

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: Aayush Bidari

Group: N2

London Met ID: 20049122

College ID: Np01NT4S210046

Assignment Due Date: 20th August, 2021

Assignment Submission Date: 20th August, 2021

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

Table of Contents

Table of Figures.....	3
Table of Tables.....	3
Introduction	4
Tools used:	4
BlueJ.....	4
MS Word	4
Class Diagram:.....	5
Pseudo code:	6
Method Description:.....	9
Test 1:.....	10
Running program through command Prompt.	10
TEST 2:	11
A: Add course for acedimic.....	11
B: register course for acedimic:	12
C: Add non_academic Course	13
D: Register non_academic Course	14
E: Remove non_academic Course	15
Test:3	16
A: Trying to add duplicate courseID.....	16
B: Trying to register already registered course	17
C: Trying to remove the non-academic course which is already removed	18
Error and Correction:	19
Conclusion:	22
Appendix1:.....	22
Appendix2:.....	41
References	48

Table of Figures

Figure 1 running through command	11
Figure 2 Adding and displaying in academic.....	12
Figure 3 registering and displaying academic.....	13
Figure 4 Add non_academic Course.....	14
Figure 5 Register non_academic Course	15
Figure 6 Remove non_academic Course	16
Figure 7 Trying to add duplicate courseID.....	17
Figure 8 Trying to register already registered course	18
Figure 9 syntax error.....	20
Figure 10 correction of syntax error.....	20

Table of Tables

Table 1 Running through command prompt	10
Table 2 Add course for academic	12
Table 3 : register course for academic	13
Table 4 Add non_academic Course	14
Table 5 Register non_academic Course	14
Table 6 : Remove non_academic Course.....	15
Table 7 Trying to add duplicate courseID	16
Table 8 Trying to register already registered course	17
Table 9 Trying to remove the non-academic course which is already removed ...	18

Introduction

Java is a network-centric, multi-platform, object-oriented programming language. Java is a class-based, object-oriented programming language with a wide range of applications.

The report is based on the java program, in which we're going to add a class to our previous project and create an interface that holds information about academic and non_academic courses in one graphical user interface (GUI). In this coursework we got to work with the concept of buttons. We got to more about GUI concept by utilizing the concept of the frame panel label buttons and many more. In this program user can add and register the course through the GUI, and the course is added and registered on the arraylist of type courses. We also can view the added course through display button.

Tools used:

BlueJ:

BlueJ is a development environment that enables you to rapidly and simply build Java apps. BlueJ is therefore a strong Java graphics shell/REPL. Portable BlueJ works on Windows, Mac OS X, Linux and other Java-running systems. It may also be operated without a USB stick installation. All the code is written in bluej.

MS Word:

A graphical word processing application that users may type with is Microsoft Word, or MS Word (commonly known as Word). It's manufactured by Microsoft computer business. It aims to enable users to type document and to save it. It includes useful document-making features, similar to other word processors.

Class Diagram:

ING College

- JLabel AcedimicCourse;JLabel lblCourseID; JTextField txtCourseID;JLabel lblCourseName;JTextField txtCourseName;JLabel lblDuration;JTextField txtDuration;JLabel lblLecturerName;JTextField txtLecturerName;JLabel lblCourseLeader;JTextField txtCourseLeader;JLabel lblLevel;JTextField txtLevel;JLabel lblCredit;JTextField txtCerdit;JLabel lblStarting_date;JTextField txtStarting_date; JLabel lblCompletion_date; JTextField txtCompletion_date;JLabel lblNumberof_assessments; JTextField txtNumberof_assessments;JLabel lblCourseID3; JTextField txtCourseID3;
- JLabel NonAcedimicCourse;JLabel lblCourseID1; JTextField txtCourseID1;JLabel lblCourseName1;JTextField txtCourseName1;JLabel lblDuration1;JTextField txtDuration1; JLabel lblInstructorName;JTextField txtInstructorName;JLabel lblCourseLeader1;JTextField txtCourseLeader1;JLabel lblExamDate;JTextField txtExamDate;JLabel lblprerequisite ; JTextField txtprerequisite ;JLabel lblStart_date;JTextField txtStart_date;JLabel lblCompletion_date1; JTextField txtCompletion_date1; JLabel lblCourseID2;JTextField txtCourseID2;
- JButton ADD;JButton Register;JButton Display;JButton Clear;JButton ADD1;JButton Register1;JButton Display1;JButton Clear1;JButton Remove;

+add(actionPeformed):event
+register(actionPeformed):event
+remove(actionPeformed):event
+clear(actionPeformed):event
+display(actionPeformed):event
+ING_Collage;
+main()

Psudo code:

```
IMPORT javax.swing.*;
IMPORT java.awt.*;
IMPORT java.awt.event.*;
IMPORT java.util.*; Instance:
CREATE class ING_College
    DEFINE ArrayList <Courses> acedemicCourseList;
    DEFINE ArrayList <Courses> nonacedimicCourseList;
    DEFINE Instance variable of acedimiccourse and nonacedimiccourse;
    CREATE constructor of class
        CREATE components
            DEFINE frame;
            DEFINE label;
            DEFINE panel;
            DEFINE button;
            DEFINE text field';
            DEFINE menu;

    DEFINE actionPerformed Method for Home button
        Frame.remove(Acedimic);
        Frame.remove(NonAcedimic);
        Frame.add(welcome);
    DEFINE actionPerformed method for exit button
        Frame.dispose()

    DEFINE actionPerformed Method for acaedimic button
        frame.remove(welcome);
        frame.remove(NonAcaedimic);
        frame.add(Acaedimic);

    DEFINE actionPerformed method for Non_acaedimic button
        frame.remove(welcome);
        frame.remove(Acaedimic);
        frame.add(NonAcaedimic);

    DEFINE actionPerformed Method for acaedimic add button
```

```

GET all the data from text field of acaedimicCourse
CHECK textfeilds
    IF (IS TextFeild Empty)
        YES: Show date required messages
    END IF
TRY
    CONVERT duration to number
    CONVERT number of assessment to number
END TRY
CATCH
    DISPLAY error message
END CATCH
CREATE object for AcedimicCourse and pass the fetched date
ADD.acedimicCourseObject to arraylist.

DISPLAY add successful message

```

```

DEFINE actionperformed Method for nonacedimic add button
    GET all the data from text field of Non_acaedimicCourse

    CHECK textfeilds
        IF (IS TextFeild Empty)
            YES: Show date required messages
        END IF
    TRY
        CONVERT duration to number
    END TRY
    CATCH
        DISPLAY error message
    END CATCH

    CREATE object for Non_AcedimicCourse and pass
the fetched date
ADD.Non_acedimicCourseObject to arraylist.

DISPLAY add successful message

```

```

DEFINE actionperformed Method for acaedimic Register button
    GET all the data from text field of acaedimicCourse
    CHECK textfeilds

```

```

        IF (IS TextFeild Empty)
        YES: Show date required messages
        END IF
    FOR (Get data of acedimicCourseList)
        IF (data id == textfeild id)
            CREATE object of acedimicCourse and add data
            IF(acedimicCourse is not registerd)
                REGISTER acedimicCourse and pass fleched data
                DISPLAY coueseregisterd methord
            ENDIF
        ELSE
            DISPLAY invalid massage
        ENDELSE
    END IF
END FOR

```

DEFINE actionperformed Method for Non_acaedimic Register button

```

    GET all the data from text field of Non_acaedimic
    CHECK textfeilds
        IF (IS TextFeild Empty)
        YES: Show date required messages
        END IF
    FOR (Get data of Non_acaedimicCourseList)
        IF (data id == textfeild id)
            CREATE object of Non_acaedimicCourse and add data
            IF(Nom_acaedimicCourse is not registerd)
                REGISTER Non_acedimicCourse and pass fleched
data
                DISPLAY coureregisterd method
            ENDIF
        ELSE
            DISPLAY invalid massage
        ENDELSE
    END IF
END FOR

```

DEFINE actionperformed Method for Non_acaedimic Remove button

```

    CHECK if CourseID is empty or not
    FOR (Get data of Non_acaedimicCourseList)
        IF (data ID == textfeild ID)

```



```

        CREATE object of Non_academicCourse and add data
        IF(Nom_academicCourse is not Removed)
            REMOVE Non_academicCourse
            DISPLAY removed message
        ENDIF
        IF(data != getcourseID)
            DISPLAY ID didn't exist
        ENDIF
    END FOR

```

```

DEFINE actionPerformed Method for Non_academic clear button
    SetText empty for every textfield

```

```

DEFINE actionPerformed Method for academic clear button

    SetText empty for every textfield

```

```

DEFINE actionPerformed Method for academic Display button

```

```

DEFINE actionPerformed Method for Non_academic Display button

```

Method Description:

- actionPerformed(method for academic add button)
This method gets the value from textfield and add the values to arraylist academiccourselist.
- actionPerformed(method for academic register button)
This method gets the value from textfield and register it to arraylist academiccourselist
- actionPerformed(method for academic display button)
This method display the values from the arraylist academiccourselist
- actionPerformed(method for academic clear button)
This method sets the value of textfields to empty
- actionPerformed(method for Non_academic add button)

This method gets the value from textfeild and add the values to araylist nonacedimiccourselist

- actionPeformed(methord for Non_acedimic register button)
This method gets the value from textfeild and register it to araylist nonacedimiccourselist
- actionPeformed(methord for Non_acedimic display button)
This method display the values from the araylist nonacedimiccourselist
- actionPeformed(methord for Non_acedimic remove button)
This method removes the enterd course id from the araylist nonacedimiccourselist
- actionPeformed(methord for Non_acedimic clear button)
This method sets the value of textfeilds to empty

Test 1:

Running program through command Prompt.

