Contents

Tables of Tables

Tables of Figures

1.Introduction

1.1 Introduction to coursework

This course work is done with help of Bluej and msWord application. This coursework is based upon the GUI where simple GUI application is supposed to be developed. Here, Ingcollege is main class which contains all the GUI component and number of methods for adding, removing, registering, displaying academic and non academic course from arraylist of course type where data boject are store. This GUI based coursework is done with by combining all three classes i.e. Course class, AcademicCourse class and NonAcademicCourse class from past coursework with IngCollege class.

Here, using difffernet component of java swing menu bar, menu, menu items, labels , text fields and buttons are created for user graphical interaction where label represent name , text field provide place for user to enter value and button to perform some event when it is clicked.

Java awt event like MouseListener,ActionListener,FoucusListener are used in button for more interactive of GUI and also perfroms It task when the event occurs. Here, Add button is for both academic and non adcademic course to add respective course in respective arraylist when this button is clicked, Register button registers both academic and non acdemic course in respective arraylist when this button is clicked, Remove button removes only non acdemic course from the respective arraylist, Display button is used to display both academic and non academic course list from their respective arraylist to give information to user when this button is clicked and Clear button is used to clear the text field in the GUI which help to remove text in the text field when this button is clicked. Some extrat button where added like academic and non academic button when help to toggle the visibility of component in panel . Menu bar with Menu having Menu Items with in it is aslo added to main frame to make more user interactive. Mene contains MenuItems like Academic Course, Non Academic course,

Exit (which help to terminate program normally when this item is clicked), About (which help to give infromation about the developer). This application is develop to make more user interactive.

1.2 Tools use in project

Tools like Bluej and ms Word application is used to acomplished this coursework.

Bluej is application which is used in development environment which allows user to develop java programs fast and in efficient manner. The main features of this appication are its simple user interface than professional enverionment like NetBeans or Eclips which help to allow beginners to get started fastly and in efficitent manner. This application also allow to interact with objects which help to inspect their value ,call method on them,passing them as parameter and more. Invoking java expression without cpmpling can be done directly here. (bluej.org, 2021)

MsWord application was very helpful in documentation of this coursework report. This Application is word processor which is published by Microsoft company. It is available for windos,maOS,android ,IOS and linux operating system platform. This application allows to create professional quality of reports, letters,documents and CVs. It also has features like spell check, grammer check, text font formatting,image support, managing page layout and more. (Hope, 2021)

2.Class Diagram

2.1 Introduction

A type of diagram which defines and gives the overview and structure of a system in terms of classes, attributes, methods and relationship exist between various classes such diagram is known as Class diagram.

It is used to illustrate and generate a functional diagram of the system classes, which can also serve as a system development resource throughout the software development life cycle. (Inc., 2021)

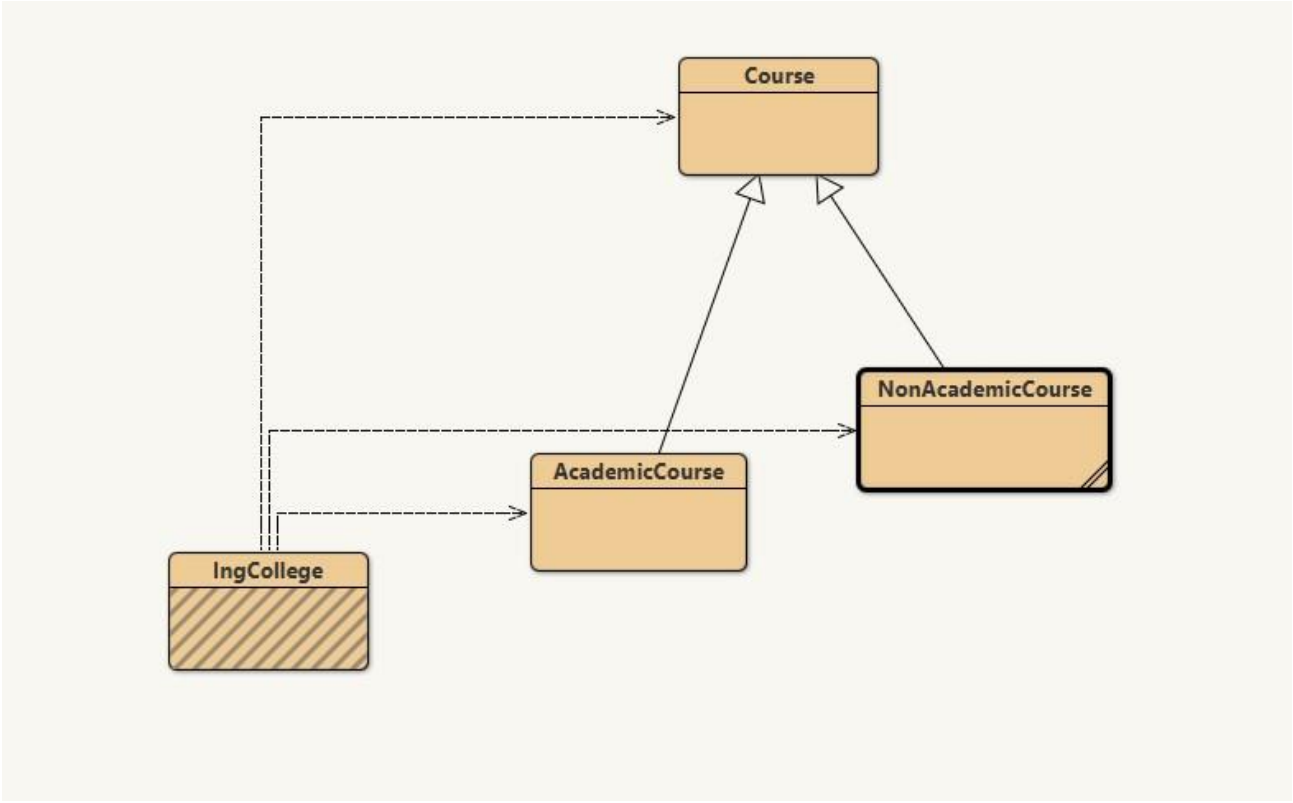

Figure 1 : Class diagram of classes in bluej.

2.2 IngCollege Class

IngCollege	
- mainFrame : JFrame	- panelCourse : JPanel
-mB : JMenuBar	-course : JMenu
-help : JMenu	-academic : JMenuItem
-nonAcademic : JMenuItem	-exit : JMenuItem
-about : JMenuItem	-flbl1: Font
-ff : Font	-ftxt: Font
-fMnu: Font	-fBtn: Font
-titleIcon : ImageIcon	-lbl_academic : JLabel
-lbl_nonAcademic : JLabel	-lbl_courseID : JLabel
-lbl_courseName : JLabel	-lbl_duration : JLabel
-lbl_lecturerName : JLabel	-lbl_level : JLabel
-lbl_credit : JLabel	-lbl_startDate : JLabel
-lbl_completionDate : JLabel	-lbl_numberOfAssessment : JLabel
-lbl_courseLeader : JLabel	-lbl_instructorName : JLabel
-lbl_examDate : JLabel	-lbl_preRequisite : JLabel
-lbl_logo : JLabel	-lbl_footer : JLabel
-lbl_academic : JLabel	-lbl_academic : JLabel
-lbl_academic : JLabel	-txt_courseID : JTextField
-txt_courseName : JTextField	-txt_duration : JTextField
-txt_lecturerName : JTextField	-txt_level : JTextField
-txt_credit : JTextField	-txt_numberOfAssessment : JTextField
-txt_courseLeader : JTextField	-txt_instructorName : JTextField
-txt_preRequisite : JTextField	-txt_examDate : JTextField
-txt_startDate : JTextField	-txt_completionDate : JTextField
-btnAcademic : JButton	-btnNonAcademic : JButton
-btnDisplay : JButton	-btnRegister : JButton
-btnAdd : JButton	-btnClear : JButton
-btnRemove : JButton	+acdObj : AcademicCourse
+nonAcdObj : NonAcademicCourse	+academicList : ArrayList<Course>
+nonAcademicList : ArrayList<Course>	

+<<Constructor>>IngCollege()	+actionPerformed(e : ActionEvent) : void
+mouseEntered(e : MouseEvent) : void	+mouseExited (e : MouseEvent) : void
+mouseClicked (e : MouseEvent) : void	+mousePressed (e : MouseEvent) : void
+mouseReleased (e : MouseEvent) : void	+about() : void
+academicComponent() : void	+nonAcademicComponent() : void

Table 1 : Class diagram of IngCollege Class

3.Pseudocode

A made-up and informal language that help programmers in the development of algorithms is known as Pseudocode. It is a detail (algorithmic) design tool which text is based. The rules of Pseudocode are very simple to understand. Below Pseudocode for IngCollege is given. (unf.edu, 2021)

START

IMPORT packages in program

CREATE IngCollege Class that implements ActionListener,MouseListener interface

DECLARE instance variables

mainframe as JFrame

panelCourse as JPanel

mB as JMenuBar

course, help as JMenu

academic, nonAcademic, exit, about as JMenuItem

lbl_academic, lbl_nonAcademic, lbl_courseID
lbl_courseName, lbl_duration, lbl_lecturerName,
lbl_level, lbl_credit, lbl_startDate, lbl_completionDate,
lbl_numberOfAssessment, lbl_courseLeader,
lbl_ instructorName, lbl_examDate,
lbl_preRequisite, lbl_logo,lbl_footer as JLabel

txt_courseID, txt_courseName, txt_duration, txt_lecturerName,

txt_level,txt_credit, txt_numberOfAssessment,
txt_courseLeader, txt_instructorName,
txt_preRequisite, txt_examDate,txt_startDate,
txt_completionDate as JTextField

btnAcademic, btnNonAcademic, btnDisplay,
btnRegister, btnAdd,btnClear,
btnRemove as JButton

flbl1, ff, ftxt, fMnu, fBtn as Font

titleIcon as ImageIcon

DECLARE staict variable

acdObj as Reference Object of AcademicCourse class

nonAcdObj as Reference Object of NonAcademicCourse class

DECLARE arraylist named academicList of Course class type

DECLARE arraylist named nonAcademicList of Course class type

CREATE a constructor IngCollege

DO

INITIALIZE JFrame to mainframe

INITIALIZE JPanel to JPanel

SET layout to null

SET background color of panel

SET Focusable to true

SET size of panel

INITIALIZE Font to flbl1 with parameter of Font

INITIALIZE Font to ff with parameter of Font

INITIALIZE Font to ftxt with parameter of Font

INITIALIZE Font to fBtn with parameter of Font

INITIALIZE Font to fMnu with parameter of Font

INITIALIZE JMenuBar to mb
INITIALIZE JMenu to course
INITIALIZE JMenuItem to academic
INITIALIZE JMenuItem to nonAcademic
INITIALIZE JMenuItem to exit
INITIALIZE JMenuItem to about
SET ActionListener interface to academic JMenuItem
S**ET** ActionListener interface to nonAcademic JMenuItem
SET ActionListener interface to exit JMenuItem
SET ActionListener interface to about JMenuItem

INITIALIZE JButton to btnAcademic with string parameter
SET position and size of btnAcademic button
SET border of btnAcademic button to null
SET font, textcolor, background color of btnAcademic button
SET visible of btnAcademic button to false
SET MouseListener interface to btnAcademic button
DEFINE actionPerformed method for btnAcademic button
DO
CALL academicComponent method
END DO

INITIALIZE JButton to btnNonAcademic with string parameter
SET position and size of btnNonAcademic button
SET border of btnNonAcademic button to null
SET font, text color, background color of btnAcademic button
SET MouseListener interface to btnNonAcademic button
DEFINE actionPerformed method for btnNonAcademic button

DO

CALL nonAcademicComponent method

END DO

INITIALIZE JButton to btnDisplay with string parameter

SET position and size of btnDisplay button

SET border of btnDisplay button to null

SET font, text color, background color of btnDisplay button

ADD MouseListener interface to btnDisplay button

DEFINE actionPerformed method for btnDisplay button

DO

IF(txt_courseID text field equals to “Enter The Course ID”)

DISPLAY message dialog box

ELSE

INITIALIZE JFrame to frame

IF(academic label or btnNonAcademic visibility is true)

DECLARE column names in object array

INITIALIZE DefaultTableModel to tabelModel

INITIALIZE JTable to table

INITIALIZE Font to font with font parameter

SET font of text in table

SET height of row in table

INITIALIZE JScrollPane to scrollPane

FOR

CONVERT object to AcademicCourse class type

DECLARE data in object array

ADD data to row of tableModel

END FOR

ADD scrollPane to frame

INITIALIZE JTabelHeader to tabelHeader

SET background color, text color,font

ELSE

IF(nonAcademic label or btnAcademic visibility is true)

DECLARE column names in object array

INITIALIZE DefaultTableModel to tabelModel

INITIALIZE JTable to table

INITIALIZE Font to font with font parameter

SET font of text in table

SET height of row in table

INITIALIZE JScrollPane to scrollPane

FOR

CONVERT object to NonAcademicCourse class type

DECLARE data in object array

ADD data to row of tableModel

END FOR

ADD scrollPane to frame

INITIALIZE JTabelHeader to tabelHeader

SET background color, text color,font

END IF

END IF

INITIALIZE ImageIcon to titleIcon

SET IconImage to frame

SET title for frame to ING College

SET size,position,visibility of frame

SET close operation of frame

END IF

END DO

INITIALIZE JButton to btnRegister with string parameter

SET position and size of btnRegister button

SET border of btnRegister button to null

SET font, text color, background color of btnRegister button

ADD MouseListener interface to btnRegister button

DEFINE actionPerformed method for btnRegister button

DO

IF(academic label or btnNonAcademic visibility is true)

IF(any text field equals placeholder text)

DISPLAY message dialog box

ELSE

GET data from all text Field for academic course

FOR

IF(object courseID equal textfield courseID)

CONVERT course type object

to AcademicCourse type

IF(course is not register)

CALL register method
From AcademicCourse class
DISPLAY message dialog box
RETURN exit method

ELSE

DISPLAY message dialog box
RETURN exit method

END IF

END IF

END FOR

DISPLAY message dialog box

RETURN exit method

END IF

ELSE

IF(nonAcademic label or btnAcademic visibility is true)

IF(any text field equals placeholder text)

```
                    DISPLAY message dialog box
                ELSE
                    GET value from all text
                     field for nonacademic course
                     FOR
                        IF(object courseID equal textfield courseID)
                            CONVERT course type object to
                            NonAcademicCourse   type
                             IF(course is not register)
                                CALL register method
                                 From AcademicCourse class
                                 DISPLAY message dialog box
                                 RETURN exit method
                             ELSE
                                 DISPLAY message dialog box
                                 RETURN exit method
                             END IF
                        END IF
                    END FOR
                    DISPLAY message dialog box
                    RETURN exit method
                END IF
            END IF
        END IF
    END DO


INITIALIZE JButton to btnAdd with string parameter
    SET position and size of btnAdd button
    SET border of btnAdd button to null
   SET font, text color, background color of btnAdd button
   ADD MouseListener interface to btnAdd  button
DEFINE actionPerformed method for btnAdd button
    DO
       IF(academic label or btnNonAcademic visibility is true )
            IF( any text field equals placeholder text)
```

DISPLAY message dialog box

ELSE

GET values from all text fields for academic course

TRY

CONVERT duration to number

END TRY

CATCH

DISPLAY message dialog box
RETURN exit method

END CATCH
TRY

CONVERT number of assessment to number

END TRY

CATCH

DISPLAY message dialog box
RETURN exit method

END CATCH

FOR
IF(object course ID equals text field courseID)
DISPLAY message dialog box
RETURN exit method
END IF
END FOR
INITIALIZE object of AcademicCourse
FETCH data from text field
ADD object of AcademicCourse to Array List
DISPLAY message dialog box

END IF

ELSE

IF(nonAcademic label or btnAcademic visibility is true)
IF(any text field equals placeholder text)
DISPLAY message dialog box
ELSE
GET values from all text fields
for non academic course
TRY

CONVERT duration to number

END TRY

CATCH

DISPLAY message dialog box
RETURN exit method
END CATCH
FOR
IF(object course ID equals text field courseID)
DISPLAY message dialog box
RETURN exit method
END IF
END FOR
INITIALIZE object of NonAcademicCourse
FETCH data from text field
ADD object of NonAcademicCourse to Array List
DISPLAY message dialog box
END IF
END IF
END DO

INITIALIZE JButton to btnClear with string parameter

SET position and size of btnClear button

SET border of btnClear button to null

SET font, text color, background color of btnClear button

ADD MouseListener interface to btnClear button

DEFINE actionPerformed method for btnClear button

DO

DECLARE response as Dialog box with yes or no option

IF (response == yes)

SET text Field of txt_courseID to

Enter The Course ID

SET text Field of txt_courseName to

Enter The Course Name..

SET text Field of txt_duration to

Enter the Duration For Course..

SET text Field of txt_lecturerName to

Enter The Name for Lecturer

SET text Field of txt_level to

Enter The Course Level

SET text Field of txt_credit to

Enter The Total credit For Course

SET text Field of txt_numberOfAssessment

to Enter Total Assessment

SET text Field of txt_courseLeader to

Enter The Name Course Leader

SET text Field of txt_startDate to dd-mm-yy

SET text Field of txt_completionDate to dd-mm-yy

SET text Field of txt_preRequisite to

Enter PreRequisite For Course

SET text Field of txt_instructorName to

Enter The Instructor Name

SET text Field of txt_examDate to dd-mm-yy

SET color of text of txt_courseID Text Field

SET color of text of txt_courseName Text Field

SET color of text of txt_duration Text Field

SET color of text of txt_courseLeader Text Field

SET color of text of txt_startDate Text Field

SET color of text of txt_completionDate Text Field

SET color of text of txt_numberOfAssessment

Text Fie

SET color of text of txt_level Text Field

SET color of text of txt_credit Text Field

SET color of text of txt_lecturerName Text Field

SET color of text of txt_instructorName Text Field

SET color of text of txt_examDate Text Field

SET color of text of txt_preRequisite Text Field

DISPLAY message dialog box

END IF

```
END DO

INITIALIZE JButton to btnRemove with string parameter
    SET position and size of btnRemove button
    SET border of btnRemove button to null
    SET font, text color, background color of btnRemove button
    ADD MouseListener interface to btnRemove  button
DEFINE actionPerformed method for btnRemove button
    DO
        IF( any text field equals placeholder text)
            DISPLAY message dialog box
        ELSE
            DECLARE response as Dialog box
             with yes or no option
            IF (response == yes)
                GET value from courseID text field
                 of non-academic course
                FOR
                    IF(object course ID equals text field courseID)
                        CONVERT Course type object
                        to NonAcademicCourse type
                        IF(course is not remove)
                            CALL remove method
                             from NonAcademicCourse class
                             CALL remove method
                              from arrayList class
                             DISPLAY message dialog box
                             RETURN exit method
                        END IF
                    END IF
                END FOR
                DISPLAY message dialog box
                RETURN exit method
            END IF
```

END IF

END DO

INITIALIZE JLabel to lbl_academic,
lbl_nonAcademic, lbl_courseID
lbl_courseName, lbl_duration, lbl_lecturerName,
lbl_level, lbl_credit, lbl_startDate, lbl_completionDate,
lbl_numberOfAssessment, lbl_courseLeader,
lbl_ instructorName, lbl_examDate,
lbl_preRequisite, lbl_logo,lbl_footer

SET size,position,font ,title,visibility of all Label

INITIALIZE JTextField to txt_courseID,
txt_courseName, txt_duration, txt_lecturerName,
txt_level,txt_credit, txt_numberOfAssessment,
txt_courseLeader, txt_instructorName,
txt_preRequisite, txt_examDate,txt_startDate,
txt_completionDate

SET size,position,font ,title,visibility of all textField

ADD focusListener to all text Field

DEFINE focusGained method for all text Field

DO

IF(any text field text equal placeholder text)

SET text field to empty

END IF

SET text of text field color

END DO

DEFINE focus Lost method for all text Field

DO

IF(any text field text equal placeholder text or is empty)

SET text color of text Field

SET placeholder text of text Field

END IF

END DO

ADD academic,nonAcademic,exit Menu Item to Course Menu
ADD about Menu Item to help Menu
ADD course,help Menu to Menu Bar

ADD all label and text Field,buttons to PanelCourse
ADD panelCourse to mainframe
ADD MenuBar to mainframe
INITIALIZE ImageIcon to titleIcon pass image to its parameter
SET image,size,position,title,layout,
visibility,close operation of main frame
END DO

DEFINE actionPerformed method as void for JMenuItem
DO
IF (Source == academic)
CALL academicComponent method
ELSE IF (Source == nonAcademic)
CALL nonAcademicComponent method
ELSE IF (Source == exit)
DECLARE response as Dialog box with yes or no option
IF (response == yes)
EXIT program normally
END IF
ELSE
IF (Source == about)
CALL about method
END IF
END IF
END DO

DEFINE mouseEntered method as void for buttons

DO

IF (Source == btnAcademic)

SET background Color of academic button

ELSE IF (Source == btnNonAcademic)

SET background Color of NonAcademic button

ELSE IF (Source == btnDisplay)

SET background Color of Display button

ELSE IF (Source == btnRegister)

SET background Color of Register button

ELSE IF (Source == btnAdd)

SET background Color of Add button

ELSE IF (Source ==btnRemove)

SET background Color of Remove button

ELSE

IF (Source ==btnClear)

SET background Color of Clear button

END IF

END IF

END DO

DEFINE mouseExited method void for buttons

DO

IF (Source == btnAcademic)

SET background Color of academic button

ELSE IF (Source == btnNonAcademic)

SET background Color of NonAcademic button

ELSE IF (Source == btnDisplay)

