException have been handle under the try and catch blocks. JOptionPane() has been used show to related messages in a dialog box.

ii. Register Academic Course

The GUI is used to input the courseID, course leader, lecturer name, starting date, and completion date. On pressing the register button, the input value of courseID is compared to the existing courseID and if the entered courseID is valid, it is used to register the Academic Course from the list. And finally, the method used to register the academic course is called from the AcademicCourse class. JOptionPane() has also been used show to related messages in a dialog box.

iii. Display Academic Course

When the display button is pressed from the Academic Course Registration GUI it calls the display button from the AcademicCourse class.

When the display button is clicked iteration occurs, and the values entered through the text field in the array list are shown calling the display method in the AcademicCourse.

iv. Clear Academic Course

When the clear button is clicked, it basically clears the text fields by setting the text field to empty string and setting the selected index to 0 for combobox.

For Non-Academic Course

i. Add course for Non-Academic Course

The courseID, course name, duration and prerequisite input values are utilized to generate a new NonAcademicCourse object, which is then added to an array list of Course class. This is performed when the add button of the Non-Academic Course is pressed. Try and catch blocks have also been used for exception handling. Generally, in this project, NumberFormatException and NullPointerException have been handle under the try and catch blocks. JOptionPane() has been used show to related messages in a dialog box.

ii. Register Non-Academic Course

The GUI is used to input the courseID, course leader, instructor name, starting date, completion date and exam date. On pressing the register button, the input value of courseID is compared to the existing courseID and if the entered courseID is valid, it is used to register the NonAcademic Course from the list. And finally, the method used to register the non-academic course is called from the AcademicCourse class. JOptionPane() has also been used show to related messages in a dialog box.

iii. Remove Non-Academic Course

When the remove button is pressed, it checks if the new courseID matches with the previous courseID entered. And if the courseID gets matched, remove method from the NonAcademicCourse class is called and if the courseID does not matches, a suitable message through a dialog box is given using the JOptionPane.

iv. Display Non- Academic Course

When the display button is pressed from the Non-Academic Course Registration GUI it calls the display button from the NonAcademicCourse class.

When the display button is clicked iteration occurs, and the values entered through the text field in the array list are shown calling the display method in the NonAcademicCourse.

v. Clear Non-Academic Course

When the clear button is clicked, it basically clears the text fields by setting the text field to empty string and setting the selected index to 0 for combobox.

5. Test:

5.1 Test 1: To test that the program can be compiled and run using the command prompt

Test No:	1
----------	---

Objective:	To test that the program can be compiled and run using the command prompt.
Action:	<ul style="list-style-type: none"> • Command prompt is opened from the file location • Command lines are given: <ul style="list-style-type: none"> i) <code>javac Course.java</code> ii) <code>javac AcademicCourse.java</code> iii) <code>javac NonAcademicCourse.java</code> iv) <code>javac INGCollege.java</code> v) <code>java INGCollege</code>
Expected result:	The java file would compile and main frame or the home frame would open.
Actual result:	The java files were complied and the main frame or the home frame was opened.
Conclusion:	Test successful.

Table 1 To test that the program can be compiled and run using the command prompt

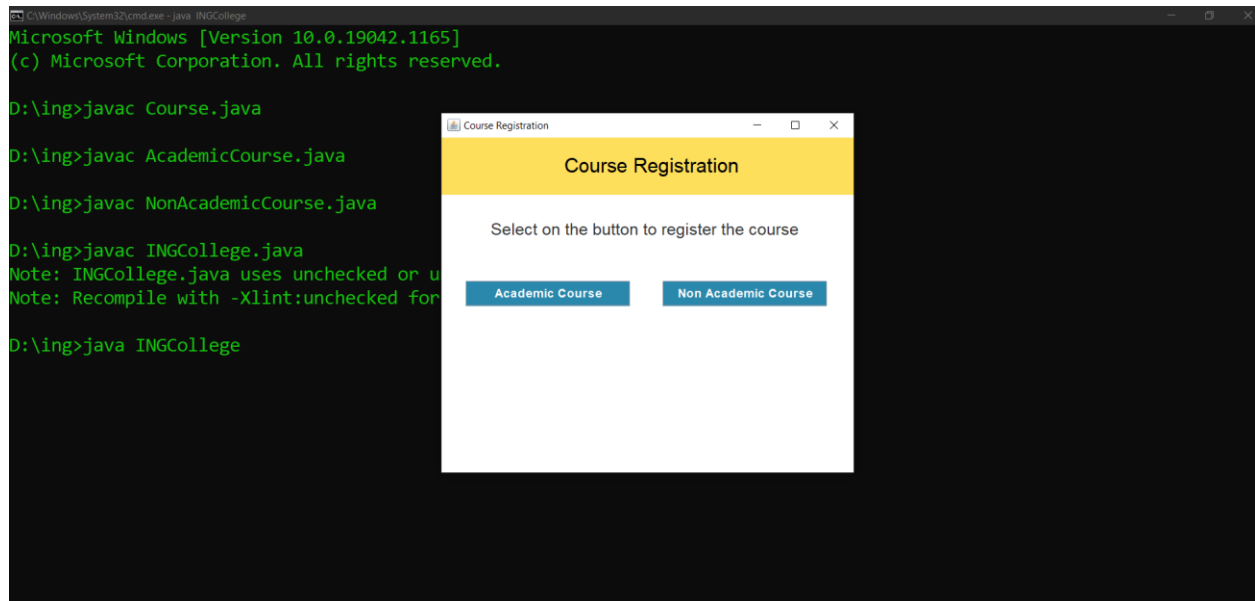


Figure 3 (Test 1) The command lines and the executed program

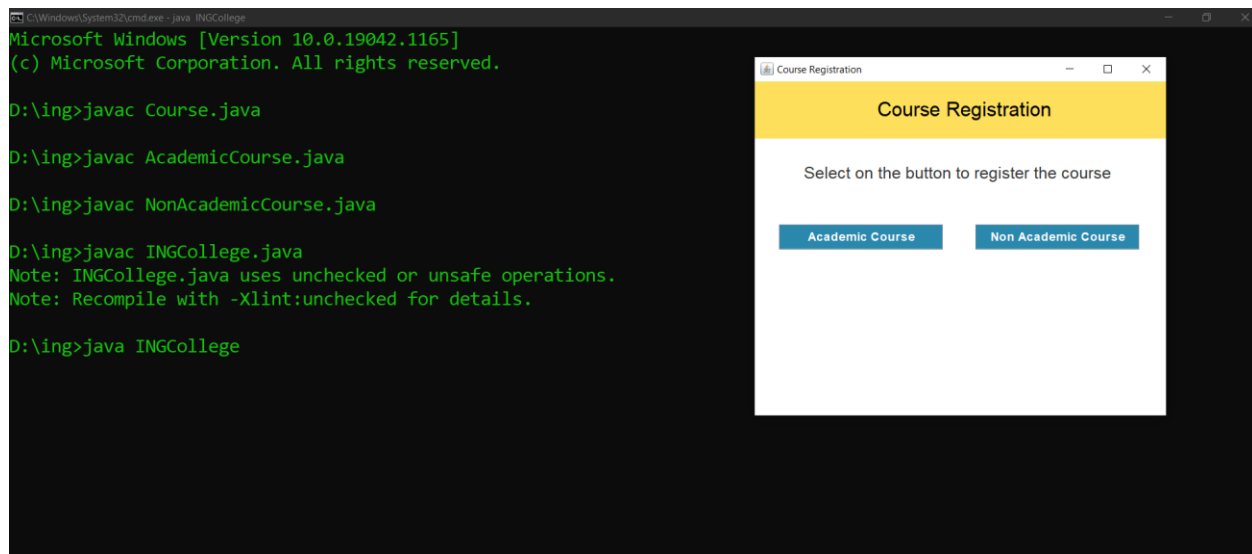


Figure 4 (Test 1) Command lines given to execute and run the program

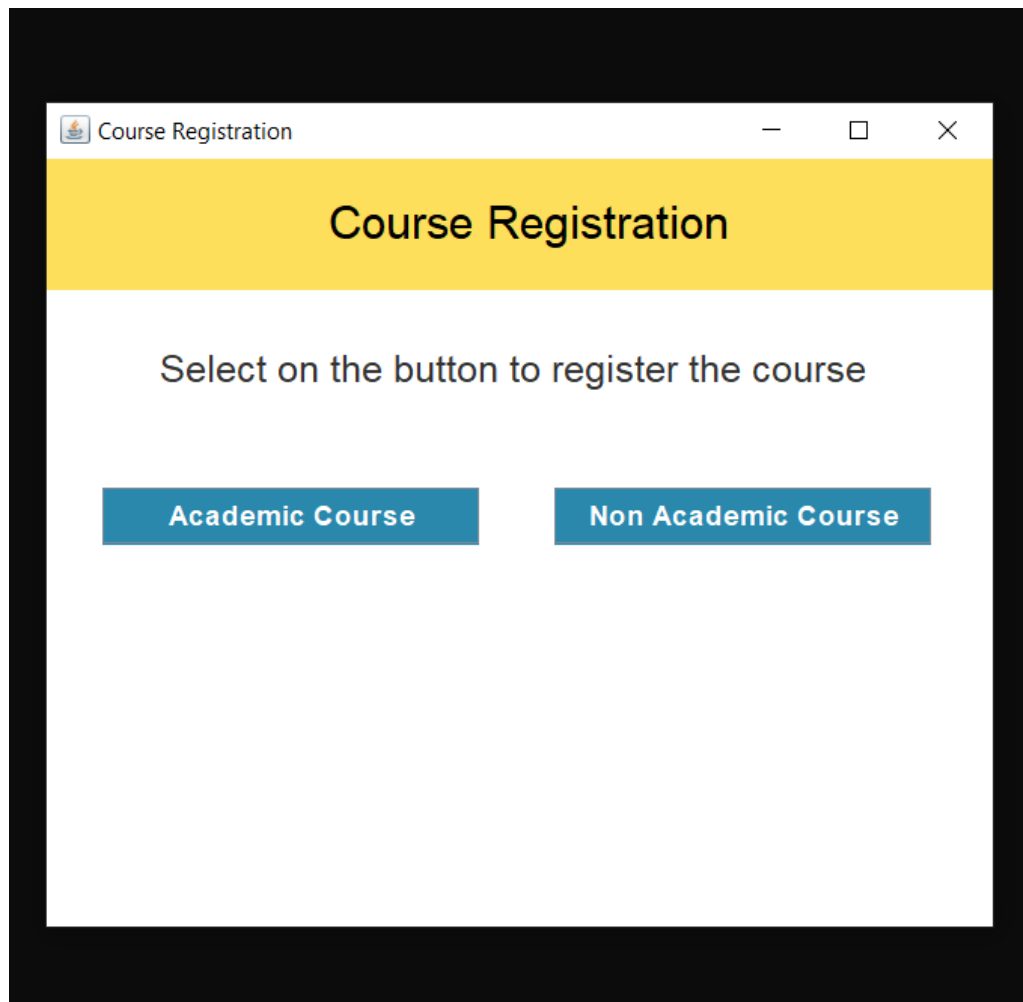


Figure 5 (Test 1) Frame

5.2 Test 2:

5.2.1 Test 2a: To add course for Academic Course

Test	2a
Objective:	To add a Course for Academic Course
Action:	<ul style="list-style-type: none">• The values are entered into the text field.<ul style="list-style-type: none">i. CourseID: 01ii. Course name: programmingiii. Duration: 3iv. Level: 4v. Credit: 120vi. Number of assessments: 5• The add button for Academic Course is pressed.
Expected result:	The course would be added and a message dialogue box would appear.
Actual result:	The course was added and a message dialogue box appeared.
Conclusion:	Test successful.

Table 2 (Test 2a) To add a Course for Academic Course

The screenshot shows a web application window titled "Academic Course Registration". The form has a yellow header with the title "Academic Course". Below the header, there are two columns of input fields. The left column contains: CourseID (01), Course name (programming), Duration (3), Level (4), Credit (120), and Number of assessments (5). The right column contains: Course Leader (empty), Lecturer Name (empty), Starting Date (2020, January, 01), and Completion Date (2023, January, 01). There are four buttons: "Change Course" (blue), "Register" (blue), "Add" (blue), "Display" (blue), and "Clear" (red). The "Add" button is highlighted.

Figure 6 (Test2a) The values are entered into the text field

The screenshot shows the same web application window as Figure 6, but with a "Message" dialog box open in the center. The dialog box has a title bar "Message" and a close button "X". It contains an information icon (i) and the following text: "Course ID: 01", "Course Name: programming", "Duration: 3", "Level: 4", "Credit: 120", "Number of assessments: 5", and "has been successfully added to Academic Course". There is an "OK" button at the bottom of the dialog box. The background form is slightly dimmed.

Figure 7 (Test 2a) The add button for Academic Course is pressed

5.2.2 Test 2b: To add a Course for Non-Academic Course

Test 2b	
Objective:	To add a Course for Non-Academic Course
Action:	<ul style="list-style-type: none">The values are entered into the text field.<ul style="list-style-type: none">vii. CourseID: 7viii. Course name: musicix. Duration: 2x. Prerequisite: basic degreeThe add button for Non-Academic Course is pressed.
Expected result:	The course would be added and a message dialogue box would appear.
Actual result:	The course was added and a message dialogue box appeared.
Conclusion:	Test successful.