TEST NO.	1
Objective	To run java program through command Prompt.
Action	Java ING_Collage.java
Expected result	ING_Collage would be compiled
Actual result	ING_Collage was compiled
conclusion	sucessful

Table 1 Running through command prompt

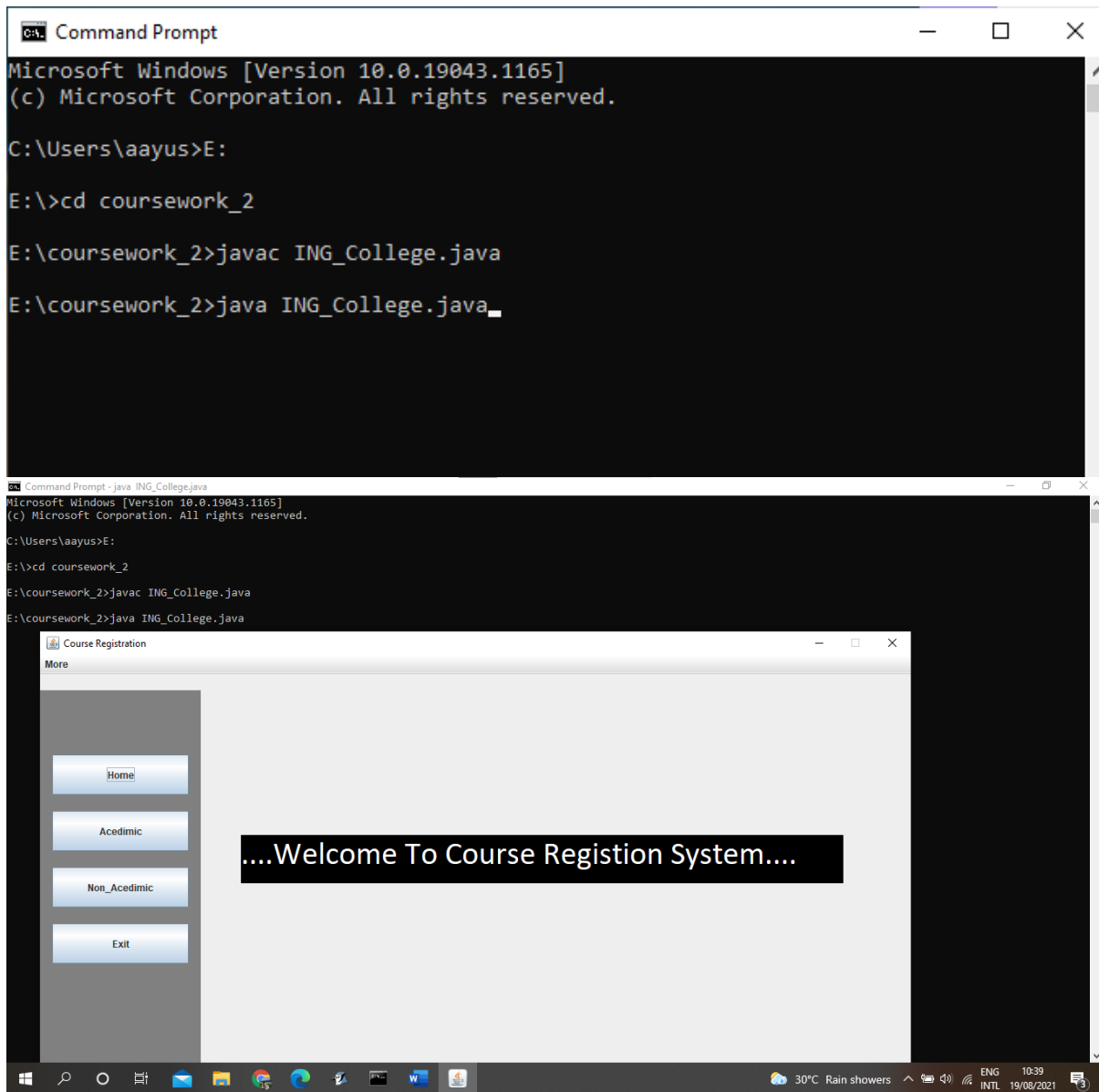


Figure 1 running through command

TEST 2:

A: Add course for acedimic

TEST NO.	2A:
Objective	To add user given value to arraylist acedimicCourselist

Action	Values from textfeild is added to arraylist
Expected result	Values would be added
Actual result	Values was added
conclusion	successful

Table 2 Add course for acedimic



Figure 2Adding and displaying in academic

B: register course for acedimic:

TEST NO.	2B:
Objective	To Register user given value to arraylist acedimicCourseList
Action	Values from textfeild is registered to

	added arraylist
Expected result	Values would be registered
Actual result	Values were registered
conclusion	successful

Table 3 : register course for acedimic

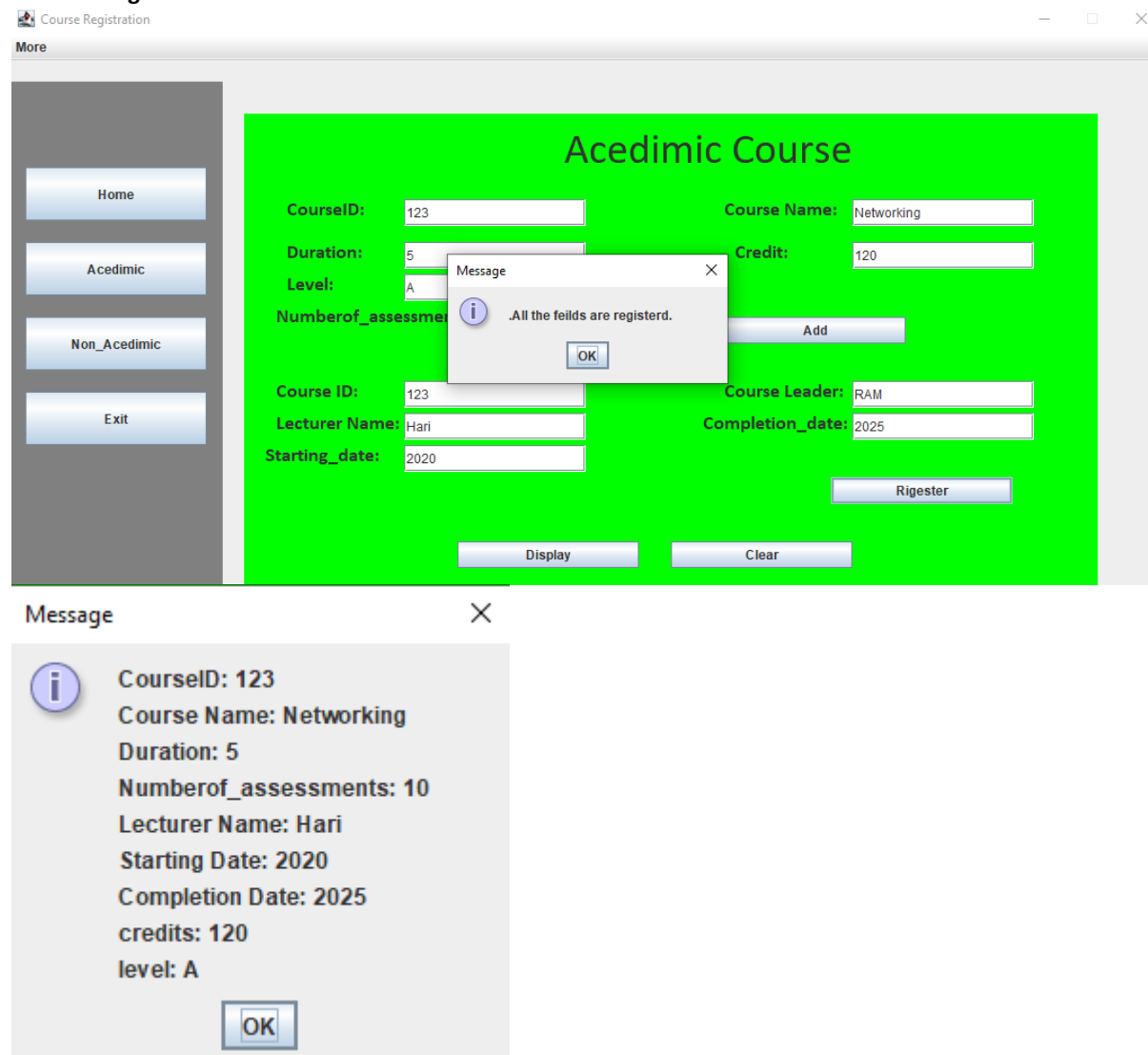


Figure 3 registering and displaying acedimic

C: Add non_academic Course

TEST NO.	2C:
Objective	To add user given value to arraylist nonacedimicCourseList
Action	Values from textfeild is added to arraylist