SET background Color of Display button

ELSE IF (Source == btnRegister)

```
                    SET background Color of Register button
                ELSE IF (Source == btnAdd)
                    SET background Color of Add button
                ELSE IF (Source ==btnRemove)
                    SET background Color of Remove button
                ELSE
                     IF (Source ==btnClear)
                          SET background Color of Clear button
                    END IF
                 END IF
            END DO

DEFINE mouseClicked method as void
DEFINE mousePressed method as void
DEFINE mouseRelesaed method as void

DEFINE about method as void
        DO
            DECLARE local variables
                Jfme as JFrame

                titleIcon as ImageIcon

                panelInfo as JPanel

                lblTitle, lblVersion,
                lblCopyRight, lblContact,
                lblEmail, lblMoblieNumber,
                lblImage as JLabel
```

btnOk as JButton

fTitle, font, fCright, fBtn as Font

INITIALIZE JFrame to jfme

INITIALIZE JPanel to panelInfo
SET background color of panel
SET Size and position of panel
SET Layout panelInfo to null

INITIALIZE Font to fTitle with parameter of Font
INITIALIZE Font to font with parameter of Font
INITIALIZE Font to fCright with parameter of Font
INITIALIZE Font to fBtn with parameter of Font

INITIALIZE JLabel to lblImage with object of ImageIcon class
SET size and position of lblImage Label

INITIALIZE JLabel to lblTitle with string parameter
SET size and position of lblTitle Label
SET font attributes of text lblTitle Label

INITIALIZE JLabel to lblVersion with string parameter
SET size and position of lblVersion Label
SET font attributes of text lblVersion Label

INITIALIZE JLabel to lblCopyRight with string parameter

SET size and position of lblCopyRightLabel

SET text color lblCopyRight Label

SET font attributes of text lblCopyRight Label

INITIALIZE JLabel to lblContact with string parameter

SET size and position of lblContact Label

SET text color lblContact Label

SET font attributes of text lblContact Label

INITIALIZE JLabel to lblEmail with string parameter

SET size and position of lblEmail Label

SET text color lblEmail JLabel

SET font attributes of text lblEmail Label

INITIALIZE JLabel to lblMobNum with string parameter

SET size and position of lblMobNum Label

SET text color lblMobNum Label

SET font attributes of text lblMobNum Label

INITIALIZE JButton to btnOk with string parameter

SET size and position of btnOk Button

SET border line to null of btnOk Button

SET background color of btnOk Button

SET text color of btnOK JButton

SET font attributes of text on btnOK Button

DEFINE actionListener method for btnOk Button

DO

Call dispose method to close this JFrame

END DO

DEFINE mouseEntered method for btnOk Button

DO

IF (Source == btnOk)

SET background color of btnOk button

END IF

END DO

DEFINE mouseExited method for btnOkButton

DO

IF (Source == btnOk)

SET background color of btnOk button

END IF

END DO

DEFINE mouseClicked method for btnOk Button

DEFINE mousePressed method for btnOk Button

DEFINE mouseReleased method for btnOk Button

SET lblTitle JLabel in to panel

SET lblVersion JLabel in to panel

SET lblCopyRight JLabel in to panel

SET lblContact JLabel in to panel

SET lblEmail JLabel in to panel

SET lblMobNum JLabel in to panel

SET btnOk JButton in to panel

SET lblImage JLabel in to JFrame

SET panelInfo JPanel in to JFrame

INITIALIZE ImageIcon to titleIcon with image as parameter

SET icon image into JFame

SET size of JFrame
SET background color of JFrame
SET position of JFrame to center
SET title of JFrame to About ING College
SET Layout to null for JFrame
SET no resize option for JFrame
SET close operation of JFrame
SET visibility of JFrame to true
END DO

DEFINE academicComponent method as void
DO
SET panel Focusable true
SET label nonAcademic visibility to false
SET button Academic visibility to false
SET button Remove visibility to false
SET label examDate visibility to false
SET text Field examDate visibility to false
SET label instructorName visibility false
SET text Field instructorName visibility false
SET label preRequisite visibility false
SET text Field preRequisite visibility false
SET label academic visibility true
SET button NonAcademic visibility true
SET label credit visibility true
SET text Field credit visibility true
SET label level visibility true
SET text Field level visibility true
SET label numberOfAssessment visibility true
SET text Field numberOfAssessments visibility true

SET label lecturerName visibility true

SET text Field of lecturerName visibility true

SET text Field of txt_courseID to Enter The Course ID.....

SET text Field of txt_courseName to Enter The Course Name..

SET text Field of txt_duration to Enter the Duration For Course..

SET text Field of txt_lecturerName to Enter The Name for Lecturer

SET text Field of txt_level to Enter The Course Level

SET text Field of txt_credit to Enter The Total credit For Course

SET text Field of txt_numberOfAssessment to Enter Total Assessment

SET text Field of txt_courseLeader to Enter The Name Course Leader

SET text Field of txt_startDate to dd-mm-yy

SET text Field of txt_completionDate to dd-mm-yy

SET color of text of txt_courseID Text Field

SET color of text of txt_courseName Text Field

SET color of text of txt_duration Text Field

SET color of text of txt_lecturerName Text Field

SET color of text of txt_level Text Field

SET color of text of txt_credit Text Field

SET color of text of txt_numberOfAssessment Text Field

SET color of text of txt_courseLeader Text Field

SET color of text of txt_startDate Text Field

SET color of text of txt_completionDate Text Field

END DO

DEFINE nonAcademicComponent method as void

DO

SET panel Focusable true

SET button NonAcademic visibility false

SET label academic visibility false

SET label level visibility false

SET text Field level visibility false

SET label numberOfAssessment visibility false

SET text Field numberOfAssessments visibility false

SET label credit visibility false

SET text Field credit visibility false

SET label lecturerName visibility false

SET text Field of lecturerName visibility false

SET label examDate visibility to true

SET text Field examDate visibility to true

SET label nonAcademic visibility to true

SET button Remove visibility to true

SET button Academic visibility to true

SET label preRequisite visibility true

SET text Field preRequisite visibility true

SET label instructorName visibility true

SET text Field instructorName visibility true

SET text Field of txt_courseID to Enter The Course ID

SET text Field of txt_courseName to Enter The Course Name

SET text Field of txt_duration to Enter the Duration For Course

SET text Field of txt_preRequisite to Enter PreRequisite For Course

SET text Field of txt_instructorName to Enter The Instructor Name

SET text Field of txt_courseLeader to Enter The Name Course Leader

SET text Field of txt_examDate to dd-mm-yy

SET text Field of txt_startDate to dd-mm-yy

SET text Field of txt_completionDate to dd-mm-yy

SET color of text of txt_courseID Text Field

SET color of text of txt_instructorName Text Field

SET color of text of txt_examDate Text Field

SET color of text of txt_preRequisite Text Field

SET color of text of txt_courseName Text Field

SET color of text of txt_duration Text

SET color of text of txt_courseLeader Text Field

SET color of text of txt_startDate Text Field

SET color of text of txt_completionDate Text Field

END DO

DEFINE main method as void

DO

CALL IngCollege constructor method of IngCollege class

END DO

END

4.Method Description

Different methods have been used while creating GUI of this program. In this program, ActionListener and MouseListener interface's methods are override for functionality of buttons and other method are also used.

4.1 IngCollege Class

- **IngCollege() :**

  In **main()** method only This constructor method of IngCollege class that is invoked
  In this method, all GUI component of this program and buttons action methods are carried out. Here in the method size, position, visibility, font, for JFrame, Jpanel, Jlabel and JText field all are define. Below given are the method for buttons present inside constructor method :-

  - **actionPerformed(ActionEvent e) for btnAcademic button :**

    This ActionListener interface's method which is override to call **academicComponent()**.

  - **actionPerformed(ActionEvent e) for btnNonAcademic button :**

    This is ActionListener interface's method which is override to call **nonAcademicComponent()**.

  - **actionPerformed(ActionEvent e) for btnAdd button :**

    This is ActionListener interface's method which accepts Event source as a parameter which is override to add the both academic course and non-academic course in its respective academic and nonacademic array list of

course type. With the help of instance object of AcademicCourse and NonAcademicCourse class their constructor is called and value from text field like course ID, course Name, duration, level, credit, numberOfAssessments for academic course and course ID, course Name, duration, preRequsite for non academic course are retrieve with the help of **get Text()** method and passed it in to constructor's parameter and that instance object of AcademicCourse and NonAcademicCourse is added to its respective academic and NonAcademic array list with the help of **add()** method one at a time. Here, every time when btnAdd is clicked academic and non-academic course is added by following above process.

- **actionPerformed(ActionEvent e) for btnRegister :**

  This is ActionListener interface's method that accepts Event source as a parameter which is override to register both academic and non-academic course in its respective academic and nonAcademic array list course type. Value from text field like courseID, course leader, lecturer name, starting date, and completion date are retrieve for academic course and Value from text field like courseID, course leader, instructor name, starting date, and completion date, exam date are retrieve for academic course with the help of **getText()** method. For loop is used to iterate and access the array list and checks course Id of each oject data in arraylist and try to match with user enterd courseID in text field .if match is found then downcating is done for converting course type object into AcademicCourse type or NonAcademicCourse type and again checks if the a particular course is register or not. If the aprticular course is not register then **register()** method is called from AcademicCourse or NonAacademicCourse class and fetch value from text field are passed in to **register()** method parameter and a particular course is registered . After that a dialog message box appears saying " course is successfully registered". If the course is already register It will display message box saying " course is already registered". If the course

ID cannot be match in arrayList it will display “course doesnot exit”. Here, every time when btnRegister is clicked academic and non-academic course is register and checks every condition by following above process .

- **actionPerformed(ActionEvent e) for btnDisplay :**

  This is ActionListener interface’s method that accepts Event source as a parameter which is override to display both academic and non-academic course details respectively in new frame in form of table by using JTabel. The data for the table are extracted from respective arraylist i.e. academicList and nonAcademicList. When the btnDisplay is clicked this method is invoked and execute.

- **actionPerformed(ActionEvent e) for btnClear button :**

  This is ActionListener interface’s method that accepts Event source as a parameter which is override to set the text for placeholder of all text field in both academic and non-academic course if user provides “yes” response in dialog message box which is shown when this button is clicked by user and a suitable dialog messages appears saying “ All text field is cleared”.

- **actionPerformed(ActionEvent e) for btnRemove button :**

  This is ActionListener interface’s method that accepts Event source as a parameter which is override to remove non-academic course in from non Academic array list when this button is clicked. When the button is clicked by user, a dialog message box appears for user to click “yes” or “no”. If user provide “yes” response then the user provided course Id is match in array list and if found, then method to remove non academic course is called from

NonAcademicCourse class and that course is also removed from array list and Then suitable dialog box appears saying" course is removed". IF the match is not found then dialog box appears saying "course does not exit". Every time when user clicks this button this method is invoked and execute.

- **actionPerfromed(ActionEvent e) for JMenuItems :**

  This is ActionListener interface's method which accepts Event source as a parameter which is override to performed task according to event source where the event occurs.
  This method is invoked and executed when button is clicked.
  Here, if the event source is academic JMenuItem then it will simply calls **academicCompenent()** method. If the event source is nonAcademic JMenuItem then it will call **nonAcademicComponent()** method. Likewise, if the event source is exit JMenuItem then a message dialog box appears for user response, if the response is "yes" then program will exit normally. Similarly, if the event source is about JMenuItem then it will invoke **about()** method and executes.

- **mouseEntered(MouseEvent e) for academic and non-academic course button :**

  This is MouseListener interface's method which accepts event source as a parameter which is override to change the background color of all button where this interface is implemented. when user hovers mouse pointer on button the background color of a button is change.

- **mouseExited(MouseEvent e) for academic and non-academic course button :**

  This is MouseListener interface's method which accepts event source as a parameter which is override to change the background color of all button where this interface is implemented. when user moves mouse pointer away from the button the background color of a button is change its original color.

- **about() :**

  This method is for showing details about the developer information in new frame when about JMenuItem is clicked this method is invoked and executed.

- **academicComponent() :**

  This method is simply for setting visibility of label and text field for academic course .it also set text color of the text in the text field and sets certain text as placeholder in text feild. This method is invoked and executed when btnAcademic button and academic JMenuItem are clicked.

- **nonAcademicComponent() :**

  This method is simply for setting visibility of label and text field for academic course .it also set text color of the text in the text field and sets certain text as placeholder in text feild. This method is invoked and executed when btnNonAcademic button and nonAcademic JMenuItem are clicked.

5.Testing

5.1. Test 1

Test No	1
Objective	To run IngCollege java file through command prompt.
Action	Initially opening command prompt and giving commands for compile and running java file.
Expected result	Java file will be compiled successfully and GUI of IngCollege will be displayed.
Actual Result	GUI of IngCollege was displayed.
Conclusion	Test was successful.

Table 2 : To run IngCollege java file through command prompt.

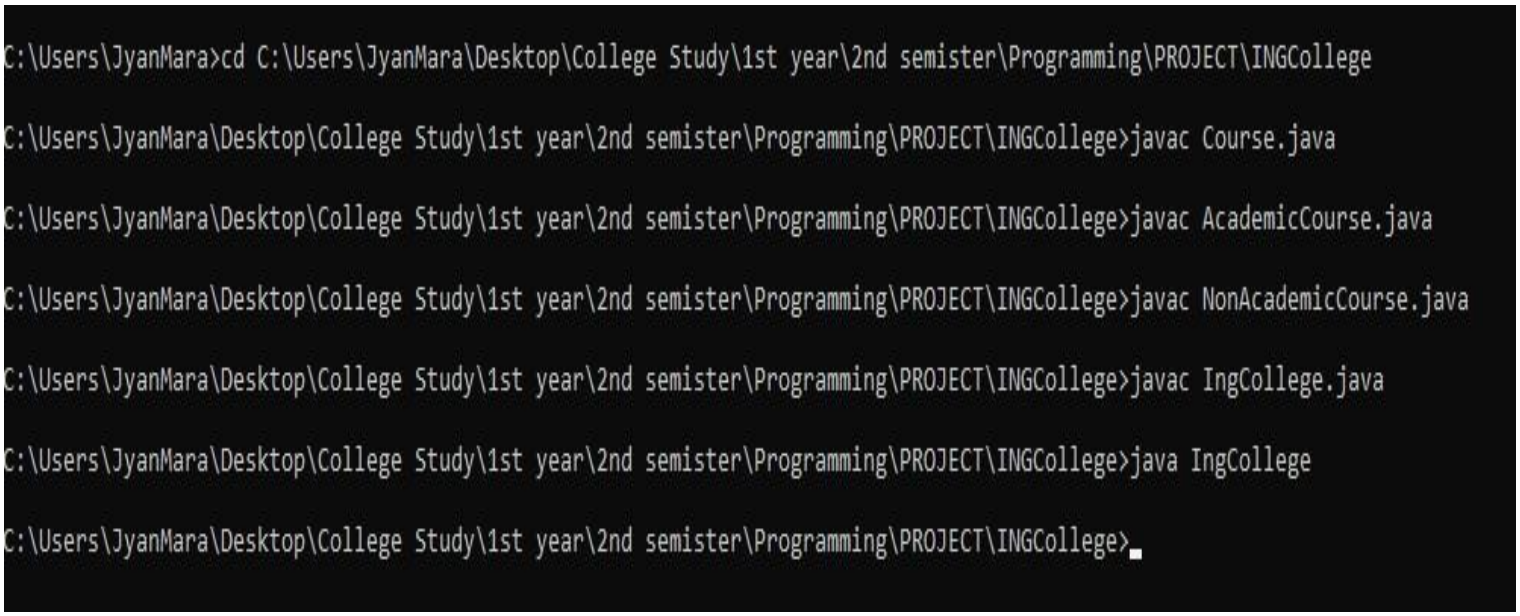

Figure 2 : Running program through command prompt.

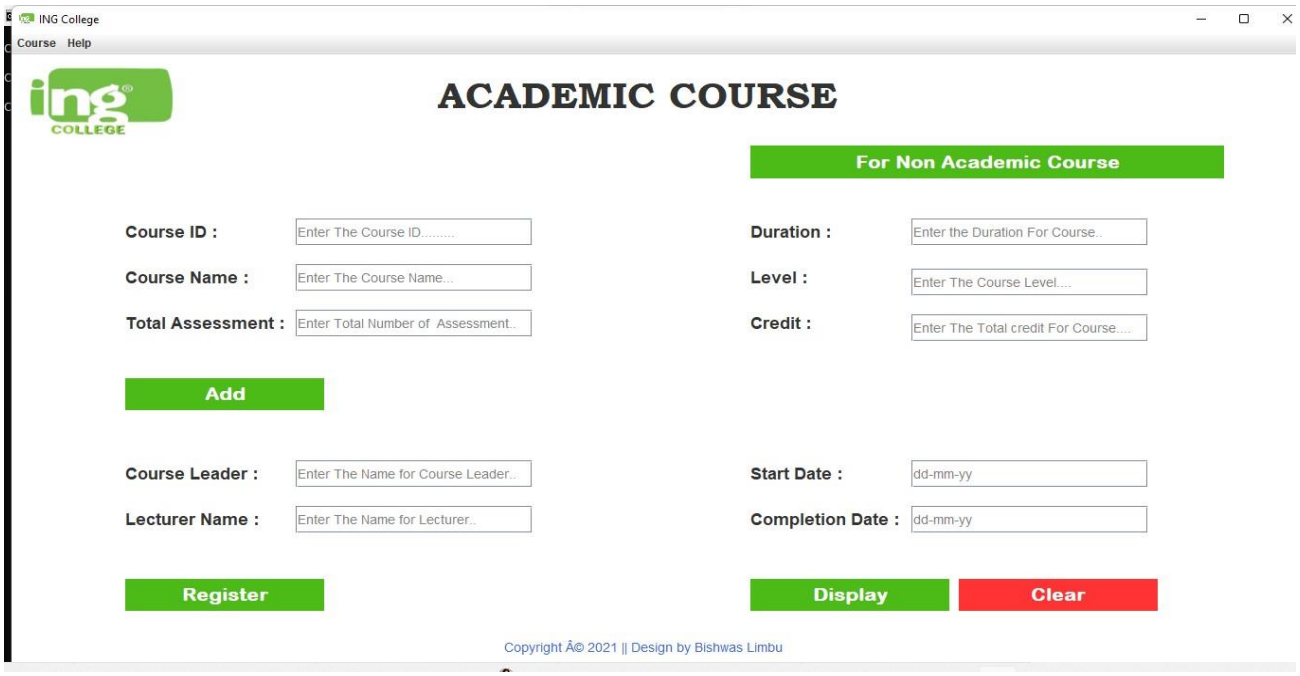

Figure 3 : Screenshot of GUI of IngCollege Class.

5.2. Test 2

5.2.1 Test 2.a

Test No	**2.a**
Objective	To Add course for Academic course
Action	- All required text field for adding Academic course are filled. Course ID: CS4100 Course Name: Programming Duration: 12 Total Assessment: 3 Level: 4 Credit: 30 -Add button was clicked.
Expected result	A Message dialog box will appear as indicating academic course is successfully added.

Actual Result	Message dialog box showing Academic course is added successfully was prompt out.
Conclusion	Test was successful.

Table 3 : To add course for Academic course

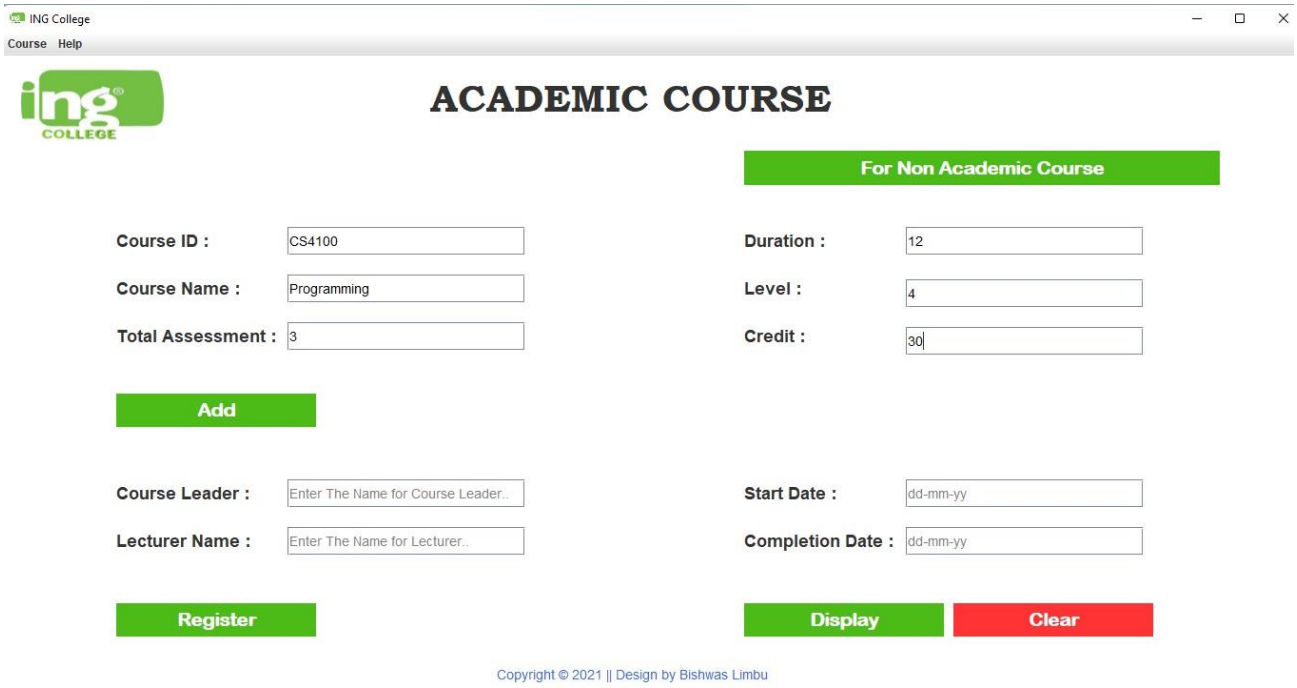

Figure 4 : Providing values to required text fields for adding Academic course.

