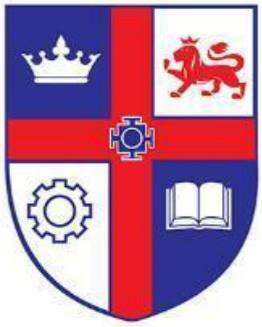
LONDON METROPOLITAN UNIVERSITY



islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS4001NI Programming

COURSEWORK-2

Assessment Weightage & Type

30% Individual Coursework

Semester and Year

Spring 2021

Student Name: Pratik Shrestha

Group: C3

London Met ID: 20048957

College ID: NP01CP4S210063

Assignment Due Date: 20th August,021

Assignment Submission Date: 20th August, 2021

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

Contents

TABLE OF FIGURE

LIST OF TABLES

CHAPTER-1 INTRODUCTION OF TOPIC

This is the second coursework from the “Introduction to Programming” module that was assigned to us. The aim of this course is to design a GUI (Graphical User Interface) for a Course Registration system. BlueJ is used to perform out this program. The main instinct of this coursework is to add a class to the project that was developed for the first part of the coursework to make a graphical user interface (GUI) for a system that stores the details of academic and non-academic course. In this course, many components of swing and awt will be used to develop GUI and will be implemented in Academic and non-academic form, and some buttons such as "add," "register," "display," and "clear," will be designed using various components. To start, we design a main form named Ing College using the "Academic Courses" and "Non-Academic Courses" buttons. And created the form of academic courses and non-academic courses. The main form's design will allow users or clients to choose academic or non-academic courses based on their preferences. The Academic Course form and the Non-Academic Course form will both be designed so that the user may enter various kinds of information.

A GUI was developed for this assignment based on the Course. This is an academic course. An academic course was created, the course leader or lecturer was selected, and information not related to the course was saved. The GUI was created using swing and awt, and several kinds of graphical components are used to take input depending on the varied information stored. It utilizes the use of lecturer, course Leader and the Non-Academic class from the previous coursework to create a software or the forms that appoints, add, displays, clear, register and remove the input data or the information. Event handling was also implemented. The awt's event handling will be used for a variety of reasons, including error handling, data entry, and validation. To ensure that the code is encapsulated, all the attributes utilized in this application will be set to private. The Action Listener was also used to make the required buttons for all classes responsive. In order to prosecute the buttons used in this program, exception handling processes like try, catch, and throw will be used.

CHAPTER-2 CLASS DIAGRAM

Class diagram is a static diagram which represents the static view of an application. A class diagram is used not only for visualizing, describing, and documenting various components of a system, but also for creating source codes for a software application. In brief, a class diagram illustrates a class's attributes and functions, as well as the constraints imposed on the system. Since class diagrams are the only UML diagrams which can be directly mapped with object-oriented languages, they are widely used in the design of object-oriented systems. The classes in the class diagrams are divided into classes with similar characteristics. A collection of classes, interfaces, associations, cooperation, and limitations can be seen in a class diagram. It is also known as a structural diagram

A class diagram is like a flowchart in which classes are presented as boxes with three rectangles inside each box. The name of the class is displayed in the top rectangle. The class's attributes are in the center rectangle. The class's methods, also referred as operations, are found in the lower rectangle. The boxes are linked by lines with arrows at one or both ends. These lines define the relationships, also called associations, between the classes.

The class diagram of each of the classes is shown below:

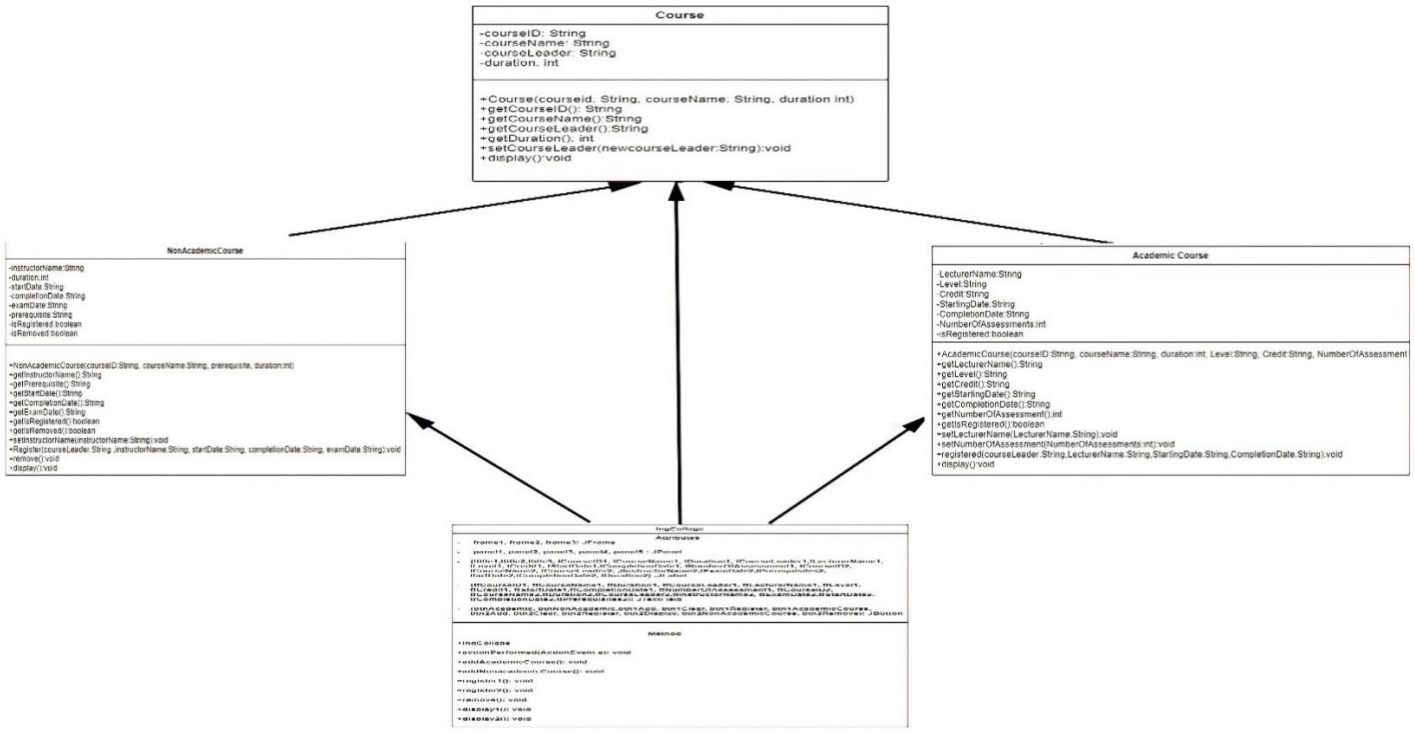


Figure 1: class diagram for course

CHAPTER-3 PSEUDOCODE OF ING COLLAGE

Pseudocode is a programming methodology that allows a programmer to represent an algorithm's implementation. Pseudocode is an unstructured technique of describing a program, not a programming language. It serves as a basic representation of a program's functions rather than having specific syntax. Since pseudocode is an informal language, it is mostly used to create a program blueprint or rough draft. Pseudocode cannot be compiled into an executable program since it is not a programming language. As a result, if pseudocode is to become a workable application, it must be translated into a specific programming language.
The pseudocode of the program is given below:

```
CREATE class IngCollage implements ActionListener
    DO
        INITIALISE frame1 as JFrame
        INITIALISE frame2 as JFrame
        INITIALISE frame3 as JFrame

        INITIALISE ltitle1 as JLabel
        INITIALISE ltitle2 as JLabel
        INITIALISE ltitle3 as JLabel

        INITIALISE lCourseID1 as JLabel
        INITIALISE lCourseName1 as JLabel
        INITIALISE lDuration1 as JLabel
        INITIALISE lCourseLeader1 as JLabel
        INITIALISE lLecturerName1 as JLabel
        INITIALISE lLevel1 as JLabel
        INITIALISE lCredit1 as JLabel
        INITIALISE lStartDate1 as JLabel
        INITIALISE lCompletionDate1 as JLabel
        INITIALISE lNumberOfAssessment1 as JLabel
        INITIALISE lCourseID2 as JLabel
        INITIALISE lCourseName2 as JLabel
```

INITIALISE lCourseLeader2 as JLabel
INITIALISE lInstructorName2 as JLabel
INITIALISE lExamDate2 as JLabel
INITIALISE lPrerequisites2 as JLabel
INITIALISE lStartDate2 as JLabel
INITIALISE lCompletionDate2 as JLabel
INITIALISE lDuration2 as JLabel

INITIALISE tfCourseID1 as JTextField
INITIALISE tfCourseName1 as JTextField
INITIALISE tfDuration1 as JTextField
INITIALISE tfCourseLeader1 as JTextField
INITIALISE tfLecturerName1 as JTextField
INITIALISE tfLevel1 as JTextField
INITIALISE tfCredit1 as JTextField
INITIALISE tfStartDate1 as JTextField
INITIALISE tfCompletionDate1 as JTextField
INITIALISE tfNumberOfAssessment1 as JTextField
INITIALISE tfCourseID2 as JTextField
INITIALISE tfCourseName2 as JTextField
INITIALISE tfCourseLeader2 as JTextField
INITIALISE tfInstructorName2 as JTextField
INITIALISE tfExamDate2 as JTextField
INITIALISE tfPrerequisites2 as JTextField
INITIALISE ltfStartDate2 as JTextField
INITIALISE tfCompletionDate2 as JTextField
INITIALISE tfDuration2 as JTextField

INITIALISE btn1Academic as JButton
INITIALISE btn2NonAcademic as JButton
INITIALISE btn1Add as JButton
INITIALISE btn1Clear as JButton

INITIALISE btn1Register as JButton
INITIALISE btn1Display as JButton
INITIALISE btn2Register as JButton
INITIALISE btn2Add as JButton
INITIALISE btn2Display as JButton
INITIALISE btn2Remove as JButton
INITIALISE btn2Clear as JButton

INITIALISE panel1 as JPanel
INITIALISE panel2 as JPanel
INITIALISE panel3 as JPanel
INITIALISE panel4 as JPanel
INITIALISE panel5 as JPanel

INITIALISE < Course> course
END DO
DO
CREATE constructor IngCollage()
CREATE object frame1 as JFrame("IngCollege") of JFrame type
CREATE object panel1 as JPanel type

DEFINE ltitle1 as JLabel type("Welcome to Islington College")
SET bound of ltitle1 as setBounds(120,50,500,100)
SET font of ltitle1 as setFont(new Font("SanSerif Bold Italic", Font.BOLD, 25))

DEFINE object btn1AcademicCourse as JButton type(“Academic Course")
SET bound of btn1AcademicCourse as setBounds(80,180,200,50)
SET font of btn1AcademicCourse as setFont(new Font("Arial", Font.BOLD, 15))
ADD action listener to btn1AcademicCourse

DEFINE object btn2AcademicCourse as JButton type(“Academic Course")
SET bound of btn2AcademicCourse as setBounds(320,180,200,50)
SET font of btn2AcademicCourse as setFont(new Font("Arial", Font.BOLD, 15))
ADD action listener to btn2AcademicCourse

ADD ltitle1 to panel1
ADD btn1AcademicCourse to panel1
ADD btn2AcademicCourse to panel1
ADD panel1 to frame1
SET frame size as setSize(600,400)
SET frame as Visible(true)

CREATE object frame2 as JFrame("Academic Course") of JFrame type
CREATE object panel2 as JPanel type
CREATE object panel4 as JPanel type

INITIALISE ltitle2 as JLabel type("Academic Course")
SET bound of ltitle2as setBounds(130,50,350,70);
SET bound of tftitle2as setBounds(10,10,100,25);
SET font of ltitle2 as setFont(new Font("Arial", Font.BOLD, 35))