Expected result	Values would be added
Actual result	Values were added
conclusion	successful

Table 4 Add non_academic Course

Figure 4 Add non_academic Course

D: Register non_academic Course

TEST NO.	2D:
Objective	To Register user given value to arraylist nonacedimicCourseIist
Action	Values from textfeild is registered to added arraylist
Expected result	Values would be registered
Actual result	Values were registered
conclusion	successful

Table 5 Register non_academic Course

Course Registration

More

Home

Acedimic

Non_Acedimic

Exit

Non Acedimic Course

CourseID: 123a Course Name: computing

Duration: 4 prerequisite: none

Course ID: 123a Course Leader: hari

Instructor Name: hari Exam Date: 2022/4/15

Start_date: 2020 Completion_date: 2023

Remove Register

Display Clear

Message

.All the feilds are registerd.

OK

Figure 5 Register non_academic Course

E: Remove non_academic Course

TEST NO.	2C:
Objective	To remove the values from courseID
Action	Get textfeild courseID and remove it from araylist.
Expected result	The courseID would be removed
Actual result	The courdeID was removed
conclusion	successful

Table 6 : Remove non_academic Course

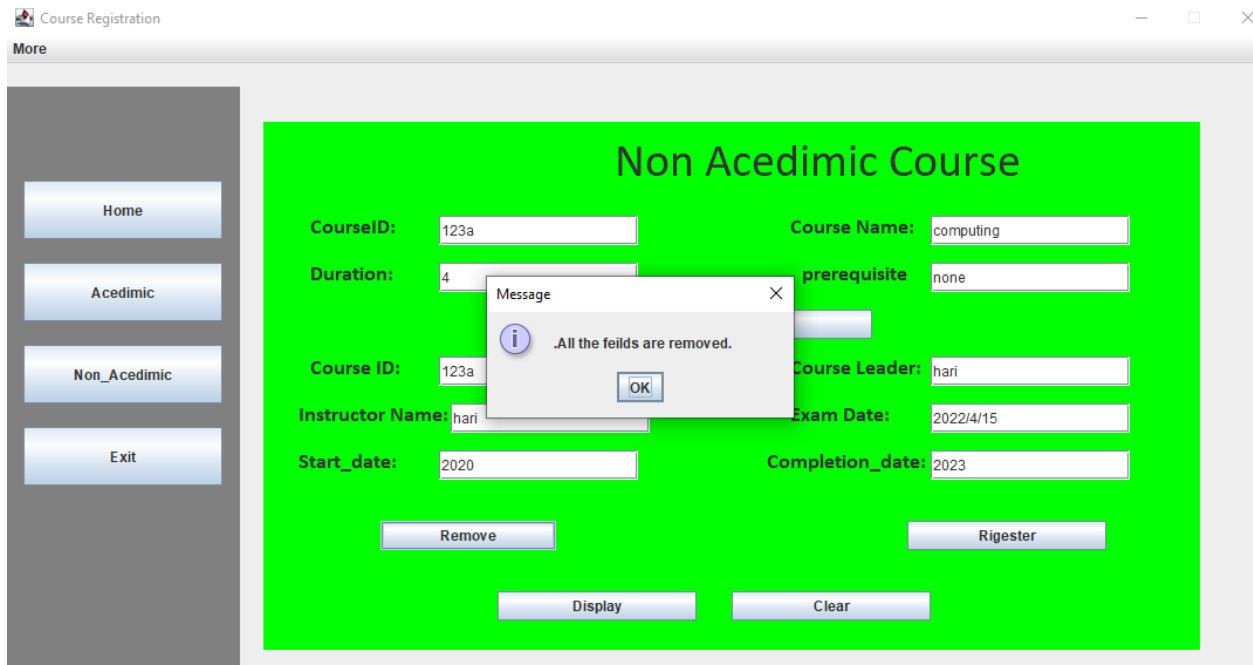


Figure 6 Remove non_academic Course

Test:3

A: Trying to add duplicate courseID

TEST NO.	3A:
Objective	To add same ID twice
Action	Same ID was tried to add twice
Expected result	Message box would appear
Actual result	Message box appears
conclusion	successful

Table 7 Trying to add duplicate courseID

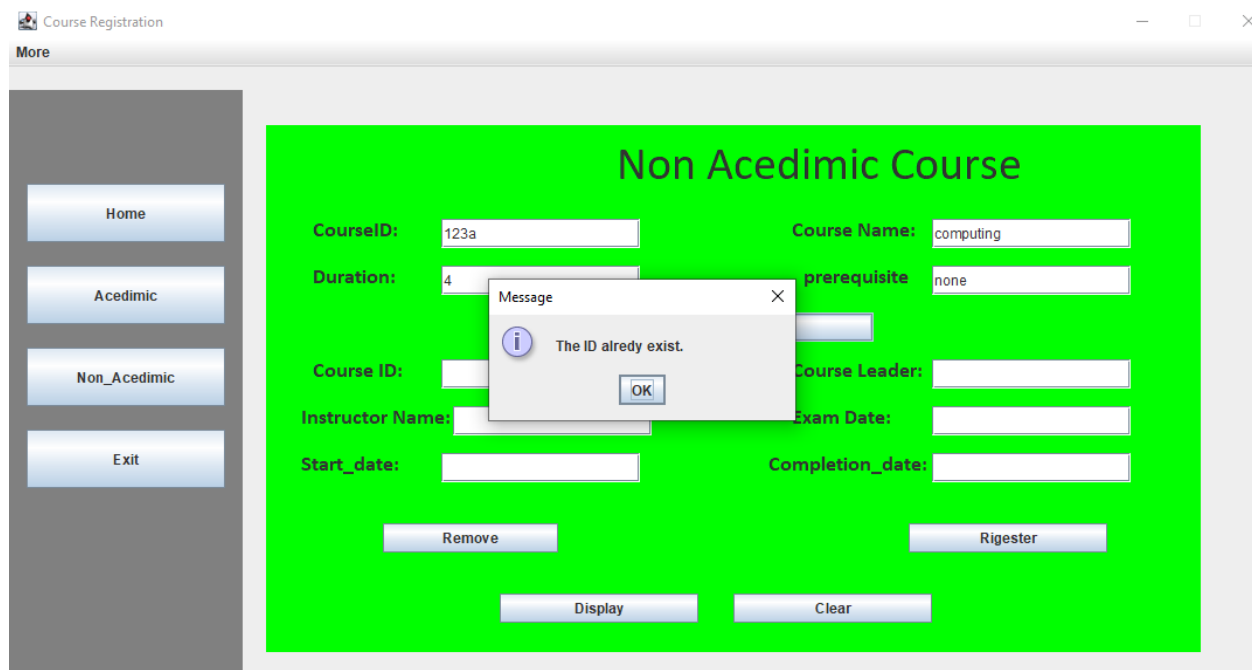


Figure 7 Trying to add duplicate courseID

B: Trying to register already registered course

TEST NO.	3B:
Objective	Regestring same course twice
Action	Giving same course ID twice
Expected result	Message box would appear
Actual result	Massage box appears
conclusion	successful