Table 3 To add a Course for Non-Academic Course

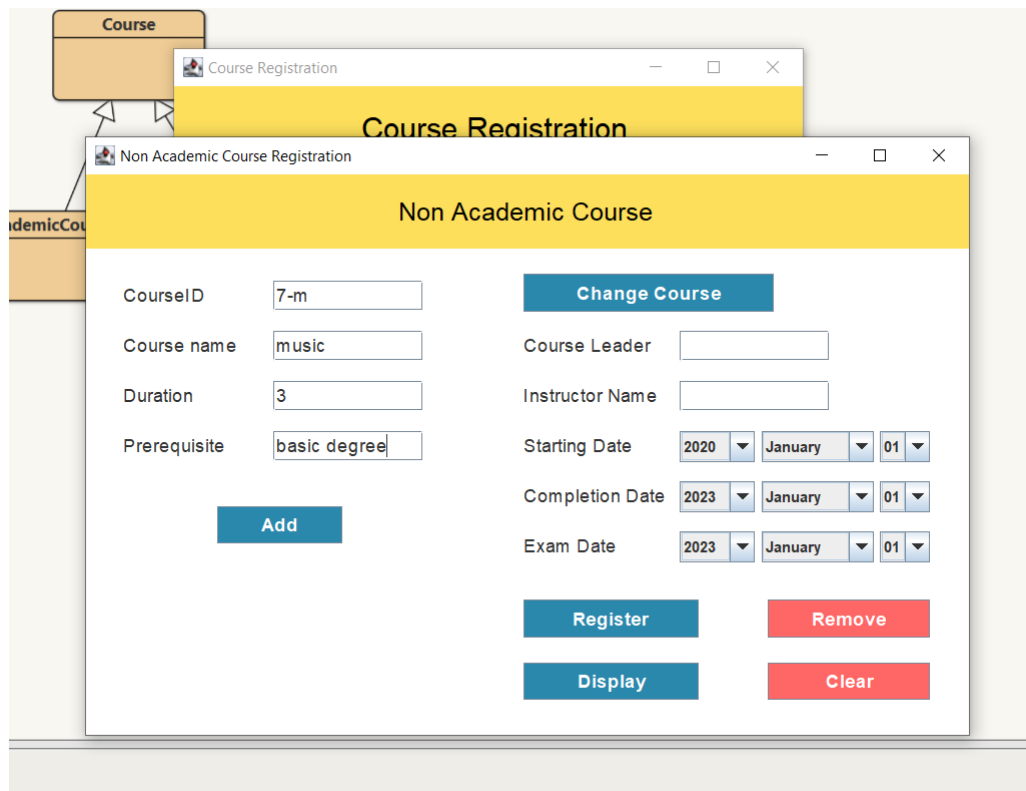


Figure 8 (Test2b) The values are entered into the text field

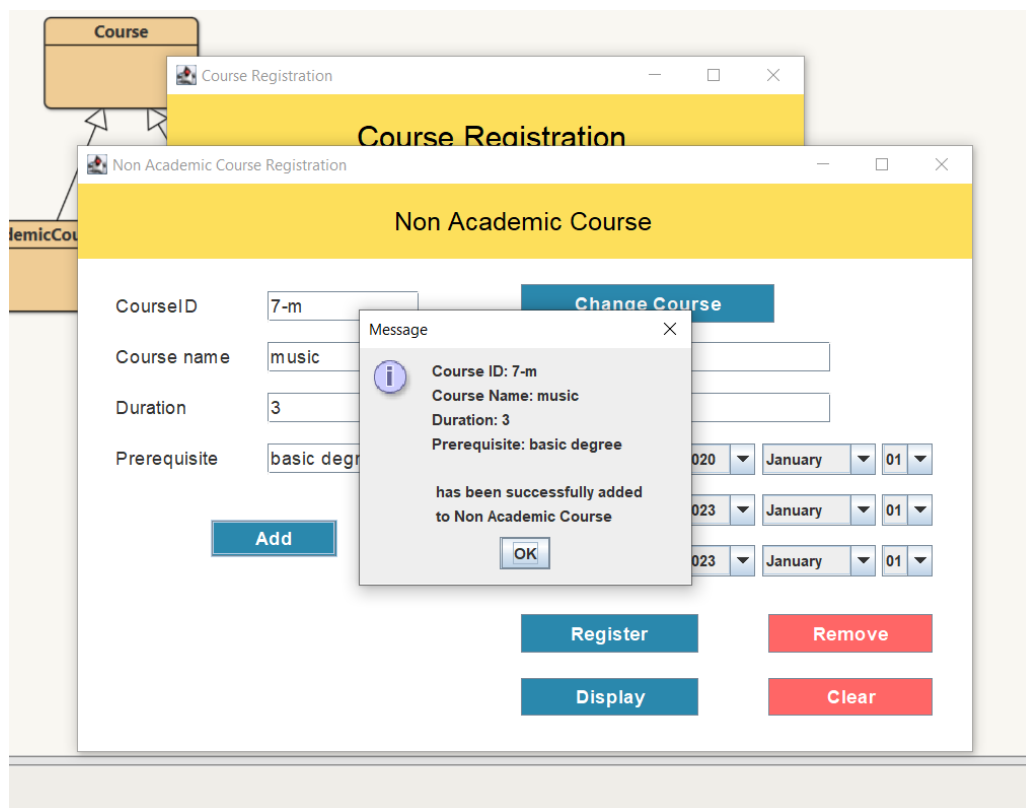


Figure 9 (Test 2b) The add button for Non-Academic Course is pressed

5.2.3 Test 2c: To register Academic Course

Test	2c
Objective:	To register Academic Course
Action:	<ul style="list-style-type: none">The values are entered into the text field.<ul style="list-style-type: none">xi. Course Leader: Eddie Vedderxii. Lecturer Name: Kurt Cobainxiii. Starting Date: 2020/March/05xiv. Completion Date: 2023/April/10The register button for Academic Course is pressed.
Expected result:	The course would be registered and a message dialogue box would appear.
Actual result:	The course was registered and a message dialogue box appeared.
Conclusion:	Test successful.

Table 4 (Test 2c) To register Academic Course

The screenshot displays a web application window titled "Academic Course Registration". The window has a yellow header bar with the text "Academic Course". Below the header, there is a form with two columns of input fields. The left column contains: "CourseID" (01), "Course name" (programming), "Duration" (3), "Level" (4), "Credit" (120), and "Number of assessments" (5). The right column contains: "Course Leader" (Eddie Vedder), "Lecturer Name" (Kurt Cobain), "Starting Date" (2020, March, 05), and "Completion Date" (2023, April, 10). There are several buttons: "Change Course" (blue) next to the CourseID field, "Register" (blue) below the form, "Add" (blue) at the bottom left, "Display" (blue) at the bottom center, and "Clear" (red) at the bottom right. The background shows other windows like "Course" and "Acad".

Field	Value
CourseID	01
Course name	programming
Duration	3
Level	4
Credit	120
Number of assessments	5
Course Leader	Eddie Vedder
Lecturer Name	Kurt Cobain
Starting Date	2020 March 05
Completion Date	2023 April 10

Figure 10 (Test 2c) The values are entered into the text field

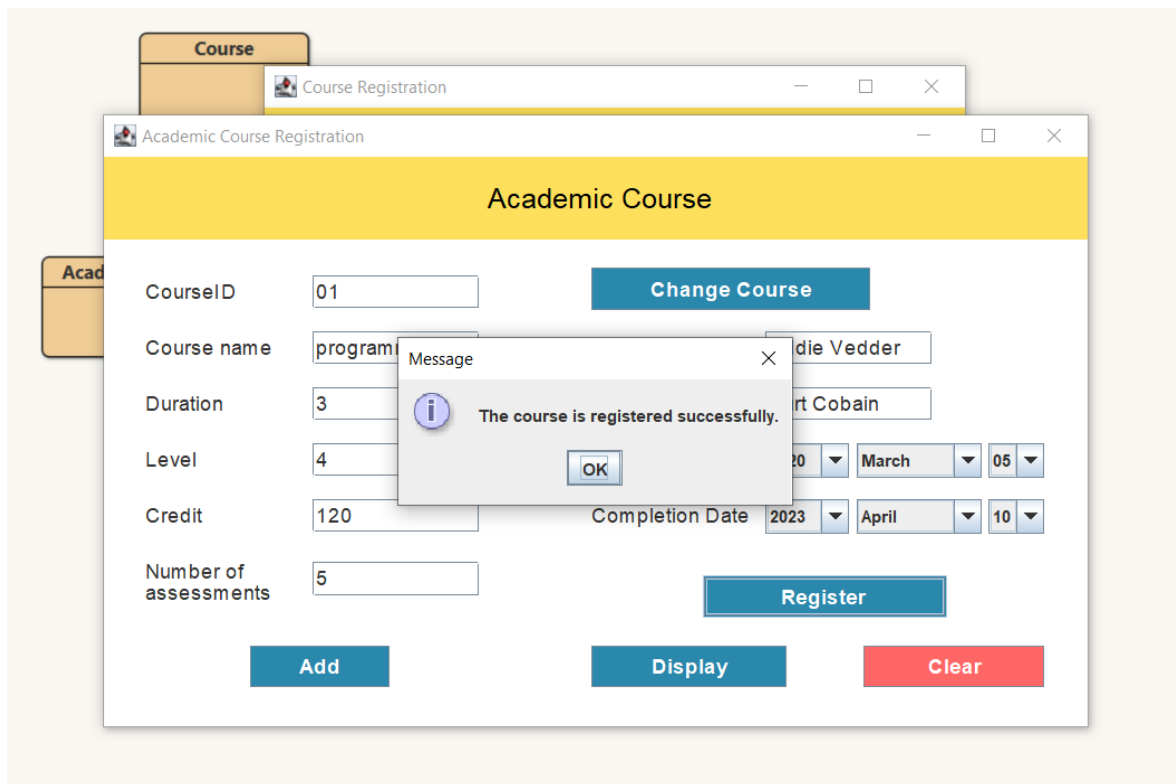


Figure 11 (Test 2c) The register button for Academic Course is pressed

5.2.4 Test 2d: To register Non-Academic Course

Test 2d	
Objective:	To register Non-Academic Course
Action:	<ul style="list-style-type: none">The values are entered into the text field.<ul style="list-style-type: none">xv. Course Leader: Eddie Vedderxvi. Lecturer Name: Kurt Cobainxvii. Starting Date: 2020/March/05xviii. Completion Date: 2023/April/10The register button for Non-Academic Course is pressed.
Expected result:	The course would be registered and a message dialogue box would appear.
Actual result:	The course was registered and a message dialogue box appeared.
Conclusion:	Test successful.

Table 5 (Test 2d) To register Non-Academic Course

The screenshot shows a web application interface for 'Non Academic Course Registration'. The form is titled 'Non Academic Course' and contains several input fields and buttons. The data entered in the form is as follows:

Field	Value
CourseID	7-m
Course name	music
Duration	3
Prerequisite	basic degree
Course Leader	John Mayer
Instructor Name	Chris Martin
Starting Date	2020 March 15
Completion Date	2023 March 05
Exam Date	2023 March 29

Buttons visible on the form include 'Change Course', 'Add', 'Register', 'Remove', 'Display', and 'Clear'.

Figure 12 (Test 2d) The values are entered into the text field

This screenshot shows the same 'Non Academic Course Registration' form as Figure 12, but with a 'Message' dialog box overlaid in the center. The dialog box contains the following text:

Message

The course is registered successfully.

OK

The background form shows the same data as Figure 12, but the 'Register' button is now highlighted, indicating it has been pressed.

Figure 13 (Test 2d) The register button for Non-Academic Course is pressed

5.2.5 Test 2e: To remove Non-Academic Course

Test	2e
Objective:	To remove Non-Academic Course
Action:	<ul style="list-style-type: none">The remove button for Non-Academic Course is pressed.
Expected result:	The course would be removed and a message dialogue box would appear.
Actual result:	The course was removed and a message dialogue box appeared.
Conclusion:	Test successful.

Table 6 (Test 2e) To remove Non-Academic Course

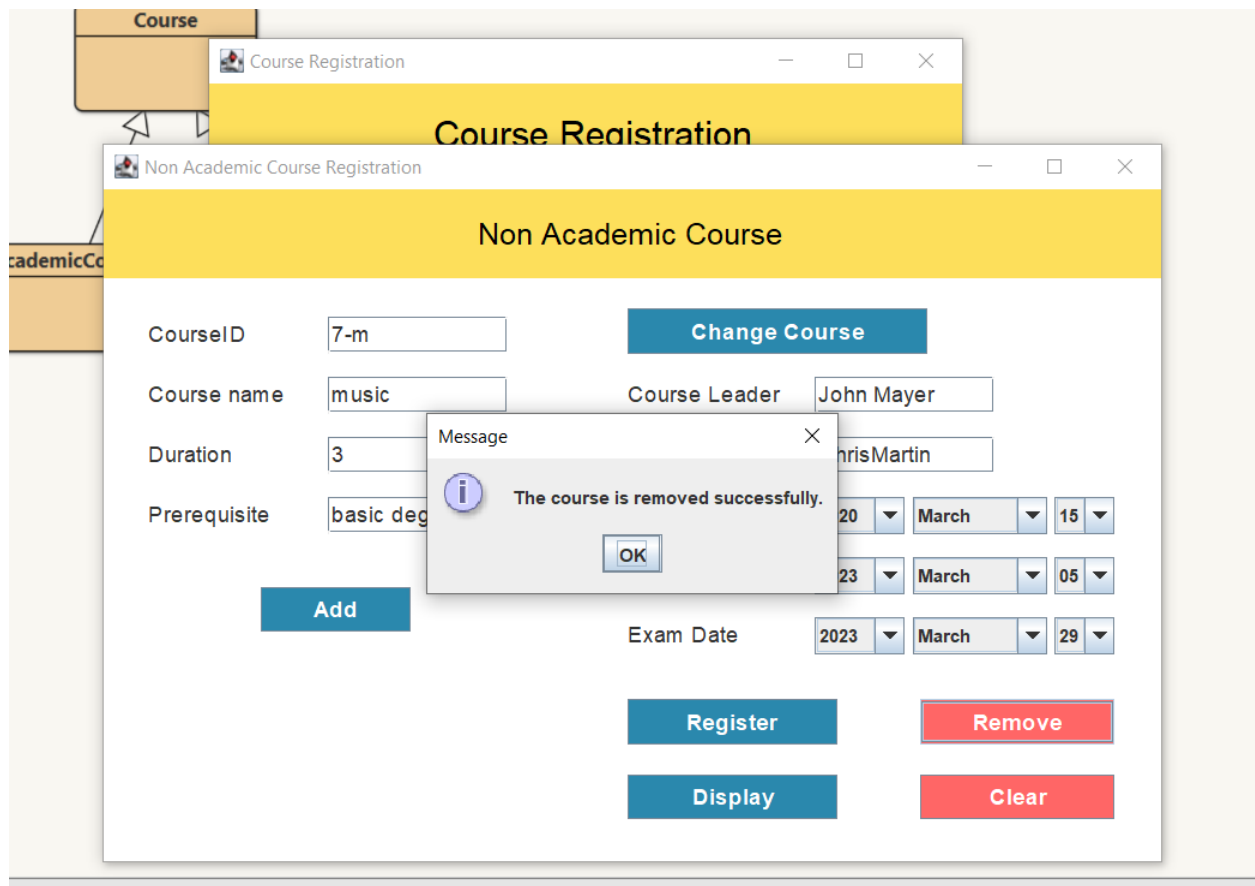


Figure 14 (Test 2e) The remove button for Non-Academic Course is pressed (the course is removed and a message dialogue box appears).

5.3 Test 3:

5.3.1 Test 3a:

i. To add duplicate CourseID (In Academic Course)

Test	3a.i.
Objective:	To add duplicate CourseID in Academic Course
Action:	<ul style="list-style-type: none">• The values are entered in the text field.• The add button is pressed.• Different values are entered in the text field with the same CourseID.• The add button is pressed.
Expected result:	A message box would appear with a message “Course ID (user entered courseID) is already added.”
Actual result:	A message box appeared with a message “Course ID (user entered courseID) is already added.”
Conclusion:	Test successful.