INITIALISE lCourseID1as JLabel type(" lCourseID* ")
SET bound of lCourseID1 as setBounds(50,150,200,25);
INITIALISE tfCourseID1as JTextField type
SET bound of tfCourseID1 as setBounds(50,180,200,40);
SET font of lCourseID1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lCourseName1 as JLabel type("Course Name*")
SET bound of lCourseName1 as setBounds(300,150,200,25);
INITIALISE tfCourseName1 as JTextField type

SET bound of tfCourseName1 as setBounds(300,180,200,40);
SET font of lCourseName1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lCourseLeader1 as JLabel type(" Course Leader* ")
SET bound of lCourseLeader1 as setBounds(50,250,200,25);
INITIALISE tfCourseLeader1 as JTextField type
SET bound of tfCourseLeader1 as setBounds(50,280,200,40);
SET font of lCourseLeader1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lLecturerName1 as JLabel type(" Lecturer Name* ")
SET bound of lLecturerName1 as setBounds(300,250,200,25);
INITIALISE tfLecturerName1 as JTextField type
SET bound of tfLecturerName1 as setBounds(300,280,200,40);
SET font of lLecturerName1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lStartDate1 as JLabel type(" StartDate* ")
SET bound of lStartDate1 as setBounds(50,350,200,25);
INITIALISE tfStartDate1 as JTextField type
SET bound of tfStartDate1 as setBounds(50,380,200,40);
SET font of lStartDate1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lCompletionDate1 as JLabel type(" Completion Date* ")
SET bound of lCompletionDate1as setBounds(300,350,200,25);
INITIALISE tfCompletionDate1 as JTextField type
SET bound of tfCompletionDate1as setBounds(300,380,200,40);
SET font of lCompletionDate1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lLevel1 as JLabel type(" Level* ")
SET bound of lLevel1 as setBounds(50,450,200,25);

INITIALISE tfLevel1 as JTextField type
SET bound of tfLevel1 as setBounds(50,480,200,40);
SET font of lLevel1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lCredit1 as JLabel type(" Credit* ")
SET bound of lCredit1 as setBounds(300,450,200,25);
INITIALISE tfCredit1 as JTextField type
SET bound of tfCredit1 as setBounds(300,480,200,40);
SET font of lCredit1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lNumberOfAssessment1 as JLabel type(" NumberOfAssessment1* ")
SET bound of lNumberOfAssessment1 as setBounds(50,550,200,25);
INITIALISE tfNumberOfAssessment1 as JTextField type
SET bound of tfNumberOfAssessment1 as setBounds(50,580,200,40);
SET font of lNumberOfAssessment1 as setFont(new Font("Arial", Font.BOLD, 15))

DEFINE object btn1Add as JButton type("Add");
SET bound of btn1Add as setBounds(50,220,120,40);
SET font of btn1Add as setFont(new Font("Arial", Font.BOLD, 15);
ADD action listener to btn1Add

DEFINE object btn1Reguster as JButton type ("Register");
SET bound of btn1Reguster as setBounds(50,320,120,40);
SET font of btn1Reguster as setFont(new Font("Arial", Font.BOLD, 15))
ADD action listener to btn1Reguster

DEFINE object btn1Clear as JButton type("Clear");
SET bound of btn1Clear as setBounds(50,520,120,40);
SET font of btn1Clear as setFont(new Font("Arial", Font.BOLD, 15))
ADD action listener to btn1Clear

DEFINE object btn1Display as JButton type("Display");
SET bound of btn1Display as setBounds(50,420,120,40);
SET font of btn1Display as setFont(new Font("Arial", Font.BOLD, 15))
ADD action listener to btn1Display

ADD lCourseID1 to Panel2
ADD lCourseName1 to Panel2
ADD lDuration1 to Panel2
ADD lCourseLeader1 to Panel2
ADD lLecturerName1 to Panel2
ADD lLevel1 to Panel2
ADD lCredit1 to Panel2
ADD lStartDate1 to Panel2
ADD lCompletionDate1 to Panel2
ADD lNumberOfTosessment1 to Panel2
ADD lCourseID2 to Panel2
ADD lCourseName2 to Panel2
ADD lCourseLeader2 to Panel2
ADD lInstructorName2 to Panel2
ADD lExamDate2 to Panel2
ADD lPrerequisites2 to Panel2
ADD lStartDate2 to Panel2
ADD lCompletionDate2 to Panel2
ADD lDuration2 to Panel2

ADD tfCourseID1 to Panel2
ADD tfCourseName1 to Panel2
ADD tfDuration1 to Panel2
ADD tfCourseLeader1 to Panel2
ADD tfLecturerName1 to Panel2
ADD tfLevel1 to Panel2
ADD tfCredit1 to Panel2

ADD tfStartDate1 to Panel2

ADD tfCompletionDate1 to Panel2

ADD tfNumberOfTosessment1 to Panel2

ADD tfCourseID2 to Panel2

ADD tfCourseName2 to Panel2

ADD tfCourseLeader2 to Panel2

ADD tfInstructorName2 to Panel2

ADD tfExamDate2 to Panel2

ADD tfPrerequisites2 to Panel2

ADD ltfStartDate2 to Panel2

ADD tfCompletionDate2 to Panel2

ADD tfDuration2 to Panel2

ADD btn1Add to Panel4

ADD btn1Clear to Panel4

ADD tbtn1Register to Panel4

ADD btn1Display to Panel4

ADD panel2 to frame2

ADD panel4 to frame2

SET panel2 size as setSize(500,700)

SET panel2 size as setLocation(0,0)

SET panel4 size as setSize(300,700)

SET panel4 size as setLocation(525,25)

SET frame2 size as setSize(570,400)

SET frame2 DefaultCloseOperation(JFrame.EXIT_ON_CLOSE)

SET frame2 Visible(true)

SET panel layout to null

CREATE object frame3 as JFrame("Non Academic Course") of JFrame type

CREATE object panel3 as JPanel type

CREATE object panel5 as JPanel type

INITIALISE ltitle3 as JLabel type("Academic Course")
SET bound of ltitle3 as setBounds(130,50,350,70);
SET bound of ltitle3 as setBounds(10,10,100,25);
SET font of ltitle3 as setFont(new Font("Arial", Font.BOLD, 35))

INITIALISE lCourseID1as JLabel type(" lCourseID* ")
SET bound of lCourseID1 as setBounds(50,150,200,25);
INITIALISE tfCourseID1as JTextField type
SET bound of lCourseID1 as setBounds(50,180,200,40);
SET font of lCourseID1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lCourseName1 as JLabel type("Course Name*")
SET bound of lCourseName1 as setBounds(300,150,200,25);
INITIALISE tfCourseName1 as JTextField type
SET bound of lCourseName1 as setBounds(300,180,200,40);
SET font of lCourseName1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lCourseLeader1 as JLabel type(" Course Leader* ")
SET bound of lCourseLeader1 as setBounds(50,250,200,25);
INITIALISE tfCourseLeader1 as JTextField type
SET bound of lCourseLeader1 as setBounds(50,280,200,40);
SET font of lCourseLeader1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lLecturerName1 as JLabel type(" Lecturer Name* ")
SET bound of lLecturerName1 as setBounds(300,250,200,25);
INITIALISE tfLecturerName1 as JTextField type
SET bound of lLecturerName1 as setBounds(300,280,200,40);
SET font of lLecturerName1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lStartDate1 as JLabel type(" StartDate* ")
SET bound of lStartDate1 as setBounds(50,350,200,25);
INITIALISE tfStartDate1 as JTextField type
SET bound of lStartDate1 as setBounds(50,380,200,40);
SET font of lStartDate1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lCompletionDate1 as JLabel type(" Completion Date* ")
SET bound of lCompletionDate1as setBounds(300,350,200,25);
INITIALISE tfCompletionDate1 as JTextField type
SET bound of lCompletionDate1as setBounds(300,380,200,40);
SET font of lCompletionDate1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lLevel1 as JLabel type(" Level* ")
SET bound of lLevel1 as setBounds(50,450,200,25);
INITIALISE tfLevel1 as JTextField type
SET bound of lLevel1 as setBounds(50,480,200,40);
SET font of lLevel1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lCredit1 as JLabel type(" Credit* ")
SET bound of lCredit1 as setBounds(300,450,200,25);
INITIALISE tfCredit1 as JTextField type
SET bound of lCredit1 as setBounds(300,480,200,40);
SET font of lCredit1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE lNumberOfAssessment1 as JLabel type(" NumberOfAssessment1* ")
SET bound of lNumberOfAssessment1 as setBounds(50,550,200,25);
INITIALISE tfNumberOfAssessment1 as JTextField type
SET bound of lNumberOfAssessment1 as setBounds(50,580,200,40);
SET font of lNumberOfAssessment1 as setFont(new Font("Arial", Font.BOLD, 15))

INITIALISE ltitle3 as JLabel type("Non Academic Course")
SET bound of ltitle3 as setBounds(230,50,350,70);
INITIALISE tfltitle3 as JTextField type
SET bound of tftitle3 as setBounds(20,20,200,25);
SET font of ltitle3 as setFont(new Font("Arial", Font.BOLD, 35))

INITIALISE lCourseID2 as JLabel type(" CourseID* ")
SET bound of lCourseID2 as setBounds(50,250,200,25);
INITIALISE tfCourseID2as JTextField type
SET bound of tfCourseID2 as setBounds(50,280,200,40);
SET font of lCourseID2 as setFont(new Font("Arial", Font.BOLD, 25))

INITIALISE lCourseName2 as JLabel type("Course Name*")
SET bound of lCourseName2 as setBounds(300,250,200,25);
INITIALISE tfCourseName2 as JTextField type
SET bound of tfCourseName2 as setBounds(300,280,200,40);
SET font of lCourseName2 as setFont(new Font("Arial", Font.BOLD, 25))

INITIALISE lCourseLeader2 as JLabel type(" Course Leader* ")
SET bound of lCourseLeader2 as setBounds(50,250,200,25);
INITIALISE tfCourseLeader2 as JTextField type
SET bound of tfCourseLeader2 as setBounds(50,280,200,40);
SET font of lCourseLeader2 as setFont(new Font("Arial", Font.BOLD, 25))

INITIALISE lInstructorName2 as JLabel type(" Instructor Name* ")
SET bound of lInstructorName2 as setBounds(300,250,200,25);
INITIALISE tfInstructorName2 as JTextField type
SET bound of tfInstructorName2 as setBounds(300,280,200,40);
SET font of lInstructorName2 as setFont(new Font("Arial", Font.BOLD, 25))

INITIALISE lStartDate2 as JLabel type(" StartDate* ")
SET bound of lStartDate2 as setBounds(50,350,200,25);
INITIALISE tfStartDate2 as JTextField type
SET bound of tfStartDate2 as setBounds(50,380,200,40);
SET font of lStartDate2 as setFont(new Font("Arial", Font.BOLD, 25))

INITIALISE lCompletionDate2 as JLabel type(" Completion Date* ")
SET bound of lCompletionDate2as setBounds(300,350,200,25);
INITIALISE tfCompletionDate2 as JTextField type
SET bound of tfCompletionDate2as setBounds(300,380,200,40);
SET font of lCompletionDate2 as setFont(new Font("Arial", Font.BOLD, 25))

INITIALISE lStartDate2 as JLabel type(" Starting Date* ")
SET bound of lStartDate2 as setBounds(50,450,200,25);
INITIALISE tfStartDate2as JTextField type
SET bound of tfStartDate2 as setBounds(50,480,200,40);
SET font of lStartDate2 as setFont(new Font("Arial", Font.BOLD, 25))

INITIALISE lCompletionDate2 as JLabel type(" Completion Date* ")
SET bound of lCompletionDate2 as setBounds(300,450,200,25);
INITIALISE tfCompletionDate2 as JTextField type
SET bound of tfCompletionDate2 as setBounds(300,480,200,40);
SET font of CompletionDate2 as setFont(new Font("Arial", Font.BOLD, 25))