Table 8 Trying to register already registered course

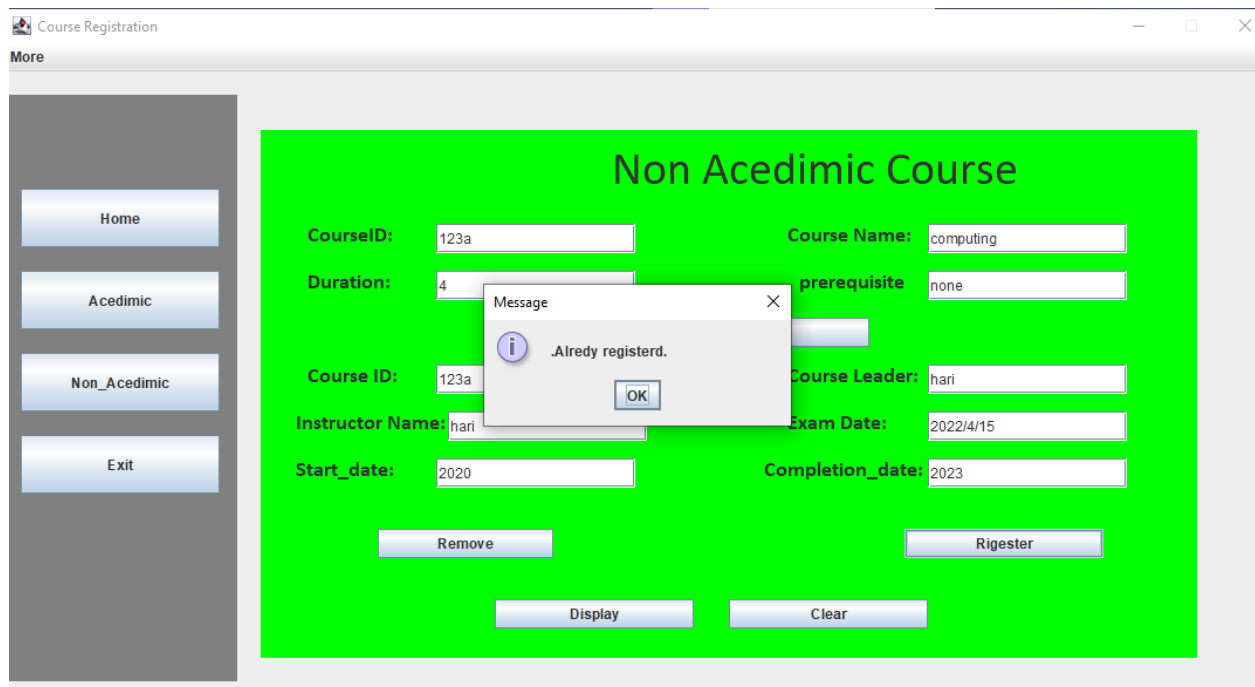


Figure 8 Trying to register already registered course

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

TEST NO.	3C:
Objective	Trying to removed course which is already removed
Action	Removing same courseID twice
Expected result	Message box would appear
Actual result	Massage box appears
conclusion	successful

Table 9 Trying to remove the non-academic course which is already removed

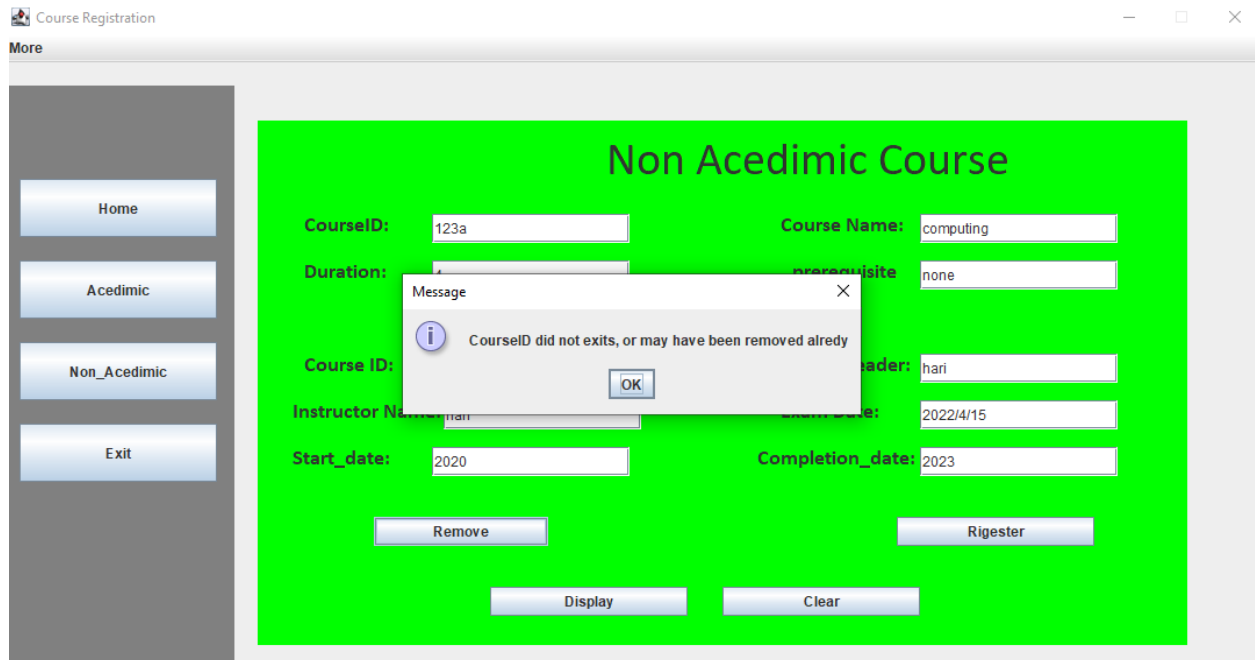


Figure 9 trying to remove removed courseID

Error and Correction:

There are basically three types of errors they are:-

- 1) Syntax error
- 2) Symmetric error
- 3) Logical error

Syntax error

The error which occurs while you write the program, while writing a program I have faced a syntax error. I missed a semicolon.

```

//...creating menu
mb = new JMenuBar();
More=new JMenu("More");
Help=new JMenuItem("Help");
Contact=new JMenuItem("Contact");
More.add(Help);
More.add(Contact);
mb.add(More);
frame.setJMenuBar(mb);

//....creating Jpanel Nagivation....//
Nav = new JPanel();

//...Home Button of Nav...//
Home = new JButton("Home");
Home.setBounds(15,80,170,50);
Nav.add(Home);

//...acedimic Button of Nav...//
acedimic = new JButton("Acedimic");
acedimic.setBounds(15,150,170,50);
Nav.add(acedimic);

//...Non_Acedimic Button of Nav...//

```

Figure 10 syntax error

I corrected by adding a semicolon on the particular place where I missed it its easy to find the syntax error as compiler will find it for you.

```

mb = new JMenuBar();
More=new JMenu("More");
Help=new JMenuItem("Help");
Contact=new JMenuItem("Contact");
More.add(Help);
More.add(Contact);
mb.add(More);
frame.setJMenuBar(mb);

//....creating Jpanel Nagivation....//
Nav = new JPanel();

//...Home Button of Nav...//
Home = new JButton("Home");
Home.setBounds(15,80,170,50);
Nav.add(Home);

//...acedimic Button of Nav...//
acedimic = new JButton("Acedimic");
acedimic.setBounds(15,150,170,50);
Nav.add(acedimic);

//...Non_Acedimic Button of Nav...//
Non_Acedimic = new JButton("Non_Acedimic");
Non_Acedimic.setBounds(15,220,170,50);
Nav.add(Non_Acedimic);

//...Exit Button of Nav...//
Exit = new JButton("Exit");
Exit.setBounds(15,290,170,50);
Nav.add(Exit);

```

Figure 11 correction of syntax error

Logical error:

The error in which compiles and run but give a unexpected result is known as logical error. I have also faced this kind of error. The error occur due to mistake on my logic while I was trying assign grater than operator but accidentally assign lesser then operator.

```
Remove.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if(txtCourseID2.getText().isEmpty())
        {
            JOptionPane.showMessageDialog(frame, "please enter the Course ID below ADD button");
            return;
        }
        //String ID = txtCourseID2.getText();
        for(int i = 0; i > nonacademicCourseList.size(); i++)
        {
            if((nonacademicCourseList.get(i).getcourse_ID()).equals(txtCourseID2.getText()))
            {
                Non_Academic_course NAC = (Non_Academic_course)(nonacademicCourseList.get(i));
                if(!NAC.getIsRemoved())
                {
                    NAC.Remove();
                    JOptionPane.showMessageDialog(frame, ".All the feilds are removed. ");
                    return;
                }
            }
        }
    }
});
```

I solved it by using propre logic ie greater then sing properly.

```
Remove.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if(txtCourseID2.getText().isEmpty())
        {
            JOptionPane.showMessageDialog(frame, "please enter the Course ID below ADD button");
            return;
        }
        //String ID = txtCourseID2.getText();
        for(int i = 0; i < nonacademicCourseList.size(); i++)
        {
            if((nonacademicCourseList.get(i).getcourse_ID()).equals(txtCourseID2.getText()))
            {
                Non_Academic_course NAC = (Non_Academic_course)(nonacademicCourseList.get(i));
                if(!NAC.getIsRemoved())
                {
                    NAC.Remove();
                    JOptionPane.showMessageDialog(frame, ".All the feilds are removed. ");
                    return;
                }
            }
        }
    }
});
```