Table 7 (Test 3a (Academic Course)) To add duplicate CourseID in Academic Course

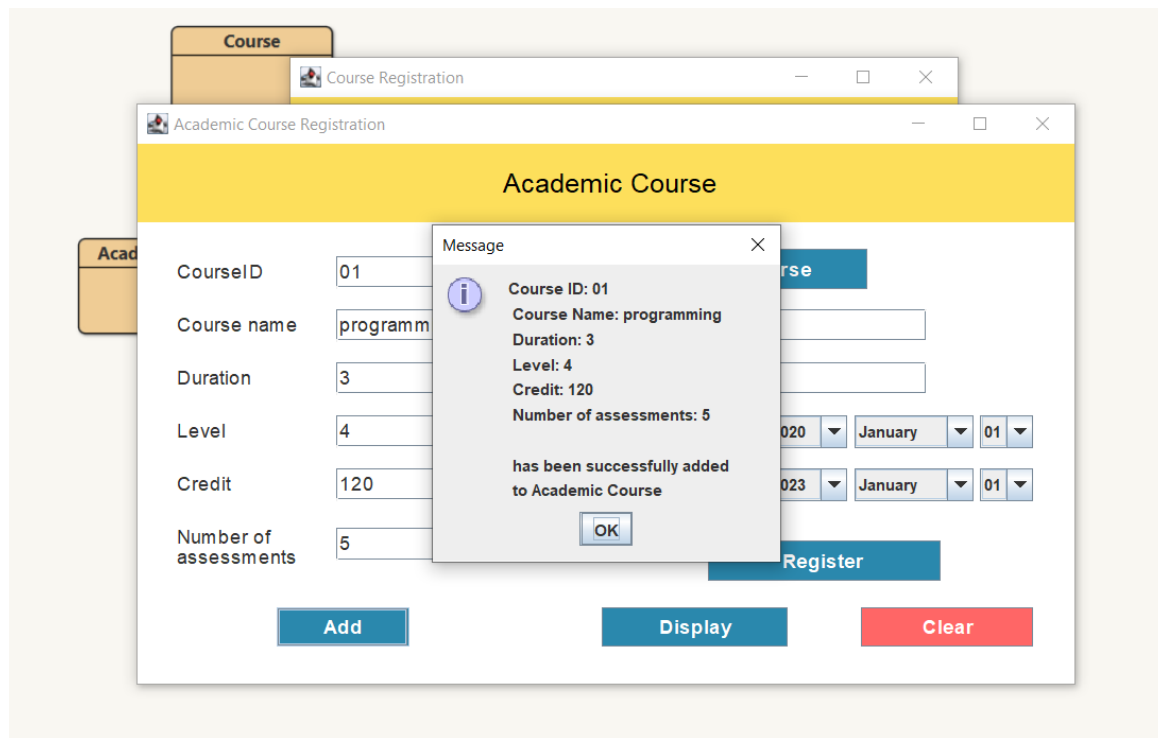


Figure 15 (Test 3a (Academic Course)) The values are entered into the text field. The add button is pressed

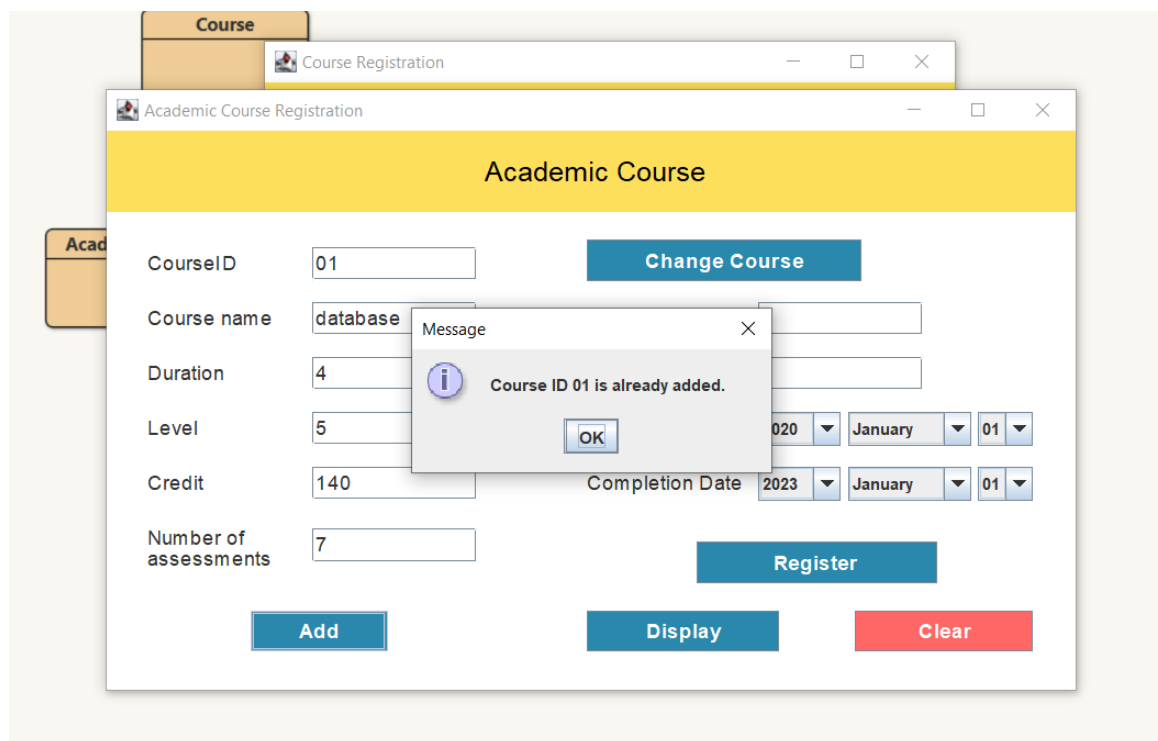


Figure 16 (Test 3 a (Academic Course)) Different values are entered in the text field with the same CourseID. The add button is pressed.

ii. To add duplicate CourseID (In Non-Academic Course)

Test 3a.ii.	
Objective:	To add duplicate CourseID in Non-Academic Course
Action:	<ul style="list-style-type: none">• The values are entered in the text field.• The add button is pressed.• Different values are entered in the text field with the same CourseID.• The add button is pressed.
Expected result:	A message box would appear with a message "Course ID (user entered courseID) is already added."
Actual result:	A message box appeared with a message "Course ID (user entered courseID) is already added."
Conclusion:	Test successful.

Table 8 (Test 3a (Non-Academic Course)) To add duplicate CourseID in Non-Academic Course

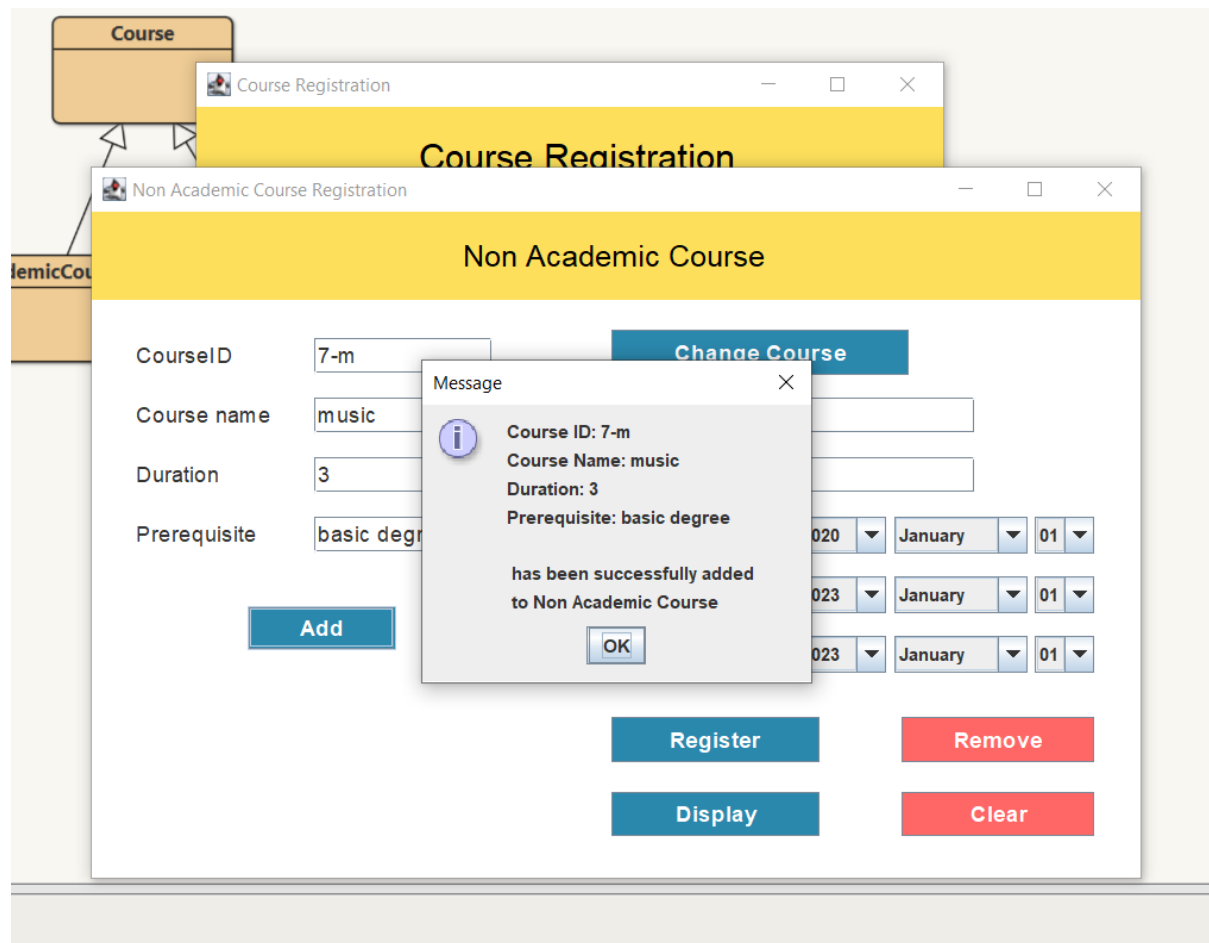


Figure 17 (Test 3a (Non-Academic Course)) The values are entered into the text field. The add button is pressed

The screenshot shows a web application window titled "Non Academic Course Registration". The form has a yellow header bar with the title. Below the header, there are several input fields and buttons. The "CourseID" field contains "7-m". The "Course name" field contains "app developer". The "Duration" field contains "4". The "Prerequisite" field contains "+2 in scier". There is a "Change Course" button next to the "CourseID" field. Below the input fields, there is an "Add" button. To the right of the "Add" button, there are three date pickers for "Exam Date", each showing "2023", "January", and "01". Below the date pickers, there are four buttons: "Register", "Remove", "Display", and "Clear". A "Message" dialog box is open in the center of the form, displaying an information icon and the text "Course ID 7-m is already added." with an "OK" button.

Course Registration

Non Academic Course Registration

Non Academic Course

CourseID: 7-m Change Course

Course name: app developer Course Leader:

Duration: 4

Prerequisite: +2 in scier

Add

Exam Date: 2023 January 01

Register Remove

Display Clear

Message: Course ID 7-m is already added. OK

Figure 18 (Test 3 a (Non-Academic Course)) Different values are entered in the text field with the same CourseID. The add button is pressed.

5.3.2 Test 3b:

i. To register already registered course (Academic Course)

Test	3.i.
Objective:	To register already registered course (Academic Course)
Action:	<ul style="list-style-type: none">• The values are entered in the text field with the same CourseID.• The register button is pressed.
Expected result:	A message box would appear with a message "Sorry, the course is already registered."
Actual result:	A message box appeared with a message "Sorry, the course is already registered."
Conclusion:	Test successful.

Table 9 (Test 3b (Academic Course)) To register already registered course (Academic Course)

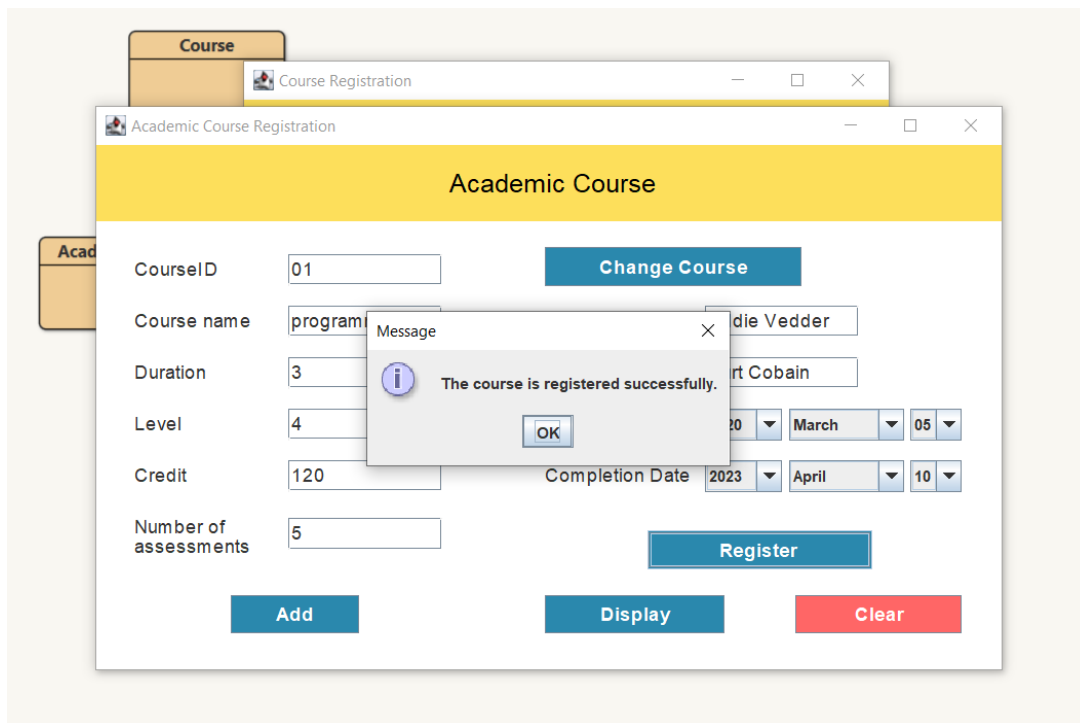


Figure 19 (Test 3b (Academic Course)) The values are entered in the text field with the same CourseID. (To register Academic Course)

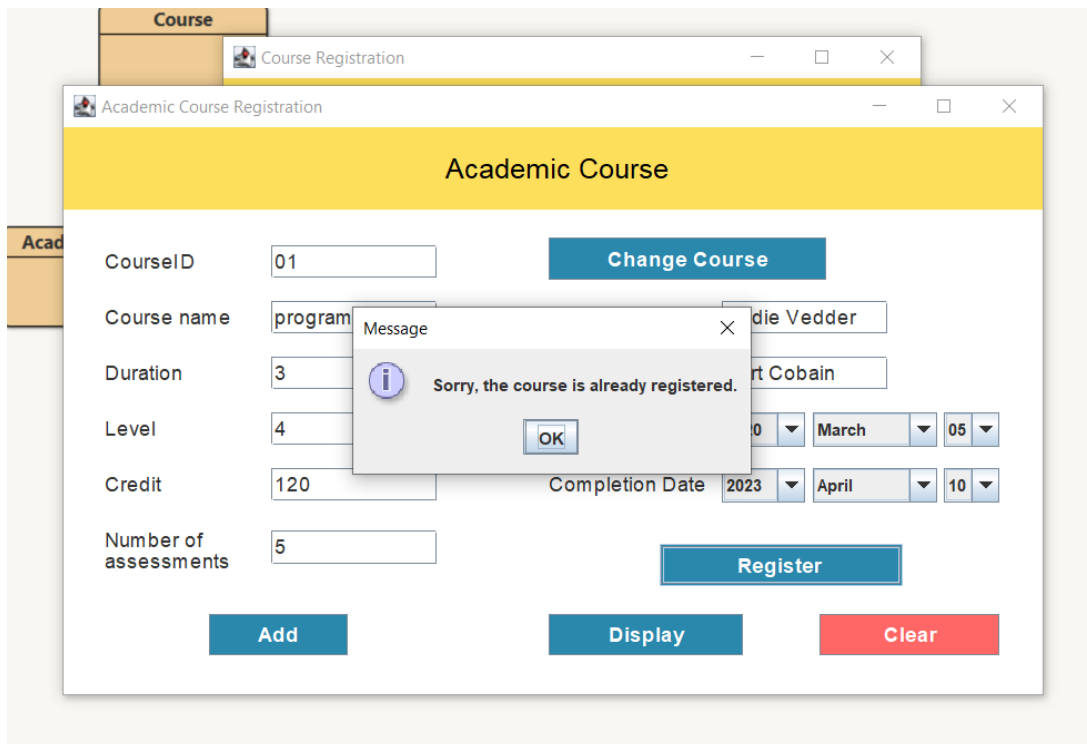


Figure 20 (Test 3b (Academic Course)) The register button is pressed. (To register already registered Academic Course)

ii. To register already registered course (Non-Academic Course)

Test	3.i.
Objective:	To register already registered course (Non-Academic Course)
Action:	<ul style="list-style-type: none"> The values are entered in the text field with the same CourseID. The register button is pressed.
Expected result:	A message box would appear with a message "Sorry, the course is already registered."
Actual result:	A message box appeared with a message "Sorry, the course is already registered."
Conclusion:	Test successful.