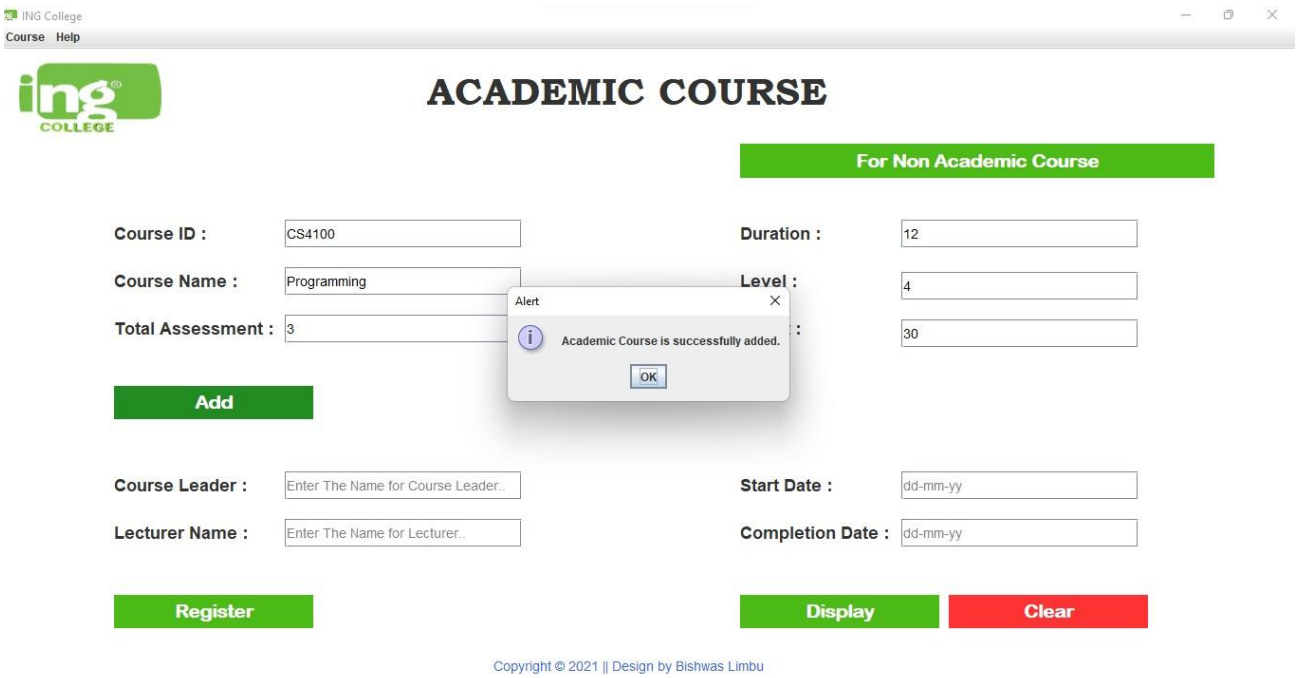

Figure 5 : Result after clicking Add button for Academic Course.

Figure 6 : Result after clicking display button to display added Academic course.

5.2.2 Test 2.b

Test No	2.b
Objective	To Add course for Non-Academic course
Action	-All required text field for adding Non Academic course are filled Course ID: AB4500 Course Name: Adobe photoshop Duration: 4 preRequisite: photoshop basic -Add button was clicked.

Expected result	A Message dialog box will appear as indicating non academic course is successfully added.
Actual Result	Message dialog box showing Non Academic course is added successfully was prompt out.
Conclusion	Test was successful.

Table 4 : To add course for Non Academic course.

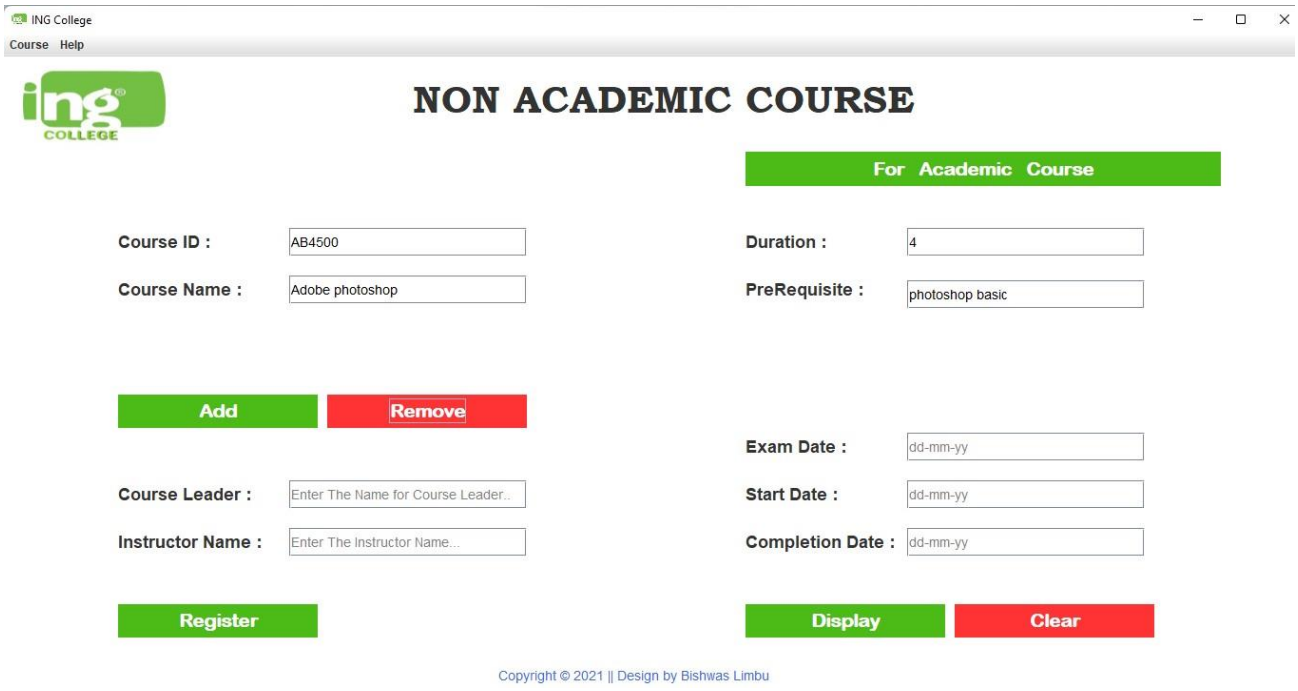

Figure 7 : Providing values to required text fields for adding Non Academic course

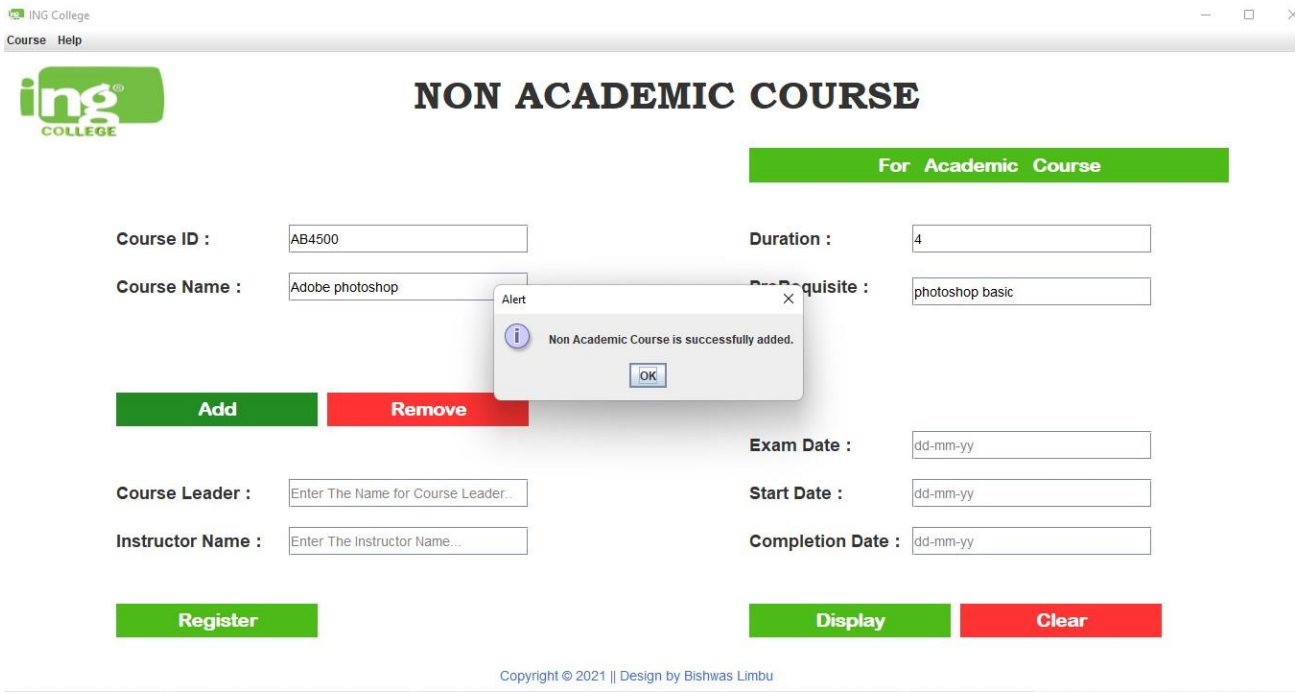

Figure 8 : Result after clicking Add button for Non Academic Course.

Figure 9 : Result after clicking display button to display added Non Academic course.

5.2.3 Test 2.c

Test No	**2.c**
Objective	To register course for Academic course.
Action	-All required text field including course ID for registering Academic course are filled Course ID: CS4100 Course Name: Venus Chhantel Lecture Name: Subash shau Start Date: 21-04-2021

	Completion Date: 02-06-2021 -Register button was clicked.
Expected result	A Message dialog box will appear as indicating academic course is successfully registered.
Actual Result	Message dialog box showing The Academic course is registered successfully was prompt out.
Conclusion	Test was successful.

Table 5 : To register course for Academic course.

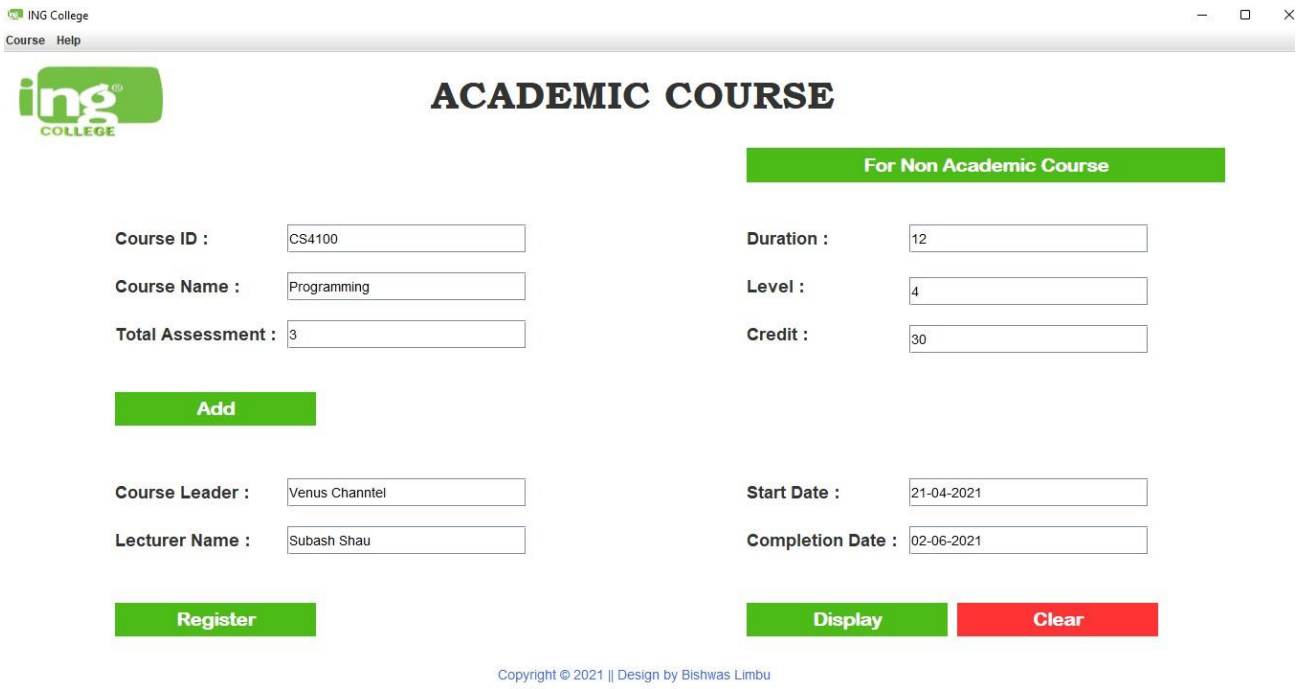

Figure 10 : Providing values to required text fields for registering Academic course

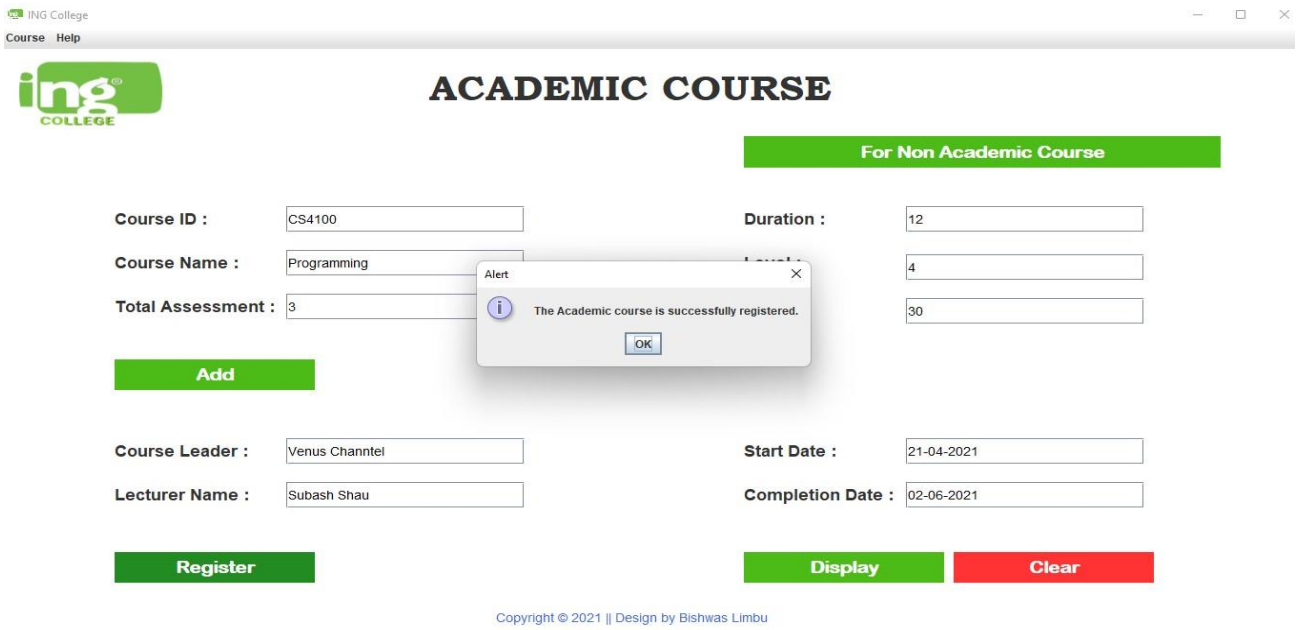

Figure 11 : Result after clicking Register button for Academic Course.

Figure 12 : Result after clicking display button to display registered Academic course.

5.2.4 Test 2.d

Test No	**2.d**
Objective	To register course for Non Academic course.
Action	-All required text field including course ID for registering Non Academic course are filled. Course ID: AB4500 Course Name: Kriti Kumari Rai Instructor Name: Pasang Lama Dolmu Exam Date: 12-02-2021 Start Date: 04-01-2021 Completion Date: 09-02-2021

	-Register button was clicked.
Expected result	A Message dialog box will appear as indicating non academic course is successfully registered.
Actual Result	Message dialog box showing The Non Academic course is registered successfully was prompt out.
Conclusion	Test was successful.

Table 6 : To register course for Non Academic course.

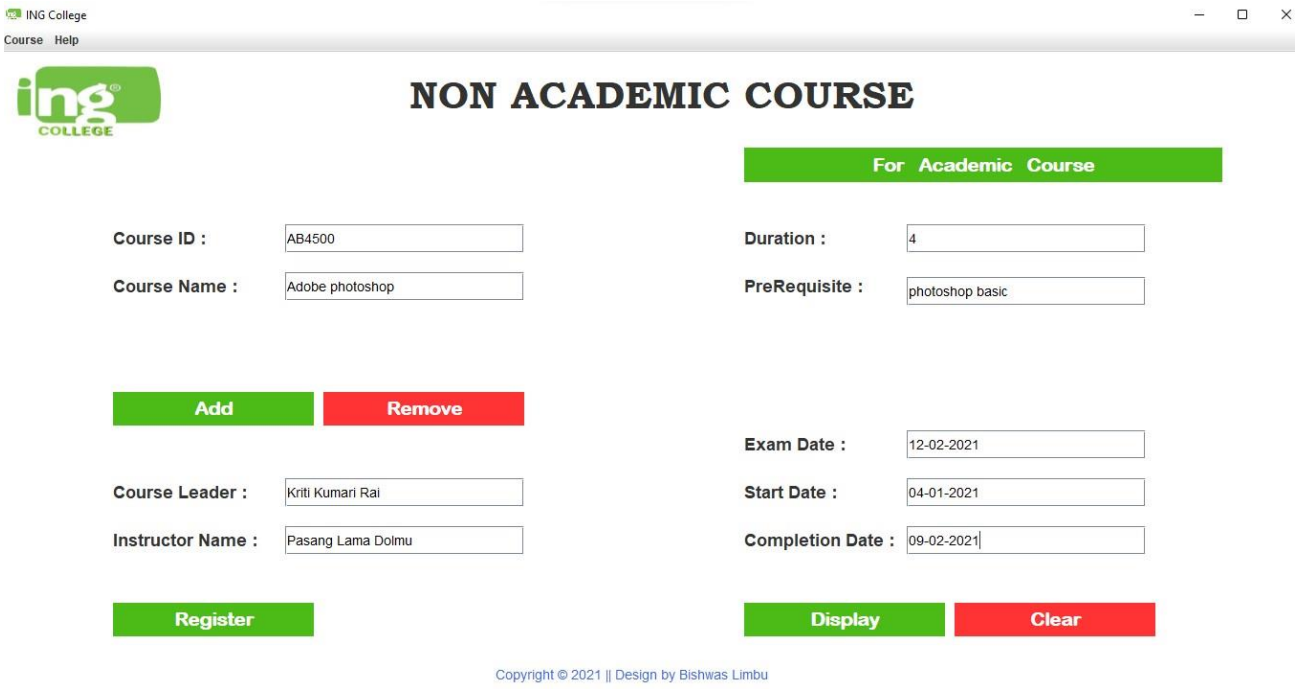

Figure 13 : Providing values to required text fields for registering Non Academic course

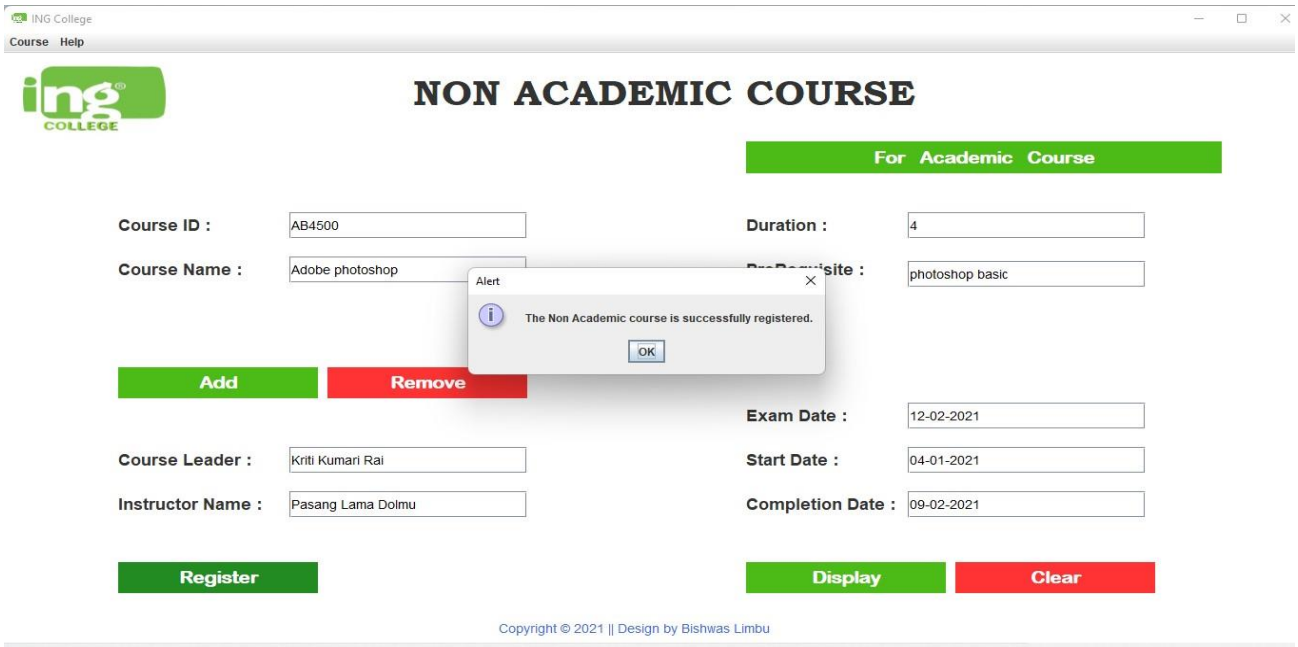

Figure 14 : Result after clicking Register button for Non Academic Course.