INITIALISE lDuration2 as JLabel type(" Duration* ")
SET bound of lDuration2 as setBounds(50,550,200,25);
INITIALISE tfDuration2 as JTextField type
SET bound of tfDuration2 as setBounds(50,580,200,40);
SET font of lDuration2 as setFont(new Font("Arial", Font.BOLD, 25))

DEFINE object btn2Add as JButton type("Add");
SET bound of btn2Add as setBounds(50,220,120,40);
SET font of btn2Add as setFont(new Font("Arial", Font.BOLD, 15);
ADD action listener to btn2Add

DEFINE object btn2Reguster as JButton type ("Register");
SET bound of btn2Reguster as setBounds(50,320,120,40);
SET font of btn2Reguster as setFont(new Font("Arial", Font.BOLD, 15))
ADD action listener to btn2Reguster

DEFINE object btn2Clear as JButton type("Clear");
SET bound of btn2Clear as setBounds(50,520,120,40);
SET font of btn2Clear as setFont(new Font("Arial", Font.BOLD, 15))
ADD action listener to btn2Clear

DEFINE object btn2Display as JButton type("Display");
SET bound of btn2Display as setBounds(50,420,120,40);
SET font of btn2Display as setFont(new Font("Arial", Font.BOLD, 15))
ADD action listener to btn2Display

ADD lCourseID2 to Panel3
ADD lCourseName2 to Panel3
ADD lDuration2 to Panel3
ADD lCourseLeader2 to Panel3
ADD lLecturerName2 to Panel3
ADD lLevel2 to Panel3
ADD lCredit2 to Panel3
ADD lStartDate2 to Panel3
ADD lCompletionDate2 to Panel3
ADD lNumberOfTosessment2 to Panel3

ADD lCourseID2 to Panel3
ADD lCourseName2 to Panel3
ADD lCourseLeader2 to Panel3
ADD lInstructorName2 to Panel3
ADD lExamDate2 to Panel3
ADD lPrerequisites2 to Panel3
ADD lStartDate2 to Panel3
ADD lCompletionDate2 to Panel3
ADD lDuration2 to Panel3

ADD tfCourseID2 to Panel3
ADD tfCourseName2 to Panel3
ADD tfDuration2 to Panel3
ADD tfCourseLeader2 to Panel3
ADD tfLecturerName2 to Panel3
ADD tfLevel2 to Panel3
ADD tfCredit2 to Panel3
ADD tfStartDate2 to Panel3
ADD tfCompletionDate2 to Panel3
ADD tfNumberOfTosessment2 to Panel3
ADD tfCourseID2 to Panel3
ADD tfCourseName2 to Panel3
ADD tfCourseLeader2 to Panel3
ADD tfInstructorName2 to Panel3
ADD tfExamDate2 to Panel3
ADD tfPrerequisites2 to Panel3
ADD ltfStartDate2 to Panel3
ADD tfCompletionDate2 to Panel3
ADD tfDuration2 to Panel3

ADD btn2Add to Panel5
ADD btn2Clear to Panel5

ADD btn2Register to Panel5

ADD btn2Display to Panel5

SET panel3 size as setSize(500,700)

SET panel3 size as setLocation(0,0)

SET panel4 size as setSize(300,700)

SET panel5 size as setLocation(525,25)

SET frame3 size as setSize(570,400)

SET frame3 DefaultCloseOperation(JFrame.EXIT_ON_CLOSE)

SET frame3 Visible(true)

SET panel layout to null

METHOD actionPerformed (ActionEvent e)

DO

IF (e.getSource() == btn1Clear)

SET tfCourseID1Textfield TO empty string (“ “)

SET tfCourseName1 Textfield TO empty string (“ “)

SET tfDuration1 Textfield TO empty string (“ “)

SET tfCourseLeader1 Textfield TO empty string (“ “)

SET tfLecturerName1 Textfield TO empty string (“ “)

SET tfLevel1Textfield TO empty string (“ “)

SET tfCredit1Textfield TO empty string (“ “)

SET tfStartDate1Textfield TO empty string (“ “)

SET tfCompletionDate1Textfield TO empty string (“ “)

SET tfNumberOfAssessment1Textfield TO empty string (“ “)

ENDIF

IF (e.getSource() == btn2Clear)

SET tfCourseID2 Textfield TO empty string (“ “)

SET tfCourseName2 Textfield TO empty string (“ “)

SET tfDuration2 Textfield TO empty string (“ “)

SET tfCourseLeader2 Textfield TO empty string (“ “)

```
        SET tfLecturerName2 Textfield TO empty string (“ “)
        SET tfLevel2 Textfield TO empty string (“ “)
        SET tfCredit2Textfield TO empty string (“ “)
        SET tfStartDate2 Textfield TO empty string (“ “)
        SETtfCompletionDate2 Textfield TO empty string (“ “)
        SET tfNumberOfAssessment2Textfield TO empty string (“ “)
ENDIF

IF (e.getSource() == btn1AcademicCourse)
        SET frame2 frame VISIBILITY TO true
        SET frame3 frame VISIBILITY TO false
ENDIF


IF (e.getSource() == NonAcademicCourse)
        SET frame2 frame VISIBILITY TO false
        SET frame3 frame VISIBILITY TO true
ENDIF

IF (e.getSource() == btn1Add)
        DO
                METHOD addAcademicCourse()
        END DO
ENDIF

IF (e.getSource() == btn2Add)
        DO
                METHOD addAcademicCourse()
        END DO
ENDIF

IF (e.getSource() == btn1Register)
```

```
        DO
            METHOD register1()
        END DO
    ENDIF

    IF (e.getSource() == btn2Register)
        DO
            METHOD register2()
        END DO
    ENDIF


    IF (e.getSource() == btn1Display)
        DO
            FOR (Course ac_co: course)
                DECLARE object ac_d of AcademicCourse and downcast aca_co
                CALL display() method from object ac_d
            ENDFOR
        ENDDO
    ENDIF

    IF (e.getSource() == btn2Display)
        DO
            FOR (Course nac_co_course: course)
                DECLARE object nac_co of NonAcademicCourse and downcast
            naca_co
                CALL display() method from object nac_d
            ENDFOR
        ENDDO

        IF (e.getSource() == btn2Remove)
            DO
```

METHOD remove()

END DO

ENDIF

METHOD register1()

DO

TRY

GET tfCourseID1 textfield and ASSIGN it to CourseID1

GET tfCourseName1 textfield and ASSIGN it to CourseName1

GET tfDuration1 textfield and ASSIGN it to Duration1

GET tfCourseLeader1 textfield and ASSIGN it to CourseLeader1

GET tfLecturerName1 textfield and ASSIGN it to LecturerName1

GET tfLevel1 textfield and ASSIGN it to Level1

GET tfCredit1 textfield and ASSIGN it to Credit1

GET tfStartDate1 textfield and ASSIGN it to StartDate1

GET tfCompletionDate1 textfield and ASSIGN it to CompletionDate1

GET tfNumberOfAssessment1 textfield and ASSIGN it to NumberOfAssessment

INT Duration1 = Integer.parseInt(Duration1)

BOOLEAN courseIsAdded1 = false

FOR (Course ac_co : course)

IF (ac_co_course.getCourseId().equals(course_ID))

DECLARE object ac_c of AcademicCourse and downcast ac_co

IF (ac_c.getIsRegistered())

SHOW message dialog ("The Academic Course is already Registered. ") on frame2

ASSIGN courseIsFound1 as true

ENDIF

ELSE

ASSIGN register method with parameters(CourseLeader, LecturerName, StartDate, CompletionDate) TO object ac_c

SHOW message dialog (“The Academic Course is Registered successfully. ") on frame2

ASSIGN courseIsFound1 as true

BREAK

ENDELSE

IF (courseIsFound1 == false)

SHOW message dialog (“The Academic Course is not Found. ") on frame2

ENDI

ENDTRY

CATCH (Exception ex1)

SHOW message dialog ("Please input valid value. ") on frame2

ENDCATCH

METHOD register2()

DO

TRY

GET tfCourseID2 textfield and ASSIGN it to CourseID

GET tfCourseName2 textfield and ASSIGN it to CourseName2

GET tfDuration2 textfield and ASSIGN it to Duration2

GET tfCourseLeader2 textfield and ASSIGN it to CourseLeader2

GET tfInstructorName2 textfield and ASSIGN it to InstructorName2

GET tfExamDate2 textfield and ASSIGN it to ExamDate2

GET tfStartDate2 textfield and ASSIGN it to StartDate2

GET tfCompletionDate2 textfield and ASSIGN it to CompletionDate2

GET tfPrerequisites2 textfield and ASSIGN it to NumberOfPrerequisites2

INT Duration2 = Integer.parseInt(Duration2)

BOOLEAN courseIsAdded2 = false

FOR (Course nac_co : course)

IF (nac_co_course.getCourseId().equals(course_ID))

DECLARE object nac_c of NonAcademicCourse and downcast nac_co

IF (nac_c.getIsRegistered())

SHOW message dialog ("The Non Academic Course is already Registered. ") on frame3

ASSIGN courseIsFound2 as true

ELSE

ASSIGN register method with parameters(CourseLeader2, InstructorName2 ,ExamDate2, StartDate2, CompletionDate2) TO object nac_c

SHOW message dialog (“The Non-Academic Course is Registered successfully. ") on frame3

ASSIGN courseIsFound2 as true

ENDELSE

IF (courseIsFound2 == false)

SHOW message dialog (“The Non-Academic Course is not Found. ") o frame3

ENDIF

ENDTRY

CATCH (Exception ex2)

SHOW message dialog ("Please input valid value. ") on frame3

ENDCATCH

METHOD remove()

DO

TRY

GET tfCourseID2textfield and ASSIGN it to CourseID2

IF (CourseID2.equals("))

SHOW message dialog ("Invalid Value") on frame3

ENDIF

ELSE

Boolean courseIsFound2 = false

FOR (Course nac_co_: course)

IF ((non_nac_co.getCourseId().equals(CourseID2))

DECLARE object nac_r of NonAcademicCourse and downcast nac_co

```
                    SHOW message dialog ("Course is already Removed.") on
                frame3
                    ASSIGN courseIsFound2 as true
                ENDIF

            ENDFOR
            IF (courseIsFound2 == false)
                    SHOW message dialog ("Non-Academic Course is not
                found.") on frame3
            ENDIF
        ENDELSE
    ENDTRY

    CATCH (Exception expr)
        SHOW message dialog ("Please enter valid value") on frame3
    ENDCATCH
ENDDO

METHOD addNonacademicCourse()
TRY
    GET tfCourseID2 textfield and ASSIGN it to CourseID
    GET tfCourseName2 textfield and ASSIGN it to CourseName2
    GET tfDuration2 textfield and ASSIGN it to Duration2
    GET tfCourseLeader2 textfield and ASSIGN it to CourseLeader2
    GET tfInstructorName2 textfield and ASSIGN it to InstructorName2
    GET tfExamDate2  textfield and ASSIGN it to ExamDate2
    GET tfStartDate2 textfield and ASSIGN it to StartDate2
    GET tfCompletionDate2 textfield and ASSIGN it to CompletionDate2
    GET tfPrerequisites2 textfield and ASSIGN it to NumberOfPrerequisites2
    INT Duration2 = Integer.parseInt(Duration2)
    BOOLEAN courseIsAdded2 = false
FOR (Course nac_co : course)
```

IF (nac_co.getCourseId().equals(CourseID))

ASSIGN courseisAdded2 as true

BREAK

ENDFOR

IF (courseIsAdded2 == false)

CREATE object of NonacademicCourse as nac_co with parameters (CourseID2, CourseName2, duration2, Prerequisites2)

ADD nac_co to course object

SHOW message dialog ("The course is added successfully to Nonacademic course") on frame3

ENDIF

ELSE IF (courseAdded2 == true)