Table 10 (Test 3b (Non-Academic Course)) To register already registered course (Non-Academic Course)

The screenshot displays a web application interface for course registration. In the background, a 'Course Registration' window is visible. The foreground window is titled 'Non Academic Course Registration' and features a yellow header. The form contains the following fields and controls:

- CourseID**: Text field with '7-m' entered.
- Course name**: Text field with 'music' entered.
- Duration**: Text field with '3' entered.
- Prerequisite**: Text field with 'basic de' entered.
- Course Leader**: Text field with 'John Mayer' entered.
- Exam Date**: Three date pickers showing '2023', 'March', and '29'.
- Buttons**: 'Change Course' (blue), 'Add' (blue), 'Register' (blue), 'Remove' (red), 'Display' (blue), and 'Clear' (red).

A 'Message' dialog box is overlaid on the form, displaying an information icon and the text: 'The course is registered successfully.' with an 'OK' button.

Figure 21 (Test 3b (Non-Academic Course)) The values are entered in the text field with the same CourseID. (To register Non-Academic Course)

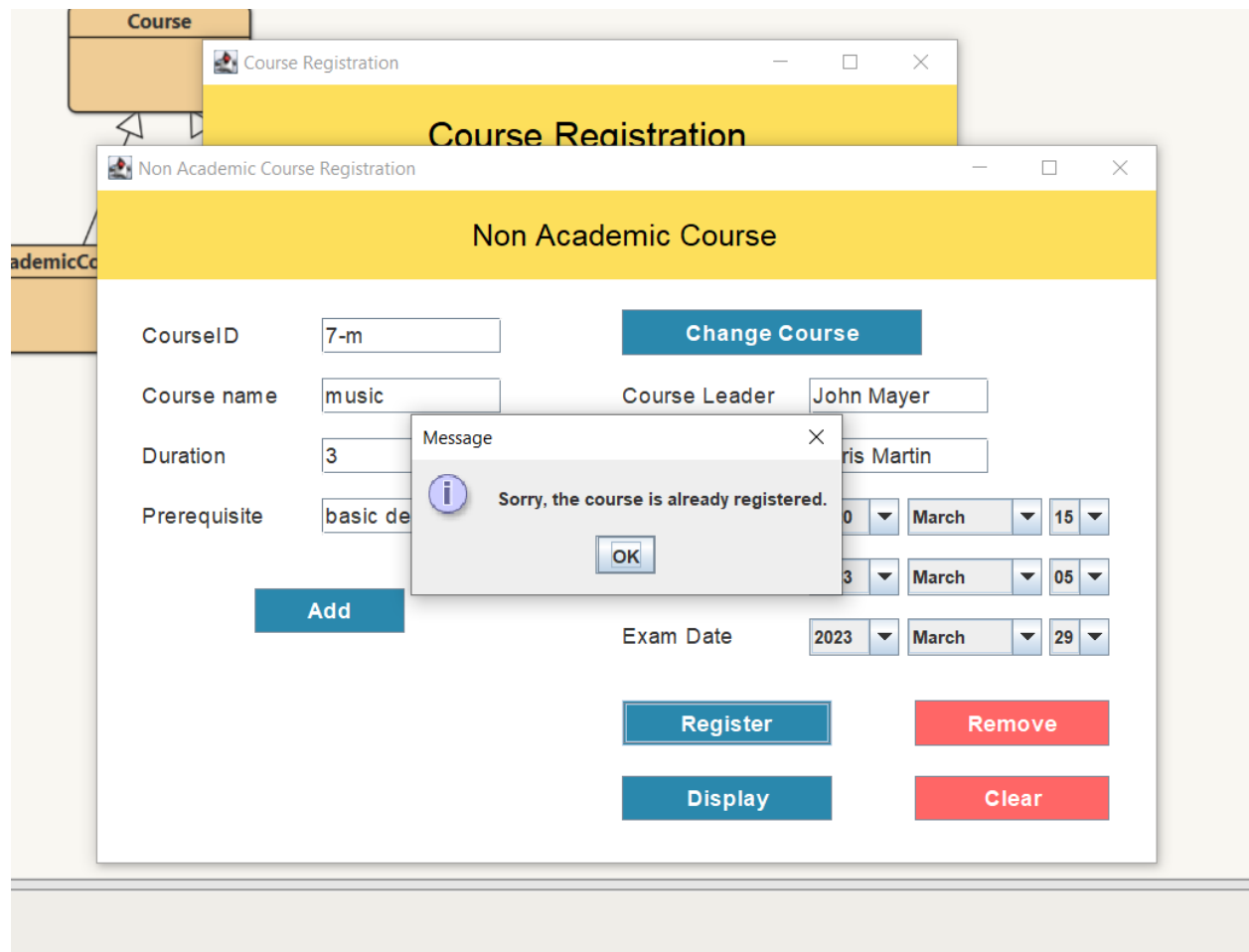


Figure 22 (Test 3b (Non-Academic Course)) The register button is pressed. (To register already registered Non-Academic Course)

5.3.3 Test 3c

Test	3c
Objective:	To remove the Non-Academic Course which is already removed.
Action:	<ul style="list-style-type: none">• The values are entered in the text field with the same CourseID.• The remove button is pressed.
Expected result:	A message box would appear with a message "Sorry, the course is already removed."
Actual result:	A message box appeared with a message "Sorry, the course is already removed."
Conclusion:	Test successful.

Table 11 To remove the Non-Academic Course which is already removed.

The image shows a web application interface for 'Non Academic Course Registration'. The form has a yellow header with the title 'Non Academic Course'. Below the header, there are several input fields and buttons. A modal message box is overlaid on the form, displaying the text 'The course is removed successfully.' with an 'OK' button.

Form Fields and Values:

- CourseID:** 7-m
- Course name:** music
- Duration:** 3
- Prerequisite:** basic deg
- Course Leader:** John Mayer
- Exam Date:** 2023, March, 29

Buttons:

- Change Course** (blue)
- Add** (blue)
- Register** (blue)
- Display** (blue)
- Remove** (red)
- Clear** (red)

Message Box:

- Title:** Message
- Text:** The course is removed successfully.
- Button:** OK

Figure 23 (Test 3b (Non-Academic Course)) The values are entered in the text field with the same CourseID. (To remove the Non-Academic Course)

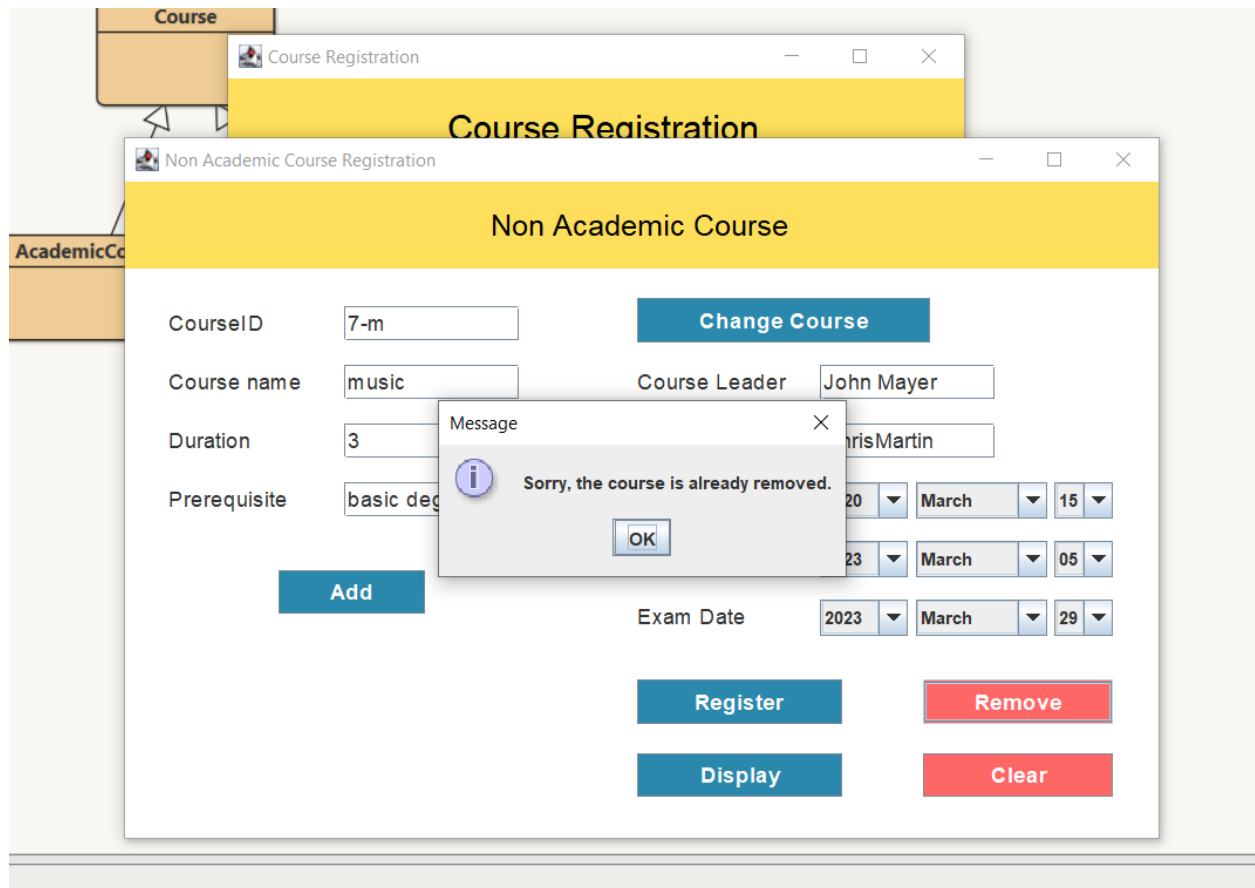


Figure 24 (Test 3b (Non-Academic Course)) The remove button is pressed. (To remove the Non-Academic Course which is already removed)

6. Error detection and correction

i. Syntax error

a) Error detection:

A syntax error is shown in the picture below. The syntax error occurred as the spelling of “JComboBox” was incorrect. The error was caused as “JCombox” was written in place of “JComboBox”.

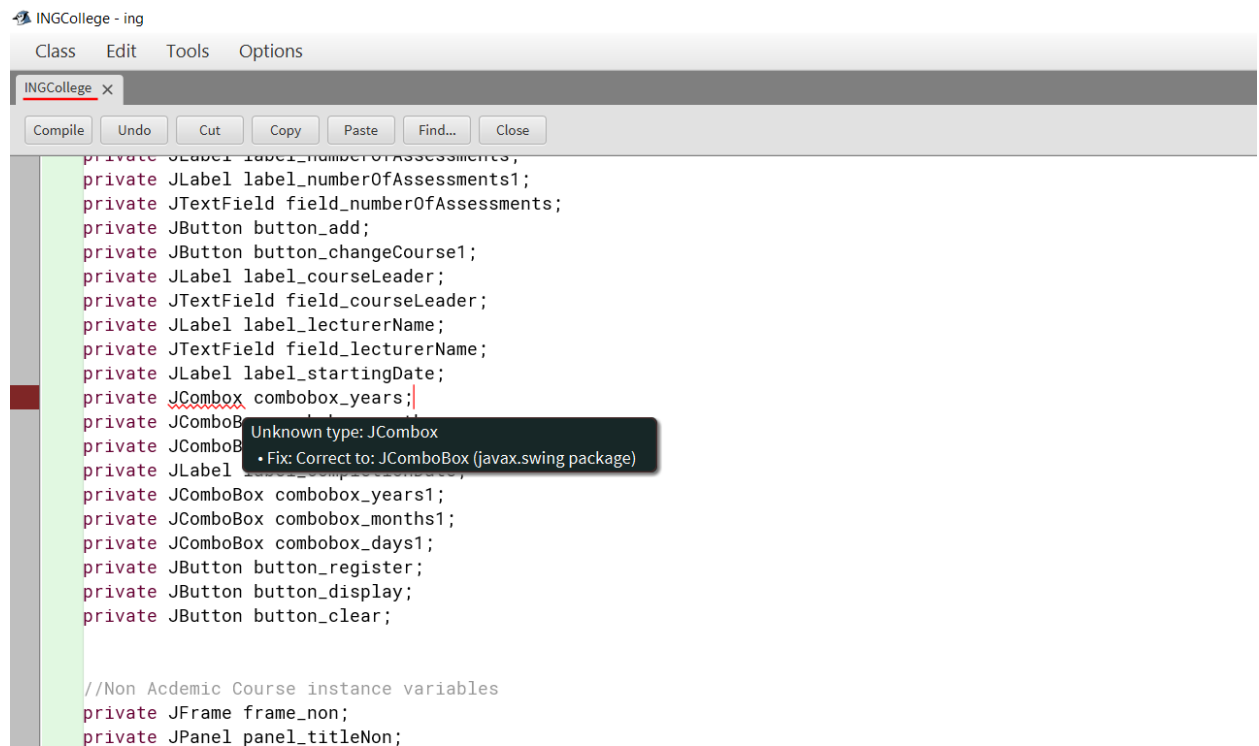
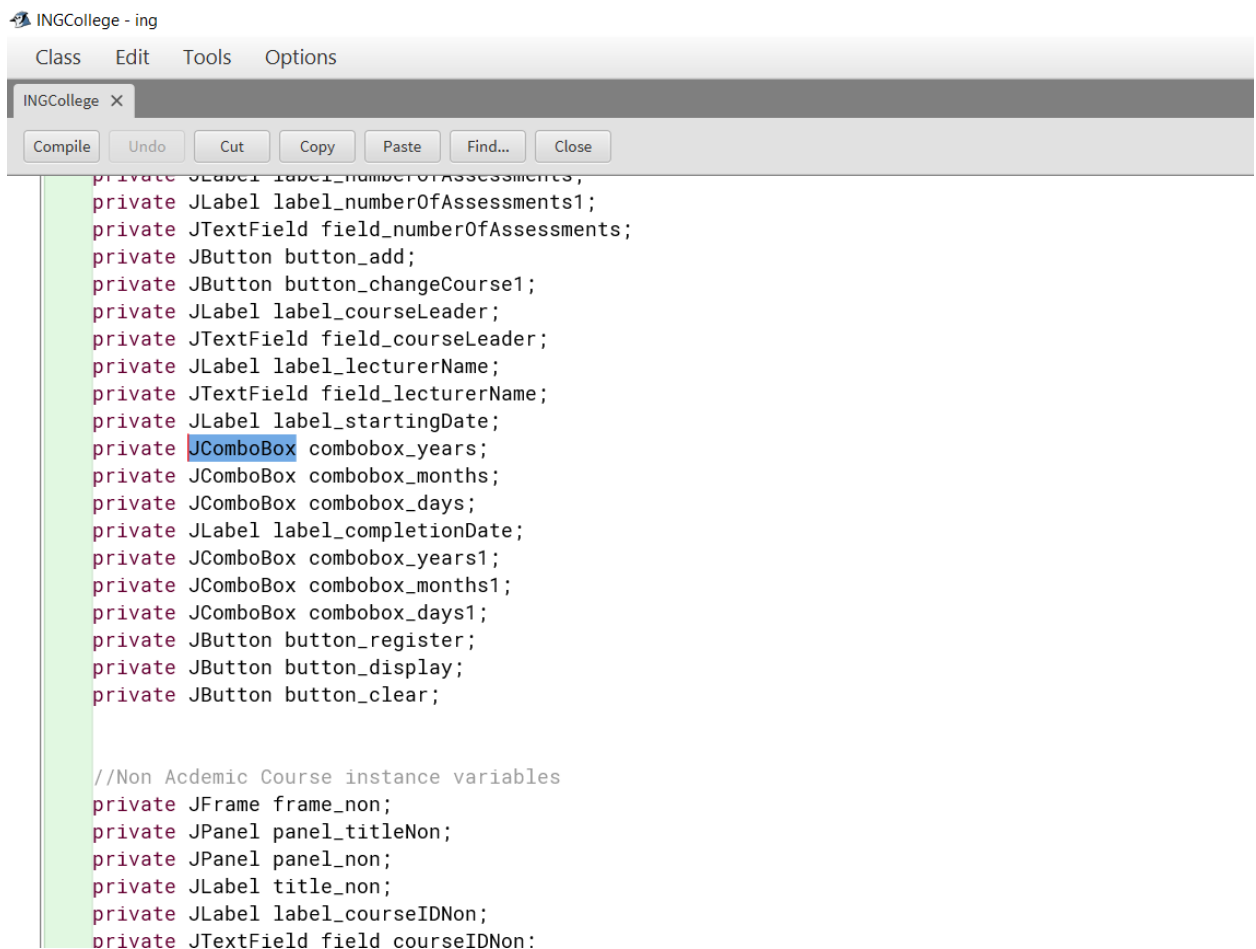