Figure 15 : Result after clicking display button to display registered Non Academic course.

5.2.5 Test 2.e

Test No	**2.e**
Objective	To remove course for Non Academic course.
Action	-Non Academic course's course ID value was filled in course ID text field Course ID: AB4500 -Remove button was clicked.
Expected result	A confirm dialog box appears asking user response. If user provide response "yes" then a message dialog indicating Non Academic course is successfully removed will appear.

Actual Result	Message dialog box showing The Non Academic course is successfully removed was appeared after response “yes” was given to confirm dialog box.
Conclusion	Test was successful.

Table 7 : To remove course for Non Academic course.

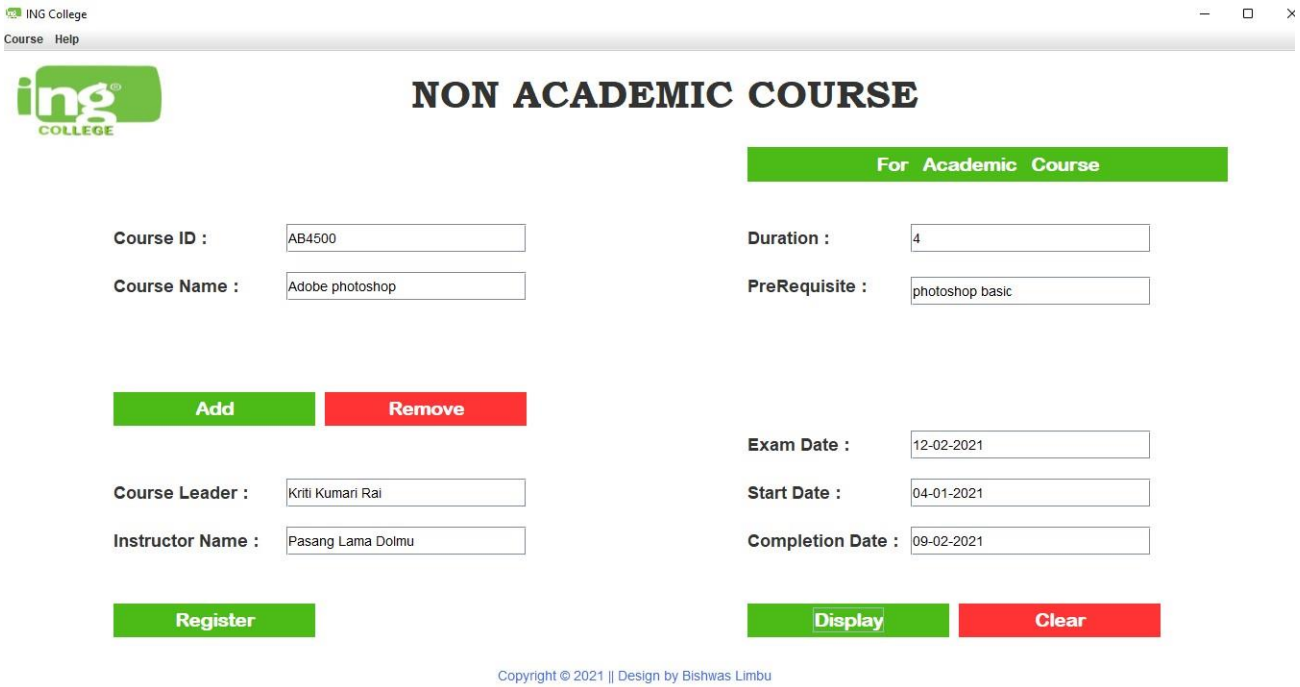

Figure 16 : Providing Non Academic course's course ID to Course ID text field.

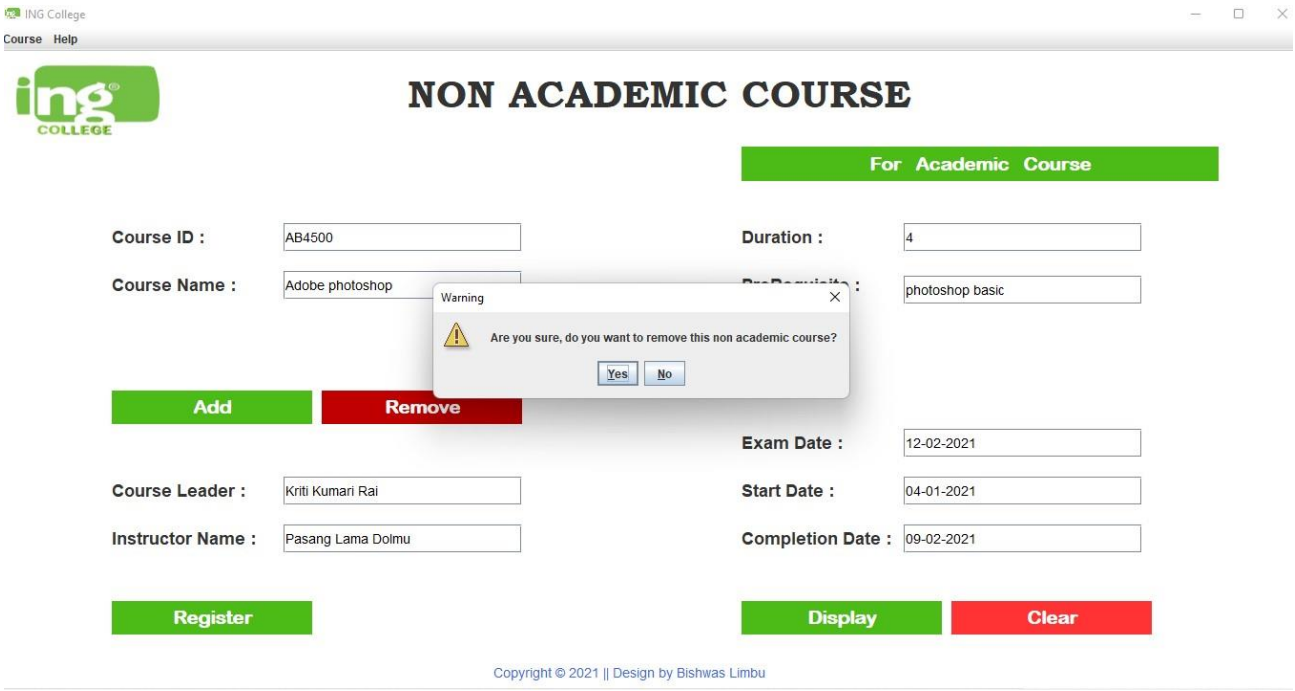

Figure 17 : Confirm dialog box appears after remove button is clicked for user response.

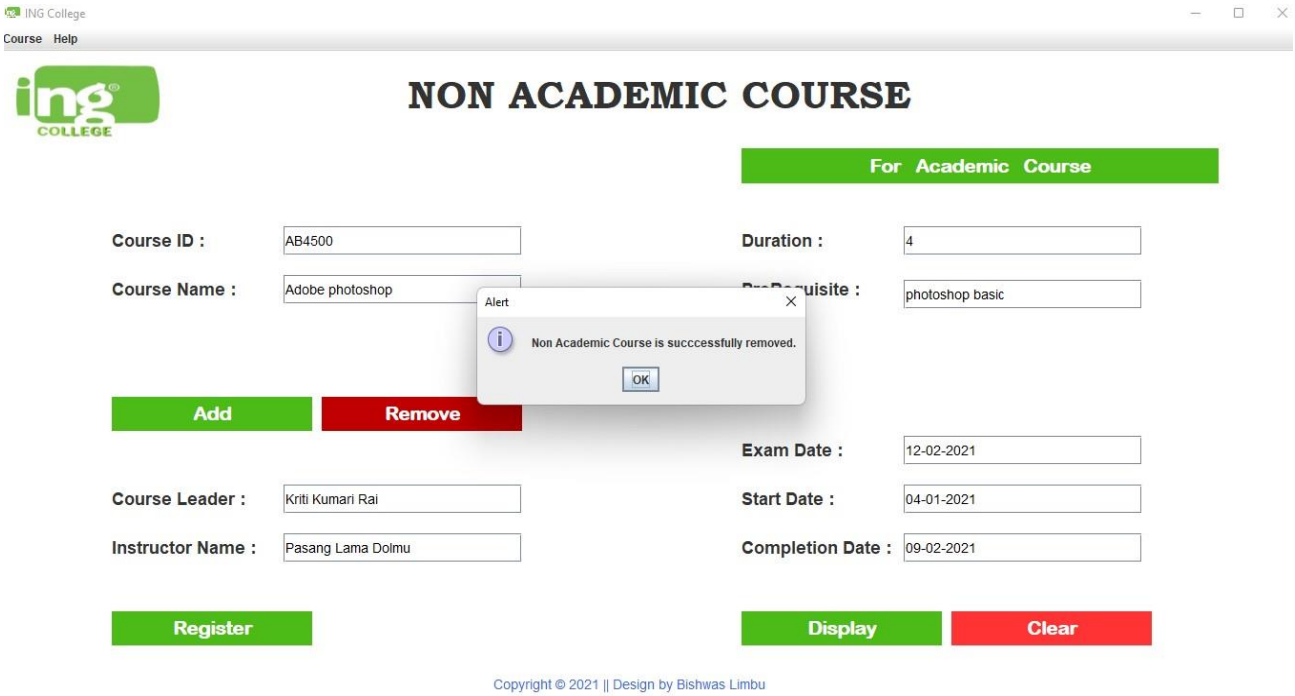

Figure 18 : "Non Academic course is successfully removed" Message dialog box appears after user response is "yes".

Figure 19 : Result after Non Academic course is removed and Display button clicked will show no data.

5.3. Test 3

5.3.1 Test 3.a

Test No	**3.a**
Objective	Tying to add duplicate course ID in both Academic and Non academic Course.
Action	Already added course ID of Academic and Non academic course respectively was provided to course ID text field as its value and other required text field are also filled. Then Add button was clicked
Expected result	A message dialog box will appears in Academic or Non Academic course indicating course ID already exist.
Actual Result	A message dialog box appears in Academic or Non Academic course showing Entered Course ID already exist.
Conclusion	Test was successful.

Table 8 : Tying to add duplicate course ID in both Academic and Non academic Course.

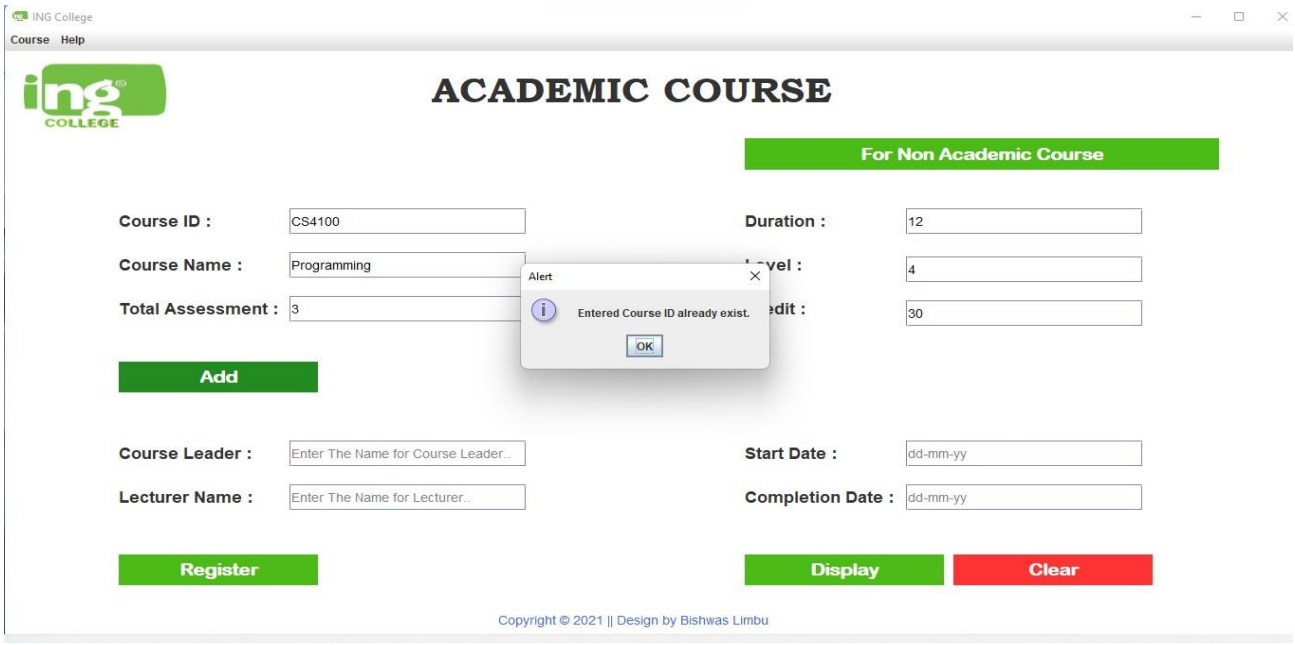

Figure 20 : Result after duplicate course Id is given to add course to Academic course.

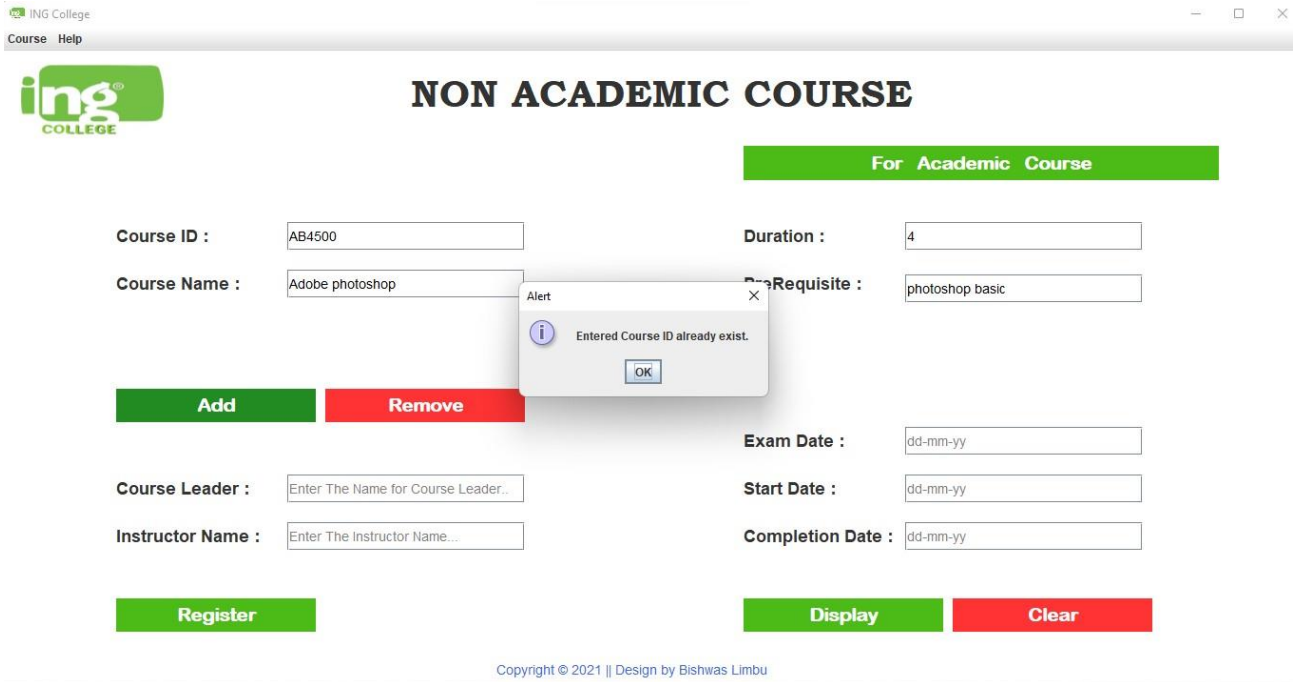

Figure 21 : Result after duplicate course Id is given to add course to Non Academic course.

5.3.2 Test 3.b.

Test No	3.b
Objective	Tying to register already register course in both Academic and Non academic Course.
Action	Course ID of Already registered course in Academic and Non academic course respectively was provided to course ID text field as its value and other required text field are filled. Then Register button was clicked
Expected result	A message dialog box will appears in Academic or Non Academic course indicating Academic or Non Academic course already exist respectively.
Actual Result	A message dialog box appears in Academic or Non Academic course showing Academic or Non Academic course already exist respectively.
Conclusion	Test was successful.

Table 9 : Tying to register already register course in both Academic and Non academic Course.

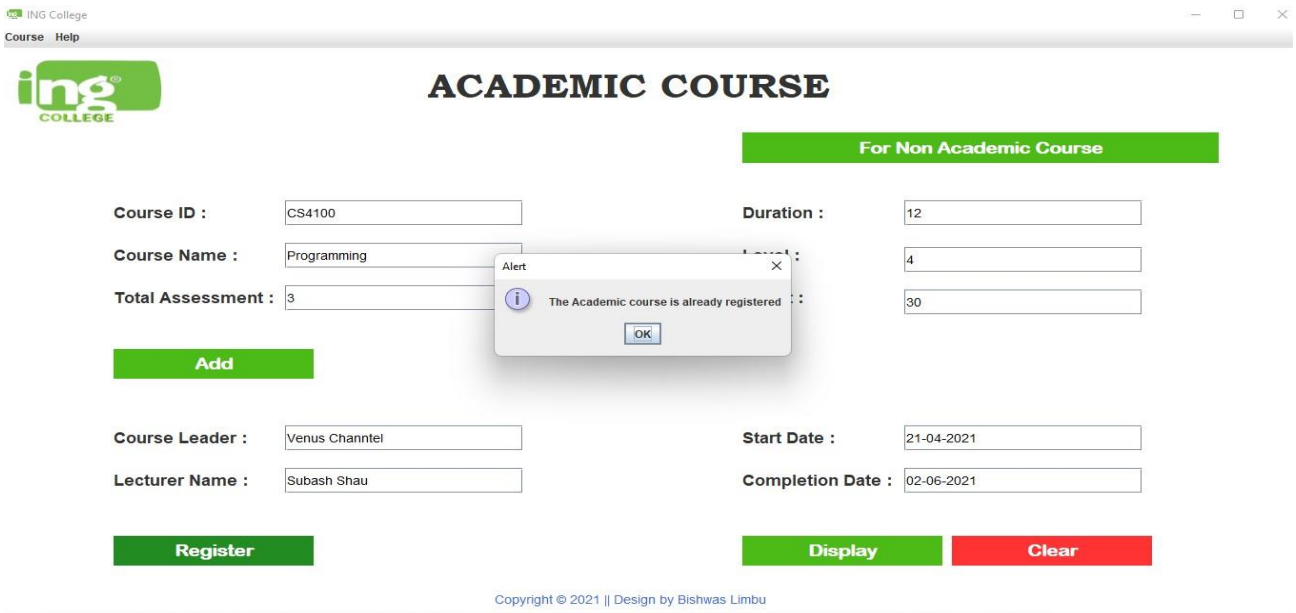

Figure 22 : Result after already registered course's course Id is given to register a course to Academic course.

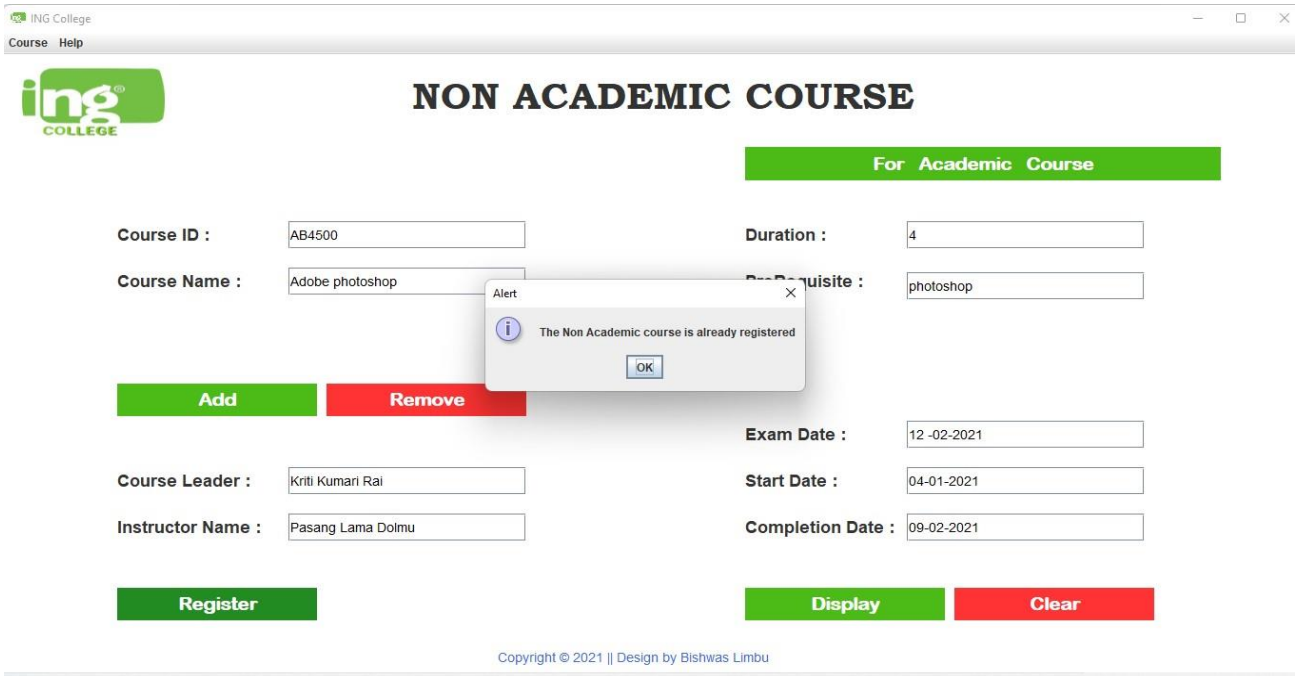

Figure 23 : Result after already registered course's course Id is given to register a course to Non Academic course.