Symmetric error:

semantic error when the syntax of your code is correct but the code usage isn't correct. Same way I also initialize wrong variable here I used String instead of int.

```
});
Display1.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if(nonacademicCourseList.isEmpty())
        {
            JOptionPane.showMessageDialog(frame, "...Nothing Added.....");
        }
        for(int i = 0; i < nonacademicCourseList.size(); i++)
        {
            String courseID = nonacademicCourseList.get(i).getcourse_ID();
            String courseName = nonacademicCourseList.get(i).getcourse_name();
            Non_Academic_course NAC = (Non_Academic_course)(nonacademicCourseList.get(i));
            String duration = NAC.getcourse_duration();
            String Prerequisite = NAC.getPrerequisite();
            String Instructor_name = NAC.getInstructor_name();
            String StartDate = NAC.getStart_date();
            String Completion_date = NAC.getcompletion_date();
            String ExamDate = NAC.getExam_date();
        }
    }
});
```

I solved this error by initialize correct variable.

```

Display1.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if(nonacedimicCourseList.isEmpty())
        {
            JOptionPane.showMessageDialog(frame, "...Nothing Added.....");
        }

        for(int i = 0; i < nonacedimicCourseList.size();i++)
        {
            String courseID = nonacedimicCourseList.get(i).getcourse_ID();
            String coursename = nonacedimicCourseList.get(i).getcourse_name();
            Non_Academic_course NAC = (Non_Academic_course)(nonacedimicCourseList.get(i));
            int duration = NAC.getcourse_duration();
            String Prerequisite = NAC.getPrerequisite();
            String Instructor_name = NAC.getInstructor_name();
            String StartDate = NAC.getStart_date();
            String Completion_date = NAC.getcompletion_date();
            String ExamDate = NAC.getExam_date();

            String emptyStr = "";

```

Conclusion:

The coursework was based on real life application of a java program. All the coursework was completed on time, by doing a lot of hard work, making a class in a java program, by making a lot of method so it can give an expected result. It was a great experience doing the coursework of java in gui, getting new things to learn and lot more other things on gui based program. By researching a lot about java as well as by practicing a lot helped me a lot to complete this coursework. I even faced a lot of errors while writing a code as well as I try harder to overcome errors. Various test was also performed according to the coursework. As well as all the method was described while writing a report. Overall, the coursework helped us a lot about understanding java.

Appendix1:

ING_Collage

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
public class ING_College
{
    JFrame frame;JMenuBar mb; JPanel Acedimic;JPanel NonAcedimic;JPanel Nav;Font
    FTA; Font lbl;JLabel welcome;
    ArrayList <Courses> acedemicCourseList;ArrayList <Courses>
    nonacedimicCourseList; Courses NAC;

    //content of JPanel Nagivation
    JButton acedimic; JButton Non_Acedimic; JButton Exit; JButton Home;
    //content of menu
    JMenu More; JMenuItem Help; JMenuItem Contact;
    //content of JPanel Acedimic
    JLabel AcedimicCourse;JLabel lblCourseID; JTextField txtCourseID;JLabel
    lblCourseName;JTextField txtCourseName;JLabel lblDuration;

```

```

    JTextField txtDuration;JLabel lblLecturerName;JTextField txtLecturerName;JLabel
    lblCourseLeader;JTextField txtCourseLeader;
    JLabel lblLevel;JTextField txtLevel;JLabel lblCredit;JTextField txtCedit;JLabel
    lblStarting_date;JTextField txtStarting_date;
    JLabel lblCompletion_date; JTextField txtCompletion_date;JLabel
    lblNumberof_assessments; JTextField txtNumberof_assessments;
    JLabel lblCourseID3; JTextField txtCourseID3;
    //content of JPanel nonacedimic
    JLabel NonAcedimicCourse;JLabel lblCourseID1; JTextField txtCourseID1;JLabel
    lblCourseName1;JTextField txtCourseName1;JLabel lblDuration1;JTextField
    txtDuration1;
    JLabel lblInstructorName;JTextField txtInstructorName;JLabel
    lblCourseLeader1;JTextField txtCourseLeader1;JLabel lblExamDate;JTextField
    txtExamDate;
    JLabel lblprerequisite ; JTextField txtprerequisite ;JLabel lblStart_date;JTextField
    txtStart_date;JLabel lblCompletion_date1; JTextField txtCompletion_date1;
    JLabel lblCourseID2;JTextField txtCourseID2;

    //Buttons used in Acedimic pannel
    JButton ADD;JButton Register;JButton Display;JButton Clear;
    //Buttons used in Non Acedimic pannel
    JButton ADD1;JButton Register1;JButton Display1;JButton Clear1;JButton Remove;

```

```

public ING_College()
{

```

```

    acedemicCourseList = new ArrayList <Courses>();
    nonacedimicCourseList = new ArrayList <Courses>();

```

```

    //....creating frame....//
    frame = new JFrame("Course Registration");
    FTA = new Font("Calibri",Font.PLAIN,40);
    lbl = new Font("Calibri",Font.BOLD,18);

```

```

    welcome = new JLabel("....Welcome To Course Regision System....");
    welcome.setForeground(Color.white);
    welcome.setBackground(Color.black);
    welcome.setBounds(250,200,750,60);

```

```
welcome.setOpaque(true);  
welcome.setFont(FTA);  
frame.add(welcome);
```

```
//...creating meanu  
mb = new JMenuBar();  
More=new JMenu("More");  
Help=new JMenuItem("Help");  
Contact=new JMenuItem("Contact");  
More.add(Help);  
More.add(Contact);  
mb.add(More);  
frame.setJMenuBar(mb);
```

```
//....creating Jpanel Nagivation....//  
Nav = new JPanel();
```

```
//...Home Button of Nav...//  
Home = new JButton("Home");  
Home.setBounds(15,80,170,50);  
Nav.add(Home);
```

```
//...acedimic Button of Nav...//  
acedimic = new JButton("Acedimic");  
acedimic.setBounds(15,150,170,50);  
Nav.add(acedimic);
```

```
//...Non_Acedimic Button of Nav...//  
Non_Acedimic = new JButton("Non_Acedimic");  
Non_Acedimic.setBounds(15,220,170,50);  
Nav.add(Non_Acedimic);
```

```
//...Exit Button of Nav...//  
Exit = new JButton("Exit");  
Exit.setBounds(15,290,170,50);  
Nav.add(Exit);
```

```
Nav.setBounds(0,20,200,500);  
Nav.setBackground(Color.gray);  
Nav.setLayout(null);  
frame.add(Nav);  
//.....///
```

```
//... creating panel Acedimic
```



```

Acedimic = new JPanel();

//JLabel Acedimic course//
AcedimicCourse = new JLabel("Acedimic Course");
AcedimicCourse.setBounds(300,20,400,40);
AcedimicCourse.setFont(FTA);
Acedimic.add(AcedimicCourse);
//.....//

//label and textfeild of course ID.
lblCourseID = new JLabel("CourseID:");
lblCourseID.setBounds(40,80,120,25);
lblCourseID.setFont(lbl);
Acedimic.add(lblCourseID);

txtCourseID = new JTextField();
txtCourseID.setBounds(150,80,170,25);
Acedimic.add(txtCourseID);
//.....//
//label and textfeild of course ID.
lblCourseName = new JLabel("Course Name:");
lblCourseName.setBounds(450,80,120,25);
lblCourseName.setFont(lbl);
Acedimic.add(lblCourseName);

txtCourseName = new JTextField();
txtCourseName.setBounds(570,80,170,25);
Acedimic.add(txtCourseName);
//.....//
//FOR DURATION
lblDuration = new JLabel("Duration:");
lblDuration.setBounds(40,120,120,25);
lblDuration.setFont(lbl);
Acedimic.add(lblDuration);

txtDuration = new JTextField();
txtDuration.setBounds(150,120,170,25);
Acedimic.add(txtDuration);

//.....//
JLabel lblLevel = new JLabel("Level:");
lblLevel.setBounds(40,150,100,25);
lblLevel.setFont(lbl);
Acedimic.add(lblLevel);

```

```

JTextField txtLevel = new JTextField();
txtLevel.setBounds(150,150,170,25);
Acedimic.add(txtLevel);

//.....//

JLabel lblCredit = new JLabel("Credit:");
lblCredit.setBounds(460,120,100,25);
lblCredit.setFont(lbl);
Acedimic.add(lblCredit);

JTextField txtCerdit = new JTextField();
txtCerdit.setBounds(570,120,170,25);
Acedimic.add(txtCerdit);
//.....//
//.....//
JLabel lblNumberof_assessments= new JLabel("Numberof_assessments:");
lblNumberof_assessments.setBounds(30,180,190,25);
lblNumberof_assessments.setFont(lbl);
Acedimic.add(lblNumberof_assessments);