Figure 25 Syntax Error (Detection)

b) Error Correction:

The error was corrected by correcting the misspelled word (“JCombox”) to correct word (“JComboBox”).



```
INGCollege - ing
Class Edit Tools Options
INGCollege X
Compile Undo Cut Copy Paste Find... Close

private JLabel label_numberOfAssessments;
private JLabel label_numberOfAssessments1;
private JTextField field_numberOfAssessments;
private JButton button_add;
private JButton button_changeCourse1;
private JLabel label_courseLeader;
private JTextField field_courseLeader;
private JLabel label_lecturerName;
private JTextField field_lecturerName;
private JLabel label_startingDate;
private JComboBox combobox_years;
private JComboBox combobox_months;
private JComboBox combobox_days;
private JLabel label_completionDate;
private JComboBox combobox_years1;
private JComboBox combobox_months1;
private JComboBox combobox_days1;
private JButton button_register;
private JButton button_display;
private JButton button_clear;

//Non Academic Course instance variables
private JFrame frame_non;
private JPanel panel_titleNon;
private JPanel panel_non;
private JLabel title_non;
private JLabel label_courseIDNon;
private JTextField field courseIDNon;
```

Figure 26 Syntax Error (Correction)

ii. Semantic Error

a) Error detection:

The picture below shows a semantic error. The error occurred as variable “add_nonAcademicCourseID” was undeclared.

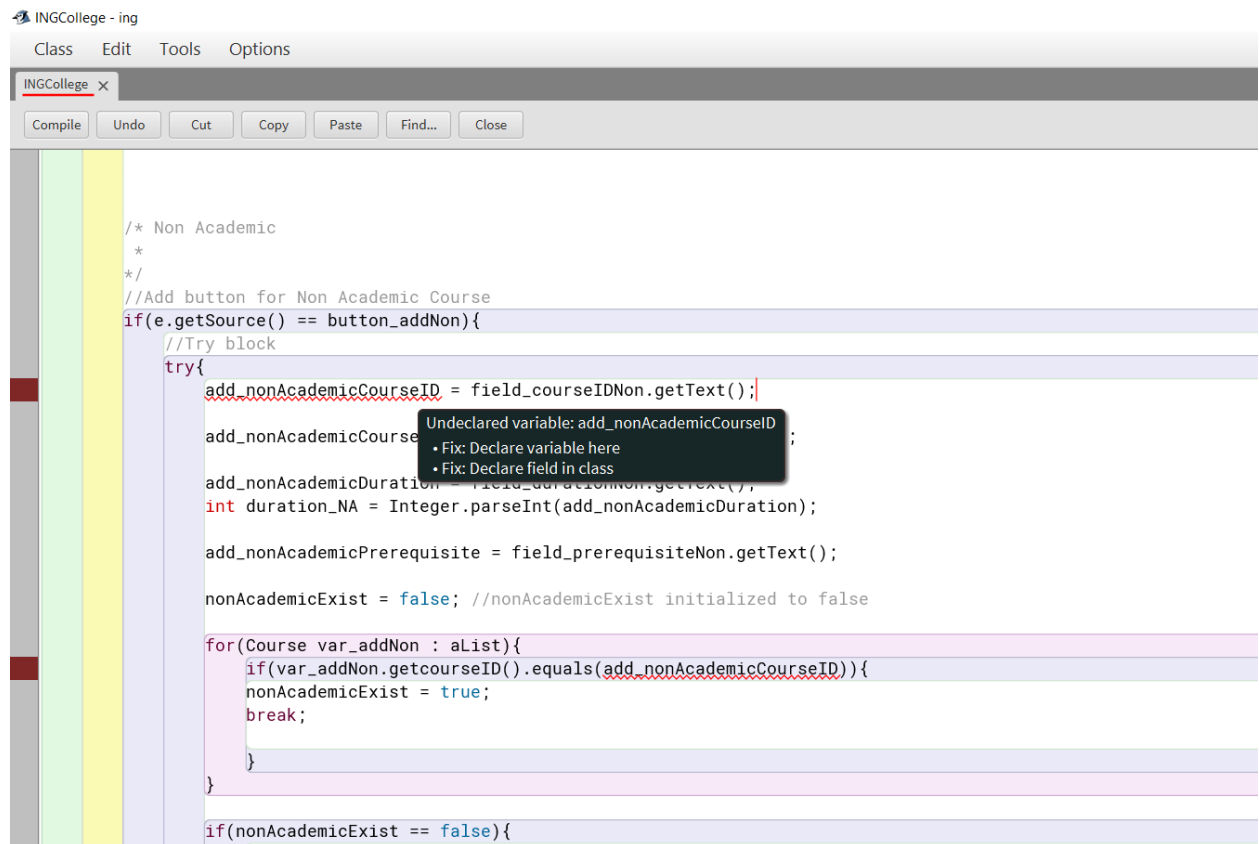
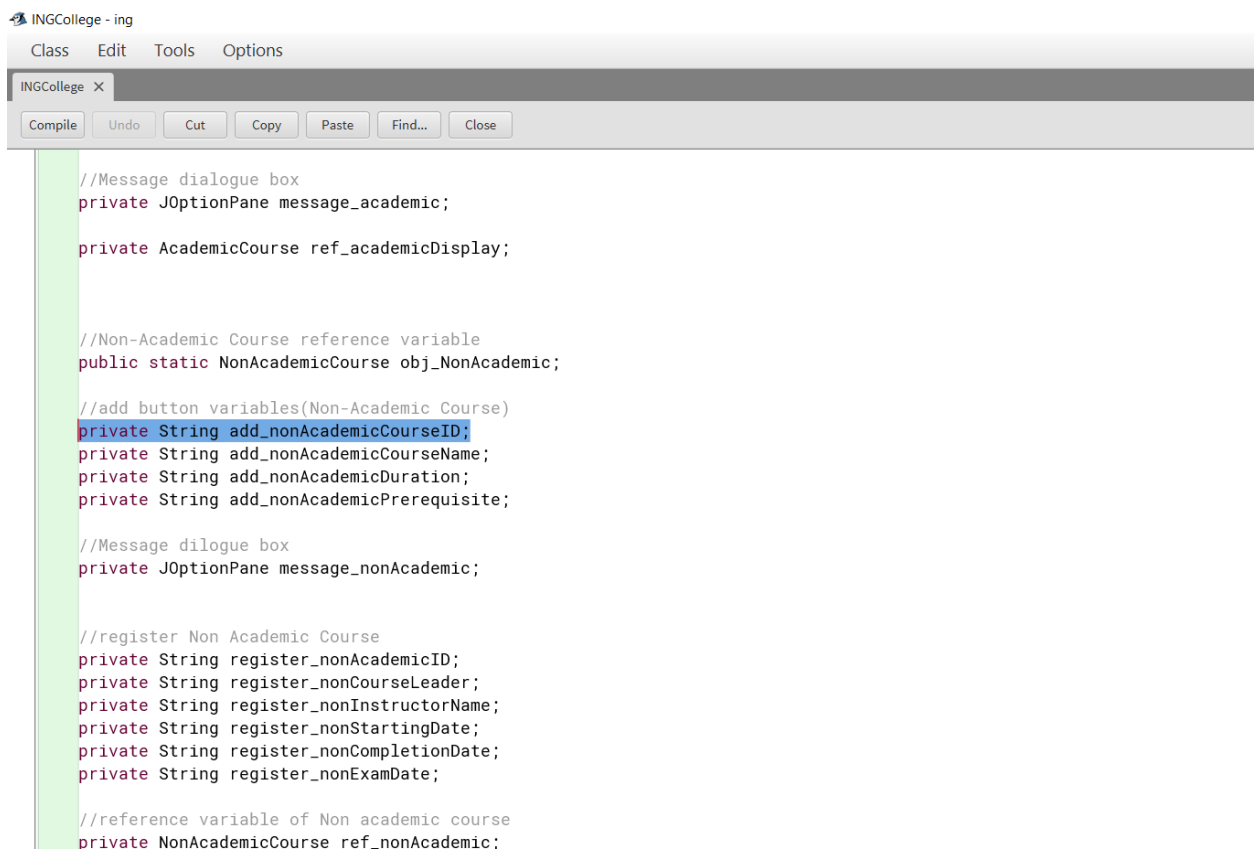


Figure 27 Semantic Error (Detection)

b) Error correction:

The error was corrected by declaring the variable “add_nonAcademicCourseID”.



```
//Message dialogue box
private JOptionPane message_academic;

private AcademicCourse ref_academicDisplay;

//Non-Academic Course reference variable
public static NonAcademicCourse obj_NonAcademic;

//add button variables(Non-Academic Course)
private String add_nonAcademicCourseID;
private String add_nonAcademicCourseName;
private String add_nonAcademicDuration;
private String add_nonAcademicPrerequisite;

//Message dialogue box
private JOptionPane message_nonAcademic;

//register Non Academic Course
private String register_nonAcademicID;
private String register_nonCourseLeader;
private String register_nonInstructorName;
private String register_nonStartingDate;
private String register_nonCompletionDate;
private String register_nonExamDate;

//reference variable of Non academic course
private NonAcademicCourse ref_nonAcademic;
```

Figure 28 Semantic Error (Correction)

iii. Logical Error

a) Error detection:

A logical error was detected in the picture below. The error occurred while checking the condition “`if(ref_nonAcademicRemove.getisRemoved() == true)`” and “`else if(ref_nonAcademicRemove.getisRemoved() == false)`”. This logical error caused unexpected output.

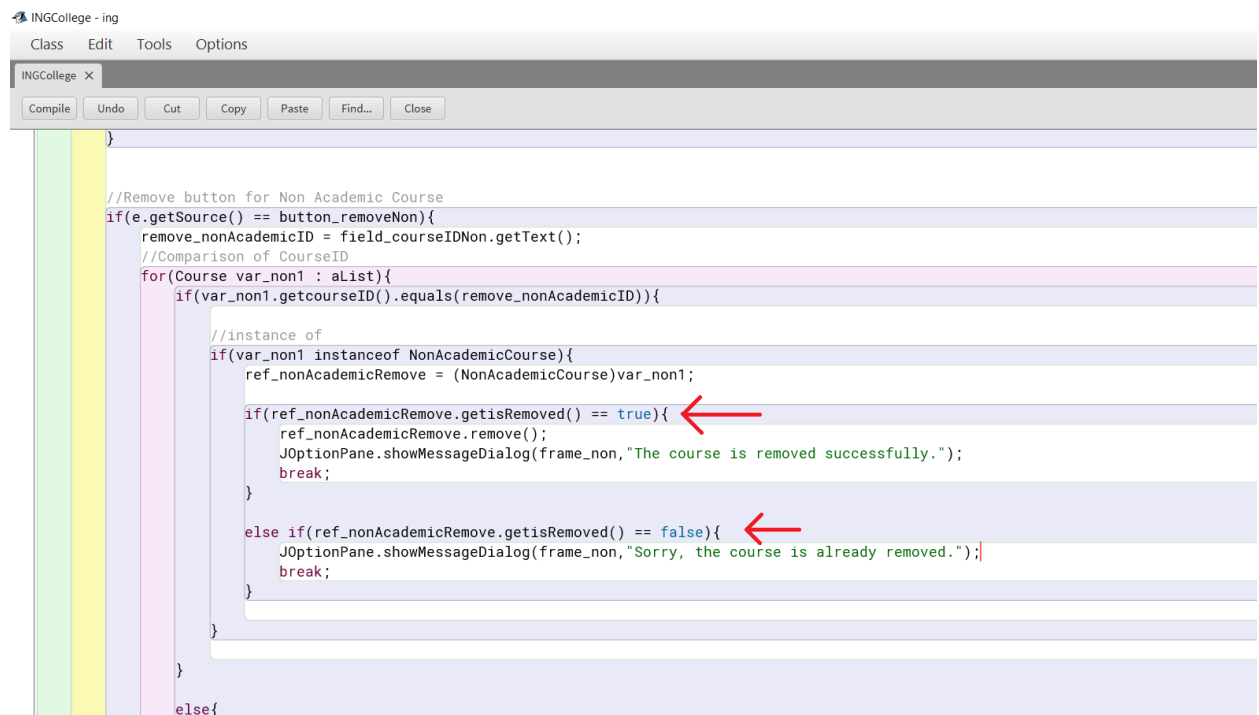


Figure 29 Logical Error (Detection)

b) Error correction:

The logical error was corrected by

changing the condition of `if(ref_nonAcademicRemove.getisRemoved() == true)` to

`if(ref_nonAcademicRemove.getisRemoved() == false)`

and

`else if(ref_nonAcademicRemove.getisRemoved() == false)` to

`else if(ref_nonAcademicRemove.getisRemoved() == true)`

```

//Remove button for Non Academic Course
if(e.getSource() == button_removeNon){
    remove_nonAcademicID = field_courseIDNon.getText();
    //Comparison of CourseID
    for(Course var_non1 : aList){
        if(var_non1.getcourseID().equals(remove_nonAcademicID)){

            //instance of
            if(var_non1 instanceof NonAcademicCourse){
                ref_nonAcademicRemove = (NonAcademicCourse)var_non1;

                if(ref_nonAcademicRemove.getisRemoved() == false){
                    ref_nonAcademicRemove.remove();
                    JOptionPane.showMessageDialog(frame_non,"The course is removed successfully.");
                    break;
                }

                else if(ref_nonAcademicRemove.getisRemoved() == true){
                    JOptionPane.showMessageDialog(frame_non,"Sorry, the course is already removed.");
                    break;
                }
            }
        }
    }
}
else{

```

Figure 30 Logical Error (Correction)

7. Conclusion:

The overall project was productive and instructive, providing us the knowledge on GUI. GUI was a new but very interesting and useful topic to learn about. The INGCollege class consisted of the GUI method. The project overall was interesting to code and even more interesting to use as GUI had made it more user friendly. In all of that, one of the major factors of the project was the project itself is real-world based. We use these interfaces all the time and I was immensely grateful to know about the functioning of these interfaces, to work with the source codes and design our form.