5.3.3 Test 3.c

Test No	3.c
Objective	Trying to remove already removed Non Academic course.
Action	Course ID of Already removed Non academic course was provided to course ID text field as its value and Remove button was clicked
Expected result	A confirm dialog box appears asking user response. If user provide response “yes” then a message dialog indicating The course ID doesn’t exist will appear.
Actual Result	Message dialog box showing The course ID doesn’t exist was appeared after response “yes” was given to confirm dialog box.
Conclusion	Test was successful.

Table 10 : Trying to remove already removed Non Academic course.

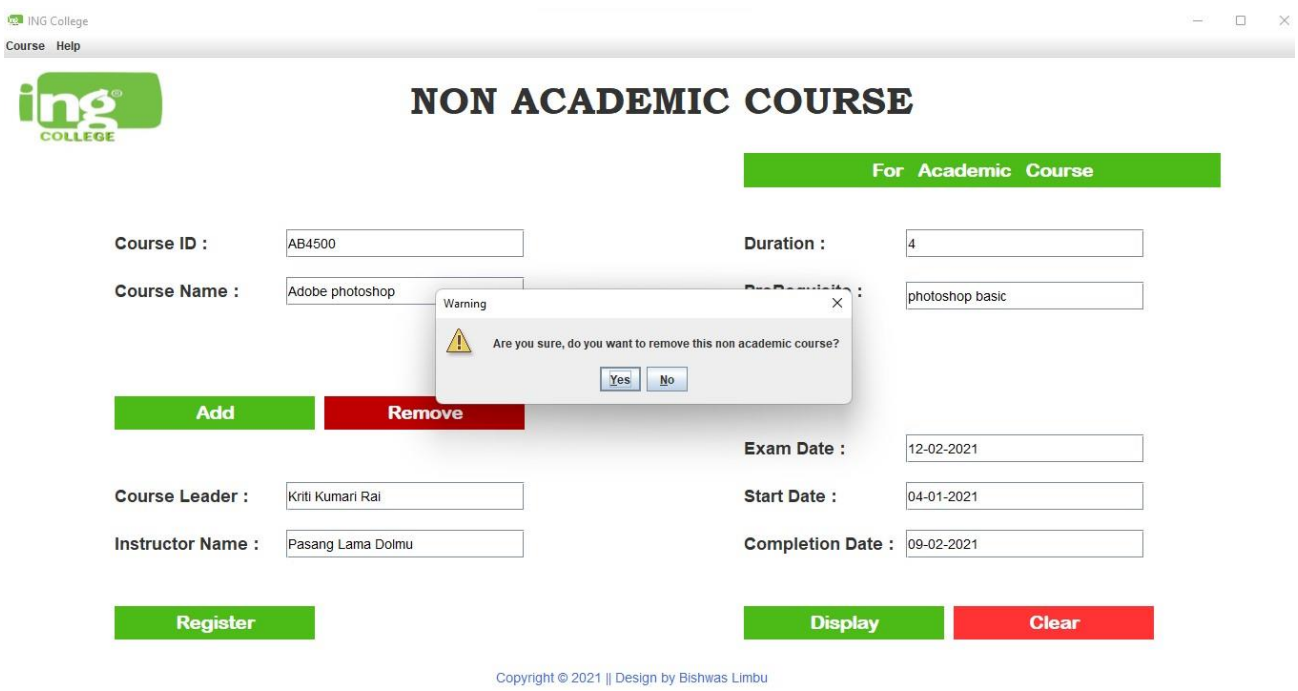

Figure 24 : Result after clicking Remove button for already removed Non Academic course.

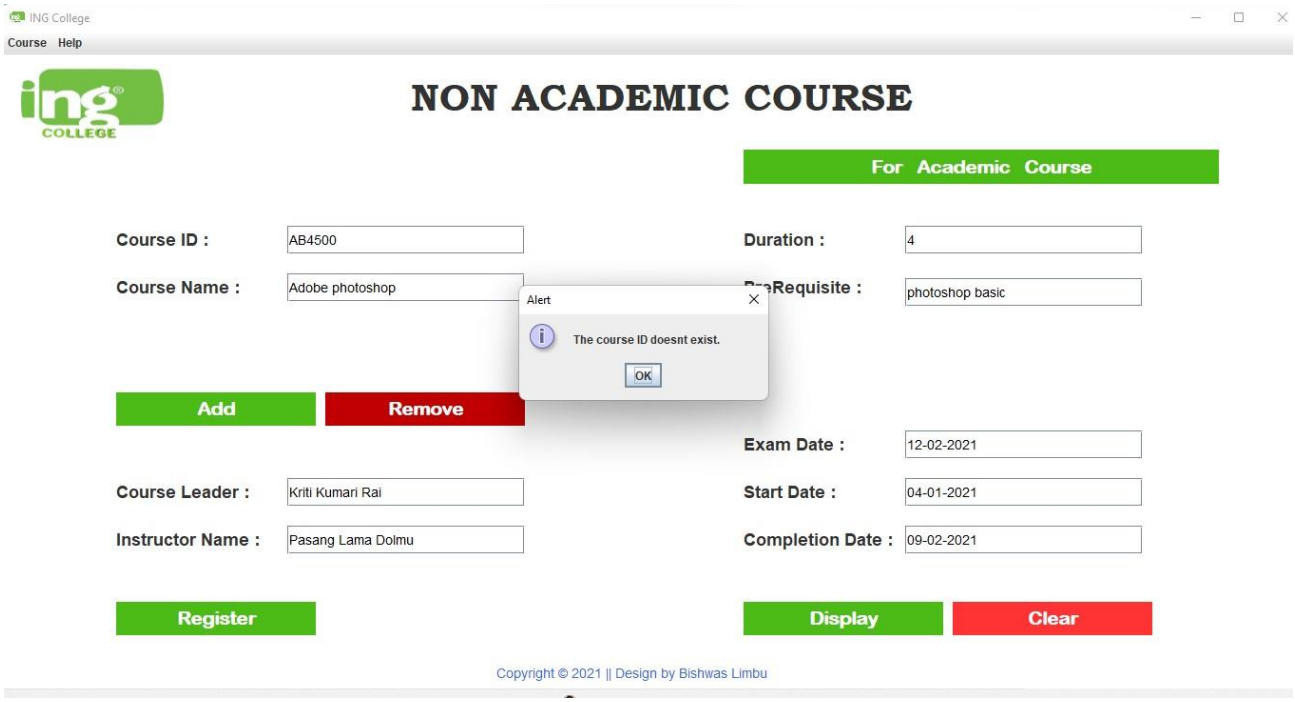

Figure 25 : Result alter user provide "yes" response to remove already removed Non Academic course.

6.Error detection and Correction

Various errors can come across while building an application. The error can be of any type like syntax error, semantic error, logical error and runtime error. This error occurs while editing codes, compiling code or at the running time. These types of error i.e., syntax error, semantic error and logical error are shown below which is faced while building this project.

6.1 Syntax error

A mistake in a program's source code known as syntax error. As computer programs must adhere to rigorous syntax in order to compile effectively, any parts of the code that do not compile the programming language's syntax will result in a syntax error. (techterms.com, 2021)

While declaring JMenuBar in the IngCollege class, its reference variable was missing in declaration of JMenuBar throwing syntax error.

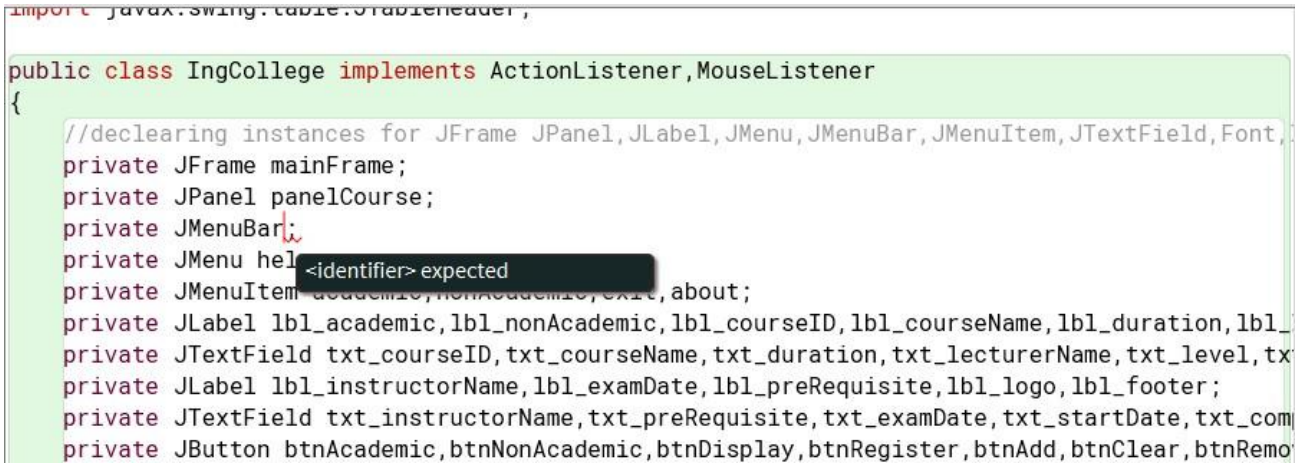

Figure 26 : Syntax error detection while declaring JMenuBar without its reference variable in class.

This error was corrected by placing reference variable to JMenuBar in class.

```
//declearing instances for JFrame JPanel,JLabel,JMenu,JMenuBar,JMenuItem,JTextField,Font,
private JFrame mainFrame;
private JPanel panelCourse;
private JMenuBar mB;
private JMenu help,course;
private JMenuItem academic,nonAcademic,exit,about;
private JLabel lbl_academic,lbl_nonAcademic,lbl_courseID,lbl_courseName,lbl_duration,lbl_
```

Figure 27 : Syntax error corrected by placing reference variable to JMenuBar.

6.2 Semantic error

Semantic error is those type of which are mostly detected at compile time. These types of error can arise when datatype incompatible or using wrong operator or executing operation in wrong order. (javatpoint.com, 2011-2021)

Here, Level is string type variable and duration is integer type variable which are passed in AcademicCourse constructor method as parameter which is invoked in add button using AcademicCourse class object for adding academic course. AcademicCourse constructor method accepts string in 1st parameter, string in 2nd parameter, integer in 3rd parameter, string in 4th parameter and string in remaining two parameters. Error occurs when level variable is passed in 3rd parameter and duration variable is passed in 4th parameter which throws incompatible data types error.

```
            return;
        }
    }
    acd = new AcademicCourse(courseID,courseName,level,duration,credit,numberOfAssessments);//passing value to parame
    academicList.add(acd);//adding AcademicCourse
    JOptionPane.showMessageDialog(null,"Academic
}
```

incompatible types: java.lang.String cannot be converted to int

Figure 28 : Semantic error due to incompatible data types.

This error is corrected by passing variable according to the types of parameters accepted by AcademicCourse constructor method.

```
}
acd = new AcademicCourse(courseID,courseName,duration,level,credit,numberOfAssessments);//passing value to parame
academicList.add(acd);//adding AcademicCourse object in arraylist
JOptionPane.showMessageDialog(null,"Academic Course is successfully added.","Alert",JOptionPane.INFORMATION_MESSA

}

/checking visibility boolean value for lable and button for using same button for  non academic course
```

Figure 29 : Semantic error is corrected by passing proper types of parameters in constructor method.

6.3 Logical error

A mistake in the source code of a program that causes it to behave incorrectly or unexpectedly is known as a logical error. It's a form of runtime error which can cause a program to crash or just generate the incorrect output. Such types of error are very hard to debug. (techterms.com, 2021)

Here, while trying to display add course in academic course, a logical error arise which did not give correct input values for level, total assessment, credit in table when displayed. Level value given in text field was 5 and total assessment was 3 and credit was 6 but at the time of displaying output incorrect output was displayed.

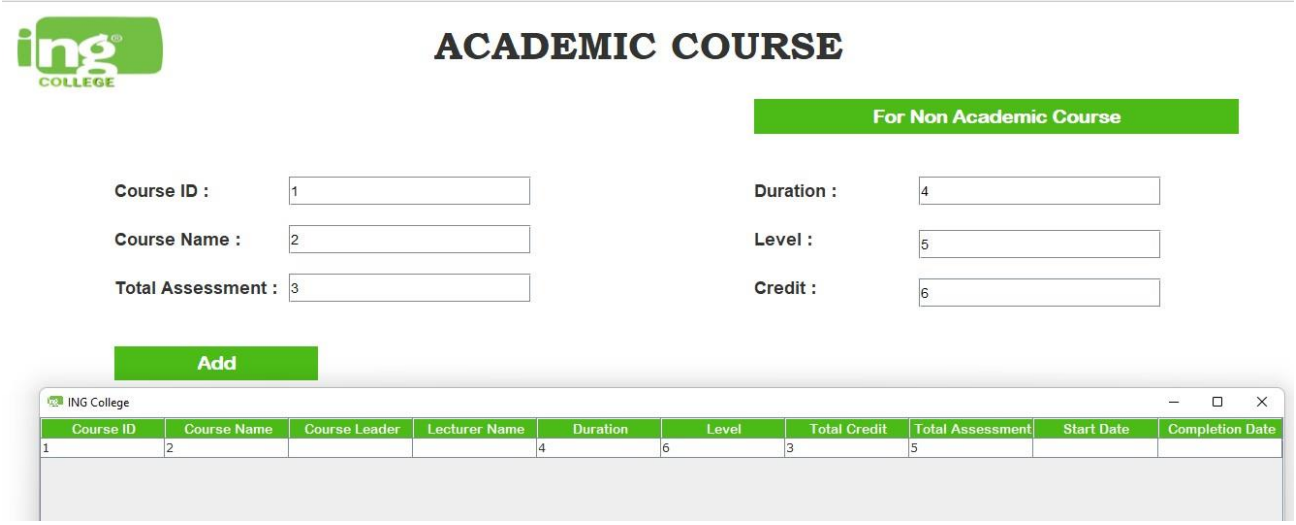

Figure 30 : Incorrect output displayed due to logical error.

```
for(int i = 0; i < academicList.size(); i++){
    //downcasting from Course to AcademicCourse type for accessing
    acd = (AcademicCourse)academicList.get(i);
    //creating data for row section of Table and accessing data fro
    Object data[]= {academicList.get(i).getCourseID(),
                    academicList.get(i).getCourseName(),
                    academicList.get(i).getCourseLeader(),
                    acd.getLecturerName(),
                    academicList.get(i).getDuration(),
                    acd.getCredit(),
                    acd.getNumberOfAssessments(),
                    acd.getLevel(),
                    acd.getStartingDate(),
                    acd.getCompletionDate()
                    };
    tableModel.addRow(data);//adding data list into row section of t
    }
```

Figure 31 : This code for fetching data from academic course list to display output.

This logical error was corrected by repositioning elements of Object data[] array according to elements of Object col[] array respectively as shown in figure below.

```
    //defining column of JTable with string value for column title
    Object col[]= {"Course ID","Course Name","Course Leader","Lecturer Name","Duration",
        "Level","Total Credit","Total Assessment","Start Date","Completion Date"};
    //intializing constructor of DefaultTableModel with two parameter i.e. column and rows c
```

Figure 32 : Positioning of elements in Object col[] array.

```
for(int i = 0; i < academicList.size(); i++){
    //downcasting from Course to AcademicCourse type for accessing child class method
    acd = (AcademicCourse)academicList.get(i);
    //creating data for row section of Table and accessing data from academiclist arraylist and academicCour
    Object data[]= {academicList.get(i).getCourseID(),
                    academicList.get(i).getCourseName(),
                    academicList.get(i).getCourseLeader(),
                    acd.getLecturerName(),
                    academicList.get(i).getDuration(),
                    acd.getLevel(),
                    acd.getCredit(),
                    acd.getNumberOfAssessments(),
                    acd.getStartingDate(),
                    acd.getCompletionDate()
                    };
    tableModel.addRow(data);//adding data list into row section of table model
    }
```

Figure 33 : After repositioning elements of Object data[] array in respect to elements of Object col[] array.

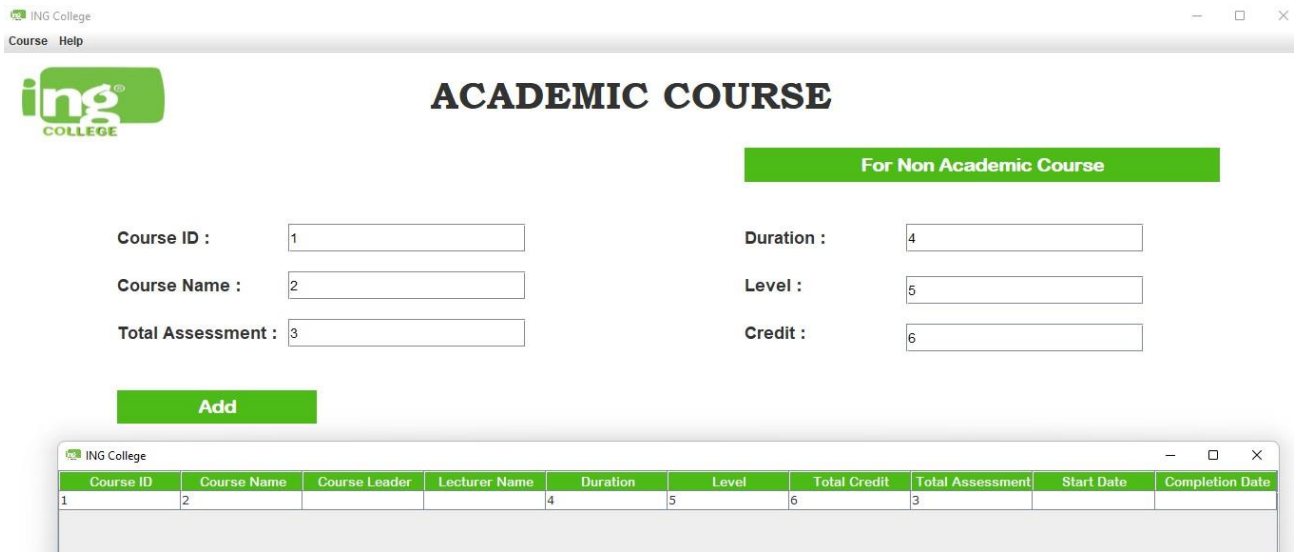

Figure 34 : Displaying output after correction of logical error.

7.Conclusion

At first, I would like to give thank you to my teacher for providing an opportunity to work on this coursework. I feel very luck to learn more about developing java application using GUI component. I have got improvement in better understanding for java GUI based application development as I know only to code in past. Here, I focus more on developing better user interface and making more user interactive. Different java swing and awt component were use to develop this GUI and fully functioning application. For, buttons on GUI different awt event like action event, mouse event and focus event were used for button functionality. Java Swing component like JLabel, Jframe, JText Field, JButton, JPanel and java awt component like color, Font, etc and image icon are uset to make GUI better. All the data are given by user which is then passed to respective text field and from the textfield data are retrieve and store in array list using different methods and event.

Here, in this coursework I got chance to know more about developing java-based GUI application. I have learned to implement many components of java Swing and awt like JFrame, JPane, Font, JButton, JLabel etc for making better GUI for application. I got knowledge on array list and how we can store Object data in array list and access them, perform operation like delete, remove, add, etc. I also came to know a lot about event handling that how we can use that in buttons, menu items etc to do certain action when they are clicked. Action event, Mouse event and Focus event are some components of awt event that I became mostly familiar. At last, I also learned how to about exception handling in program using try catch block and implementation of it in program.

Here, while developing this project I came across various errors like syntax, semantic, logical errors and run time error. I was having confuse to of developing pseudocode for the GUI component in application. I was also stuck at display button action performed method to display all added and registered courses in table form. I was having difficulties in understanding down casting of an object.

I overcome all such difficulties and completed my coursework. I did get ideas from our module teacher and did online research for writing pseudocode for GUI component of our project. I came to know about detection and correction of different types of error in program through module lecture guidance and online website like techopedia.com, javapoint.com etc. I was able to display my application output in table form by doing online research in javapoint.com and stackoverflow.com website. For down casting of object, I did guidance from my module teacher and did research on lecture slide and video materials.