SHOW message dialog ("The Nonacademic Course has already added. ") on frame3

ENDELSEIF

CATCH (exp1 exception)

DO

SHOW message dialog ("Please enter valid value") on frame2

ENDDO

METHOD addAcademicCourse()

TRY

GET tfCourseID1textfield and ASSIGN it to CourseID

GET tfCourseName1textfield and ASSIGN it to CourseName

GET tfDuration1textfield and ASSIGN it to Duration

GET tfCourseLeader1textfield and ASSIGN it to CourseLeader

GET tfLecturerName1textfield and ASSIGN it to LecturerName

GET tfLevel1 textfield and ASSIGN it to Level

GET tfCredit1 textfield and ASSIGN it to Credit

GET tfStartDate1textfield and ASSIGN it to StartDate

GET tfCompletionDate1 textfield and ASSIGN it to CompletionDate

GET tfNumberOfAssessment1 textfield and ASSIGN it to NumberOfAssessment
INT Duration1 = Integer.parseInt(Duration)
INT numberOfAssessment1 = Integer.parseInt(NumberOfAssessment)
BOOLEAN courseIsAdded1 = false

FOR (Course ac_co : course)
IF (ac_co.getCourseId().equals(CourseID))
ASSIGN courseIsAdded1 as true
BREAK
ENDFOR
IF (courseIsAdded1 == false)
CREATE object of AcademicCourse as ac_co with parameters (CourseID, CourseName, duration1, Level, Credit,numberOfAssessment1)
ADD ac_co to course object
SHOW message dialog ("The course is added successfully to academic course") on frame2
ENDIF
ELSE IF (courseIsAdded1 == true)
SHOW message dialog ("The Academic Course has already added. ") on frame2
ENDELSEIF

CATCH (exp1 exception)
DO
SHOW message dialog ("Please enter valid value") on frame2
ENDDO

METHOD main (String args[])
DO
CALL method IngCollege()
ENDDO

CHAPTER-4 METHOD DESCRIPTION

A method is a group of statements that perform a task and provide the result to the user. Without returning anything, a method can perform a specific task. Methods allow us to reuse code without having to retype it. Every method in Java must be part of a class, which is unique from all other languages. Each method's functions are explained in the Method Description. It simply defines the component of a class that can be found in any of the class's objects. The following is a list of the project's method descriptions:

i. addAcademicCourse ():

The public access modifier is applied to this method, and the return type is void. This method is called when the add button in the GUI of an academic course is clicked to add a course, and all the course's entered data is saved as a list in the class course and academic course as an object.

ii. addNonAcademicCourse ():

The public access modifier is added to this method, as well as the return type is void. When the add button in the GUI of a non-academic course is pressed to add a course, all of the entered data for the course is stored as a list in class course and non-academic course as an object.

iii. register1 ():

The public access modifier is added to this method, as well as the return type is void. This function is invoked when the register button in the academic course's GUI is clicked to register the course. If all the data is accurate, the course is successfully registered; otherwise, an appropriate result is displayed to the user.

iv. register2 ():

The public access modifier is added to this method, as well as the return type is void. When the register button in the GUI of a non-academic course is clicked to register the course, this function is called. If all the data is accurate, the course is successfully registered; otherwise, an appropriate result is displayed to the user.

v. remove ():

The public access modifier is applied to this method, and the return type is void. When the delete button in the GUI of the non-academic course is pushed to remove the existing course, this function is invoked. If the course data is valid, it is successfully deleted; otherwise, an appropriate output is displayed to the user.

vi. display 1():

The public access modifier is added to this method, as well as the return type is void. This method is called when the display button in the academic course's GUI is clicked to display data from classes Course and Academic Course.

vii. display2 ():

The public access modifier is added to this method, as well as the return type is void. , This method is called when the display button in the non-academic course's GUI is clicked to display data from classes Course and Non-academic Course.

CHAPTER-5 TESTING

Test 1:

Test that the program can be compiled and run using the command prompt, including a screenshot like Figure 1 from the command prompt.

Objective	To compile and run the program using command prompt.
Action:	Compile the program IngCollege.java using the javac ➔ Javac AcademicCourse.java ➔ Javac NonacademicCourse.java ➔ Javac Course.java Run the program or code command java IngCollage.
Expected Result:	The program would be compiled and run in the command prompt.
Actual Result:	The program was compiled and run in the command prompt.
Conclusion:	The test is successful.

Table 1: TEST course program in CMD

OUTPUT:

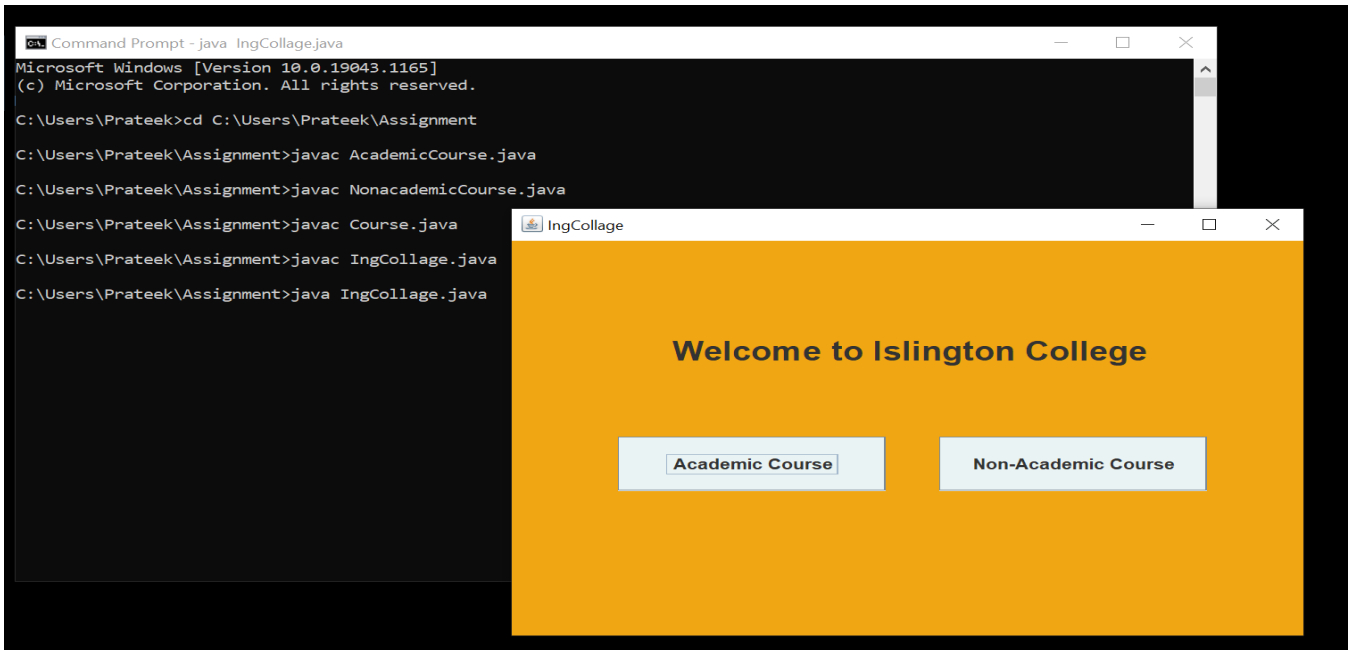


Figure 2: TEST to run program in Cmd

Test 2:

Evidences should be shown of Add course for Academic course:

Objective	To Add course for Academic course
Action:	The following data were entered in order to add course for the Academic course list ➔ courseID: CS4001NI ➔ course name: Programming ➔ duration: 10 days ➔ level: Bachelor in IT ➔ Credit: 360 points ➔ number of assessments: 12 The "Add" button was clicked.
Expected Result:	The Academic Course would be added.
Actual Result:	The Academic Course was added.
Conclusion:	The test is successful.

Table 2: TEST to add academic course

Output:

Figure 3: TEST to add academic course

Test 3:

Evidences should be shown of Add course for Non-Academic course:

Objective	To Add course for Non-Academic course
Action:	The following data were entered in order to add course for the Non-Academic course list ➔ courseID: CS4002NI ➔ course name: Fundamental of Computing ➔ duration: 10 days ➔ prerequisites: XYZ The "Add" button was clicked. .
Expected Result:	The Non-Academic Course would be added.
Actual Result:	The Non-Academic Course was added.
Conclusion:	The test is successful.

Table 3: TEST to add non academic course

Output:

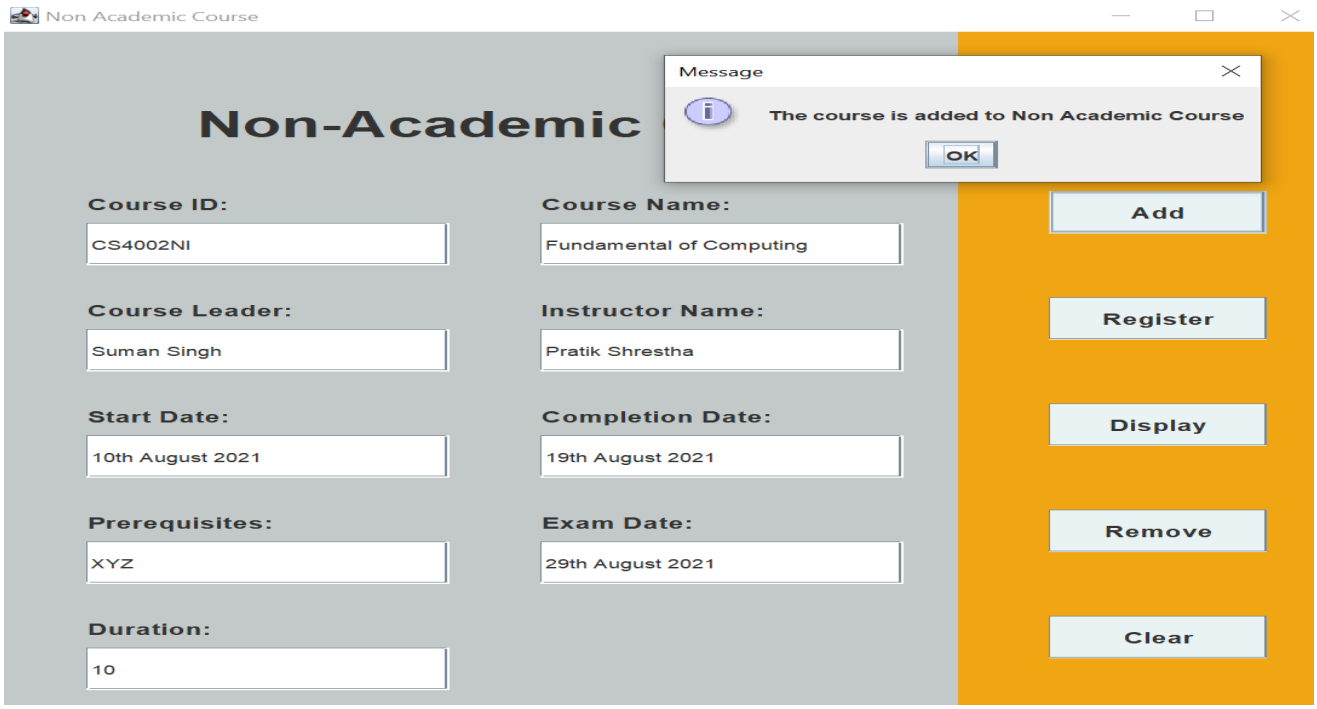


Figure 4: TEST to add non-academic Course

Test 4:

Evidences should be shown to register academic course:

Objective	To register academic course
Action:	The following data are entered to register academic course. ➔ courseID: CS4001NI ➔ course leader: Dhurba Sen ➔ lecturer name: Roshan Shrestha ➔ starting date: 10th August 2021 ➔ completion date: 19th August 2021 The register button is clicked.
Expected Result:	The Academic Course would be registered.
Actual Result:	The Academic Course was registered.
Conclusion:	The test is successful.