Working with a high-level programming language such as java has been really informative and fruitful in a theoretical and practical way. This coursework enlightened us about how the Java programming language can be used to make a graphical interface and how it is constrained and arranged. We acknowledged about all the various source codes utilized in making a GUI like JFrame, JPanel, JLabel and so on. Utilizing all the mentioned functionality of Swing and AWT to form the main enclosure of the structure and including the content fields and buttons was an interesting way to work with the aspects of coding and make the program autonomous. The concept of event handling was a very useful and productive thing to learn. Event handling was the core of the GUI as it decided what to perform when a certain event occurred.

There are always some difficulties we have to face doing a project. Setting the correct position of the JLabel, JTextField and other components was one of the difficult things to do. I also faced some difficulties putting the logic in the program and had to face logical errors a couple of times. I also faced some logical errors while condition checking where the Boolean value I kept was wrong.

The whole GUI concept being new there were some difficulties but with time and practice put into the project, the difficulties were outshined. I overcame the difficulties by doing some research on a particular topic or by taking guidelines from the teacher and putting it into practice.

Bibliography

- Anon., n.d. *diagrams.net*. [Online]
Available at: <https://www.diagrams.net/>
[Accessed 18 August 2021].
- Beal, V., 2021. *webopedia*. [Online]
Available at: <https://www.webopedia.com/definitions/java/>
[Accessed 17 August 2021].
- Christensson, P., 2006. *TechTerms*. [Online]
Available at: <https://techterms.com/definition/gui>
[Accessed 17 August 2021].

Appendix 1:

INGCollege

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
import java.util.*;
```



```
public class INGCollege implements ActionListener  
{
```

```
    private JFrame frame_main;  
    private JPanel panel_main;  
    private JPanel panel_titleMain;  
    private JLabel titleMain;  
    private JLabel label_info;  
    private JButton button_mainAcademic;  
    private JButton button_mainNonAcademic;
```

```
    private JFrame frame_academic;  
    private JPanel panel_titleAcademic;  
    private JPanel panel_academic;  
    private JLabel title_academic;  
    private JLabel label_courseID;  
    private JTextField field_courseID;  
    private JLabel label_courseName;  
    private JTextField field_courseName;  
    private JLabel label_duration;  
    private JTextField field_duration;  
    private JLabel label_level;  
    private JTextField field_level;  
    private JLabel label_credit;  
    private JTextField field_credit;  
    private JLabel label_numberOfAssessments;  
    private JLabel label_numberOfAssessments1;
```

```
private JTextField field_numberOfAssessments;
private JButton button_add;
private JButton button_changeCourse1;
private JLabel label_courseLeader;
private JTextField field_courseLeader;
private JLabel label_lecturerName;
private JTextField field_lecturerName;
private JLabel label_startingDate;
private JComboBox combobox_years;
private JComboBox combobox_months;
private JComboBox combobox_days;
private JLabel label_completionDate;
private JComboBox combobox_years1;
private JComboBox combobox_months1;
private JComboBox combobox_days1;
private JButton button_register;
private JButton button_display;
private JButton button_clear;

private JFrame frame_non;
private JPanel panel_titleNon;
private JPanel panel_non;
private JLabel title_non;
private JLabel label_courseIDNon;
private JTextField field_courseIDNon;
private JLabel label_courseNameNon;
private JTextField field_courseNameNon;
```

```
private JLabel label_durationNon;  
private JTextField field_durationNon;  
private JLabel label_prerequisiteNon;  
private JTextField field_prerequisiteNon;  
private JButton button_addNon;  
private JButton button_changeCourse2;  
private JLabel label_courseLeaderNon;  
private JTextField field_courseLeaderNon;  
private JLabel label_instructorNameNon;  
private JTextField field_instructorNameNon;  
private JLabel label_startingDateNon;  
private JComboBox combobox_yearsNon;  
private JComboBox combobox_monthsNon;  
private JComboBox combobox_daysNon;  
private JLabel label_completionDateNon;  
private JComboBox combobox_yearsNon1;  
private JComboBox combobox_monthsNon1;  
private JComboBox combobox_daysNon1;  
private JLabel label_examDateNon;  
private JComboBox combobox_yearsNon2;  
private JComboBox combobox_monthsNon2;  
private JComboBox combobox_daysNon2;  
private JButton button_displayNon;  
private JButton button_clearNon;  
private JButton button_registerNon;  
private JButton button_removeNon;
```

```
public static INGCollege obj_ing = new INGCollege();
```

```
public static AcademicCourse obj_Academic;
```

```
public ArrayList<Course> aList = new ArrayList<Course>();
```

```
private String add_academicID;
```

```
private String add_academicName;
```

```
private String add_academicDuration;
```

```
private String add_academicLevel;
```

```
private String add_academicCredit;
```

```
private String add_numberOfAssessments;
```

```
private String register_academicID;
```

```
private String register_courseLeader;
```

```
private String register_lecturerName;
```

```
private String register_startingDate;
```

```
private String register_completionDate;
```

```
private boolean academicExist;
```

```
private AcademicCourse ref_academic;
```

```
private JOptionPane message_academic;
```

```
private AcademicCourse ref_academicDisplay;
```

```
public static NonAcademicCourse obj_NonAcademic;
```

```
private String add_nonAcademicCourseID;
```

```
private String add_nonAcademicCourseName;
```

```
private String add_nonAcademicDuration;
```

```
private String add_nonAcademicPrerequisite;
```

```
private JOptionPane message_nonAcademic;
```

```
private String register_nonAcademicID;
```

```
private String register_nonCourseLeader;
```

```
private String register_nonInstructorName;
```

```
private String register_nonStartingDate;
```

```
private String register_nonCompletionDate;
```

```
private String register_nonExamDate;
```

```
private NonAcademicCourse ref_nonAcademic;
```

```
private boolean nonAcademicExist;
```

```
private String remove_nonAcademicID;  
  
private NonAcademicCourse ref_nonAcademicRemove;  
  
private NonAcademicCourse ref_nonAcademicDisplay;
```

```
public void mainGUI()  
{  
  
    frame_main = new JFrame();  
    frame_main.setTitle("Course Registration");  
    frame_main.setVisible(true);  
    frame_main.setResizable(true);  
    frame_main.setBounds(520,130,517,445);  
  
  
    panel_main = new JPanel();  
    panel_main.setLayout(null);  
    panel_main.setBackground(Color.white);  
    panel_main.setBounds(0,0,503,430);
```

```
panel_titleMain = new JPanel();  
panel_titleMain.setLayout(null);  
panel_titleMain.setBounds(0,0,503,70);
```

```
Color color_titleMain = new Color(253,223,91);  
panel_titleMain.setBackground(color_titleMain);
```

```
titleMain = new JLabel();  
titleMain.setText("Course Registration");  
titleMain.setBounds(150,20,250,30);  
titleMain.setForeground(Color.black);  
Font fontMain = new Font("Arial",Font.PLAIN,25);  
titleMain.setFont(fontMain);
```

```
panel_titleMain.add(titleMain);
```

```
label_info = new JLabel();
```

```
label_info.setText("Select on the button to register the course");
label_info.setBounds(60,100,450,25);
Font font_info = new Font("Arial",Font.PLAIN,20);
label_info.setFont(font_info);

panel_main.add(label_info);
```

```
button_mainAcademic = new JButton();
button_mainAcademic.setText("Academic Course");
button_mainAcademic.setBounds(30,175,200,30);

Font font_mainAcademic = new Font("Arial",Font.BOLD,15);
button_mainAcademic.setFont(font_mainAcademic);
Color color_mainAcademic = new Color(42,136,173);
button_mainAcademic.setBackground(color_mainAcademic);
button_mainAcademic.setForeground(Color.white);

button_mainAcademic.addActionListener(this);

panel_main.add(button_mainAcademic);
```



```
button_mainNonAcademic = new JButton();
button_mainNonAcademic.setText("Non Academic Course");
button_mainNonAcademic.setBounds(270,175,200,30);
Font font_mainNonAcademic = new Font("Arial",Font.BOLD,15);
button_mainNonAcademic.setFont(font_mainNonAcademic);
Color color_mainNonAcademic = new Color(42,136,173);
button_mainNonAcademic.setBackground(color_mainNonAcademic);
button_mainNonAcademic.setForeground(Color.white);

button_mainNonAcademic.addActionListener(this);

panel_main.add(button_mainNonAcademic);


frame_main.add(panel_titleMain);
frame_main.add(panel_main);

}

public void academicCourseGUI()
{
```

```
frame_academic = new JFrame();  
frame_academic.setTitle("Academic Course Registration");  
frame_academic.setVisible(true);  
frame_academic.setResizable(true);  
frame_academic.setBounds(405,165,720,445);
```

```
panel_academic = new JPanel();  
panel_academic.setLayout(null);  
panel_academic.setBackground(Color.white);  
panel_academic.setBounds(0,0,710,430);
```

```
panel_titleAcademic = new JPanel();  
panel_titleAcademic.setLayout(null);  
panel_titleAcademic.setBounds(0,0,710,60);
```

```
Color color_titleAcademic = new Color(253,223,91);  
panel_titleAcademic.setBackground(color_titleAcademic);
```

```
title_academic = new JLabel();
```

```
title_academic.setText("Academic Course");
title_academic.setBounds(275,15,250,30);
title_academic.setForeground(Color.black);

Font font_academic = new Font("Arial",Font.PLAIN,20);
title_academic.setFont(font_academic);

panel_titleAcademic.add(title_academic);


label_courseID = new JLabel();
label_courseID.setText("CourseID");
label_courseID.setBounds(30,85,150,25);
Font font_one = new Font("Arial",Font.PLAIN,15);
label_courseID.setFont(font_one);

panel_academic.add(label_courseID);


field_courseID = new JTextField();
field_courseID.setBounds(150,85,120,25);
field_courseID.setFont(font_one);

panel_academic.add(field_courseID);
```

```
label_courseName = new JLabel();  
label_courseName.setText("Course name");  
label_courseName.setBounds(30,125,120,25);  
Font font_two = new Font("Arial",Font.PLAIN,15);  
label_courseName.setFont(font_two);
```

```
panel_academic.add(label_courseName);
```

```
field_courseName = new JTextField();  
field_courseName.setBounds(150,125,120,25);  
field_courseName.setFont(font_two);
```

```
panel_academic.add(field_courseName);
```

```
label_duration = new JLabel();  
label_duration.setText("Duration");  
label_duration.setBounds(30,165,120,25);
```

```
Font font_three = new Font("Arial",Font.PLAIN,15);  
label_duration.setFont(font_three);
```

```
panel_academic.add(label_duration);
```

```
field_duration = new JTextField();  
field_duration.setBounds(150,165,120,25);  
field_duration.setFont(font_three);
```

```
panel_academic.add(field_duration);
```

```
label_level = new JLabel();  
label_level.setText("Level");  
label_level.setBounds(30,205,120,25);
```

```
Font font_four = new Font("Arial",Font.PLAIN,15);  
label_level.setFont(font_four);
```

```
panel_academic.add(label_level);
```

```
field_level = new JTextField();  
field_level.setBounds(150,205,120,25);  
field_level.setFont(font_four);
```

```
panel_academic.add(field_level);
```

```
label_credit = new JLabel();
label_credit.setText("Credit");
label_credit.setBounds(30,245,120,25);

Font font_five = new Font("Arial",Font.PLAIN,15);
label_credit.setFont(font_five);

panel_academic.add(label_credit);

field_credit = new JTextField();
field_credit.setBounds(150,245,120,25);
field_credit.setFont(font_five);

panel_academic.add(field_credit);


label_numberOfAssessments = new JLabel();
label_numberOfAssessments.setText("Number of");
label_numberOfAssessments.setBounds(30,285,190,25);

Font font_six = new Font("Arial",Font.PLAIN,15);
label_numberOfAssessments.setFont(font_six);
```

```
panel_academic.add(label_numberOfAssessments);
```

```
label_numberOfAssessments1= new JLabel();  
label_numberOfAssessments1.setText("assessments");  
label_numberOfAssessments1.setBounds(30,300,190,25);  
label_numberOfAssessments1.setFont(font_six);
```

```
panel_academic.add(label_numberOfAssessments1);
```

```
field_numberOfAssessments = new JTextField();  
field_numberOfAssessments.setBounds(150,290,120,25);  
field_numberOfAssessments.setFont(font_six);
```

```
panel_academic.add(field_numberOfAssessments);
```

```
button_add = new JButton();  
button_add.setText("Add");  
button_add.setBounds(105,350,100,30);
```

```
Font font_seven = new Font("Arial",Font.BOLD,15);  
button_add.setFont(font_seven);  
Color color_add = new Color(42,136,173);  
button_add.setBackground(color_add);  
button_add.setForeground(Color.white);
```

```
button_add.addActionListener(this);
```

```
panel_academic.add(button_add);
```

```
button_changeCourse1 = new JButton();
```

```
button_changeCourse1.setText("Change Course");
```

```
button_changeCourse1.setBounds(350,80,200,30);
```

```
Font font_eight = new Font("Arial",Font.BOLD,15);
```

```
button_changeCourse1.setFont(font_eight);
```

```
Color color_changeCourse1 = new Color(42,136,173);
```

```
button_changeCourse1.setBackground(color_changeCourse1);
```

```
button_changeCourse1.setForeground(Color.white);
```

```
button_changeCourse1.addActionListener(this);
```

```
panel_academic.add(button_changeCourse1);
```

```
label_courseLeader = new JLabel();
```



```
label_courseLeader.setText("Course Leader");  
label_courseLeader.setBounds(350,125,150,25);  
Font font_nine = new Font("Arial",Font.PLAIN,15);  
label_courseLeader.setFont(font_nine);
```

```
panel_academic.add(label_courseLeader);
```

```
field_courseLeader = new JTextField();  
field_courseLeader.setBounds(475,125,120,25);  
field_courseLeader.setFont(font_nine);
```

```
panel_academic.add(field_courseLeader);
```

```
label_lecturerName = new JLabel();  
label_lecturerName.setText("Lecturer Name");  
label_lecturerName.setBounds(350,165,150,25);
```

```
Font font_ten = new Font("Arial",Font.PLAIN,15);  
label_lecturerName.setFont(font_ten);
```

```
panel_academic.add(label_lecturerName);
```

```
field_lecturerName = new JTextField();  
field_lecturerName.setBounds(475,165,120,25);  
field_lecturerName.setFont(font_ten);
```

```
panel_academic.add(field_lecturerName);
```

```
label_startingDate = new JLabel();  
label_startingDate.setText("Starting Date");  
label_startingDate.setBounds(350,205,150,25);
```

```
Font font_eleven = new Font("Arial",Font.PLAIN,15);  
label_startingDate.setFont(font_eleven);
```

```
panel_academic.add(label_startingDate);
```

```
String years[] = {"2020","2021","2022","2023"};  
combobox_years = new JComboBox(years);  
combobox_years.setBounds(475,205,60,25);
```

```
panel_academic.add(combobox_years);
```

```
String months[] =  
{ "January", "February", "March", "April", "May", "June", "July", "August", "September", "October",  
  "November", "December"};
```

```
combobox_months = new JComboBox(months);
```

```
combobox_months.setBounds(540,205,90,25);
```

```
panel_academic.add(combobox_months);
```

```
String days[] =  
{ "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15", "16", "17", "18",  
  "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31"};
```