JTextField txtNumberof_assessments = new JTextField();
txtNumberof_assessments.setBounds(230,180,170,25);
Acedimic.add(txtNumberof_assessments);
//.....//

//.ADD BUTTON.//

ADD = new JButton("Add");
ADD.setBounds(450,190,170,25);
Acedimic.add(ADD);

//.....//
lblCourseID3 = new JLabel("Course ID:");
lblCourseID3.setBounds(30,250,120,25);
lblCourseID3.setFont(lbl);
Acedimic.add(lblCourseID3);

txtCourseID3 = new JTextField();
txtCourseID3.setBounds(150,250,170,25);
Acedimic.add(txtCourseID3);
//.....//

//..FOR LECTURE NAME.....//

```

```

lblLecturerName = new JLabel("Lecturer Name:");
lblLecturerName.setBounds(30,280,120,25);
lblLecturerName.setFont(lbl);
Acedimic.add(lblLecturerName);

txtLecturerName = new JTextField();
txtLecturerName.setBounds(150,280,170,25);
Acedimic.add(txtLecturerName);

//..for course leader..//

lblCourseLeader = new JLabel("Course Leader:");
lblCourseLeader.setBounds(450,250,120,25);
lblCourseLeader.setFont(lbl);
Acedimic.add(lblCourseLeader);

txtCourseLeader = new JTextField();
txtCourseLeader.setBounds(570,250,170,25);
Acedimic.add(txtCourseLeader);

//.....//

//.....//
lblStarting_date = new JLabel("Starting_date:");
lblStarting_date.setBounds(20,310,120,25);
lblStarting_date.setFont(lbl);
Acedimic.add(lblStarting_date);

txtStarting_date = new JTextField();
txtStarting_date.setBounds(150,310,170,25);
Acedimic.add(txtStarting_date);

//.....//
lblCompletion_date = new JLabel("Completion_date:");
lblCompletion_date.setBounds(430,280,140,25);
lblCompletion_date.setFont(lbl);
Acedimic.add(lblCompletion_date);

txtCompletion_date = new JTextField();
txtCompletion_date.setBounds(570,280,170,25);
Acedimic.add(txtCompletion_date);

Register = new JButton("Rigester");

```

```
Register.setBounds(550,340,170,25);
Acedimic.add(Register);
```

```
Display = new JButton("Display");
Display.setBounds(200,400,170,25);
Acedimic.add(Display);
```

```
Clear = new JButton("Clear");
Clear.setBounds(400,400,170,25);
Acedimic.add(Clear);
```

```
Acedimic.setBounds(220,50,800,450);
Acedimic.setBackground(Color.green);
Acedimic.setLayout(null);
```

```
//... creating panel NonAcedimic...//
NonAcedimic = new JPanel();
```

```
//Jlabel Acedimic course//
NonAcedimicCourse = new JLabel("Non Acedimic Course");
NonAcedimicCourse.setBounds(300,20,400,40);
NonAcedimicCourse.setFont(FTA);
NonAcedimic.add(NonAcedimicCourse);
//.....//
```

```
//label and textfeild of course ID.
lblCourseID1 = new JLabel("CourseID:");
lblCourseID1.setBounds(40,80,120,25);
lblCourseID1.setFont(lbl);
NonAcedimic.add(lblCourseID1);
```

```
txtCourseID1 = new JTextField();
txtCourseID1.setBounds(150,80,170,25);
NonAcedimic.add(txtCourseID1);
//.....//
```

```
//label and textfeild of course ID.
lblCourseName1 = new JLabel("Course Name:");
lblCourseName1.setBounds(450,80,120,25);
lblCourseName1.setFont(lbl);
NonAcedimic.add(lblCourseName1);
```

```
txtCourseName1 = new JTextField();
txtCourseName1.setBounds(570,80,170,25);
```

```

NonAcedimic.add(txtCourseName1);
//.....//
//FOR DURATION
lblDuration1 = new JLabel("Duration:");
lblDuration1.setBounds(40,120,120,25);
lblDuration1.setFont(lbl);
NonAcedimic.add(lblDuration1);

txtDuration1 = new JTextField();
txtDuration1.setBounds(150,120,170,25);
NonAcedimic.add(txtDuration1);

//.....//

lblprerequisite = new JLabel("prerequisite ");
lblprerequisite.setBounds(460,120,100,25);
lblprerequisite.setFont(lbl);
NonAcedimic.add(lblprerequisite);

txtprerequisite = new JTextField();
txtprerequisite.setBounds(570,120,170,25);
NonAcedimic.add(txtprerequisite);

//.....//

//.ADD BUTTON.//

ADD1 = new JButton("Add");
ADD1.setBounds(350,160,170,25);
NonAcedimic.add(ADD1);
//.....//
//For Course Leader//
lblCourseID2= new JLabel("Course ID:");
lblCourseID2.setBounds(40,200,140,25);
lblCourseID2.setFont(lbl);
NonAcedimic.add(lblCourseID2);

txtCourseID2 = new JTextField();
txtCourseID2.setBounds(150,200,170,25);
NonAcedimic.add(txtCourseID2);
//.....//
//For INSTRUCTOR NAME//
lblInstructorName= new JLabel("Instructor Name:");
lblInstructorName.setBounds(30,240,140,25);
lblInstructorName.setFont(lbl);

```

```

NonAcedimic.add(lblInstructorName);

txtInstructorName = new JTextField();
txtInstructorName.setBounds(160,240,170,25);
NonAcedimic.add(txtInstructorName);
//.....//
//For Course Leader//
lblCourseLeader1= new JLabel("Course Leader:");
lblCourseLeader1.setBounds(450,200,140,25);
lblCourseLeader1.setFont(lbl);
NonAcedimic.add(lblCourseLeader1);

txtCourseLeader1 = new JTextField();
txtCourseLeader1.setBounds(570,200,170,25);
NonAcedimic.add(txtCourseLeader1);
//.....//

lblExamDate = new JLabel("Exam Date:");
lblExamDate.setBounds(450,240,100,25);
lblExamDate.setFont(lbl);
NonAcedimic.add(lblExamDate);

txtExamDate = new JTextField();
txtExamDate.setBounds(570,240,170,25);
NonAcedimic.add(txtExamDate);

lblStart_date = new JLabel("Start_date:");
lblStart_date.setBounds(30,280,140,25);
lblStart_date.setFont(lbl);
NonAcedimic.add(lblStart_date);

txtStart_date = new JTextField();
txtStart_date.setBounds(150,280,170,25);
NonAcedimic.add(txtStart_date);

//.....//
lblCompletion_date1 = new JLabel("Completion_date:");
lblCompletion_date1.setBounds(430,280,140,25);
lblCompletion_date1.setFont(lbl);
NonAcedimic.add(lblCompletion_date1);

txtCompletion_date1 = new JTextField();
txtCompletion_date1.setBounds(570,280,170,25);
NonAcedimic.add(txtCompletion_date1);
//.....//

```

```
//.....//
Register1 = new JButton("Rigester");
Register1.setBounds(550,340,170,25);
NonAcedimic.add(Register1);

Display1 = new JButton("Display");
Display1.setBounds(200,400,170,25);
NonAcedimic.add(Display1);

Clear1 = new JButton("Clear");
Clear1.setBounds(400,400,170,25);
NonAcedimic.add(Clear1);

Remove = new JButton("Remove");
Remove.setBounds(100,340,150,25);
NonAcedimic.add(Remove);

NonAcedimic.setBounds(220,50,800,450);
NonAcedimic.setBackground(Color.green);
NonAcedimic.setLayout(null);
```

```
Help.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        JOptionPane.showMessageDialog(frame, "You can register your Acedimic
and Nonacedimic Course... For futher query plz contact us.");
    }
});
```

```
Contact.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        JOptionPane.showMessageDialog(frame, "E-mail = lol12@gmail.com \\\\\\
phone=989878985820");
    }
});
```

```
acedimic.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        frame.remove(welcome);
        frame.remove(NonAcedimic);
        frame.add(Acedimic);
    }
});
```

```
Non_Acedimic.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        frame.remove(welcome);
        frame.remove(Acedimic);
        frame.add(NonAcedimic);
    }
});
```

```
Home.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        frame.remove(Acedimic);
        frame.remove(NonAcedimic);
        frame.add(welcome);
    }
});
```

```
Exit.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        frame.dispose();
    }
});
```

///Buttons of Acedimic panel.

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