Table 4: TEST to register academic course:

Output:

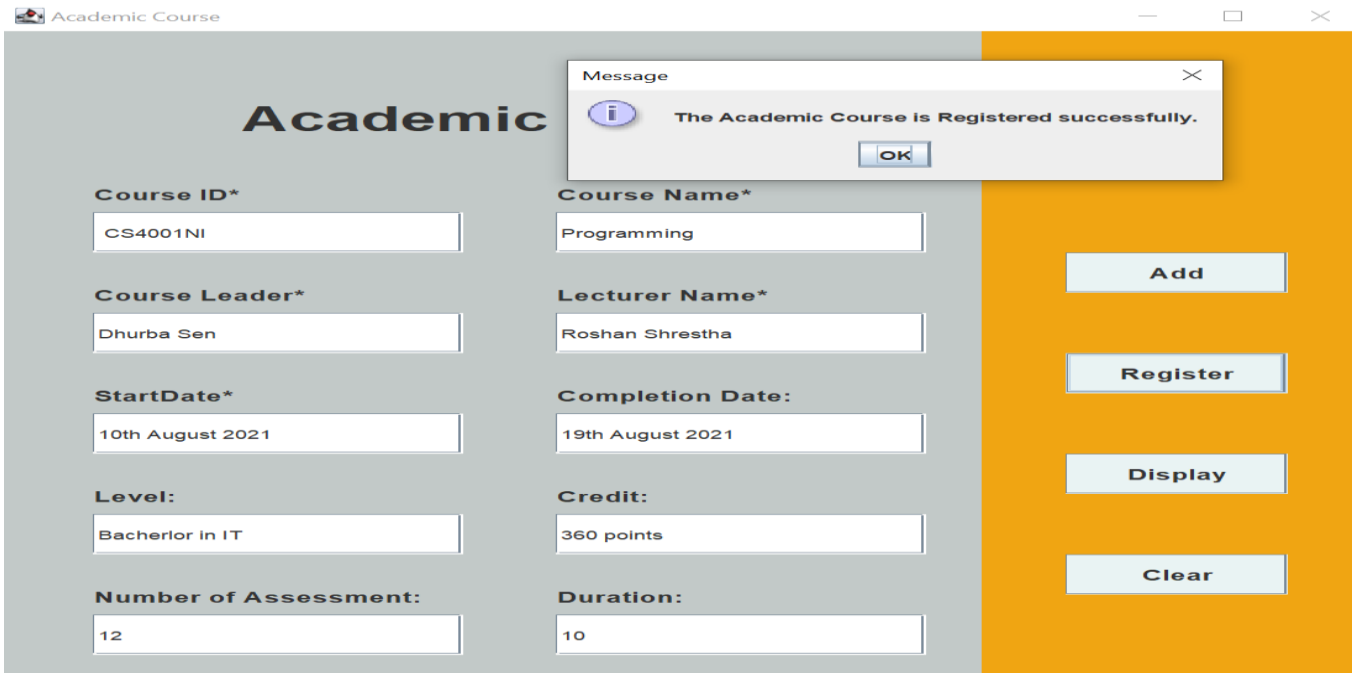


Figure 5: TEST to register academic course

Test 5:

Evidences should be shown to register Non-academic course:

Objective	To register Non-academic course
Action:	The following data are entered to register Non-academic course. ➔ courseID: CS4002NI ➔ course leader: Suman Singh ➔ instructor name: Pratik Shrestha ➔ starting date: 10th August 2021 ➔ completion date: 19th August 2021 ➔ exam date: 29th August 2021 The register button is clicked.
Expected Result:	The Non-Academic Course would be registered.
Actual Result:	The Non-Academic Course was registered.
Conclusion:	The test is successful.

Table 5: TEST to register non-academic class

Output:

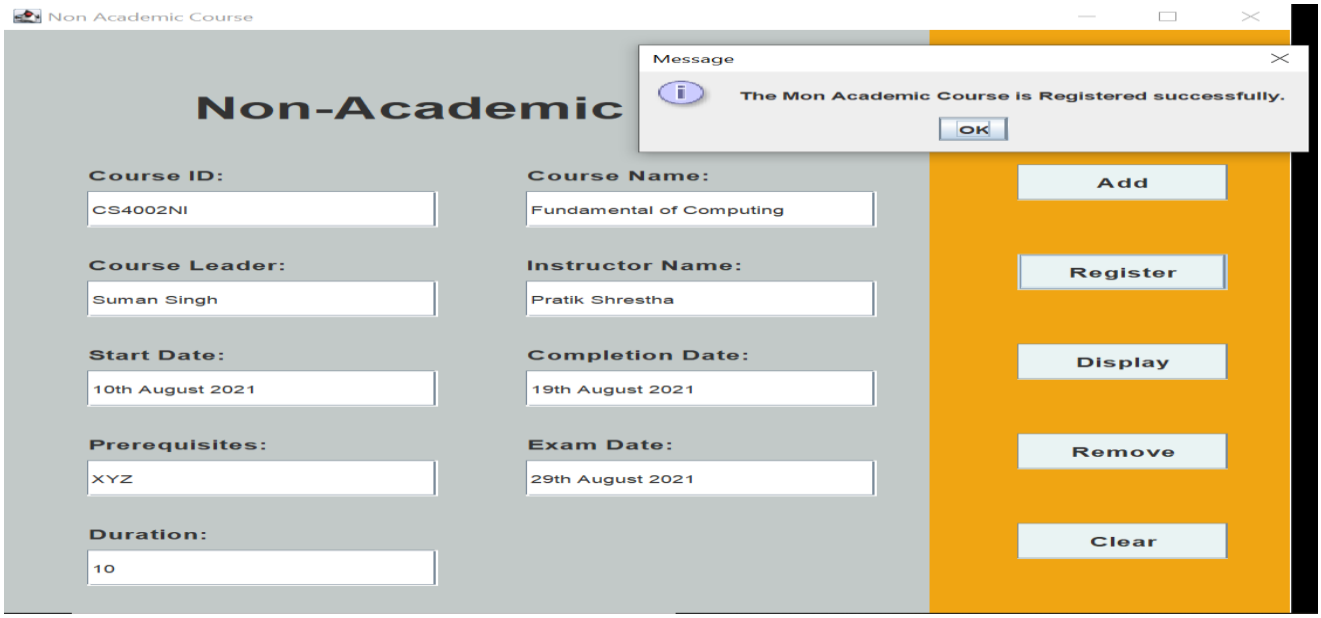


Figure 6: TEST to register Non-academic course

Test 6:

Evidences should be shown remove non-academic course

Objective	To Remove non-academic course
Action:	The following data were entered in order to remove course for the Non-Academic course list courseID: CS4002NI course name: Fundamental of Computing duration: 10 days prerequisites: XYZ The “Remove” button was clicked. .
Expected Result:	The Non-Academic Course would be removed.
Actual Result:	The Non-Academic Course was removed.
Conclusion:	The test is successful.

Table 6: TEST to remove non-academic course

Output:

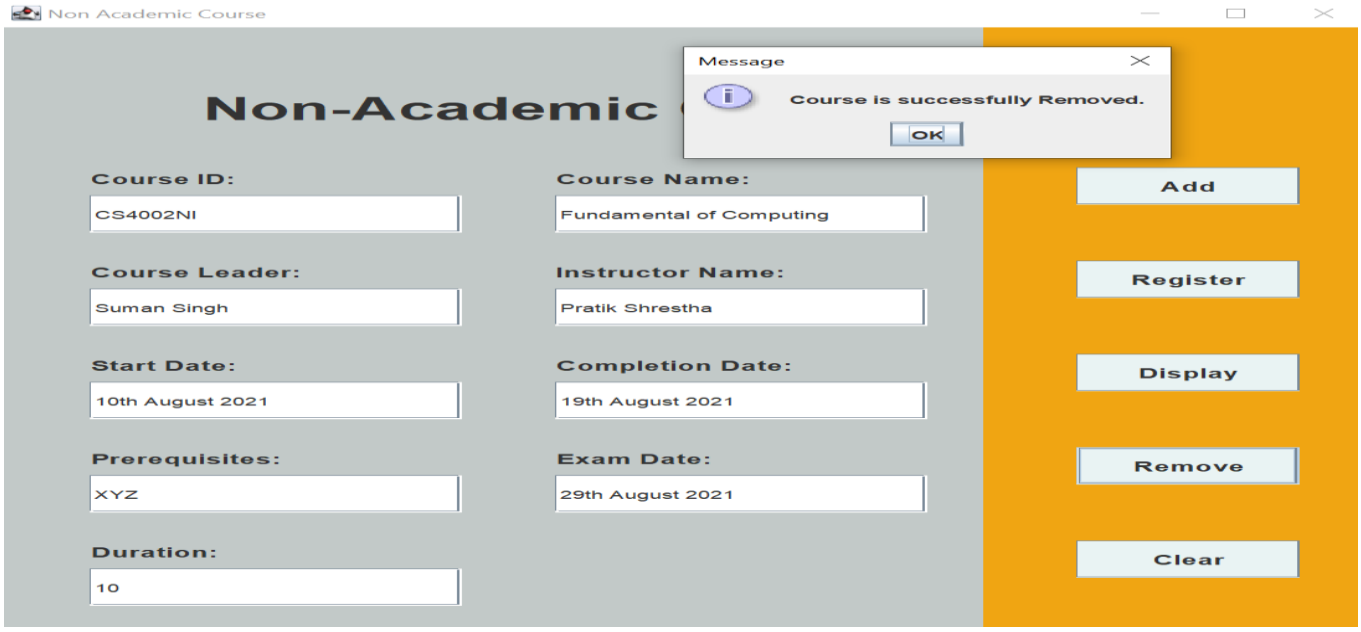


Figure 7: TEST to remove Non-academic course

Test 6:

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

Objective	To remove already removed non-academic course.
Action:	After Adding the course after entering data, registering the course and then removing the course and again trying to re-remove the same course again.
Expected Result:	A message dialog should be displayed as “Course is already Removed.”.
Actual Result:	The message was displayed.
Conclusion:	The test is successful.

Table 7: TEST to remove non-academic course which is already removed.