```
combobox_days = new JComboBox(days);
```

```
combobox_days.setBounds(635,205,40,25);
```

```
panel_academic.add(combobox_days);
```

```
label_completionDate = new JLabel();
```

```
label_completionDate.setText("Completion Date");
```

```
label_completionDate.setBounds(350,245,150,25);
```

```
Font font_twelve = new Font("Arial",Font.PLAIN,15);  
label_completionDate.setFont(font_twelve);
```

```
panel_academic.add(label_completionDate);
```

```
String years1[] = {"2023","2024","2025","2026","2027","2028","2029","2030"};  
combobox_years1 = new JComboBox(years1);  
combobox_years1.setBounds(475,245,60,25);
```

```
panel_academic.add(combobox_years1);
```

```
String months1[] =  
{ "January","February","March","April","May","June","July","August","September","October",  
  "November","December"};
```

```
combobox_months1 = new JComboBox(months1);  
combobox_months1.setBounds(540,245,90,25);
```

```
panel_academic.add(combobox_months1);
```

```
String days1[] =  
{ "01","02","03","04","05","06","07","08","09","10","11","12","13","14","15","16","17","18","19",  
  "20","21","22","23","24","25","26","27","28","29","30","31"};
```

```
combobox_days1 = new JComboBox(days1);  
combobox_days1.setBounds(635,245,40,25);
```

```
panel_academic.add(combobox_days1);
```

```
button_register = new JButton();  
button_register.setText("Register");  
button_register.setBounds(430,300,175,30);
```

```
Font font_thirteen = new Font("Arial",Font.BOLD,15);  
button_register.setFont(font_thirteen);
```

```
Color color_register1 = new Color(42,136,173);  
button_register.setBackground(color_register1);  
button_register.setForeground(Color.white);
```

```
button_register.addActionListener(this);
```

```
panel_academic.add(button_register);
```

```
button_display = new JButton();
```

```
button_display.setText("Display");
button_display.setBounds(350,350,140,30);

Font font_fourteen = new Font("Arial",Font.BOLD,15);
button_display.setFont(font_fourteen);

Color color_display1 = new Color(42,136,173);
button_display.setBackground(color_display1);
button_display.setForeground(Color.white);

button_display.addActionListener(this);

panel_academic.add(button_display);


button_clear = new JButton();
button_clear.setText("Clear");
button_clear.setBounds(545,350,130,30);

Font font_fifteen = new Font("Arial",Font.BOLD,15);
button_clear.setFont(font_fifteen);

Color color_clear1 = new Color(255,102,102);
button_clear.setBackground(color_clear1);
```

```
button_clear.setForeground(Color.white);
```

```
button_clear.addActionListener(this);
```

```
panel_academic.add(button_clear);
```

```
frame_academic.add(panel_titleAcademic);
```

```
frame_academic.add(panel_academic);
```

```
}
```

```
public void nonAcademicCourseGUI()
```

```
{
```

```
frame_non = new JFrame();
```

```
frame_non.setTitle("Non Academic Course Registration");  
frame_non.setVisible(true);  
frame_non.setResizable(true);  
frame_non.setBounds(450,200,720,485);
```

```
panel_non = new JPanel();  
panel_non.setLayout(null);  
panel_non.setBackground(Color.white);  
panel_non.setBounds(0,0,710,400);
```

```
panel_titleNon = new JPanel();  
panel_titleNon.setLayout(null);  
panel_titleNon.setBounds(0,0,710,60);
```

```
Color color_titleNon = new Color(253,223,91);  
panel_titleNon.setBackground(color_titleNon);
```

```
title_non = new JLabel();  
title_non.setText("Non Academic Course");  
title_non.setBounds(250,15,250,30);
```



```
title_non.setForeground(Color.black);
```

```
Font font_1 = new Font("Arial",Font.PLAIN,20);
```

```
title_non.setFont(font_1);
```

```
panel_titleNon.add(title_non);
```

```
label_courseIDNon = new JLabel();
```

```
label_courseIDNon.setText("CourseID");
```

```
label_courseIDNon.setBounds(30,85,150,25);
```

```
Font font_2 = new Font("Arial",Font.PLAIN,15);
```

```
label_courseIDNon.setFont(font_2);
```

```
panel_non.add(label_courseIDNon);
```

```
field_courseIDNon = new JTextField();
```

```
field_courseIDNon.setBounds(150,85,120,25);
```

```
field_courseIDNon.setFont(font_2);
```

```
panel_non.add(field_courseIDNon);
```

```
label_courseNameNon = new JLabel();  
label_courseNameNon.setText("Course name");  
label_courseNameNon.setBounds(30,125,120,25);
```

```
Font font_3 = new Font("Arial",Font.PLAIN,15);  
label_courseNameNon.setFont(font_3);
```

```
panel_non.add(label_courseNameNon);
```

```
field_courseNameNon = new JTextField();  
field_courseNameNon.setBounds(150,125,120,25);  
field_courseNameNon.setFont(font_3);
```

```
panel_non.add(field_courseNameNon);
```

```
label_durationNon = new JLabel();
```

```
label_durationNon.setText("Duration");
label_durationNon.setBounds(30,165,120,25);

Font font_4 = new Font("Arial",Font.PLAIN,15);
label_durationNon.setFont(font_4);

panel_non.add(label_durationNon);

field_durationNon = new JTextField();
field_durationNon.setBounds(150,165,120,25);
field_durationNon.setFont(font_4);

panel_non.add(field_durationNon);

label_prerequisiteNon = new JLabel();
label_prerequisiteNon.setText("Prerequisite");
label_prerequisiteNon.setBounds(30,205,120,25);

Font font_5 = new Font("Arial",Font.PLAIN,15);
label_prerequisiteNon.setFont(font_5);
```

```
panel_non.add(label_prerequisiteNon);
```

```
field_prerequisiteNon = new JTextField();  
field_prerequisiteNon.setBounds(150,205,120,25);  
field_prerequisiteNon.setFont(font_5);
```

```
panel_non.add(field_prerequisiteNon);
```

```
button_addNon = new JButton();  
button_addNon.setText("Add");  
button_addNon.setBounds(105,265,100,30);
```

```
Font font_6 = new Font("Arial",Font.BOLD,15);  
button_addNon.setFont(font_6);
```

```
Color color_addNon = new Color(42,136,173);  
button_addNon.setBackground(color_addNon);  
button_addNon.setForeground(Color.white);
```

```
button_addNon.addActionListener(this);
```

```
panel_non.add(button_addNon);
```

```
button_changeCourse2 = new JButton();  
button_changeCourse2.setText("Change Course");  
button_changeCourse2.setBounds(350,80,200,30);
```

```
Font font_7 = new Font("Arial",Font.BOLD,15);  
button_changeCourse2.setFont(font_7);
```

```
Color color_changeCourse2 = new Color(42,136,173);  
button_changeCourse2.setBackground(color_changeCourse2);  
button_changeCourse2.setForeground(Color.white);
```

```
button_changeCourse2.addActionListener(this);
```

```
panel_non.add(button_changeCourse2);
```

```
label_courseLeaderNon = new JLabel();  
label_courseLeaderNon.setText("Course Leader");
```

```
label_courseLeaderNon.setBounds(350,125,150,25);
```

```
Font font_8 = new Font("Arial",Font.PLAIN,15);
```

```
label_courseLeaderNon.setFont(font_8);
```

```
panel_non.add(label_courseLeaderNon);
```

```
field_courseLeaderNon = new JTextField();
```

```
field_courseLeaderNon.setBounds(475,125,120,25);
```

```
field_courseLeaderNon.setFont(font_8);
```

```
panel_non.add(field_courseLeaderNon);
```

```
label_instructorNameNon = new JLabel();
```

```
label_instructorNameNon.setText("Instructor Name");
```

```
label_instructorNameNon.setBounds(350,165,150,25);
```

```
Font font_9 = new Font("Arial",Font.PLAIN,15);
```

```
label_instructorNameNon.setFont(font_9);
```

```
panel_non.add(label_instructorNameNon);
```

```
field_instructorNameNon = new JTextField();  
field_instructorNameNon.setBounds(475,165,120,25);  
field_instructorNameNon.setFont(font_9);
```

```
panel_non.add(field_instructorNameNon);
```

```
label_startingDateNon = new JLabel();  
label_startingDateNon.setText("Starting Date");  
label_startingDateNon.setBounds(350,205,150,25);
```

```
Font font_10 = new Font("Arial",Font.PLAIN,15);  
label_startingDateNon.setFont(font_10);
```

```
panel_non.add(label_startingDateNon);
```

```
String yearsNon[] = {"2020","2021","2022","2023"};  
combobox_yearsNon = new JComboBox(yearsNon);  
combobox_yearsNon.setBounds(475,205,60,25);
```

```
panel_non.add(combobox_yearsNon);
```

```
String monthsNon[] =  
{ "January", "February", "March", "April", "May", "June", "July", "August", "September", "October",  
  "November", "December" };
```

```
combobox_monthsNon = new JComboBox(monthsNon);
```

```
combobox_monthsNon.setBounds(540,205,90,25);
```

```
panel_non.add(combobox_monthsNon);
```

```
String daysNon[] =  
{ "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15", "16", "17", "18",  
  "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31" };
```

```
combobox_daysNon = new JComboBox(daysNon);
```

```
combobox_daysNon.setBounds(635,205,40,25);
```

```
panel_non.add(combobox_daysNon);
```

```
label_completionDateNon = new JLabel();
```

```
label_completionDateNon.setText("Completion Date");
```

```
label_completionDateNon.setBounds(350,245,150,25);
```



```
Font font_11 = new Font("Arial",Font.PLAIN,15);  
label_completionDateNon.setFont(font_11);
```

```
panel_non.add(label_completionDateNon);
```

```
String yearsNon1[] =  
{ "2023", "2024", "2025", "2026", "2027", "2028", "2029", "2030" };  
combobox_yearsNon1 = new JComboBox(yearsNon1);  
combobox_yearsNon1.setBounds(475,245,60,25);
```

```
panel_non.add(combobox_yearsNon1);
```

```
String monthsNon1[] =  
{ "January", "February", "March", "April", "May", "June", "July", "August", "September", "October",  
  "November", "December" };  
combobox_monthsNon1 = new JComboBox(monthsNon1);  
combobox_monthsNon1.setBounds(540,245,90,25);
```

```
panel_non.add(combobox_monthsNon1);
```

```
String daysNon1[] =
{"01","02","03","04","05","06","07","08","09","10","11","12","13","14","15","16","17","18","
19","20","21","22","23","24","25","26","27","28","29","30","31"};

combobox_daysNon1 = new JComboBox(daysNon1);
combobox_daysNon1.setBounds(635,245,40,25);

panel_non.add(combobox_daysNon1);
```

```
label_examDateNon = new JLabel();
label_examDateNon.setText("Exam Date");
label_examDateNon.setBounds(350,285,150,25);
```

```
Font font_12 = new Font("Arial",Font.PLAIN,15);
label_examDateNon.setFont(font_12);
```

```
panel_non.add(label_examDateNon);
```

```
String yearsNon2[] =
{"2023","2024","2025","2026","2027","2028","2029","2030"};

combobox_yearsNon2 = new JComboBox(yearsNon2);
combobox_yearsNon2.setBounds(475,285,60,25);
```

```
panel_non.add(combobox_yearsNon2);
```

```
String monthsNon2[] =  
{ "January", "February", "March", "April", "May", "June", "July", "August", "September", "October",  
  "November", "December"};
```

```
combobox_monthsNon2 = new JComboBox(monthsNon2);
```

```
combobox_monthsNon2.setBounds(540,285,90,25);
```

```
panel_non.add(combobox_monthsNon2);
```

```
String daysNon2[] =  
{ "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15", "16", "17", "18",  
  "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31"};
```

```
combobox_daysNon2 = new JComboBox(daysNon2);
```

```
combobox_daysNon2.setBounds(635,285,40,25);
```

```
panel_non.add(combobox_daysNon2);
```

```
button_registerNon = new JButton();
```

```
button_registerNon.setText("Register");
```

```
button_registerNon.setBounds(350,340,140,30);
```

```
Font font_13 = new Font("Arial",Font.BOLD,15);  
button_registerNon.setFont(font_13);
```

```
Color color_register2 = new Color(42,136,173);  
button_registerNon.setBackground(color_register2);  
button_registerNon.setForeground(Color.white);
```

```
button_registerNon.addActionListener(this);
```

```
panel_non.add(button_registerNon);
```

```
button_removeNon = new JButton();  
button_removeNon.setText("Remove");  
button_removeNon.setBounds(545,340,130,30);
```

```
Font font_14 = new Font("Arial",Font.BOLD,15);  
button_removeNon.setFont(font_14);
```

```
Color color_remove2 = new Color(255,102,102);  
button_removeNon.setBackground(color_remove2);  
button_removeNon.setForeground(Color.white);
```

```
button_removeNon.addActionListener(this);
```

```
panel_non.add(button_removeNon);
```

```
button_displayNon = new JButton();
```

```
button_displayNon.setText("Display");
```

```
button_displayNon.setBounds(350,390,140,30);
```

```
Font font_15 = new Font("Arial",Font.BOLD,15);
```

```
button_displayNon.setFont(font_15);
```

```
Color color_display2 = new Color(42,136,173);
```

```
button_displayNon.setBackground(color_display2);
```

```
button_displayNon.setForeground(Color.white);
```

```
button_displayNon.addActionListener(this);
```

```
panel_non.add(button_displayNon);
```

```

button_clearNon = new JButton();
button_clearNon.setText("Clear");
button_clearNon.setBounds(545,390,130,30);

Font font_16 = new Font("Arial",Font.BOLD,15);
button_clearNon.setFont(font_16);

Color color_clear2 = new Color(255,102,102);
button_clearNon.setBackground(color_clear2);
button_clearNon.setForeground(Color.white);

button_clearNon.addActionListener(this);

panel_non.add(button_clearNon);


frame_non.add(panel_titleNon);
frame_non.add(panel_non);

}

public static void main(String[]args)

```

```
{  
    obj_ing.mainGUI();  
}
```

```
public void actionPerformed(ActionEvent e)  
{  
  
    if(e.getSource() == button_mainAcademic){  
        academicCourseGUI();  
    }  
  
    else if(e.getSource() == button_mainNonAcademic){  
        nonAcademicCourseGUI();  
    }  
  
  
    if(e.getSource() == button_changeCourse1){  
        nonAcademicCourseGUI();  
  
        frame_academic.dispose();  
    }  
  
    else if(e.getSource() == button_changeCourse2){  
        academicCourseGUI();  
    }  
}
```

```
frame_non.dispose();  
}
```

```
if(e.getSource() == button_add){  
  
    try{  
  
        add_academicID = field_courseID.getText();  
  
        add_academicName = field_courseName.getText();  
  
        add_academicDuration = field_duration.getText();  
        int duration_A = Integer.parseInt(add_academicDuration);  
  
        add_academicLevel = field_level.getText();  
  
        add_academicCredit = field_credit.getText();  
  
        add_numberOfAssessments = field_numberOfAssessments.getText();  
        int numberOfAssessments_A =  
Integer.parseInt(add_numberOfAssessments);  
  
        academicExist = false;  
  
        for(Course var_add : aList){
```



```

        if(var_add.getCourseID().equals(add_academicID)){
            academicExist = true;
            break;
        }
    }

    if(academicExist == false){

        obj_Academic = new AcademicCourse(add_academicID,
        add_academicName, duration_A,add_academicLevel, add_academicCredit,
        numberOfAssessments_A);

        aList.add(obj_Academic);

        message_academic = new JOptionPane();