References

bluej.org, 2021. *about.html.* [Online]
Available at: https://www.bluej.org/about.html (bluej intro)
[Accessed 19 august 2021].
Hope, C., 2021. *microsoft-word.htm.* [Online]
Available at: https://www.computerhope.com/jargon/m/microsoft-word.htm
[Accessed 19 august 2021].
Inc., T., 2021. *www.techopedia.com.* [Online]
Available at: https://www.techopedia.com/definition/16466/class-diagram
[Accessed 19 august 2021].
javatpoint.com, 2011-2021. *semantic-error.* [Online]
Available at: https://www.javatpoint.com/semantic-error
[Accessed 19 august 2021].
techterms.com, 2021. *logic_error.* [Online]
Available at: https://techterms.com/definition/logic_error
[Accessed 19 august 2021].
techterms.com, 2021. *syntax_error.* [Online]
Available at: https://techterms.com/definition/syntax_error
[Accessed 19 august 2021].
unf.edu, 2021. *www.unf.edu.* [Online]
Available at: ttps://www.unf.edu/~broggio/cop2221/2221pseu.htm
[Accessed 19 august 2021].

Appendix

- **IngCollege Class**

```
/**
 * IngCollege class for GUI .
 * @author (Bishwas Limbu)
 * @Group N3
 * @College ID (NP01NT4S210077)
 * Report File Name : 20049443 BISHWAS LIMBU
 */
//importing packages from java libaries
import javax.swing.*;
import java.awt.Font;
import java.awt.Color;
import java.awt.event.*;
import java.util.ArrayList;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.JTableHeader;

public class IngCollege implements ActionListener,MouseListener
{
    //declearing instances for JFrame
JPanel,JLabel,JMenu,JMenuBar,JMenuItem,JTextField,Font,ImageIcon
    private JFrame mainFrame;
    private JPanel panelCourse;
    private JMenuBar mB;
    private JMenu help,course;
    private JMenuItem academic,nonAcademic,exit,about;
    private JLabel
lbl_academic,lbl_nonAcademic,lbl_courseID,lbl_courseName,lbl_duration,lbl_lecturerNam
e,lbl_level,lbl_credit,lbl_startDate,lbl_completionDate,lbl_numberOfAssessment,lbl_course
Leader;
    private JTextField
txt_courseID,txt_courseName,txt_duration,txt_lecturerName,txt_level,txt_credit,txt_number
OfAssessment,txt_courseLeader;
    private JLabel lbl_instructorName,lbl_examDate,lbl_preRequisite,lbl_logo,lbl_footer;
    private JTextField
txt_instructorName,txt_preRequisite,txt_examDate,txt_startDate,txt_completionDate;
    private JButton
btnAcademic,btnNonAcademic,btnDisplay,btnRegister,btnAdd,btnClear,btnRemove;
    private Font flbl1,ff,ftxt,fMnu,fBtn;
    private ImageIcon titleIcon;
```

```
    //Creating reference variable of AcademicCourse and NonAcademicCourse
    public static AcademicCourse acd;
    public static NonAcademicCourse nacd;
    //creating reference object of ArrayList of Course class type
    public ArrayList<Course>academicList = new ArrayList<Course>();
    public ArrayList<Course>nonAcademicList = new ArrayList<Course>();

    public IngCollege(){
        mainFrame = new JFrame();//initializing JFrame

        //initializing and setting panel's layout,position,size,background color,focusable
        panelCourse = new JPanel();
        panelCourse.setLayout(null);
        panelCourse.setBackground(new Color(255,255,255));
        panelCourse.setFocusable(true);//sets foucs of mouse to panel when frame is loaded
        panelCourse.setSize(1650,1080);

        //defining Font attibutes as parameter in Font class constructor for
textfield,label,button
        flbl1 = new Font("Bookman Old Style",Font.BOLD,40);
        ff = new Font("Arial",Font.BOLD,18);
        ftxt = new Font("Calibri (Body)",Font.PLAIN,14);
        fMnu = new Font("Calibri (Body)",Font.PLAIN,14);
        fBtn = new Font("Arial Rounded MT Bold",Font.BOLD,20);

        //creating JMenuBar
        mB = new JMenuBar();

        //creating and adding JMenu to JMenuBar
        course = new JMenu("Course");
        help = new JMenu("Help");

       //creating adding JMenuItem to JMenu and implementing Action listener to
JMenuItems
        academic = new JMenuItem("Academic Course");
        nonAcademic = new JMenuItem("Non Academic Course");
        exit = new JMenuItem("Exit");
        about = new JMenuItem("about");
        //adding action listener to JMenuItems
        academic.addActionListener(this);
        nonAcademic.addActionListener(this);
        exit.addActionListener(this);
        about.addActionListener(this);
```

```
//initializing and setting title,fonts,size, color, visiblity,position,functionality of buttons//
btnAcademic = new JButton("For   Academic   Course");
btnAcademic.setBounds(780,100,500,36);
btnAcademic.setBorder(null);
btnAcademic.setFont(fBtn);
btnAcademic.setBackground(new Color(76,187,23));
btnAcademic.setForeground(Color.WHITE);
btnAcademic.setVisible(false);
//adding MouseListener and ActionListener interface
btnAcademic.addMouseListener(this);

btnAcademic.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent e){
      academicComponent();//calling method for academic component
   }
});


btnNonAcademic = new JButton(" For Non Academic Course ");
btnNonAcademic.setBounds(780,100,500,36);
btnNonAcademic.setBorder(null);
btnNonAcademic.setFont(fBtn);
btnNonAcademic.setBackground(new Color(76,187,23));
btnNonAcademic.setForeground(Color.WHITE);
//adding MouseListener and ActionListener interface
btnNonAcademic.addMouseListener(this);

btnNonAcademic.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent e){
     nonAcademicComponent();//calling method for non academic component
   }
});

btnDisplay = new JButton("Display");
btnDisplay.setBounds(780,575,210,35);
btnDisplay.setBorder(null);
btnDisplay.setBackground(new Color(76,187,23));
btnDisplay.setForeground(Color.WHITE);
btnDisplay.setFont(fBtn);
//adding MouseListener and ActionListener interface
btnDisplay.addMouseListener(this);

btnDisplay.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent e){
      if(txt_courseID.getText().equals("Enter The Course ID.....")){
```

```
                //displays message dialog box when courseId text field equals "Enter The
Course ID....."
                JOptionPane.showMessageDialog(null,"Please, \nFill  Course ID text field
present....","Alert",JOptionPane.WARNING_MESSAGE);
              }
              else{
                JFrame frame = new JFrame();//initializing JFrame for display purpose
                 //chceking visibility boolean value for lable and button for using same button
for academic  course
                if(lbl_academic.isVisible() == true || btnNonAcademic.isVisible() == true){
                   //defining column of JTable with string value for column title
                   Object col[]= {"Course ID","Course Name","Course Leader","Lecturer
Name","Duration",
                      "Level","Total Credit","Total Assessment","Start Date","Completion
Date"};
                   //intializing constructor of DefaultTableModel with two parameter i.e. column
and rows count values.
                   DefaultTableModel tableModel = new DefaultTableModel(col, 0);
                   JTable table = new JTable(tableModel);//initializing JTable with tabelModle
instance object as parameter for creating table
                   Font font = new Font("Verdana", Font.PLAIN, 12);
                   table.setFont(font);//setting font
                   table.setRowHeight(20);//setting height of row
                   JScrollPane scrollPane = new JScrollPane(table);//initializing JScrollPane in
table

                   for(int i = 0; i < academicList.size(); i++){
                      //downcasting from Course to AcademicCourse type for accessing child
class method
                      acd = (AcademicCourse)academicList.get(i);
                      //creating data for row section of Table and accessing data from
academiclist arraylist and academicCourse
                      Object data[]= {academicList.get(i).getCourseID(),
                                  academicList.get(i).getCourseName(),
                                  academicList.get(i).getCourseLeader(),
                                  acd.getLecturerName(),
                                  academicList.get(i).getDuration(),
                                  acd.getLevel(),
                                  acd.getCredit(),
                                  acd.getNumberOfAssessments(),
                                  acd.getStartingDate(),
                                  acd.getCompletionDate()
                                  };
                      tableModel.addRow(data);//adding data list into row section of table
model
```

```
                }
            frame.getContentPane().add(scrollPane); //adding scrollpane to frame
            JTableHeader tableHeader = table.getTableHeader();//intializing
JTabelHeader to access table header
            tableHeader.setBackground(new Color(76,187,23));//setting background
color
            tableHeader.setForeground(Color.white);//setting foreground color
            Font headerFont = new Font("Arial Rounded MT Bold", Font.PLAIN,
14);//untializing and define font attributes
            tableHeader.setFont(headerFont);//setting font of table header
          }
          else { //chceking visibility boolean value for lable and button for using same
button for  non academic course
            if(lbl_nonAcademic.isVisible() == true || btnAcademic.isVisible() == true){
              //defining column of JTable with string value for column title
              Object col[] = {"Course ID","Course Name","Course Leader","Instructor
Name","Duration",
                    "preRequisite","Exam Date","Start Date","Completion Date"};
              //intializing constructor of DefaultTableModel with two parameter i.e.
column and rows count values.
              DefaultTableModel tableModel = new DefaultTableModel(col, 0);
              JTable table = new JTable(tableModel);//initializing JTable with
tabelModle instance object as parameter for creating table
              Font font = new Font("Verdana", Font.PLAIN, 12);
              table.setFont(font);//setting font
              table.setRowHeight(20);
              JScrollPane scrollPane = new JScrollPane(table);//initializing JScrollPane
in table

              for(int i = 0; i < nonAcademicList.size(); i++){
                //downcasting from Course to NonAcademicCourse type for accessing
child class method
                nacd = (NonAcademicCourse)nonAcademicList.get(i);
                //creating data for row section of Table and accessing data from
nonacademiclist arraylist and NonacademicCourse
                Object data[] = {nonAcademicList.get(i).getCourseID(),
                          nonAcademicList.get(i).getCourseName(),
                          nonAcademicList.get(i).getCourseLeader(),
                          nacd.getInstructorName(),
                          nonAcademicList.get(i).getDuration(),
                          nacd.getPreRequisite(),
                          nacd.getExamDate(),
                          nacd.getStartDate(),
                          nacd.getCompletionDate()
                          };
```

```
                    tableModel.addRow(data);//adding data list into row section of table
model
                }
                frame.getContentPane().add(scrollPane); //adding scrollpane to frame
                JTableHeader tableHeader = table.getTableHeader();//intializing
JTabelHeader to access table header
                tableHeader.setBackground(new Color(76,187,23));//setting background
color
                tableHeader.setForeground(Color.white);//setting foreground color
                Font headerFont = new Font("Arial Rounded MT Bold", Font.PLAIN,
14);//untializing and define font attributes
                tableHeader.setFont(headerFont);//setting font of table header
              }
            }
            //setting icon image,title,location,size,visible,colseoperation of frame
            titleIcon = new ImageIcon("icon.jpg");//initializing reference object of
ImageIcon class with image location as parameter
            frame.setIconImage(titleIcon.getImage());
            frame.setTitle("ING College");
            frame.setLocation(40,50);
            frame.setSize(1300,600);
            frame.setVisible(true);
            frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

          }
        }
    });

    btnRegister = new JButton("Register");
    btnRegister.setBounds(120,575,210,35);
    btnRegister.setBackground(new Color(76,187,23));
    btnRegister.setBorder(null);
    btnRegister.setForeground(Color.WHITE);
    btnRegister.setFont(fBtn);
    //adding MouseListener and ActionListener interface
    btnRegister.addMouseListener(this);

    btnRegister.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e){
            //chceking visibility boolean value for lable and button for using same button for
academic  course
          if(lbl_academic.isVisible() == true || btnNonAcademic.isVisible() == true){
            if(txt_courseID.getText().equals("Enter The Course
ID.....")||txt_courseLeader.getText().equals("Enter The Name for Course
Leader..")||txt_lecturerName.getText().equals("Enter The Name for Lecturer..")
```

```
                ||txt_startDate.getText().equals("dd-mm-
yy")||txt_completionDate.getText().equals("dd-mm-yy")){
                    //displays message dialog if condition is met
                    JOptionPane.showMessageDialog(null,"Please, \nFill all the required text
field present....","Alert",JOptionPane.WARNING_MESSAGE);
                }
                else{
                    //declaring variable and assigning text field value to same variable
                    String courseID = txt_courseID.getText();
                    String startDate = txt_startDate.getText();
                    String completionDate =  txt_completionDate.getText();
                    String courseLeader = txt_courseLeader.getText();
                    String lecturerName = txt_lecturerName.getText();

                    for(int i = 0; i < academicList.size(); i++){
                        //checks if user entered course id is already in arraylist
                        if(academicList.get(i).getCourseID().equals(courseID)){
                            AcademicCourse ac =
(AcademicCourse)academicList.get(i);//downcasting from Course to AcademicCourse type
for accessing child class method
                            if(acd.getIsRegistered() == false){

acd.register(courseLeader,lecturerName,startDate,completionDate);//passing value to
parameter of register method
                                JOptionPane.showMessageDialog(null," The Academic course is
successfully registered. ","Alert",JOptionPane.INFORMATION_MESSAGE);
                                return;

                            }
                            else{
                                JOptionPane.showMessageDialog(null,"The Academic course is
already registered","Alert",JOptionPane.INFORMATION_MESSAGE);
                                return;
                            }

                        }

                    }
                    JOptionPane.showMessageDialog(null,"The course ID doesnot
exist","Alert",JOptionPane.ERROR_MESSAGE);
                    return;
                }
            }
            else{
```

```
                //chceking visibility boolean value for lable and button for using same button
for non academic course
                if(lbl_nonAcademic.isVisible() == true || btnAcademic.isVisible() == true){
                    if(txt_courseID.getText().equals("Enter The Course
ID.....")||txt_courseLeader.getText().equals("Enter The Name for Course
Leader..")||txt_instructorName.getText().equals("Enter The Instructor Name...")
                ||txt_startDate.getText().equals("dd-mm-
yy")||txt_completionDate.getText().equals("dd-mm-
yy")||txt_examDate.getText().equals("dd-mm-yy")){
                    //displays message dialog if condition is met
                    JOptionPane.showMessageDialog(null,"Please, \nFill all the required text
field present....","Alert",JOptionPane.WARNING_MESSAGE);
                }
                else{
                    //declaring variable and assigning text field value to same variable
                    String courseID = txt_courseID.getText();
                    String startDate = txt_startDate.getText();
                    String examDate = txt_examDate.getText();
                    String completionDate =  txt_completionDate.getText();
                    String courseLeader = txt_courseLeader.getText();
                    String instructorName = txt_instructorName.getText();

                    for(int i = 0; i < nonAcademicList.size(); i++){
                        //checks if user entered course id is already in arraylist
                        if(nonAcademicList.get(i).getCourseID().equals(courseID)){
                            nacd = (NonAcademicCourse)nonAcademicList.get(i);//downcasting
from Course to NonAcademicCourse type for accessing child class method
                            if(nacd.getIsRegistered() == false){

nacd.register(courseLeader,instructorName,startDate,completionDate,examDate);//passing
value to parameter of register method
                                JOptionPane.showMessageDialog(null," The Non Academic course
is successfully registered. ","Alert",JOptionPane.INFORMATION_MESSAGE);
                              return;

                            }
                            else{
                              JOptionPane.showMessageDialog(null,"The Non Academic course is
already registered","Alert",JOptionPane.INFORMATION_MESSAGE);
                              return;
                            }

                        }
                    }
```

```
                    JOptionPane.showMessageDialog(null,"The course ID doesnot
exist","Alert",JOptionPane.ERROR_MESSAGE);
                    return;
                   }
                 }
              }
           }
        });


        btnAdd = new JButton("Add");
        btnAdd.setBounds(120,355,210,35);
        btnAdd.setBorder(null);
        btnAdd.setBackground(new Color(76,187,23));
        btnAdd.setForeground(Color.WHITE);
        btnAdd.setFont(fBtn);
        //adding MouseListener and ActionListener interface
        btnAdd.addMouseListener(this);
        btnAdd.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
               //checking visibility boolean value for lable and button for using same button for
academic course
              if(lbl_academic.isVisible() == true || btnNonAcademic.isVisible() == true){
                  if(txt_courseID.getText().equals("Enter The Course
ID.....")||txt_courseName.getText().equals("Enter The Course
Name...")||txt_duration.getText().equals("Enter the Duration For Course..")
                    ||txt_numberOfAssessment.getText().equals("Enter Total Number of
Assessment..")||txt_credit.getText().equals("Enter The Total credit For
Course....")||txt_level.getText().equals("Enter The Course Level....")){
                       //displays message dialog if condition is met
                       JOptionPane.showMessageDialog(null,"Please, \nFill all the required text
field present....","Alert",JOptionPane.WARNING_MESSAGE);
                    }
                   else{
                      //declaring variable and assigning text field value to same variable
                     String courseID = txt_courseID.getText();
                     String courseName =  txt_courseName.getText();
                     String credit = txt_credit.getText();
                     String level = txt_level.getText();
                     int duration = 0;
                     int numberOfAssessments = 0;
                     //using try and catch block NumberFormatException error handling
                     try{duration = Integer.parseInt(txt_duration.getText()); }//chaning string
datatype to integer datatype
                     catch(NumberFormatException d){
```

```
                    JOptionPane.showMessageDialog(null,"Please, \n give valid user
input...","Invalid duration datatype",JOptionPane.ERROR_MESSAGE);
                    return;
                }
                try{numberOfAssessments =
Integer.parseInt(txt_numberOfAssessment.getText()); }//chaning string datatype to integer
datatype
                catch(NumberFormatException n){
                    JOptionPane.showMessageDialog(null,"Please, \n give valid user
input...","Invalid numberOfAssessments datatype",JOptionPane.ERROR_MESSAGE);
                    return;
                }
                //iterating academicList
                for(Course c : academicList){
                    //checks if user entered course id is already in arraylist
                    if(c.getCourseID().equals(courseID)){
                        //displays message dialog if conditon is met
                        JOptionPane.showMessageDialog(null," Entered Course ID already
exist. ","Alert",JOptionPane.INFORMATION_MESSAGE);
                        return;
                    }
                }
                acd = new
AcademicCourse(courseID,courseName,duration,level,credit,numberOfAssessments);//pas
sing value to parameter of AcademicCourse constructor
                academicList.add(acd);//adding AcademicCourse object in arraylist
                JOptionPane.showMessageDialog(null,"Academic Course is successfully
added.","Alert",JOptionPane.INFORMATION_MESSAGE);

                }
            }
            else{ //checking visibility boolean value for lable and button for using same
button for  non academic course
                if(lbl_nonAcademic.isVisible() == true || btnAcademic.isVisible() == true){
                    if(txt_courseID.getText().equals("Enter The Course
ID.....")||txt_courseName.getText().equals("Enter The Course
Name...")||txt_duration.getText().equals("Enter the Duration For Course..")
                ||txt_preRequisite.getText().equals("Enter The  PreRequisite For Course..")){
                    //displays message dialog if condition is met
                    JOptionPane.showMessageDialog(null,"Please, \nFill all the required text
field present....","Alert",JOptionPane.WARNING_MESSAGE);
                }
                else{//declaring variable and assigning text field value to same variable
                    String courseID = txt_courseID.getText();
                    String courseName =  txt_courseName.getText();
```

```
                String preRequisite = txt_preRequisite.getText();
                int duration = 0;
                 //using try and catch block NumberFormatException error handling
                try{duration = Integer.parseInt(txt_duration.getText()); }//chaning string
datatype to integer datatype
                catch(NumberFormatException d){
                    JOptionPane.showMessageDialog(null,"Please, \n give valid user
input...","Invalid duration datatype",JOptionPane.ERROR_MESSAGE);
                    return;
                }
                 //iterating nonacademicList
                for(Course c : nonAcademicList){
                    //checks if user entered course id is already in arraylist
                    if(c.getCourseID().equals(courseID)){
                        //displays message dialog if conditon is met
                         JOptionPane.showMessageDialog(null," Entered Course ID already
exist. ","Alert",JOptionPane.INFORMATION_MESSAGE);
                            return;
                    }
                }
                nacd = new
NonAcademicCourse(courseID,courseName,duration,preRequisite);//passing value to
parameter of NonAcademicCourse constructor
                nonAcademicList.add(nacd);//adding NonAcademicCourse object in
arraylist
                JOptionPane.showMessageDialog(null,"Non Academic Course is
successfully added.","Alert",JOptionPane.INFORMATION_MESSAGE);

                }

            }
        }
    }
});


btnClear = new JButton("Clear");
btnClear.setBounds(1000,575,210,35);
btnClear.setBorder(null);
btnClear.setBackground(new Color(255,51,51));
btnClear.setForeground(Color.WHITE);
btnClear.setFont(fBtn);
 //adding MouseListener and ActionListener interface
btnClear.addMouseListener(this);
btnClear.addActionListener(new ActionListener(){
```

```
            public void actionPerformed(ActionEvent e){
                int response = JOptionPane.showConfirmDialog(null," Are you sure, do you
want to clear all text field? ",
"Warning",JOptionPane.YES_NO_OPTION,JOptionPane.WARNING_MESSAGE);
                 //clears all textields if condition is met4
                if(response == JOptionPane.YES_OPTION){
                    //setting some text in the text Field which act as place holder
                    txt_courseID.setText("Enter The Course ID.....");
                    txt_courseName.setText("Enter The Course Name...");
                    txt_duration.setText("Enter the Duration For Course..");
                    txt_lecturerName.setText("Enter The Name for Lecturer..");
                    txt_level.setText("Enter The Course Level....");
                    txt_credit.setText("Enter The Total credit For Course....");
                    txt_numberOfAssessment.setText("Enter Total Number of  Assessment..");
                    txt_courseLeader.setText("Enter The Name for Course Leader..");
                    txt_startDate.setText("dd-mm-yy");
                    txt_completionDate.setText("dd-mm-yy");
                    txt_preRequisite.setText("Enter The  PreRequisite For Course..");
                    txt_instructorName.setText("Enter The Instructor Name...");
                    txt_examDate.setText("dd-mm-yy");

                    //setting color to text in the text Field
                    txt_courseID.setForeground(Color.GRAY);
                    txt_courseName.setForeground(Color.GRAY);
                    txt_duration.setForeground(Color.GRAY);
                    txt_courseLeader.setForeground(Color.GRAY);
                    txt_startDate.setForeground(Color.GRAY);
                    txt_completionDate.setForeground(Color.GRAY);
                    txt_numberOfAssessment.setForeground(Color.GRAY);
                    txt_level.setForeground(Color.GRAY);
                    txt_credit.setForeground(Color.GRAY);
                    txt_lecturerName.setForeground(Color.GRAY);
                    txt_instructorName.setForeground(Color.GRAY);
                    txt_examDate.setForeground(Color.GRAY);
                    txt_preRequisite.setForeground(Color.GRAY);

                    JOptionPane.showMessageDialog(null,"All Text Field are succcessfully
Cleared.","Alert",JOptionPane.INFORMATION_MESSAGE);
                }
            }
        });

        btnRemove = new JButton("Remove");
        btnRemove.setBounds(340,355,210,35);
        btnRemove.setBorder(null);
```

```
        btnRemove.setBackground(new Color(255,51,51));
        btnRemove.setForeground(Color.WHITE);
        btnRemove.setFont(fBtn);
        btnRemove.setVisible(false);
         //adding MouseListener and ActionListener interface
        btnRemove.addMouseListener(this);

        btnRemove.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){

                        if(txt_courseID.getText().equals("Enter The Course ID.....")){
                        //displays message dialog if condition is met 4
                       JOptionPane.showMessageDialog(null,"Please, \nFill Course ID text field
present....","Alert",JOptionPane.WARNING_MESSAGE);
                    }
                    else{
                        int response = JOptionPane.showConfirmDialog(null," Are you sure, do you
want to remove this non academic course? ",
"Warning",JOptionPane.YES_NO_OPTION,JOptionPane.WARNING_MESSAGE);
                        if(response == JOptionPane.YES_OPTION){
                           //declaring variable and assigning text field value to same variable
                            String courseID = txt_courseID.getText();
                            //iterating nonacademicList
                            for(int i = 0; i < nonAcademicList.size(); i++){
                               //checks if user entered course id is already in arraylist
                               if(nonAcademicList.get(i).getCourseID().equals(courseID)){
                                  nacd = (NonAcademicCourse)nonAcademicList.get(i);//downcasting
from Course to NonAcademicCourse type for accessing child class method
                                   if(nacd.getIsRemoved() == false){
                                      nacd.remove();//calling remove method of NonAcademicCourse
                                      nonAcademicList.remove(i);//removing object in i index
                                      JOptionPane.showMessageDialog(null,"Non Academic Course is
succcessfully removed.","Alert",JOptionPane.INFORMATION_MESSAGE);
                                      return;
                                   }

                               }

                           }
                           JOptionPane.showMessageDialog(null,"The course ID doesnt
exist.","Alert",JOptionPane.INFORMATION_MESSAGE);
                           return;

                    }
```

```
            }
        }
    });

    //initializing and setting title,fonts,size, color, visiblity,position of JLabel//
    lbl_academic = new JLabel("ACADEMIC COURSE");
    lbl_academic.setBounds(450,30,500,30);
    lbl_academic.setFont(flbl1);


    lbl_nonAcademic = new JLabel("NON ACADEMIC COURSE");
    lbl_nonAcademic.setBounds(430,30,600,30);
    lbl_nonAcademic.setFont(flbl1);
    lbl_nonAcademic.setVisible(false);

    lbl_logo = new JLabel((new ImageIcon("icon1.png")));//inserting image in label and
adding it to panel
    lbl_logo .setBounds(20,0,150,100);

    lbl_footer = new JLabel("Copyright Â© 2021 || Design by Bishwas Limbu");
    lbl_footer .setBounds(520,600,700,100);
    Font fFooter = new Font("Arial",Font.PLAIN,14);
    lbl_footer.setForeground(new Color(65, 105, 225));
    lbl_footer .setFont(fFooter);

    lbl_courseID = new JLabel("Course ID : ");
    lbl_courseID.setBounds(120,185,200,20);
    lbl_courseID.setFont(ff);

    lbl_courseName = new JLabel("Course Name : ");
    lbl_courseName.setBounds(120,235,300,20);
    lbl_courseName.setFont(ff);

    lbl_duration = new JLabel("Duration : ");
    lbl_duration.setBounds(780,185,100,20);
    lbl_duration.setFont(ff);


    lbl_lecturerName = new JLabel("Lecturer Name : ");
    lbl_lecturerName.setBounds(120,500,200,20);
    lbl_lecturerName.setFont(ff);


    lbl_level = new JLabel("Level : ");
```

```
lbl_level.setBounds(780,235,90,20);
lbl_level.setFont(ff);

lbl_credit = new JLabel("Credit : ");
lbl_credit.setBounds(780,285,90,20);
lbl_credit.setFont(ff);

lbl_startDate = new JLabel("Start Date : ");
lbl_startDate.setBounds(780,450,150,20);
lbl_startDate.setFont(ff);

lbl_completionDate = new JLabel("Completion Date : ");
lbl_completionDate.setBounds(780,500,200,20);
lbl_completionDate.setFont(ff);

lbl_numberOfAssessment = new JLabel("Total Assessment : ");
lbl_numberOfAssessment.setBounds(120,285,200,20);
lbl_numberOfAssessment.setFont(ff);

lbl_instructorName = new JLabel("Instructor Name : ");
lbl_instructorName.setBounds(120,500,200,20);
lbl_instructorName.setFont(ff);
lbl_instructorName.setVisible(false);

lbl_examDate = new JLabel("Exam Date : ");
lbl_examDate.setBounds(780,400,120,20);
lbl_examDate.setFont(ff);