OUTPUT:

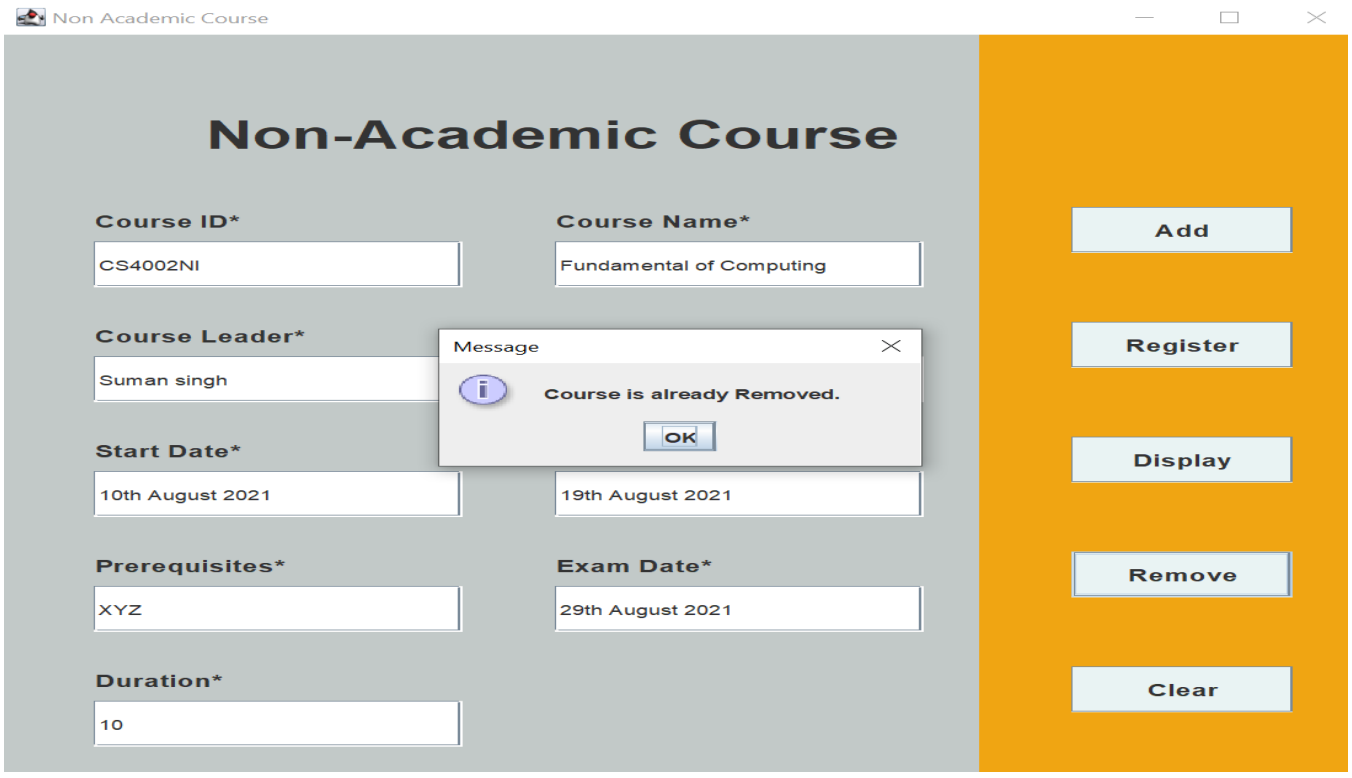


Figure 8: TEST To remove already removed non-academic course

CHAPTER-6 Error Detection and Correction

1. Syntax Error Detection

The error occurred as the parentheses or () is missing here after register method. To remove these error parentheses or () should be added after register method.

Error

```
public void addNonAcademicCourse{
    try{
        //creating obj of user input value
        CourseID2=tfCourseID2.getText();
        CourseName2=tfCourseName2.getText();
        CourseLeader2=tfCourseLeader2.getText();
        InstructorName2 = tfInstructorName2.getText();
        Duration2=tfDuration2.getText();
        StartDate2 = tfStartDate2.getText();
        CompletionDate2 = tfCompletionDate2.getText();
        ExamDate2 = tfExamDate2.getText();
        Prerequisites2 = tfPrerequisites2.getText();
        //converting String to Integer
        int duration2 = Integer.parseInt(Duration2);
        boolean courseIsAdded2 = false;
```

Figure 9: Syntax error

Output

```
    }
}
public void addNonAcademicCourse(){
    try{
        //creating obj of user input value
        CourseID2=tfCourseID2.getText();
        CourseName2=tfCourseName2.getText();
        CourseLeader2=tfCourseLeader2.getText();
        InstructorName2 = tfInstructorName2.getText();
        Duration2=tfDuration2.getText();
        StartDate2 = tfStartDate2.getText();
```

Figure 10: Syntax error correction

2. Logical error

The logical error indicates that the program did not function as expected. The program fails to complete the task as expected. Since logical errors in Java programming do not reflect any sort of coding problem or an error in the use of Java language elements, they can distinguish. The code works well as written; it just is not doing what we want it to do

.

Output:

```
//adding component of main class to panel
panel1.add(ltitle1);
panel1.add(btn1AcademicCourse);
panel1.add(btn2NonAcademicCourse);
panel1.setLayout(null);
panel1.setBackground(smallbox);
//adding panel to frame
frame1.add(panel1);
frame1.setSize(600,400);
```

Class compiled - no syntax errors

Figure 11:Logical error

Correction of logical error:

```
//adding panel to frame
frame1.add(panel1);
frame1.setSize(600,400);
frame1.setVisible(true);

//Creating JFrame and panel for academic class
```

Class compiled - no syntax errors

Figure 12: Correction of logical error

3. Sematic error:

Runtime error occurs even when the program compiles so it's difficult to avoid runtime error previously before compiler does. However, after the program runs, we can find the error. Here the error occurred as the data type of the "InstructorName" should be String as it takes the string value but it is is set as int and the types string and int are not compatible

OUTPUT:

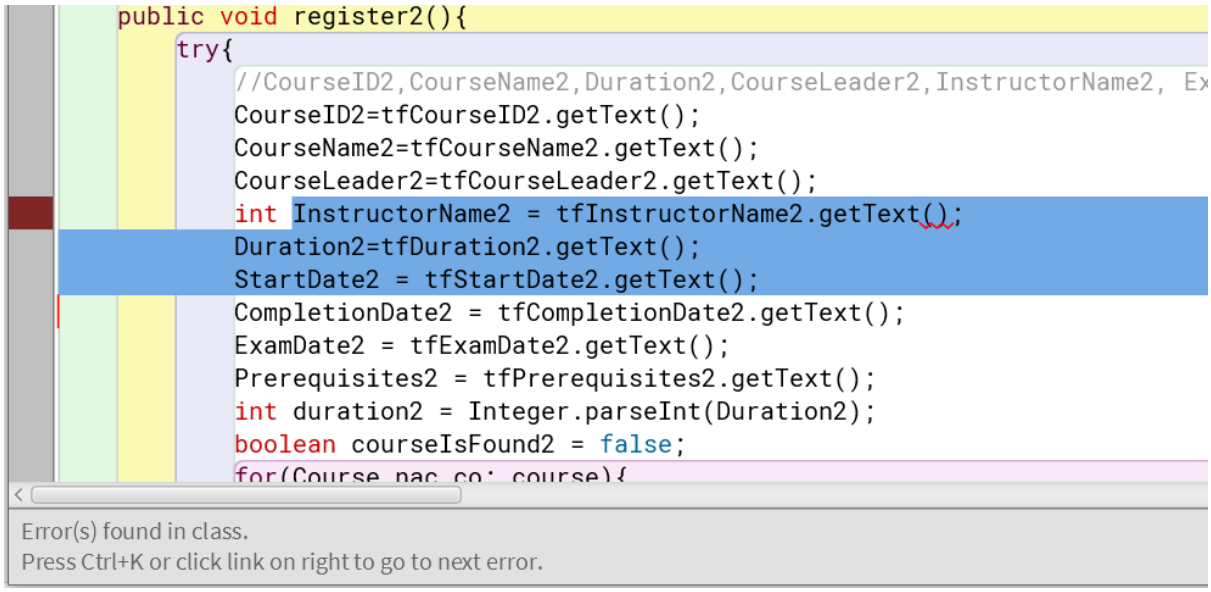


Figure 13: Sematic error

Correction of sematic error:

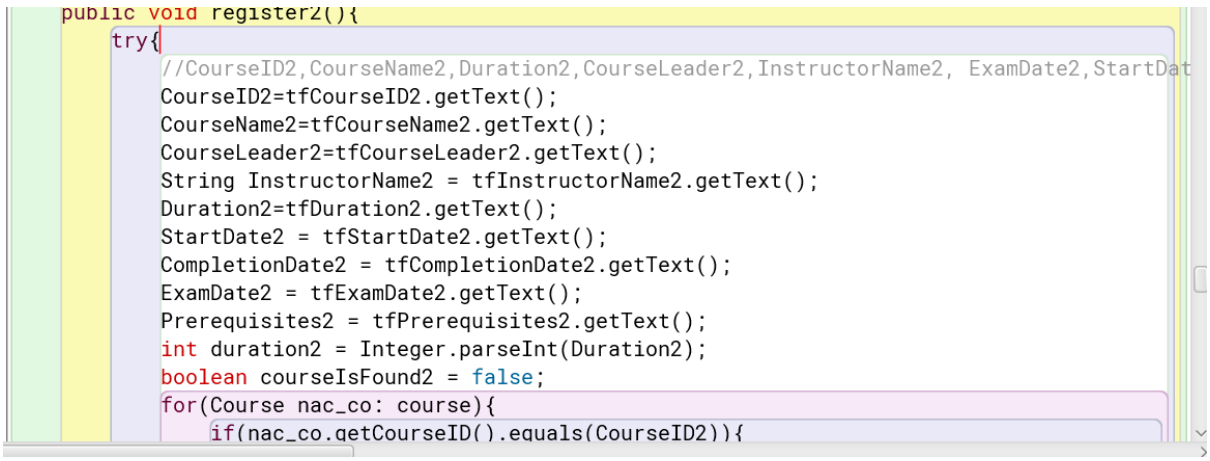


Figure 14: Sematic error correction

CHAPTER-7 CONCLUSION

To summarize, we were assigned this coursework to develop a GUI software that was related to previous coursework. The main form, the Academic course form, and the Non-Academic form were all designed separately. This course seems to be easy at first, but when it came to the coding portion, there were numerous obstacles and confusion. There were an unusual number of mistakes, and it was difficult to identify them out. The graphical user interface (GUI) was a completely new concept. It all began with the graphical user interface's fundamental concepts and terminology. I found that the GUI collects data from the client or user by providing a platform on which they can enter various data or information, which may later be stored in an array and presented according to the user's instructions.

Learning only in class was insufficient to finish this work. At first, there were several errors and challenges with the coding. However, in order to solve the challenges and decrease the misunderstanding, several researches on the relevant issues were conducted. The tasks assigned in this class were completed with 100% effort. From the initial concept, research on java programming GUI was done, and it was then implemented in this coursework, which significantly helped in the completion of the assignments. Regular engagement with lecturers, constant effort, and many researches, and reading over the slides provided by the lecturers and browsing, all helped in developing a deep understanding of GUI and its purpose. In this coursework, the aim of exception handling was also learned and implemented.

Although it was tough at beginning, it was completed on time and submitted on time because to the constant dedication to this project. I learned a lot of new ideas and topics that I wasn't aware of that before. It was a great learning experience to develop a program in Java, and it was also a lot of pleasure to work on this project.