        String final_msgAcademic = "Course ID: " + add_academicID + "\n
        Course Name: " + add_academicName + "\n Duration: " + add_academicDuration + "\n
        Level: " + add_academicLevel + "\n Credit: " + add_academicCredit + "\n Number of
        assessments: " + add_numberOfAssessments + "\n \n has been successfully added \n
        to Academic Course";

        message_academic.showMessageDialog(frame_academic,
        final_msgAcademic);
    }

    else if(academicExist == true){
        JOptionPane.showMessageDialog(frame_academic, "Course ID " +
        add_academicID + " is already added.");
    }

```

```

    }

    catch(NumberFormatException numberFormat_one){

        try{

            if(add_academicID.equals("") && add_academicName.equals("") &&
            add_academicDuration.equals("") && add_academicLevel.equals("") &&
            add_academicCredit.equals("") && add_numberOfAssessments.equals("")) {

                JOptionPane.showMessageDialog(frame_academic,"Please don not
                leave the text field empty or fill the text field properly.", "ERROR",
                JOptionPane.ERROR_MESSAGE);

            }

        }

        catch(NullPointerException nullPointer_one){

            JOptionPane.showMessageDialog(frame_academic,"Please don not
            leave the text field empty or fill the text field properly.", "ERROR",
            JOptionPane.ERROR_MESSAGE);

        }

    }

}

```

```

if(e.getSource() == button_register){
    register_academicID = field_courseID.getText();

    register_courseLeader = field_courseLeader.getText();

    register_lecturerName = field_lecturerName.getText();


    String years = (combobox_years.getSelectedItem()).toString();
    String months = (combobox_months.getSelectedItem()).toString();
    String days = (combobox_days.getSelectedItem()).toString();

    register_startingDate = years + months + days;


    String years1 = (combobox_years1.getSelectedItem()).toString();
    String months1 = (combobox_months1.getSelectedItem()).toString();
    String days1 = (combobox_days1.getSelectedItem()).toString();

    register_completionDate = years1 + months1 + days1;


    for(Course var : aList){
        if(var.getCourseID().equals(register_academicID)){

```

```

        if(var instanceof AcademicCourse){
            ref_academic = (AcademicCourse)var;

            if(ref_academic.getisRegistered() == true){
                JOptionPane.showMessageDialog(frame_academic,"Sorry, the
course is already registered.");
                break;
            }

            else if(ref_academic.getisRegistered() == false){
                ref_academic.register(register_courseLeader,
register_lecturerName, register_startingDate, register_completionDate);
                JOptionPane.showMessageDialog(frame_academic,"The course
is registered successfully.");
                break;
            }

        }

    }

    else{
        JOptionPane.showMessageDialog(frame_academic,"The Course ID
entered to add the course does not matches \n with the Course ID entered to register
the course.", "ERROR", JOptionPane.ERROR_MESSAGE);
        break;
    }
}

```

```

    }

}

for(Course var1 : aList){
    if(e.getSource() == button_display){

        if(var1 instanceof AcademicCourse){
            ref_academicDisplay = (AcademicCourse)var1;
            ref_academicDisplay.display();
        }

    }

}

```

```

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

    field_courseID.setText("");
    field_courseName.setText("");
}

```

```

field_duration.setText("");
field_level.setText("");
field_credit.setText("");
field_numberOfAssessments.setText("");
field_courseID.setText("");
field_courseLeader.setText("");
field_lecturerName.setText("");

combobox_years.setSelectedIndex(0);
combobox_months.setSelectedIndex(0);
combobox_days.setSelectedIndex(0);
combobox_years1.setSelectedIndex(0);
combobox_months1.setSelectedIndex(0);
combobox_days1.setSelectedIndex(0);
}

```

```

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

    try{
        add_nonAcademicCourseID = field_courseIDNon.getText();

        add_nonAcademicCourseName = field_courseNameNon.getText();

        add_nonAcademicDuration = field_durationNon.getText();
        int duration_NA = Integer.parseInt(add_nonAcademicDuration);
    }
}

```

```

add_nonAcademicPrerequisite = field_prerequisiteNon.getText();

nonAcademicExist = false;

for(Course var_addNon : aList){
    if(var_addNon.getCourseID().equals(add_nonAcademicCourseID)){
        nonAcademicExist = true;
        break;
    }
}

if(nonAcademicExist == false){

    obj_NonAcademic = new
NonAcademicCourse(add_nonAcademicCourseID , add_nonAcademicCourseName,
duration_NA, add_nonAcademicPrerequisite);

    aList.add(obj_NonAcademic);

    message_nonAcademic = new JOptionPane();

    String final_messageNon = "Course ID: " + add_nonAcademicCourseID
+ "\nCourse Name: " + add_nonAcademicCourseName + "\nDuration: " +
add_nonAcademicDuration + "\nPrerequisite: " + add_nonAcademicPrerequisite + "\n \n
has been successfully added \n to Non Academic Course";

    message_nonAcademic.showMessageDialog(frame_non,
final_messageNon);

```

```

    }

    else if(nonAcademicExist == true){
        JOptionPane.showMessageDialog(frame_non, "Course ID " +
add_nonAcademicCourseID+ " is already added.");
    }
}

catch(NumberFormatException numberFormat_two){

    try{
        if(add_nonAcademicCourseID.equals("") &&
add_nonAcademicCourseName.equals("") && add_nonAcademicDuration.equals("") &&
add_nonAcademicPrerequisite.equals("")) {

            JOptionPane.showMessageDialog(frame_academic,"Please don not
leave the text field empty or fill the text field properly.", "ERROR",
JOptionPane.ERROR_MESSAGE);

        }
    }

    catch(NullPointerException nullPointer_two){

        JOptionPane.showMessageDialog(frame_academic,"Please don not
leave the text field empty or fill the text field properly.", "ERROR",
JOptionPane.ERROR_MESSAGE);

    }

}

```



```
}
```

```
if(e.getSource() == button_registerNon){  
    register_nonAcademicID = field_courseIDNon.getText();  
  
    register_nonCourseLeader = field_courseLeaderNon.getText();  
  
    register_nonInstructorName = field_instructorNameNon.getText();  
  
    String years = (combobox_yearsNon.getSelectedItem()).toString();  
    String months = (combobox_monthsNon.getSelectedItem()).toString();  
    String days = (combobox_daysNon.getSelectedItem()).toString();  
  
    register_nonStartingDate = years + months + days;  
  
    String years1 = (combobox_yearsNon1.getSelectedItem()).toString();  
    String months1 = (combobox_monthsNon1.getSelectedItem()).toString();  
    String days1 = (combobox_daysNon1.getSelectedItem()).toString();  
  
    register_nonCompletionDate = years1 + months1 + days1;
```

```

String years2 = (combobox_yearsNon2.getSelectedItemAt()).toString();
String months2 = (combobox_monthsNon2.getSelectedItemAt()).toString();
String days2 = (combobox_daysNon2.getSelectedItemAt()).toString();

register_nonExamDate = years2 + months2 + days2;


for(Course var_non : aList){
    if(var_non.getCourseID().equals(register_nonAcademicID)){

        if(var_non instanceof NonAcademicCourse){
            ref_nonAcademic = (NonAcademicCourse)var_non;

            if(ref_nonAcademic.getisRegistered() == true){
                JOptionPane.showMessageDialog(frame_non,"Sorry, the course
is already registered.");
                break;
            }

            else if(ref_nonAcademic.getisRegistered() == false){
                ref_nonAcademic.register(register_nonCourseLeader,
register_nonInstructorName, register_nonStartingDate, register_nonCompletionDate,
register_nonExamDate);

                JOptionPane.showMessageDialog(frame_non,"The course is
registered successfully.");
                break;
            }
        }
    }
}

```

```

        }

    }

    else{

        JOptionPane.showMessageDialog(frame_non,"The Course ID entered
to add the course does not matches \n with the Course ID entered to register the
course.", "ERROR", JOptionPane.ERROR_MESSAGE);

        break;

    }

}

}

```

```

if(e.getSource() == button_removeNon){
    remove_nonAcademicID = field_courseIDNon.getText();

    for(Course var_non1 : aList){
        if(var_non1.getCourseID().equals(remove_nonAcademicID)){

            if(var_non1 instanceof NonAcademicCourse){
                ref_nonAcademicRemove = (NonAcademicCourse)var_non1;

                if(ref_nonAcademicRemove.getisRemoved() == false){
                    ref_nonAcademicRemove.remove();
                }
            }
        }
    }
}

```

```

        JOptionPane.showMessageDialog(frame_non,"The course is
removed successfully.");
        break;
    }

    else if(ref_nonAcademicRemove.getisRemoved() == true){
        JOptionPane.showMessageDialog(frame_non,"Sorry, the course is
already removed.");
        break;
    }

}

}

else{
    JOptionPane.showMessageDialog(frame_non,"The Course ID entered
to add the course does not matches \n with the Course ID entered to remove the
course.", "ERROR", JOptionPane.ERROR_MESSAGE);
    break;
}

}

}

```

```

for(Course var_non2 : aList){
    if(e.getSource() == button_displayNon){

```

```

        if(var_non2 instanceof NonAcademicCourse){
            ref_nonAcademicDisplay = (NonAcademicCourse)var_non2;
            ref_nonAcademicDisplay.display();
        }
    }
}

```

```

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

```

```

    field_courseIDNon.setText("");
    field_courseNameNon.setText("");
    field_durationNon.setText("");
    field_prerequisiteNon.setText("");

```

```

    field_courseIDNon.setText("");
    field_courseLeaderNon.setText("");
    field_instructorNameNon.setText("");

```

```

    combobox_yearsNon.setSelectedIndex(0);
    combobox_monthsNon.setSelectedIndex(0);
    combobox_daysNon.setSelectedIndex(0);

```

```
        combobox_yearsNon1.setSelectedIndex(0);
        combobox_monthsNon1.setSelectedIndex(0);
        combobox_daysNon1.setSelectedIndex(0);
        combobox_yearsNon2.setSelectedIndex(0);
        combobox_monthsNon2.setSelectedIndex(0);
        combobox_daysNon2.setSelectedIndex(0);
    }

}

}
```

Appendix2:

Course, Academic Course, NonAcademicCourse

i. Course class

```
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=courseLeader;

        courseLeader="";
    }

    public String getcourseID()
    {
```

```

        return courseID;
    }

    public String getcourseName()
    {
        return courseName;
    }

    public int getduration()
    {
        return duration;
    }

    public String getcourseLeader()
    {
        return courseLeader;
    }

    public void setcourseLeader(String newcourseLeader)
    {
        this.courseLeader=newcourseLeader;
    }

    public void display()
    {
        System.out.println("The course ID is " + courseID);
        System.out.println("The course name is " + courseName);
    }

```



```
System.out.println("The duration of the course is " + duration + " years");

if(courseLeader!= ""){
    System.out.println("The course leader is " + courseLeader);
}

}

}
```

ii. AcademicCourse class

```
public class AcademicCourse extends Course
{
    private String lecturerName;
    private String level;
    private String credit;
    private String startingDate;
    private String completionDate;
    private int numberOfAssessments;
    private boolean isRegistered;

    public AcademicCourse(String courseID,String courseName,int duration,String
level,String credit,int numberOfAssessments)
    {
        super(courseID,courseName,duration);

        this.level=level;
        this.credit=credit;
        this.numberOfAssessments=numberOfAssessments;
```

```
this.lecturerName=lecturerName;
lecturerName="";

this.startingDate=startingDate;
startingDate="";

this.completionDate=completionDate;
completionDate="";

this.isRegistered=isRegistered;

isRegistered= false;

}

public String getlevel()
{
    return level;
}

public String getcredit()
{
    return credit;
}

public String getstartingDate()
{
    return startingDate;
}
```

```
}
```

```
public String getcompletionDate()
```

```
{
```

```
    return completionDate;
```

```
}
```

```
public int getnumberOfAssessments()
```

```
{
```

```
    return numberOfAssessments;
```

```
}
```

```
public String getlecturerName()
```

```
{
```

```
    return lecturerName;
```

```
}
```

```
public boolean getisRegistered()
```

```
{
```

```
    return isRegistered;
```

```
}
```

```
public void setlecturerName(String newlecturerName)
```

```
{
```

```
    this.lecturerName= newlecturerName;
```

```
}
```

```
public void setnumberOfAssessments(int newnumberOfAssessments)
```

```

{
    this.numberOfAssessments= newnumberOfAssessments;
}

public void register(String courseLeader,String lecturerName,String
startingDate,String completionDate)
{

    if(isRegistered==true){
        System.out.println("The instructor is " + lecturerName);
        System.out.println("The starting date is " + startingDate);
        System.out.println("The completion date is " + completionDate);
    }

    else{
        super.setcourseLeader(courseLeader);
        this.lecturerName=lecturerName;
        this.startingDate=startingDate;
        this.completionDate=completionDate;
        this.isRegistered=isRegistered;
        isRegistered = true;
    }

}

public void display()
{

    super.display();
}

```

```

        if(isRegistered==true){
            System.out.println("The lecturer name is " + lecturerName);
            System.out.println("The level for the course is " + level);
            System.out.println("The credit for the course is " + credit);
            System.out.println("The starting date is " + startingDate);
            System.out.println("The completion date is " + completionDate);
            System.out.println("The number of assessments are " + numberOfAssessments);
        }
    }
}

```

iii. NonAcademicCourse class

```

public class NonAcademicCourse extends Course
{
    private String instructorName;
    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.instructorName=instructorName;

        this.prerequisite=prerequisite;


        this.startDate=startDate;
        startDate="";

        this.completionDate=completionDate;
        completionDate="";

        this.examDate=examDate;
        examDate="";

        isRegistered=false;

        isRemoved=true;
    }

    public String getinstructorName()

```

```
{  
    return instructorName;  
}  
  
public String getstartDate()  
{  
    return startDate;  
}  
  
public String getcompletionDate()  
{  
    return completionDate;  
}  
  
public String getexamDate()  
{  
    return examDate;  
}  
  
public String getprerequisite()  
{  
    return prerequisite;  
}  
  
public boolean getisRegistered()  
{  
    return isRegistered;  
}
```



```

public boolean getisRemoved()
{
    return isRemoved;
}

public void setinstructorName(String newinstructorName)
{
    this.instructorName=newinstructorName;

    if(isRegistered==false){
        this.instructorName=newinstructorName;
        instructorName=newinstructorName;
    }

    else{
        System.out.println("It is therefore not possible to change the instructor name as
non-academic course is registered .");
    }
}

public void register(String courseLeader,String instructorName,String startDate,String
completionDate,String examDate)
{
    if(isRegistered==false){
        setinstructorName(instructorName);
        setcourseLeader(courseLeader);
        this.startDate=startDate;
        this.completionDate=completionDate;
    }
}

```

```

        this.examDate=examDate;

        isRegistered=true;
    }

    else{
        System.out.println("Course is already registered.");
    }
}

public void remove()
{
    if(isRemoved==true){
        System.out.println("The non-academic course is already removed");
    }

    else{
        super.setcourseLeader("");

        instructorName="";
        startDate="";
        completionDate="";
        examDate="";

        isRegistered=false;
        isRemoved=true;
    }
}

```

```
public void display()
{
    super.display();

    if(isRegistered==true){
        System.out.println("The instructor name is " + instructorName);
        System.out.println("The starting date is " + startDate);
        System.out.println("The completion date is " + completionDate);
        System.out.println("The exam date is " + examDate);
    }

}

}
```