lbl_examDate.setVisible(false);


lbl_preRequisite = new JLabel("PreRequisite : ");
lbl_preRequisite.setBounds(780,235,130,20);
lbl_preRequisite.setFont(ff);
lbl_preRequisite.setVisible(false);

lbl_courseLeader = new JLabel("Course Leader : ");
lbl_courseLeader.setBounds(120,450,200,20);
lbl_courseLeader.setFont(ff);


//setting title,fonts,size, color, visiblity,position of JTextField//
txt_courseID = new JTextField(20);
txt_courseID.setText("Enter The Course ID.....");
txt_courseID.setBounds(300,180,250,30);
txt_courseID.setFont(ftxt);
```

```
        txt_courseID.setForeground(Color.GRAY);
        //adding FoucusListener to textField
        txt_courseID.addFocusListener(new FocusListener(){
            //performs action when mouse foucus in gained by textfield
            public void focusGained(FocusEvent evt) {
                if(txt_courseID.getText().equals("Enter The Course ID.....")){
                    txt_courseID.setText(""); //clears or make empty text field if condition is ture
                }
                txt_courseID.setForeground(Color.BLACK); //changes color of text in text field
            }
            //performs action when mouse foucus in lost by textfield
            public void focusLost(FocusEvent evt) {
                if(txt_courseID.getText().equals("") || txt_courseID.getText().equals("Enter The
Course ID....."))
                {
                    txt_courseID.setForeground(Color.GRAY);//changes color of text in text field
                    txt_courseID.setText("Enter The Course ID.....");//set text field with text
                }
            }
        });

        txt_courseName = new JTextField(20);
        txt_courseName.setBounds(300,230,250,30);
        txt_courseName.setFont(ftxt);
        txt_courseName.setText("Enter The Course Name...");
        txt_courseName.setForeground(Color.GRAY);
        //adding FoucusListener to textField
        txt_courseName.addFocusListener(new FocusListener(){
            //performs action when mouse foucus in gained by textfield
            public void focusGained(FocusEvent evt) {
                if(txt_courseName.getText().equals("Enter The Course Name...")){
                    txt_courseName.setText(""); //clears or make empty text field if condition is
ture
                }
                txt_courseName.setForeground(Color.BLACK); //changes color of text in text
field
            }
            //performs action when mouse foucus in lost by textfield
            public void focusLost(FocusEvent evt) {
                if(txt_courseName.getText().equals("") ||
txt_courseName.getText().equals("Enter The Course Name..."))
                {
                    txt_courseName.setForeground(Color.GRAY);//changes color of text in text
field
                    txt_courseName.setText("Enter The Course Name...");//set text field with text
```

```
            }
        }
    });

    txt_duration = new JTextField(20);
    txt_duration.setBounds(950,180,250,30);
    txt_duration.setFont(ftxt);
    txt_duration.setText("Enter the Duration For Course..");
    txt_duration.setForeground(Color.GRAY);
    //adding FoucusListener to textField
    txt_duration.addFocusListener(new FocusListener(){
        //performs action when mouse foucus in gained by textfield
        public void focusGained(FocusEvent evt) {
            if(txt_duration.getText().equals("Enter the Duration For Course..")){
                txt_duration.setText("");//clears or make empty text field if condition is ture
            }
            txt_duration.setForeground(Color.BLACK); //changes color of text in text field
        }
        //performs action when mouse foucus in lost by textfield
        public void focusLost(FocusEvent evt) {
            if(txt_duration.getText().equals("") || txt_duration.getText().equals("Enter the
Duration For Course.."))
            {
                txt_duration.setForeground(Color.GRAY);//changes color of text in text field
                txt_duration.setText("Enter the Duration For Course..");//set text field with text
            }
        }
    });

    txt_lecturerName = new JTextField(20);
    txt_lecturerName.setBounds(300,495,250,30);
    txt_lecturerName.setFont(ftxt);
    txt_lecturerName.setText("Enter The Name for Lecturer..");
    txt_lecturerName.setForeground(Color.GRAY);
    //adding FoucusListener to textField
    txt_lecturerName.addFocusListener(new FocusListener(){
        //performs action when mouse foucus in gained by textfield
        public void focusGained(FocusEvent evt) {
            if(txt_lecturerName.getText().equals("Enter The Name for Lecturer..")){
                txt_lecturerName.setText("");//clears or make empty text field if condition is
ture
            }
            txt_lecturerName.setForeground(Color.BLACK);//changes color of text in text
field
        }
```

```
        //performs action when mouse foucus in lost by textfield
        public void focusLost(FocusEvent evt) {
            if(txt_lecturerName.getText().equals("") ||
txt_lecturerName.getText().equals("Enter the Total Duration For Course.."))
            {
                txt_lecturerName.setForeground(Color.GRAY);//changes color of text in text
field
                txt_lecturerName.setText("Enter The Name for Lecturer..");//set text field with
text
            }
        }
    });

    txt_instructorName = new JTextField(20);
    txt_instructorName.setBounds(300,495,250,30);
    txt_instructorName.setFont(ftxt);
    txt_instructorName.setText("Enter The Instructor Name...");
    txt_instructorName.setVisible(false);
    txt_instructorName.setForeground(Color.GRAY);
    //adding FoucusListener to textField
    txt_instructorName.addFocusListener(new FocusListener(){
        //performs action when mouse foucus in gained by textfield
        public void focusGained(FocusEvent evt) {
            if(txt_instructorName.getText().equals("Enter The Instructor Name...")){
                txt_instructorName.setText("");//clears or make empty text field if condition is
ture
            }
            txt_instructorName.setForeground(Color.BLACK); //changes color of text in text
field
        }
        //performs action when mouse foucus in lost by textfield
        public void focusLost(FocusEvent evt) {
            if(txt_instructorName.getText().equals("") ||
txt_instructorName.getText().equals("Enter The Instructor Name..."))
            {
                txt_instructorName.setForeground(Color.GRAY);//changes color of text in text
field
                txt_instructorName.setText("Enter The Instructor Name...");//set text field with
text
            }
        }
    });

    txt_level = new JTextField(20);
    txt_level.setBounds(950,235,250,30);
```

```
        txt_level.setFont(ftxt);
        txt_level.setText("Enter The Course Level....");
        txt_level.setForeground(Color.GRAY);
        //adding FoucusListener to textField
        txt_level.addFocusListener(new FocusListener(){
            //performs action when mouse foucus in gained by textfield
            public void focusGained(FocusEvent evt) {
                if(txt_level.getText().equals("Enter The Course Level....")){
                    txt_level.setText("");//clears or make empty text field if condition is ture
                }
                txt_level.setForeground(Color.BLACK);//changes color of text in text field
            }
            //performs action when mouse foucus in lost by textfield
            public void focusLost(FocusEvent evt) {
                if(txt_level.getText().equals("") || txt_level.getText().equals("Enter The Course
Level...."))
                {
                    txt_level.setForeground(Color.GRAY);//changes color of text in text field
                    txt_level.setText("Enter The Course Level....");//set text field with text
                }
            }
        });

        txt_credit = new JTextField(20);
        txt_credit.setBounds(950,285,250,30);
        txt_credit.setFont(ftxt);
        txt_credit.setText("Enter The Total credit For Course....");
        txt_credit.setForeground(Color.GRAY);
        //adding FoucusListener to textField
        txt_credit.addFocusListener(new FocusListener(){
            //performs action when mouse foucus in gained by textfield
            public void focusGained(FocusEvent evt) {
                if(txt_credit.getText().equals("Enter The Total credit For Course....")){
                    txt_credit.setText("");//clears or make empty text field if condition is ture
                }
                txt_credit.setForeground(Color.BLACK); //changes color of text in text field
            }
            //performs action when mouse foucus in lost by textfield
            public void focusLost(FocusEvent evt) {
                if(txt_credit.getText().equals("") || txt_credit.getText().equals("Enter The Total
credit For Course...."))
                {
                    txt_credit.setForeground(Color.GRAY);//changes color of text in text field
                    txt_credit.setText("Enter The Total credit For Course....");//set text field with
text
```

```
            }
        }
    });

    txt_numberOfAssessment = new JTextField(20);
    txt_numberOfAssessment.setBounds(300,280,250,30);
    txt_numberOfAssessment.setFont(ftxt);
    txt_numberOfAssessment.setText("Enter Total Number of  Assessment..");
    txt_numberOfAssessment.setForeground(Color.GRAY);
    //adding FoucusListener to textField
    txt_numberOfAssessment.addFocusListener(new FocusListener(){
        //performs action when mouse foucus in gained by textfield
        public void focusGained(FocusEvent evt) {
            if(txt_numberOfAssessment.getText().equals("Enter Total Number of
Assessment..")){
                txt_numberOfAssessment.setText("");//clears or make empty text field if
condition is ture
            }
            txt_numberOfAssessment.setForeground(Color.BLACK);//changes color of text
in text field
        }
        //performs action when mouse foucus in lost by textfield
        public void focusLost(FocusEvent evt) {
            if(txt_numberOfAssessment.getText().equals("") ||
txt_numberOfAssessment.getText().equals("Enter Total Number of  Assessment.."))
            {
                txt_numberOfAssessment.setForeground(Color.GRAY);//changes color of text
in text field
                txt_numberOfAssessment.setText("Enter Total Number of
Assessment..");//set text field with text
            }
        }
    });

    txt_preRequisite = new JTextField(20);
    txt_preRequisite.setBounds(950,235,250,30);
    txt_preRequisite.setFont(ftxt);
    txt_preRequisite.setText("Enter The  PreRequisite For Course..");
    txt_preRequisite.setVisible(false);
    txt_preRequisite.setForeground(Color.GRAY);
    //adding FoucusListener to textField
    txt_preRequisite.addFocusListener(new FocusListener(){
        //performs action when mouse foucus in gained by textfield
        public void focusGained(FocusEvent evt) {
            if(txt_preRequisite.getText().equals("Enter The  PreRequisite For Course..")){
```

```
                txt_preRequisite.setText(""); //clears or make empty text field if condition is
ture
            }
            txt_preRequisite.setForeground(Color.BLACK); //changes color of text in text
field
        }
        //performs action when mouse foucus in lost by textfield
        public void focusLost(FocusEvent evt) {
            if(txt_preRequisite.getText().equals("") || txt_preRequisite.getText().equals("Enter
The  PreRequisite For Course.."))
            {
                txt_preRequisite.setForeground(Color.GRAY);//changes color of text in text
field
                txt_preRequisite.setText("Enter The  PreRequisite For Course..");//set text
field with text

            }
        }
    });

    txt_courseLeader = new JTextField(20);
    txt_courseLeader.setBounds(300,445,250,30);
    txt_courseLeader.setFont(ftxt);
    txt_courseLeader.setText("Enter The Name for Course Leader..");
    txt_courseLeader.setForeground(Color.GRAY);
    //adding FoucusListener to textField
    txt_courseLeader.addFocusListener(new FocusListener(){
        public void focusGained(FocusEvent evt) {
            //performs action when mouse foucus in gained by textfield
            if(txt_courseLeader.getText().equals("Enter The Name for Course Leader..")){
                txt_courseLeader.setText("");//clears or make empty text field if condition is
ture
            }
            txt_courseLeader.setForeground(Color.BLACK);//changes color of text in text
field
        }
        //performs action when mouse foucus in lost by textfield
        public void focusLost(FocusEvent evt) {
            if(txt_courseLeader.getText().equals("") ||
txt_courseLeader.getText().equals("Enter The Name for Course Leader.."))
            {
                txt_courseLeader.setForeground(Color.GRAY);//changes color of text in text
field
                txt_courseLeader.setText("Enter The Name for Course Leader..");//set text
field with text
```

```
            }
        }
    });

    txt_examDate = new JTextField(20);
    txt_examDate.setBounds(950,395,250,30);
    txt_examDate.setFont(ftxt);
    txt_examDate.setText("dd-mm-yy");
    txt_examDate.setVisible(false);
    txt_examDate.setForeground(Color.GRAY);
    //adding FoucusListener to textField
    txt_examDate.addFocusListener(new FocusListener(){
        //performs action when mouse foucus in gained by textfield
        public void focusGained(FocusEvent evt) {
            if(txt_examDate.getText().equals("dd-mm-yy")){
                txt_examDate.setText("");//clears or make empty text field if condition is ture
            }
            txt_examDate.setForeground(Color.BLACK);//changes color of text in text field
        }
        //performs action when mouse foucus in lost by textfield
        public void focusLost(FocusEvent evt) {
            if(txt_examDate.getText().equals("") || txt_examDate.getText().equals("dd-mm-
yy"))
            {
                txt_examDate.setForeground(Color.GRAY);//changes color of text in text field
                txt_examDate.setText("dd-mm-yy");//set text field with text
            }
        }
    });

    txt_startDate = new JTextField(20);
    txt_startDate.setBounds(950,445,250,30);
    txt_startDate.setFont(ftxt);
    txt_startDate.setText("dd-mm-yy");
    txt_startDate.setForeground(Color.GRAY);
    //adding FoucusListener to textField
    txt_startDate.addFocusListener(new FocusListener(){
        //performs action when mouse foucus in gained by textfield
        public void focusGained(FocusEvent evt) {
            if(txt_startDate.getText().equals("dd-mm-yy")){
              txt_startDate.setText("");//clears or make empty text field if condition is ture
            }
            txt_startDate.setForeground(Color.BLACK);//changes color of text in text field
        }
```

```
        //performs action when mouse foucus in lost by textfield
        public void focusLost(FocusEvent evt) {

            if(txt_startDate.getText().equals("") || txt_startDate.getText().equals("dd-mm-yy"))
            {
                txt_startDate.setForeground(Color.GRAY);//changes color of text in text field
                txt_startDate.setText("dd-mm-yy");//set text field with text

            }
        }
    });

    txt_completionDate = new JTextField(20);
    txt_completionDate.setBounds(950,495,250,30);
    txt_completionDate.setFont(ftxt);
    txt_completionDate.setText("dd-mm-yy");
    txt_completionDate.setForeground(Color.GRAY);
    //adding FoucusListener to textField
    txt_completionDate.addFocusListener(new FocusListener(){
        //performs action when mouse foucus in gained by textfield
        public void focusGained(FocusEvent evt) {
            if(txt_completionDate.getText().equals("dd-mm-yy")){
               txt_completionDate.setText("");//clears or make empty text field if condition is
ture
            }
            txt_completionDate.setForeground(Color.BLACK); //changes color of text in text
field
        }
        //performs action when mouse foucus in lost by textfield
        public void focusLost(FocusEvent evt) {
            if(txt_completionDate.getText().equals("") ||
txt_completionDate.getText().equals("dd-mm-yy"))
            {
                txt_completionDate.setForeground(Color.GRAY);//changes color of text in text
field
                txt_completionDate.setText("dd-mm-yy"); //set text field with text

            }
        }
    });

    //Adding JMenuItem academic,nonAcademic and exit to JMenu course
    course.add(academic);
    course.addSeparator();//seprates academic and nonacademic JMenuItem
    course.add(nonAcademic);
```

```
course.addSeparator();//seprates nonacademic and exit  JMenuItem
course.add(exit);

//Adding JMenuItem about to JMenu help
help.add(about);

//Adding JMenu  to JMenuBar
mB.add(course);
mB.add(help);

//adding Labels and TextFields to panel
panelCourse.add(lbl_academic);
panelCourse.add(lbl_nonAcademic);
panelCourse.add(lbl_courseID);
panelCourse.add(txt_courseID);
panelCourse.add(lbl_duration);
panelCourse.add(txt_duration);
panelCourse.add(lbl_courseName);
panelCourse.add(txt_courseName);
panelCourse.add(lbl_lecturerName);
panelCourse.add(txt_lecturerName);
panelCourse.add(lbl_instructorName);
panelCourse.add(txt_instructorName);
panelCourse.add(lbl_courseLeader);
panelCourse.add(txt_courseLeader);
panelCourse.add(lbl_level);
panelCourse.add(txt_level);
panelCourse.add(lbl_credit);
panelCourse.add(txt_credit);
panelCourse.add(lbl_numberOfAssessment);
panelCourse.add(txt_numberOfAssessment);
panelCourse.add(lbl_startDate);
panelCourse.add(lbl_preRequisite);
panelCourse.add(txt_preRequisite);
panelCourse.add(lbl_startDate);
panelCourse.add(lbl_completionDate);
panelCourse.add(lbl_examDate);
panelCourse.add(txt_examDate);
panelCourse.add(txt_startDate);
panelCourse.add(txt_completionDate);
panelCourse.add(lbl_logo );
panelCourse.add(lbl_footer );
//adding buttons to panel
panelCourse.add(btnAcademic);
panelCourse.add(btnNonAcademic);
```

```
        panelCourse.add(btnRegister);
        panelCourse.add(btnDisplay);
        panelCourse.add(btnClear);
        panelCourse.add(btnRemove);
        panelCourse.add(btnAdd);


        mainFrame.add(panelCourse);//adding Jpanel  to JFrame

        mainFrame.setJMenuBar(mB);//adding JMenuBar to JFrame

        //setting height,width,location,visiblity,title,close operation of JFrame
        titleIcon = new ImageIcon("icon.jpg");//initializing reference object of ImageIcon class
with image location as parameter
        mainFrame.setIconImage(titleIcon.getImage());// adding image to title of JFrame
        mainFrame.setSize(1650,1080);
        mainFrame.setLocationRelativeTo(null);//centers the Freame location
        mainFrame.setTitle("ING College");
        mainFrame.setLayout(null);
        mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mainFrame.setVisible(true);


    }

    //overriging method of ActionListener interface for JMenuItems
    public void actionPerformed(ActionEvent e){
        if(e.getSource() == academic){
            academicComponent();//calling method for academic component
       }
       else if(e.getSource() == nonAcademic){
            nonAcademicComponent();//calling method for nonacademic component
       }
       else if(e.getSource() == exit){
             //here "null"  value help to position dialouge box in center of JFrame
             int response = JOptionPane.showConfirmDialog(null," Are you sure, do you want
to exit? ",
"Warning",JOptionPane.YES_NO_OPTION,JOptionPane.WARNING_MESSAGE);
              if(response == JOptionPane.YES_OPTION){
              System.exit(0);//exits the program normally when exit JMenuItem is clicked
             }
       }
       else{
           if(e.getSource() == about){
               about();//calling about method when about JMenuItem is clicked
```

```
        }
      }
    }

    //changes color of button when mouse curser is placed on button
    public void mouseEntered(MouseEvent e) {
        if(e.getSource() == btnAcademic){
          btnAcademic.setBackground(new Color(34,139,34));
        }

        else if(e.getSource() == btnNonAcademic){
          btnNonAcademic.setBackground(new Color(34,139,34));
        }
        else if(e.getSource() == btnDisplay){
          btnDisplay.setBackground(new Color(34,139,34));
        }
        else if(e.getSource() == btnRegister){

          btnRegister.setBackground(new Color(34,139,34));
        }
         else if(e.getSource() == btnAdd){

          btnAdd.setBackground(new Color(34,139,34));
        }
         else if(e.getSource() == btnRemove){

          btnRemove.setBackground(new Color(192, 0, 0));

        }
        else{
            if(e.getSource() == btnClear){

          btnClear.setBackground(new Color(192, 0, 0));

        }
       }
    }

    //changes color of button when mouse curser is moved away from button
    public void mouseExited(MouseEvent e) {
        if(e.getSource() == btnAcademic){
          btnAcademic.setBackground(new Color(76,187,23));
        }

        else if(e.getSource() == btnNonAcademic){
```

```
        btnNonAcademic.setBackground(new Color(76,187,23));
       }
      else if(e.getSource() == btnDisplay){
        btnDisplay.setBackground(new Color(76,187,23));
       }
       else if(e.getSource() == btnRegister){

        btnRegister.setBackground(new Color(76,187,23));
       }
        else if(e.getSource() == btnAdd){

        btnAdd.setBackground(new Color(76,187,23));
       }
        else if(e.getSource() == btnRemove){

        btnRemove.setBackground(new Color(255,51,51));

       }
       else{
          if(e.getSource() == btnClear){

        btnClear.setBackground(new Color(255,51,51));

       }
      }
  }

  public void mouseClicked(MouseEvent e) { }

  public void mousePressed(MouseEvent e) { }

  public void mouseReleased(MouseEvent e) { }

  //for JMenuItem about frame
  public void about(){
     //declaring instances of JFrame,ImageIcon,JLabel,JPanel,JButton,Font
     JFrame jfme;
     ImageIcon titleIcon;
     JPanel panelInfo;
     JLabel lblTitle,lblVersion,lblCopyRight,lblContact,lblEmail,lblMobNum_,lblImage;
     JButton btnOk;
     Font fTitle,font,fCright,fBtn;
     //initializing JFrame constructor to its object
     jfme = new JFrame();
     // initializing and setting color,layout,location and size of panel
```

```
        panelInfo = new JPanel();
        panelInfo.setBackground(Color.WHITE);
        panelInfo.setBounds(200,0,400,350);
        panelInfo.setLayout(null);

        //defining Font attibutes as parameter in Font class constructor for label,button
        fTitle = new Font("Arial",Font.BOLD,21);
        font = new Font("Arial",Font.BOLD,17);
        fCright = new Font("Arial",Font.PLAIN,14);
        fBtn = new Font("Arial Rounded MT Bold",Font.BOLD,14);

        //setting title,location,size,font,color of label
        lblImage = new JLabel(new ImageIcon("icon1.png"));//inserting image in label and
adding it directly to frame
        lblImage.setBounds(0,0,200,300);

        lblTitle = new JLabel("ING College Registration Software");
        lblTitle.setBounds(20,20,500,25);
        lblTitle.setFont(fTitle);

        lblVersion = new JLabel("Version 0.0.1");
        lblVersion.setBounds(20,70,500,25);
        lblVersion.setFont(font);

        lblCopyRight = new JLabel("Copyright Â© 2021 Design by Bishwas Limbu");//65, 105,
225
        lblCopyRight.setForeground(new Color(65, 105, 225));
        lblCopyRight.setBounds(20,200,500,25);
        lblCopyRight.setFont(fCright);

        lblContact = new JLabel("Contact Details :");
        lblContact.setBounds(20,110,500,25);
        lblContact.setFont(font);

        lblEmail = new JLabel("bisparov@gmail.com");
        lblEmail.setBounds(20,140,500,25);
        lblEmail.setFont(fCright);

        lblMobNum_ = new JLabel("+977-980-8348-036");
        lblMobNum_.setBounds(20,160,500,25);
        lblMobNum_.setFont(fCright);

        //setting title,location,size,font,color of button
        btnOk = new JButton("OK");
        btnOk.setBounds(20,250,90,27);
```

```
        btnOk.setBorder(null);
        btnOk.setBackground(new Color(255,51,51));
        btnOk.setForeground(Color.WHITE);
        btnOk.setFont(fBtn);

        //adding action listener event function to button
        btnOk.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                jfme.dispose();//calling dispose method for closing a particular frame
            }
        });

        //adding mouse listener event function to button
        btnOk.addMouseListener(new MouseListener(){
            //list of method for MouseListener interface which is must needed to override
            public void mouseEntered(MouseEvent e) {
                if(e.getSource() == btnOk){
                    btnOk.setBackground(new Color(192, 0, 0));//changes color of button when
mouse curser is placed on button
                }
            }

            public void mouseExited(MouseEvent e) {
                if(e.getSource() == btnOk){
                    btnOk.setBackground(new Color(255,51,51));//changes color of button when
mouse curser is moved away from button
                }
            }

            public void mouseClicked(MouseEvent e)
            { }

            public void mousePressed(MouseEvent e)
            { }

            public void mouseReleased(MouseEvent e)
            { }
        });

        //adding label and button to panel
        panelInfo.add(lblTitle);
        panelInfo.add(lblVersion);
        panelInfo.add(lblCopyRight);
        panelInfo.add(lblContact);
        panelInfo.add(lblEmail);
```

```
        panelInfo.add(lblMobNum_);
        panelInfo.add(btnOk);

        //adding panel and label in JFrame
        jfme.add(lblImage);
        jfme.add(panelInfo);

        //setting JFrame size,location,layout,visibility,title,image icon
        titleIcon = new ImageIcon("icon.jpg");//initializing reference object of ImageIcon class
with image location as parameter
        jfme.setIconImage(titleIcon.getImage());// adding image to title of JFrame
        jfme.setSize(600,350);
        jfme.getContentPane().setBackground(Color.WHITE);
        jfme.setLocationRelativeTo(null);//centers the Frame location
        jfme.setTitle("About ING College");
        jfme.setLayout(null);
        jfme.setResizable(false);
        jfme.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE );
        jfme.setVisible(true);
    }

    public void academicComponent(){
            //changes the compontnet visibility,color,and set text when academic JMenuItem is
clicked
            panelCourse.setFocusable(true);//sets foucs of mouse to panel when frame is
loaded
            // setting visibility False of textFields,labels adn button for Academic
            lbl_nonAcademic.setVisible(false);
            btnAcademic.setVisible(false);
            btnRemove.setVisible(false);
            lbl_examDate.setVisible(false);
            txt_examDate.setVisible(false);
            lbl_instructorName.setVisible(false);
            txt_instructorName.setVisible(false);
            lbl_preRequisite.setVisible(false);
            txt_preRequisite.setVisible(false);

            // setting visibility True of textFields,labels adn button for Academic
            lbl_academic.setVisible(true);
            btnNonAcademic.setVisible(true);
            lbl_credit.setVisible(true);
            txt_credit.setVisible(true);
            lbl_level.setVisible(true);
            txt_level.setVisible(true);
            lbl_numberOfAssessment.setVisible(true);
```

```
        txt_numberOfAssessment.setVisible(true);
        lbl_lecturerName.setVisible(true);
        txt_lecturerName.setVisible(true);
        //setting text in placeholer in Textfields
        txt_courseID.setText("Enter The Course ID.....");
        txt_courseName.setText("Enter The Course Name...");
        txt_duration.setText("Enter the Duration For Course..");
        txt_lecturerName.setText("Enter The Name for Lecturer..");
        txt_level.setText("Enter The Course Level....");
        txt_credit.setText("Enter The Total credit For Course....");
        txt_numberOfAssessment.setText("Enter Total Number of  Assessment..");
        txt_courseLeader.setText("Enter The Name for Course Leader..");
        txt_startDate.setText("dd-mm-yy");
        txt_completionDate.setText("dd-mm-yy");
        //setting color to text
        txt_courseID.setForeground(Color.GRAY);
        txt_courseName.setForeground(Color.GRAY);
        txt_duration.setForeground(Color.GRAY);
        txt_courseLeader.setForeground(Color.GRAY);
        txt_startDate.setForeground(Color.GRAY);
        txt_completionDate.setForeground(Color.GRAY);
        txt_numberOfAssessment.setForeground(Color.GRAY);
        txt_level.setForeground(Color.GRAY);
        txt_credit.setForeground(Color.GRAY);
        txt_lecturerName.setForeground(Color.GRAY);
    }

    public void nonAcademicComponent(){
        //changes the compontnet visibility,color,and set text when nonAcademic JMenuItem
is clicked
        panelCourse.setFocusable(true);//sets foucs of mouse to panel when frame is loaded
        // setting visibility False of textFields,labels adn button for Non Academic
        btnNonAcademic.setVisible(false);
        lbl_academic.setVisible(false);
        lbl_level.setVisible(false);
        txt_level.setVisible(false);
        lbl_numberOfAssessment.setVisible(false);
        txt_numberOfAssessment.setVisible(false);
        lbl_credit.setVisible(false);
        txt_credit.setVisible(false);
        lbl_lecturerName.setVisible(false);
        txt_lecturerName.setVisible(false);
        //setting visibility True of textFields,labels adn button for Non Academic
        lbl_examDate.setVisible(true);
        txt_examDate.setVisible(true);
```

```
        lbl_nonAcademic.setVisible(true);
        btnRemove.setVisible(true);
        btnAcademic.setVisible(true);
        lbl_preRequisite.setVisible(true);
        txt_preRequisite.setVisible(true);
        lbl_instructorName.setVisible(true);
        txt_instructorName.setVisible(true);
        //setting text in placeholer in Textfields
        txt_courseID.setText("Enter The Course ID.....");
        txt_courseName.setText("Enter The Course Name...");
        txt_duration.setText("Enter the Duration For Course..");
        txt_preRequisite.setText("Enter The  PreRequisite For Course..");
        txt_instructorName.setText("Enter The Instructor Name...");
        txt_courseLeader.setText("Enter The Name for Course Leader..");
        txt_examDate.setText("dd-mm-yy");
        txt_startDate.setText("dd-mm-yy");
        txt_completionDate.setText("dd-mm-yy");
       //setting color to text
        txt_courseID.setForeground(Color.GRAY);
        txt_instructorName.setForeground(Color.GRAY);
        txt_examDate.setForeground(Color.GRAY);
        txt_preRequisite.setForeground(Color.GRAY);
        txt_courseName.setForeground(Color.GRAY);
        txt_duration.setForeground(Color.GRAY);
        txt_courseLeader.setForeground(Color.GRAY);
        txt_startDate.setForeground(Color.GRAY);
        txt_completionDate.setForeground(Color.GRAY);
    }
    public static void main(String[]args){
        new IngCollege();//calling contructor of IngCollege class
    }
}
```