CHAPTER-8 APPENDIX

```
import javax.swing.*; //importing the javax.swing packages
import java.awt.*; //importing java.awt packages
import java.awt.event.*; //importing java.awt.event for event handling
import java.util.*; //importing java.util class
public  class IngCollage implements ActionListener{
JFrame frame1, frame2, frame3; //Declaring frame for the form
JButton btnAcademic, btnNonAcademic; //declaring button for main form

//Declaring  for creating the formprivate JLabel ltitle1,ltitle2, ltitle3, lCourseID1,
lCourseName1,lCourseID2, lCourseName2, lDuration1, lCourseLeader1,
lCourseLeader2, lLecturerName1, lLevel1, lCredit1, lStartDate1, lCompletionDate1,
lNumberOfAssessment1, lInstructorName2, lExamDate2,lPrerequisites2, lStartDate2,
lCompletionDate2, lDuration2;

//Declaring textfield for creating the form
private JTextField tfCourseID1, tfCourseName1, tfDuration1, tfCourseLeader1,
tfLecturerName1, tfLevel1, tfCredit1, tfStartDate1,tfCompletionDate1,
tfNumberOfAssessment1,tfCourseID2,tfCourseName2, tfDuration2,
tfCourseLeader2, tfInstructorName2, tfExamDate2, tfStartDate2, tfCompletionDate2,
tfPrerequisites2;

//Declaring button for Non Academic Course
 JButton btn1Add,btn2Add, btn1Clear,btn2Clear, btn1Register, btn1Display,
btn2Register,btn2Display,btn1AcademicCourse,btn2NonAcademicCourse,
btn2Remove;

 //Declaring panel for form
JPanel panel1, panel2, panel3, panel4, panel5;

        //Creating arrayList for storing value
         ArrayList<Course> course = new ArrayList <Course>();
```

```
        private String CourseID, CourseName, Duration, CourseLeader, LecturerName
        , Level, Credit, StartDate, CompletionDate, NumberOfAssessment;
    private String CourseID2, CourseName2, Duration2, CourseLeader2,
InstructorName2, ExamDate2, StartDate2, CompletionDate2,  Prerequisites2;
    //Creating constructor of main class
    public IngCollage(){
        //creating object for color
        Color btnclr = new Color(233, 243, 243);
        Color smallbox = new Color(240, 165, 18);
        Color bigbox = new Color(194, 201, 200);

        //creating Frame and panel for main class
        frame1 = new JFrame("IngCollage");
        panel1 = new JPanel();

        //Title for main class
      ltitle1= new JLabel("Welcome to Islington College"); //defining ltitle as JLabel type
      ltitle1.setBounds(120,50,500,100);
      ltitle1.setFont(new Font("SanSerif Bold Italic", Font.BOLD, 25));

      //JButton for main class
      //academic course button on main frame
      btn1AcademicCourse = new JButton(" Academic Course ");
      btn1AcademicCourse.setFont(new Font("Arial", Font.BOLD, 14));
      btn1AcademicCourse.setBounds(80,180,200,50);
      btn1AcademicCourse.addActionListener(this);
      btn1AcademicCourse.setBackground(btnclr);

      //Non academic course button on main frame
      btn2NonAcademicCourse = new JButton(" Non-Academic Course ");
      btn2NonAcademicCourse.setFont(new Font("Arial", Font.BOLD, 14));
      btn2NonAcademicCourse.setBounds(320,180,200,50);
      btn2NonAcademicCourse.addActionListener(this);
      btn2NonAcademicCourse.setBackground(btnclr);
```

```
//adding component of main class to panel
panel1.add(ltitle1);
panel1.add(btn1AcademicCourse);
panel1.add(btn2NonAcademicCourse);
panel1.setLayout(null);
panel1.setBackground(smallbox);
//adding panel to frame
frame1.add(panel1);
frame1.setSize(600,400);
frame1.setVisible(true);

//Creating JFrame and panel for academic class
frame2 = new JFrame("Academic Course");
panel2 = new JPanel();
panel4 = new JPanel();

//Creating JLabel for academic class.....
ltitle2 = new JLabel("Academic Course");
ltitle2.setBounds(130,50,350,70);
ltitle2.setFont(new Font("san-serif",Font.BOLD,35));

//Course ID for academicCourse
lCourseID1 = new JLabel("Course ID* ");
tfCourseID1 = new JTextField();
lCourseID1.setBounds(50,150,200,25);
tfCourseID1.setBounds(50,180,200,40);
lCourseID1.setFont(new Font("san-serif",Font.BOLD,15));

//Course Name for academicCourse
lCourseName1 = new JLabel("Course Name* ");
tfCourseName1  = new JTextField();
lCourseName1.setBounds(300,150,200,25);
tfCourseName1.setBounds(300,180,200,40);
```

```
lCourseName1.setFont(new Font("san-serif",Font.BOLD,15));

//CourseLeader for academicCourse
lCourseLeader1 = new JLabel("Course Leader* ");
tfCourseLeader1 = new JTextField();
lCourseLeader1.setBounds(50,250,200,25);
tfCourseLeader1.setBounds(50,280,200,40);
lCourseLeader1.setFont(new Font("san-serif",Font.BOLD,15));

//Lecturer Name for academicCourse
lLecturerName1 = new JLabel("Lecturer Name* ");
tfLecturerName1 = new JTextField();
lLecturerName1.setBounds(300,250,200,25);
tfLecturerName1.setBounds(300,280,200,40);
lLecturerName1.setFont(new Font("san-serif",Font.BOLD,15));

// Start Date for academicCourse
lStartDate1= new JLabel("StartDate* ");
tfStartDate1 = new JTextField();
lStartDate1.setBounds(50,350,200,25);
tfStartDate1.setBounds(50,380,200,40);
lStartDate1.setFont(new Font("san-serif",Font.BOLD,15));

//Completion date for academicCourse
lCompletionDate1 = new JLabel("Completion Date:");
tfCompletionDate1 = new JTextField();
lCompletionDate1.setBounds(300,350,200,25);
tfCompletionDate1.setBounds(300,380,200,40);
lCompletionDate1.setFont(new Font("san-serif",Font.BOLD,15));

// Level for academicCourse
lLevel1= new JLabel("Level: ");
tfLevel1 = new JTextField();
lLevel1.setBounds(50,450,200,25);
```

```
tfLevel1.setBounds(50,480,200,40);
lLevel1.setFont(new Font("san-serif",Font.BOLD,15));

//Credit for academicCourse
lCredit1 = new JLabel("Credit: ");
tfCredit1 = new JTextField();
lCredit1.setBounds(300,450,200,25);
tfCredit1.setBounds(300,480,200,40);
lCredit1.setFont(new Font("san-serif",Font.BOLD,15));

// Number of assessment for academicCourse
lNumberOfAssessment1 = new JLabel("Number of Assessment: ");
tfNumberOfAssessment1 = new JTextField();
lNumberOfAssessment1.setBounds(50,550,200,25);
tfNumberOfAssessment1.setBounds(50,580,200,40);
lNumberOfAssessment1.setFont(new Font("san-serif",Font.BOLD,15));

//Duration for academicCourse
lDuration1 = new JLabel("Duration: ");
tfDuration1 = new JTextField();
lDuration1.setBounds(300,550,200,25);
tfDuration1.setBounds(300,580,200,40);
lDuration1.setFont(new Font("san-serif",Font.BOLD,15));

//Button for Academic Course
//Add Button
btn1Add=new JButton("Add");
btn1Add.setBounds(50,220,120,40);
btn1Add.setFont(new Font("san-serif",Font.BOLD,15));
btn1Add.addActionListener(this);
btn1Add.setBackground(btnclr);

//Register Button
btn1Register=new JButton("Register");
```

```
btn1Register.setBounds(50,320,120,40);
btn1Register.setFont(new Font("san-serif",Font.BOLD,15));
btn1Register.addActionListener(this);
btn1Register.setBackground(btnclr);

//Display Button
btn1Display=new JButton("Display");
btn1Display.setBounds(50,420,120,40);
btn1Display.setFont(new Font("san-serif",Font.BOLD,15));
btn1Display.addActionListener(this);
btn1Display.setBackground(btnclr);

//Clear Button
btn1Clear=new JButton("Clear");
btn1Clear.setBounds(50,520,120,40);
btn1Clear.setFont(new Font("san-serif",Font.BOLD,15));
btn1Clear.addActionListener(this);
btn1Clear.setBackground(btnclr);


//panel set design
panel2.setSize(530,700);
panel2.setLocation(0,0);
panel2.setLayout(null);
panel4.setSize(300,700);
panel4.setLocation(525,0);
panel4.setLayout(null);
panel4.setBackground(smallbox);
panel2.setBackground(bigbox);

//adding component on panel2 for Academic class
panel2.add(ltitle2);
panel2.add(lCourseID1);
panel2.add(lCourseName1);
```

```
panel2.add(lDuration1);
panel2.add(lCourseLeader1);
panel2.add(lLecturerName1);
panel2.add(lLevel1);
panel2.add(lCredit1);
panel2.add(lCompletionDate1);
panel2.add(lNumberOfAssessment1);
panel2.add(lStartDate1);
//adding JTextfield on panel
panel2.add(tfCourseID1);
panel2.add(tfCourseName1);
panel2.add(tfDuration1);
panel2.add(tfCourseLeader1);
panel2.add(tfLecturerName1);
panel2.add(tfLevel1);
panel2.add(tfCredit1);
panel2.add(tfStartDate1);
panel2.add(tfCompletionDate1);
panel2.add(tfNumberOfAssessment1);

//Adding component on panel4 (academic Course)
panel4.add(btn1Add);
panel4.add(btn1Register);
panel4.add(btn1Display);
panel4.add(btn1Clear);

//adding panel on frame
frame2.add(panel2);
frame2.add(panel4);
frame2.setLayout(null);
frame2.setSize(750,700); //setting the size of frame

//Creating JFrame and panel for Non academic class
frame3 = new JFrame("Non Academic Course");
```

```
//creating panels for non academic class
panel3 = new JPanel();
panel5 = new JPanel();
//Components for Non Academic Course

//Tittle for Non Academic Course
ltitle3 = new JLabel("Non-Academic Course");
ltitle3.setBounds(110,50,500,70);
ltitle3.setFont(new Font("san-serif",Font.BOLD,35));

//Course ID for Non Academic Course
lCourseID2 = new JLabel("Course ID: ");
tfCourseID2 = new JTextField();
lCourseID2.setBounds(50,150,200,25);
tfCourseID2.setBounds(50,180,200,40);
lCourseID2.setFont(new Font("san-serif",Font.BOLD,15));

//course Name for Non Academic Course
lCourseName2 = new JLabel("Course Name: ");
tfCourseName2  = new JTextField();
lCourseName2.setBounds(300,150,200,25);
tfCourseName2.setBounds(300,180,200,40);
lCourseName2.setFont(new Font("san-serif",Font.BOLD,15));

// Course Leader for Non Academic Course
lCourseLeader2 = new JLabel("Course Leader: ");
tfCourseLeader2 = new JTextField();
lCourseLeader2.setBounds(50,250,200,25);
tfCourseLeader2.setBounds(50,280,200,40);
lCourseLeader2.setFont(new Font("san-serif",Font.BOLD,15));
//Instructor for Non Academic Course
lInstructorName2 = new JLabel("Instructor Name: ");
tfInstructorName2 = new JTextField();
lInstructorName2.setBounds(300,250,200,25);
```

```
tfInstructorName2.setBounds(300,280,200,40);
lInstructorName2.setFont(new Font("san-serif",Font.BOLD,15));

//StartDate for Non Academic Course
lStartDate2 = new JLabel("Start Date: ");
tfStartDate2 = new JTextField();
lStartDate2.setBounds(50,350,200,25);
tfStartDate2.setBounds(50,380,200,40);
lStartDate2.setFont(new Font("san-serif",Font.BOLD,15));

//Completion Date for Non Academic Course
lCompletionDate2 = new JLabel("Completion Date: ");
tfCompletionDate2 = new JTextField();
lCompletionDate2.setBounds(300,350,200,25);
tfCompletionDate2.setBounds(300,380,200,40);
lCompletionDate2.setFont(new Font("san-serif",Font.BOLD,15));

//Prerequisites for Non Academic Course
lPrerequisites2 = new JLabel("Prerequisites: ");
tfPrerequisites2 = new JTextField();
lPrerequisites2.setBounds(50,450,200,25);
tfPrerequisites2.setBounds(50,480,200,40);
lPrerequisites2.setFont(new Font("san-serif",Font.BOLD,15));

//Exam Date for Non Academic Course
lExamDate2 = new JLabel("Exam Date: ");
tfExamDate2 = new JTextField();
lExamDate2.setBounds(300,450,200,25);
tfExamDate2.setBounds(300,480,200,40);
lExamDate2.setFont(new Font("san-serif",Font.BOLD,15));
//Duration for Non Academic Course
lDuration2 = new JLabel("Duration: ");
tfDuration2 = new JTextField();
lDuration2.setBounds(50,550,200,25);
```

```
tfDuration2.setBounds(50,580,200,40);
lDuration2.setFont(new Font("san-serif",Font.BOLD,15));

//Button for Non Academic Course
//Add Button
btn2Add=new JButton("Add");
btn2Add.setBounds(50,150,120,40);
btn2Add.setFont(new Font("san-serif",Font.BOLD,15));
btn2Add.addActionListener(this);
btn2Add.setBackground(btnclr);

//Register Button
btn2Register=new JButton("Register");
btn2Register.setBounds(50,250,120,40);
btn2Register.setFont(new Font("san-serif",Font.BOLD,15));
btn2Register.addActionListener(this);
btn2Register.setBackground(btnclr);

//Display Button
btn2Display=new JButton("Display");
btn2Display.setBounds(50,350,120,40);
btn2Display.setFont(new Font("san-serif",Font.BOLD,15));
btn2Display.addActionListener(this);
btn2Display.setBackground(btnclr);

//Remove Button
btn2Remove = new JButton("Remove");
btn2Remove.setBounds(50,450,120,40);
btn2Remove.setFont(new Font("san-serif",Font.BOLD,15));
btn2Remove.addActionListener(this);
btn2Remove.setBackground(btnclr);

//Clear Button
btn2Clear = new JButton("Clear");
```

```
btn2Clear.setBounds(50,550,120,40);