```

        {
            if(txtCourseID.getText().isEmpty() || txtCourseName.getText().isEmpty() ||
               txtDuration.getText().isEmpty() ||
txtNumberof_assessments.getText().isEmpty() || txtCerdit.getText().isEmpty() ||
txtLevel.getText().isEmpty())
            {
                JOptionPane.showMessageDialog(frame, "All the feilds are required. ");
                return;
            }

            int NumberOfAssesment;
            int duration;
            try{
                NumberOfAssesment =
Integer.parseInt(txtNumberof_assessments.getText());
                duration = Integer.parseInt(txtDuration.getText());
            }

            catch(Exception E)
            {
                JOptionPane.showMessageDialog(frame, ".Invalid Input.");
                return;
            }
            String ID = txtCourseID.getText();
            for (Courses AC : acedemicCourseList)
            {
                if (AC.getcourse_ID().equals(ID))
                {
                    JOptionPane.showMessageDialog(frame,"The ID alredy exist.");
                    return;
                }
            }

            String name = txtCourseName.getText();
            String level = txtLevel.getText();
            String cerdit = txtCerdit.getText();

```

```

        Courses AC = new
Academic_Course(name,ID,duration,level,cerdit,NumberOfAssesment);
        acedemicCourseList.add(AC);

        if(acedemicCourseList.add(AC)== true)

```

```

        {
            JOptionPane.showMessageDialog(frame, "Course is added
Sucessfully.");
        }
    }
});

Register.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if(txtCourseID3.getText().isEmpty() || txtCourseLeader.getText().isEmpty()
|| txtStarting_date.getText().isEmpty() ||
        txtLecturerName.getText().isEmpty() ||
txtCompletion_date.getText().isEmpty())
        {
            JOptionPane.showMessageDialog(frame, "All the feilds are required. ");
            return;
        }

        String ID = txtCourseID3.getText();
        for(int i = 0; i<acedemicCourseList.size();i++)
        {
            if((acedemicCourseList.get(i).getcourse_ID()).equals(ID))
            {
                Academic_Course AC =
(Academic_Course)(acedemicCourseList.get(i));
                if(!AC.getisRegistered())
                {

AC.register(txtLecturerName.getText(),txtStarting_date.getText(),txtCompletion_date.ge
tText(),txtCourseLeader.getText());
                    JOptionPane.showMessageDialog(frame, ".All the feilds are
registerd. ");
                }
            }
            else
            {
                JOptionPane.showMessageDialog(frame, ".Alredy registerd. ");
                return;
            }
        }
    }
}
else
{

```

```

JOptionPane.showMessageDialog(frame, "...Invalid
Course_ID.....");
    }

```

```

    }

```

```

    }
});

```

```

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

```

```

        for(int i = 0; i < acedemicCourseList.size();i++)
        {
            String courseID = acedemicCourseList.get(i).getcourse_ID();
            String coursename = acedemicCourseList.get(i).getcourse_name();
            Academic_Course AC =
(Academic_Course)(acedemicCourseList.get(i));
            int duration = AC.getcourse_duration();
            int Numberof_assessments = AC.getNumberof_assessments();
            String lecturer_name = AC.getlecturer_name();
            String StartingDate = AC.getStarting_date();
            String Completion_date = AC.getCompletion_date();
            String credits = AC.getcredits();
            String level = AC.getlevel();

            String emptyStr = "";

            //if (AC.getisRegistered())
            {
                String display = emptyStr.concat("CourseID: "+courseID + "\n" +
"Course Name: "+coursename
                + "\n" + "Duration: "+duration+"\n" + "Numberof_assessments:
"+Numberof_assessments
                + "\n" + "Lecturer Name: " +lecturer_name + "\n" + "Starting Date: "
+StartingDate +
                "\n" + "Completion Date: " + Completion_date + "\n" + "credits: " +
credits + "\n" + "level: "+level);

```

```

        JOptionPane.showMessageDialog(frame,display);
    }
    //else
    {
        JOptionPane.showMessageDialog(frame,"Please register course for
more details on course");
    }

}

/*
if(acedemicCourseList.size()==0)
{
    JOptionPane.showMessageDialog(frame, "The list is empty");
}
for(int i =0; i<acedemicCourseList.size();i++)
{
    Academic_Course AC =
(Academic_Course)(acedemicCourseList.get(i));

    AC.display();
}
*/

}
});

```

```

Clear.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        txtCourseID.setText("");
        txtCourseName.setText("");
        txtDuration.setText("");
        txtNumberof_assessments.setText("");
        txtCerdit.setText("");
        txtLevel.setText("");
        txtCourseID3.setText("");
        txtCourseLeader.setText("");
    }
}

```

```

        txtStarting_date.setText("");
        txtCompletion_date.setText("");
        txtLecturerName.setText("");
    }
});

///Buttons of Acedimic panel.
ADD1.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if(txtCourseID1.getText().isEmpty() || txtCourseName1.getText().isEmpty()
||
        txtDuration1.getText().isEmpty() || txtprerequisite.getText().isEmpty())
        {
            JOptionPane.showMessageDialog(frame, "All the feilds are required. ");
            return;
        }
        int Duration = 0;
        try{
            Duration = Integer.parseInt(txtDuration1.getText());
        }
        catch(Exception a)
        {
            JOptionPane.showMessageDialog(frame, ".Invalid Input.");
            return;
        }
        String id = txtCourseID1.getText();
        for (Courses NAC : nonacedimicCourseList)
        {
            if (NAC.getcourse_ID().equals(id))
            {
                JOptionPane.showMessageDialog(frame,"The ID alredy exist.");
                return;
            }
        }
        String Name = txtCourseName1.getText();
        String Prerequisite = txtprerequisite.getText();

        Courses NAC = new
Non_Academic_course(Name,id,Duration,Prerequisite);
        nonacedimicCourseList.add(NAC);
        if(nonacedimicCourseList.add(NAC)== true)
        {

```

```

        JOptionPane.showMessageDialog(frame, ".course added.");
    }

}

});

Register1.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if(txtCourseID2.getText().isEmpty() || txtCourseLeader1.getText().isEmpty()
||
        txtInstructorName.getText().isEmpty() || txtExamDate.getText().isEmpty() ||
txtCompletion_date1.getText().isEmpty() || txtStart_date.getText().isEmpty())
        {
            JOptionPane.showMessageDialog(frame, "All the feilds are required. ");
            return;
        }
        String ID = txtCourseID2.getText();
        for(int i = 0; i<nonacedimicCourseList.size();i++)
        {
            if((nonacedimicCourseList.get(i).getcourse_ID()).equals(ID))
            {
                Non_Academic_course NAC =
(NON_Academic_course)(nonacedimicCourseList.get(i));
                if(!NAC.getisRegistered())
                {

NAC.register(txtCourseLeader1.getText(),txtInstructorName.getText(),txtExamDate.get
Text(),txtStart_date.getText(),txtCompletion_date1.getText());
                    JOptionPane.showMessageDialog(frame, ".All the feilds are
registerd. ");
                }
                else
                {
                    JOptionPane.showMessageDialog(frame, ".Alredy registerd. ");
                    return;
                }
            }
        }
        else
        {
            JOptionPane.showMessageDialog(frame, "...Invalid
Course_ID.....");
        }
    }
}

```

```

    }
}
});

Display1.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if(nonacedimicCourseList.isEmpty())
        {
            JOptionPane.showMessageDialog(frame, "...Nothing Added.....");
        }

        for(int i = 0; i < nonacedimicCourseList.size();i++)
        {
            String courseID = nonacedimicCourseList.get(i).getcourse_ID();
            String coursename = nonacedimicCourseList.get(i).getcourse_name();
            Non_Academic_course NAC =
            (Non_Academic_course)(nonacedimicCourseList.get(i));
            int duration = NAC.getcourse_duration();
            String Prerequisite = NAC.getPrerequisite();
            String Instructor_name = NAC.getInstructor_name();
            String StartDate = NAC.getStart_date();
            String Completion_date = NAC.getcompletion_date();
            String ExamDate = NAC.getExam_date();

            String emptyStr = "";

            //if (NAC.getisRegistered())
            {
                String display = emptyStr.concat("CourseID: "+courseID + "\n" +
"Course Name: "+coursename
                + "\n" + "Duration: "+duration+"\n" + "Prerequisite: " +Prerequisite
                +"\n" + "Instructor Name: " +Instructor_name +"\n" + "Start Date: "
+StartDate +
                "\n" +"Completion Date: " + Completion_date + "\n" + "Exam Date: " +
ExamDate);

                JOptionPane.showMessageDialog(frame,display);
            }
            //else

```

```

        {
            JOptionPane.showMessageDialog(frame, "Please register course for
more details on course");
        }
    }
});

```

```

Remove.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if(txtCourseID2.getText().isEmpty())
        {
            JOptionPane.showMessageDialog(frame, "please enter the Course ID
below ADD button");
            return;
        }
        //String ID = txtCourseID2.getText();
        for(int i = 0; i<nonacedimicCourseList.size();i++)
        {
            if((nonacedimicCourseList.get(i).getcourse_ID()).equals(txtCourseID2.getText()))
            {
                Non_Academic_course NAC =
                (Non_Academic_course)(nonacedimicCourseList.get(i));
                if(!NAC.getisRemoved())
                {
                    NAC.Remove();
                    JOptionPane.showMessageDialog(frame, ".All the feilds are
removed. ");
                    return;
                }
            }
        }
    }
}

```