- **Course Class**

```
/**
 * Course class is Super class.
 * @author (Bishwas Limbu)
 * @College ID (NP01NT4S210077)
 * Report File Name : 20049443 BISHWAS LIMBU
 */
public class Course
{
    /*declares  private instance variables*/
    private String courseID;
    private String courseName;
    private String courseLeader;
    private int duration;

    /*creates Course constucter with three parameter*/
    public Course(String courseID,String courseName,int duration){
        this.courseID = courseID;
        this.courseName =  courseName;
        this.duration = duration;
        this.courseLeader=""; //assigns emptystring to instance variable
    }

    /*accessor methods for given atrributes.*/
    /*allows user to access private instance variable.*/
    /*returns value of private instance variable*/

    public String getCourseID(){
        return this.courseID;
    }

    public String getCourseName(){
        return this.courseName;
    }

    public String getCourseLeader(){
        return this.courseLeader;
    }

    public int getDuration(){
        return this.duration;
```

```
    }

    /*mutator method for attributes.*/
    /*allows user to set new values to private instance variable.*/
    public void setCourseLeader(String courseLeader){
        this.courseLeader= courseLeader;
    }

    /*displays course details and also courseLeader name if course leader is not
empty*/
    public void display(){
        System.out.println("Course Details:-");
        System.out.println("Course ID is "+ this.courseID +".");
        System.out.println("Course Title is "+ this.courseName +".");
        System.out.println("The Duration of Course is "+ this.duration+" Year.");
            if (!this.courseLeader.equals("")){
                System.out.println("Course Leader name is "+ this.courseLeader +".");
            }
    }
}
```

- **AcademicCourse Class**

```
/**
 * AcademicCourse class is the Sub class inherited from Super class.
 * @author (Bishwas Limbu)
 * @College ID (NP01NT4S210077)
 * Report File Name : 20049443 BISHWAS LIMBU
 */
public class AcademicCourse extends Course {
    /*declares   private instance variables*/
    private String lecturerName;
    private String level;
    private String credit;
    private String startingDate;
    private String completionDate;
    private int numberOfAssessments;
    private boolean isRegistered;

    /*creates AcademicCourse constructor with 6 parameters*/
    public AcademicCourse(String courseID, String courseName, int duration, String
level, String credit, int numberOfAssessments) {
```

```
        super(courseID, courseName, duration);//calls constructor with 3 parameters
    from super class Course using super key
        this.level = level;
        this.credit = credit;
        this.numberOfAssessments = numberOfAssessments;
        this.lecturerName = ""; // assigns empty string value to instance variable
        this.startingDate = "";
        this.completionDate = "";
        this.isRegistered = false;
    }


    /*accessor method for given atrributes.*/
    /*allows user to access private instance variable.*/
    /*returns value of private instance variables*/
    public String getLecturerName() {
            return this.lecturerName;
    }

    public String getLevel() {
        return this.level;
    }

    public String getCredit() {
        return this.credit;
    }

    public String getStartingDate() {
        return this.startingDate;
    }

    public String getCompletionDate() {
           return this.completionDate;
    }

    public int getNumberOfAssessments() {
        return this.numberOfAssessments;
    }

    public boolean getIsRegistered() {
        return this.isRegistered;
    }

    /*mutator method for attributes.*/
    /*allows user to set new values to private instance variable.*/
```

```
    public void setLecturerName(String lecturerName) {
        this.lecturerName = lecturerName;
    }

    public void setNumberOfAssessments(int numberOfAssessments) {
        this.numberOfAssessments = numberOfAssessments;
    }

    /*registers particular academic course if not register*/
    /*displays lecture name,start and completion date of course,information on course
already register*/
    /*accepts 4 string values for its parameters*/
    public void register(String courseLeader, String lecturerName, String
startingDate, String completionDate) {
        if (this.isRegistered == true) {
            System.out.println("The Course is already registered.");
            System.out.println("The name of lecturer for this course is " + lecturerName
+".");
            System.out.println("This Course will be commencing from " + startingDate
+".");
            System.out.println("This Course will be completed by " + completionDate
+".");
        }
            else {
                super.setCourseLeader(courseLeader);
                this.lecturerName = lecturerName;
                this.startingDate = startingDate;
                this.completionDate = completionDate;
                this.isRegistered = true;
        }
    }

    /*displays the details of the course*/
    /*displays the lecturer name,credit,start,completion and number of assessments if
registered*/
    public void display(){
        super.display();  //calls display method  from super class Course using super
key
            if(this.isRegistered == true){
                System.out.println("The name of lecturer for this course is " +
this.lecturerName +".");
                System.out.println("The Level for this course is " + this.level +".");
                System.out.println("This total credit hours for this course is " + this.credit
+".");
                System.out.println("This Course will be commencing from " +
```

```
this.startingDate +".");
            System.out.println("This Course will be completed by " +
this.completionDate +".");
            System.out.println("The total number of assessments for this course is " +
this.numberOfAssessments +".");
        }
    }
}
```

- **NonAcademic Class**

```
/**
 * NonAcademicCourse class is the Sub class inherited from Super class.
 * @author (Bishwas Limbu)
 * @College ID (NP01NT4S210077)
 * Report File Name : 20049443 BISHWAS LIMBU
 */
public class NonAcademicCourse extends Course
{
    /*declares  private instance variables*/
    private String instructorName;
    private String startDate;
    private String completionDate;
    private String examDate;
    private String preRequisite;
    private boolean isRegistered;
    private boolean isRemoved;

    /*creates NonAcademicCourse constructor with 4 parameters*/
    public NonAcademicCourse(String courseID,String courseName, int duration,
String preRequisite){
        super(courseID, courseName, duration);//calls constructor with 3 parameters
from super class Course using super key
        this.preRequisite = preRequisite;
        this.startDate = "";
        this.completionDate = "";// assigning empty string value to instance variable
        this.examDate = "";
        this.isRegistered = false;
        this.isRemoved = false;
    }

    /*accessor method for given atrributes.*/
```

```
    /*allows user to access private instance variable.*/
    /*returns value of private instance variables*/
    public String getInstructorName() {
        return this.instructorName;
    }

    public String getStartDate() {
        return this.startDate;
    }

    public String getCompletionDate() {
        return this.completionDate;
    }

    public String getExamDate() {
        return this.examDate;
    }

    public String getPreRequisite() {
        return this.preRequisite;
    }

    public boolean getIsRegistered() {
        return this.isRegistered;
    }

    public boolean getIsRemoved() {
        return this.isRemoved;
    }

    /*mutator method for attributes.*/
    /*allows user to set new values to private instance variable.*/
    public void setInstructorName(String instructorName){
        if(this.isRegistered == false){       //checks condition if course registered.
            this.instructorName = instructorName;
        }
            else {
                System.out.println("Course is already Registered.Instructor cannot be
changed.");
        }
    }

    /*registers particular nonacademic course if not register*/
    /*displays information on course already registered if registered*/
    /*accepts 5 string values for its parameters*/
```

```
    public void register(String courseLeader, String instructorName, String startDate,
String completionDate, String examDate){
        if(this.isRegistered == false){
            super.setCourseLeader(courseLeader);
            setInstructorName(instructorName);
            this.startDate = startDate;
            this.completionDate = completionDate;
            this.examDate = examDate;
            this.isRegistered = true;
        }
            else{
                System.out.println("This course has been  already registered.");
        }
    }

    /*removes a particular course details by assiging  empty string value to instance
variables*/
    /*displays information that course is removed if condition is ture*/
    public void remove() {
        if (this.isRemoved == true) { //checks condition if isremoved value is true
             System.out.println("The Course you looking for is removed already.");
        }
            else {
                super.setCourseLeader("");
                this.instructorName = "";
                this.startDate = "";
                this.completionDate = "";
                this.examDate = "";
                this.isRegistered = false;
                this.isRemoved = true;
        }
    }

    /*displays the details of the course*/
    /*displays the instruction name,start,completion and exam date if registered*/
    public void display(){
        super.display(); //calls display method from super class Course using super key
        if(this.isRegistered == true){
            System.out.println("The name of Instructor for this course is " +
this.instructorName +".");
            System.out.println("This Course will be commencing from " + this.startDate
+".");
            System.out.println("This Course will be completed by " + this.completionDate
+".");
            System.out.println("The Examination date for this course will start form " +
```

```
        this.examDate +".");
            }
        }
    }
```