btn2Clear.setFont(new Font("san-serif",Font.BOLD,15));
btn2Clear.addActionListener(this);
btn2Clear.setBackground(btnclr);

//panel3 set design
panel3.setBackground(bigbox);
panel5.setBackground(smallbox);
panel3.setSize(530,700);
panel3.setLocation(0,0);
panel3.setLayout(null);

//Panel5 set Design
panel5.setSize(300,700);
panel5.setLocation(530,0);
panel5.setLayout(null);

//Adding component on panel3
panel3.add(ltitle3);
panel3.add(lCourseID2);
panel3.add(lCourseName2);
panel3.add(lCourseLeader2);
panel3.add(lInstructorName2);
panel3.add(lStartDate2);
panel3.add(lCompletionDate2);
panel3.add(lPrerequisites2);
panel3.add(lExamDate2);
panel3.add(lDuration2);
panel3.add(tfCourseID2);
panel3.add(tfCourseName2);
panel3.add(tfCourseLeader2);
panel3.add(tfInstructorName2);
panel3.add(tfStartDate2);
panel3.add(tfCompletionDate2);
```

```
        panel3.add(tfPrerequisites2);
        panel3.add(tfExamDate2);
        panel3.add(tfDuration2);
        //Adding components on panel5
        panel5.add(btn2Add);
        panel5.add(btn2Register);
        panel5.add(btn2Display);
        panel5.add(btn2Remove);
        panel5.add(btn2Clear);
        //adding panel to frame
        frame3.add(panel3);
        frame3.add(panel5);
        frame3.setLayout(null);
        frame3.setSize(750,700); //setting the size of frame


    }
    public void actionPerformed(ActionEvent e){//Using Action Listener
        if ( e.getSource() == btn1Clear ) {
                    tfCourseID1.setText("");
                    tfCourseName1.setText("");
                    tfCourseLeader1.setText("");
                    tfLecturerName1.setText("");
                    tfStartDate1.setText("");
                    tfCompletionDate1.setText("");
                     tfLevel1.setText("");
                    tfCredit1.setText("");
                     tfDuration1.setText("");
                    tfNumberOfAssessment1.setText("");
        }
        else if ( e.getSource() == btn2Clear ) {
                    tfCourseID2.setText("");
                    tfCourseName2.setText("");
                    tfCourseLeader2.setText("");
```

```
            tfInstructorName2.setText("");
            tfStartDate2.setText("");
            tfCompletionDate2.setText("");
            tfPrerequisites2.setText("");
            tfExamDate2.setText("");
            tfDuration2.setText("");
        }
        else if(e.getSource().equals(btn1AcademicCourse)){//to open Academic Course
form
            frame2.setVisible(true);
            frame3.setVisible(false);
        }
        else if(e.getSource().equals(btn2NonAcademicCourse)){// to open Non
Academic Course form
            frame2.setVisible(false);
            frame3.setVisible(true);
        }

        else if(e.getSource().equals(btn1Add)){ //to add Value of Acadmeic Course
            addAcademicCourse();
        }
        else if(e.getSource().equals(btn2Add)){ //to add Value of Non Acadmeic Course
            addNonAcademicCourse();
        }
        else if(e.getSource() == btn1Register){//to Register value of Acadmeic Course
            register1();
        }
        else if(e.getSource() == btn2Register){//to Register value of Non-Acadmeic
Course
            register2();
        }
        else if(e.getSource()== btn2Remove){
            remove();
        }
```

```
    else if(e.getSource() == btn1Display){
            for(Course ac_co: course){
        AcademicCourse ac_d = (AcademicCourse) ac_co;
        ac_d.Display();
       }
    }
    else if(e.getSource() == btn2Display){
            for(Course nac_co: course){
                NonacademicCourse nac_d = (NonacademicCourse) nac_co;
                nac_d.Display();
            }
      }
}
public void register1(){
    try{
            CourseID=tfCourseID1.getText();
            CourseName=tfCourseName1.getText();
            CourseLeader=tfCourseLeader1.getText();
            LecturerName = tfLecturerName1.getText();
            Duration=tfDuration1.getText();
            Level=tfLevel1.getText();
            Credit=tfCredit1.getText();
            NumberOfAssessment=tfNumberOfAssessment1.getText();
            StartDate = tfStartDate1.getText();
             CompletionDate = tfCompletionDate1.getText();
             int duration1 = Integer.parseInt(Duration);
             int numberOfAssessment1 = Integer.parseInt(NumberOfAssessment);
            boolean courseIsFound1 = false;
            for(Course ac_co: course){
             if(ac_co.getCourseID().equals(CourseID)){
                AcademicCourse ac_c = (AcademicCourse) ac_co;
              if(ac_c.getIsRegistered()){
                JOptionPane.showMessageDialog(frame2,"The Academic
                Course is already Registered. ");
```

```
                courseIsFound1 = true;
            }
            else{
                ac_c.Registered(CourseLeader, LecturerName, StartDate,
                CompletionDate);
                JOptionPane.showMessageDialog(frame2,"The Academic
                Course is Registered successfully. ");
                courseIsFound1 = true;
                break;
            }
        }
    }
            if(courseIsFound1 == false){
                JOptionPane.showMessageDialog(frame2,"The Academic
                Course is not Found. ");
            }
    }
catch(Exception ex1){
        JOptionPane.showMessageDialog(frame2,"Please input valid value. ");
    }
}
  public void register2(){
    try{
        CourseID2=tfCourseID2.getText();
        CourseName2=tfCourseName2.getText();
        CourseLeader2=tfCourseLeader2.getText();
        InstructorName2 = tfInstructorName2.getText();
        Duration2=tfDuration2.getText();
        StartDate2 = tfStartDate2.getText();
        CompletionDate2 = tfCompletionDate2.getText();
        ExamDate2 = tfExamDate2.getText();
        Prerequisites2 = tfPrerequisites2.getText();
        int duration2 = Integer.parseInt(Duration2);
        boolean courseIsFound2 = false;
```

```
        for(Course nac_co: course){
            if(nac_co.getCourseID().equals(CourseID2)){
                    NonacademicCourse nac_c = (NonacademicCourse) nac_co;
                if(nac_c.getIsRegistered()){
                        JOptionPane.showMessageDialog(frame3,"The Non
                        Academic Course is already Registered. ");
                     courseIsFound2 = true;
                }
                else{
                        nac_c.register(CourseLeader2, InstructorName2, ExamDate2,
                        StartDate2, CompletionDate2);
                        JOptionPane.showMessageDialog(frame3,"The Non Academic
                        Course is Registered successfully. ");
                        courseIsFound2 = true;
                        break;
                        }
                 }
        }
                 if(courseIsFound2 == false){
                        JOptionPane.showMessageDialog(frame3,"The Non Academic
                        Course is not Found. ");
                         }
                }
    catch(Exception ex2){
                JOptionPane.showMessageDialog(frame3,"Please input valid value. ");
                }
    }
  public void addAcademicCourse(){
    try{
        CourseID=tfCourseID1.getText();
        CourseName=tfCourseName1.getText();
        CourseLeader=tfCourseLeader1.getText();
        LecturerName = tfLecturerName1.getText();
        Duration=tfDuration1.getText();
```

```
        Level=tfLevel1.getText();
        Credit=tfCredit1.getText();
        NumberOfAssessment=tfNumberOfAssessment1.getText();
        StartDate = tfStartDate1.getText();
        CompletionDate = tfCompletionDate1.getText();
        int duration1 = Integer.parseInt(Duration);
        int numberOfAssessment1 = Integer.parseInt(NumberOfAssessment);
        boolean courseIsAdded1 = false;
        for(Course ac_co: course){
            if(ac_co.getCourseID().equals(CourseID)){
                courseIsAdded1 = true;
                break;
            }
        }
        if(courseIsAdded1 == false){
            AcademicCourse ac_co = new AcademicCourse(CourseID,
            CourseName, duration1, Level, Credit,numberOfAssessment1);
            course.add(ac_co);
            JOptionPane.showMessageDialog(frame2,"The course is added to
            Academic Course");
        }
        else if(courseIsAdded1 == true){
            JOptionPane.showMessageDialog(frame2,"The Academic Course is
            already added. ");
        }
      }
      catch(Exception exp1){
            JOptionPane.showMessageDialog(frame2,"Please enter valid value");
            }
        }
    public void addNonAcademicCourse(){
      try{
        //creating obj of user input value
        CourseID2=tfCourseID2.getText();
```

```
        CourseName2=tfCourseName2.getText();
        CourseLeader2=tfCourseLeader2.getText();
        InstructorName2 = tfInstructorName2.getText();
        Duration2=tfDuration2.getText();
        StartDate2 = tfStartDate2.getText();
        CompletionDate2 = tfCompletionDate2.getText();
        ExamDate2 = tfExamDate2.getText();
        Prerequisites2 = tfPrerequisites2.getText();
        //converting String to Integer
        int duration2 = Integer.parseInt(Duration2);
        boolean courseIsAdded2 = false;
        for(Course nac_co: course){//creating obj of non academic course from Course
class
           if(nac_co.getCourseID().equals(CourseID2)){
                courseIsAdded2 = true;
                break;
                }
        }
        if(courseIsAdded2 == false){//if course is not added
                NonacademicCourse nac_co = new NonacademicCourse(CourseID2,
                CourseName2, duration2, Prerequisites2);
                course.add(nac_co);//adding non academic class
                OptionPane.showMessageDialog(frame3,"The course is added to Non
                Academic Course");//JOptionPanel  for adding non academic course
        }
        else if(courseIsAdded2 == true){
                JOptionPane.showMessageDialog(frame3,"The Non-Academic Course
                is already added. ");//JOptionPanel  if  non academic course already
                added
                }
      }
      catch(Exception exp2){
                JOptionPane.showMessageDialog(frame3,"Please enter valid value");
                //JOptionPanel  for non academic course if value is invalid
```

```
                }
    }
    public void remove(){
       try{
                CourseID2 = tfCourseID2.getText();
                 if(CourseID2.equals("")){
                 JOptionPane.showMessageDialog(frame3,"Invalid Value.");
          }
          else{
             boolean courseIsFound2 = false;
             for(Course nac_co: course){
                          if(nac_co.getCourseID().equals(CourseID2)){
                           NonacademicCourse nac_r = (NonacademicCourse) nac_co;
                          nac_r.getIsRemoved();
                           if(nac_r.getIsRemoved() == true){
                          JOptionPane.showMessageDialog(frame3,"Course  is  already
                          Removed."); //JOptionPanel for frame3 remove
                          courseIsFound2 = true;
                    }
                    else{
                           nac_r.remove();
                          JOptionPane.showMessageDialog(frame3,"Course is successfully
                          Removed.");
                           courseIsFound2 = true;
                           }
                 }
           }
             if(courseIsFound2 == false){
                JOptionPane.showMessageDialog(frame3,"Non  Academic  Course  is
                not found.");
             }
          }
       }
       catch(Exception expr){//Exception catch for frame3
```

```
                JOptionPane.showMessageDialog(frame3,"Please enter Valid value.");
                //JOptionPanel for frame3
            }
        }
        //static main method
        public static void main(String[] args){
            new IngCollage();//invoking constructor
        }
}
```