```
JOptionPane.showMessageDialog(frame,"CourseID did not exists, or may have been removed already");
```

```
    }  
    });  
    Clear1.addActionListener(new ActionListener()  
    {  
        public void actionPerformed(ActionEvent e)  
        {  
            txtCourseID1.setText("");  
            txtCourseName1.setText("");  
            txtDuration1.setText("");  
            txtInstructorName.setText("");  
            txtCourseLeader1.setText("");  
            txtExamDate.setText("");  
            txtprerequisite.setText("");  
            txtStart_date.setText("");  
            txtCompletion_date1.setText("");  
            txtCourseID2.setText("");  
        }  
    });
```

```
    frame.setLayout(null);  
    frame.setSize(1100,600);  
    frame.repaint();  
    frame.setResizable(false);  
    frame.setVisible(true);
```

```
}
```

```
public static void main(String [] args)  
{  
    new ING_College();  
  
}  
}
```

Appendix2:

Course:

```
public class Courses    //developing class Courses  
{  
    private String course_name;    //attribute course name  
    private String course_ID;    //attribute course ID  
    private String course_leader;    //attribute course leader
```

```

private int course_duration; //attribute course duration

public Courses (String course_name, String course_ID, int course_duration) //
developing constructor
{
    this.course_name = course_name;
    this.course_ID = course_ID;
    this.course_duration = course_duration;
    this.course_leader = ""; // constructor paramaters
}

public String getcourse_ID()
{
    return this.course_ID;
} //this method gets course id and return to the constructor.

public String getcourse_name()
{
    return this.course_name;
} //this method gets course name and return to the constructor.

public int getcourse_duration()
{
    return this.course_duration;
} //this method gets course duration and return to the constructor.

public void setcourse_leader(String course_leader)
{
    this.course_leader = course_leader;
} //this methord will set the courseLeader to a new name by accepting the new name
of the course leader as a parameter.

public void display()
{
    System.out.println("Course ID :" + course_ID);
    System.out.println("Course NAME :" + course_name);
    System.out.println("course DURATION :" + course_duration); //suitably annotated
values will be displayed

    if (course_leader != "") //if course leader is not empty it will display name of course
leader.
    {
        System.out.print("course LEADER" + course_leader);
    }
}

```

```
}
```

Academic_Course:

//Academic_course is sub class of Courses

public class Academic_Course extends Courses //developing sub class Courses

```
{
```

```
    private String lecturer_name;
```

```
    private String level;
```

```
    private String Starting_date;
```

```
    private String credits;
```

```
    private String Completion_date;
```

```
    private int Numberof_assessments;
```

```
    private boolean isRegistered;    // attribute of class Academic_Course
```

```
    public Academic_Course(String course_name, String course_ID, int course_duration,  
String level,String credits,int Numberof_assessments )
```

```
    {
```

```
        super(course_name,course_ID,course_duration); //calling super class Course
```

```
        lecturer_name = "";
```

```
        this.level = level;
```

```
        this.Numberof_assessments = Numberof_assessments;
```

```
        this.credits = credits;
```

```
        Starting_date = "";
```

```
        Completion_date = "";
```

```
        isRegistered = false;    // constructor parameters
```

```
    }
```

```
    public String getlecturer_name()
```

```
    {
```

```
        return lecturer_name;
```

```
    } //this method gets name of lecturer and return to the constructor.
```

```
    public String getlevel()
```

```
    {
```

```
        return level;
```

```
    } //this method gets level and return to the constructor.
```

```
    public String getStarting_date()
```

```
    {
```

```
        return Starting_date;
```

```
    } //this method gets starting date and return to the constructor.
```

```
    public String getcredits()
```

```
    {
```

```
        return credits;
```

}//this method gets credits and return to the constructor.

```
public int getNumberOf_assessments()
{
    return Numberof_assessments;
} //this method gets total number of assessments and return to the constructor.
```

```
public boolean getisRegistered()
{
    return isRegistered ;
} //this method gets value in boolean either true and false and return to the
constructor.
```

```
public String getCompletion_date()
{
    return Completion_date;
} //this method gets date of Completion and return to the constructor.
```

```
public void setlecturer_name(String lecturer_name)
{
    this.lecturer_name = lecturer_name;
} //The method accepts a new lecturer name as a parameter and assigns the new
value to the lecturer name.
```

```
public void setNumberOf_assessments(int Numberof_assessments)
{
    this.Numberof_assessments = Numberof_assessments;
} //The method accepts a new number of assessments as a parameter and assigns
the new value to the number of assessments attribute.
```

```
public void register(String lecturer_name,String Starting_date, String
Completion_date, String course_leader)
{
    if (isRegistered == true)
    {
        System.out.println("Academic course is already registered");
        System.out.println("Name of lecturer is :" + lecturer_name);
        System.out.println("Your course starts from :" + Starting_date);
        System.out.println("you will complete your course on :" + Completion_date);
    }
    else
    {
        super.setcourse_leader(course_leader);
        this.lecturer_name = lecturer_name;
        this.Starting_date = Starting_date;
    }
}
```

```

        this.Completion_date = Completion_date;
        isRegistered = true;
    }

}

public void display()
{
    super.display();
    if(isRegistered == true)
    {
        System.out.println("Name of lecturer : " + lecturer_name);
        System.out.println(" your level is : " + level);
        System.out.println("your cradits is:" + credits);
        System.out.println("Starts on : " + Starting_date);
        System.out.println("ends on : " + Completion_date);
        System.out.println("number of assessments : " + Numberof_assessments);
    }
}
}

```

Non_Aceademic_Course:

public class Non_Academic_course extends Courses

```

{
    // class Attributes
    String Instructor_name;
    String Start_date;
    String Completion_date;
    String Exam_date;
    String Prerequisite;
    boolean isRegistered;
    boolean isRemoved;
    public Non_Academic_course(String course_name, String course_ID, int
course_duration,String Prerequisite)
    {
        //constructor parameters
        super(course_name,course_ID,course_duration);
        Instructor_name = Instructor_name;
        Prerequisite = Prerequisite;
        this.Start_date = "";
        this.Completion_date = "";
        this.Exam_date = "";
        this.isRegistered = false;
        this.isRemoved = false;
    }
}

```

```

}
//accessor methods for each attributes
public String getInstructor_name()
{
    return Instructor_name;
}
public String getStart_date()
{
    return Start_date;
}
public String getExam_date()
{
    return Exam_date;
}
public String getcompletion_date()
{
    return Completion_date;
}
public String getPrerequisite()
{
    return Prerequisite;
}

public boolean getisRegistered()
{
    return isRegistered;
}
public boolean getisRemoved()
{
    return isRemoved;
}

public void setInstructor_name(String Instructor_name)
{
    if(isRegistered == false)
    {
        this.Instructor_name = Instructor_name;
    }
    else{
        System.out.println("Instructor Name can't be changed");
    }
}

```

```

    public void register(String course_leader, String Instructor_name,String
Exam_date,String Start_date,String Completion_date)
    {
        if (isRegistered == false)
        {
            setInstructor_name(Instructor_name);
            isRegistered = true;
            this.Start_date = Start_date;
            this.Exam_date = Exam_date;
            this.Completion_date = Completion_date;
        }
        else{
            System.out.println("the course which you are trying to register is already
REGISTERD.");
        }
    }

    public void Remove()
    {
        if(isRegistered = true)
        {
            System.out.println("course is REMOVED.");
        }
        else{
            super.setcourse_leader("");
            Instructor_name = Instructor_name;
            Start_date = "";
            Exam_date = "";
            Completion_date = "";
            isRegistered = false;
            isRemoved = true;
        }
    }

    public void display()
    {
        System.out.println("COURSE ID : " + super.getcourse_ID());
        System.out.println("COURSE NAME : " + super.getcourse_name());
        System.out.println("COURSE DURATION : " + super.getcourse_duration());

        if(isRegistered == true)
        {
            System.out.println("Name of Instructor is : " + Instructor_name);
            System.out.println("Starting DATE : " + Start_date);
            System.out.println("Completion DATE : " + Completion_date);
            System.out.println("Exam Date : " + Exam_date);
        }
    }

```

}
}
}

References

Anon., n.d. *About blueJ*. [Online]

Available at: <https://bluej.org/about.html>

[Accessed 15 08 2021].

Computer hope, 2021/12/4. *What is microfoft word?*. [Online]

Available at: <https://www.computerhope.com/jargon/m/microsoft-word.htm>

[Accessed 15 08 2022].