COMP1752 – Object-Oriented Programming

[Willian Cieslak - 000990795]

University of Greenwich

3/31/2018

Table of Contents

1 Introduction ... 4

2 Class Diagram ... 5

2.1 The Class Diagram ... 5

2.2 A Brief Explanation of the Diagram ... 6

3 The Code for the Extra Classes Created ... 7

3.1 AllCourses.java ... 7

3.2 AllCoursework.java ... 11

3.3 Coursework.java ... 17

3.4 CourseworkScreen.java ... 22

3.5 Main.java ... 42

4 White Box Testing ... 62

4.1 Testing for class … ... 62

4.2 Testing for class … ... 62

4.3 Testing for class … ... 62

4.4 Testing for class … ... 62

5 Screen Shots of the Program Working ... 63

5.1 Screen 1 – Adding a Course ... 63

5.2 Screen 1 – A brief description ... 64

5.3 Screen 2 – Adding Notes ... 65

5.4 Screen 2 – A brief description ... 67

5.5 Screen 3 – Reading Saved Notes ... 68

5.6 Screen 3 – A brief description ... 69

5.7 Screen 4 – Editing Notes ... 70

5.8 Screen 4 – A brief description ... 71

5.9 Screen 5 – Deleting a Course ... 72

5.10 Screen 5 – A brief description ... 74

5.11 Screen 6 – Opening Coursework Screen ... 75

5.12 Screen 6 – A brief description ... 76

5.13 Screen 7 – Adding a Coursework ... 77

5.14 Screen 7 – A brief description ... 78

5.15 Screen 8 – Editing a Coursework ... 79

5.16 Screen 8 – A brief description 79
6 The Evaluation 80
6.1 What went well? 80
6.2 What went less well? 80
6.3 What was learned? 80
6.4 How would a similar task be completed differently? 81
6.5 How could the course be improved? 81
7 Self-Grading 82

1 INTRODUCTION

I have decided to have another approach to the coursework. Even though we had the view done I have decided to create a new one using a JFrame Form from Swing GUI. To my understanding, using JFrame forms is easier to manipulate and create new objects (JTextField, JTextArea, Buttons, etc) and to make everything work as expected a few changes were made to the original code and new classes were added. They are:

AllCourses: save, edit, create and display the courses that you want to add some notes.

AllCoursework: save, edit, create and display the courseworks that you might create.

Coursework: a class defining coursework as an object with int max, String name, String date, String dueDate, String requirement as attributes.

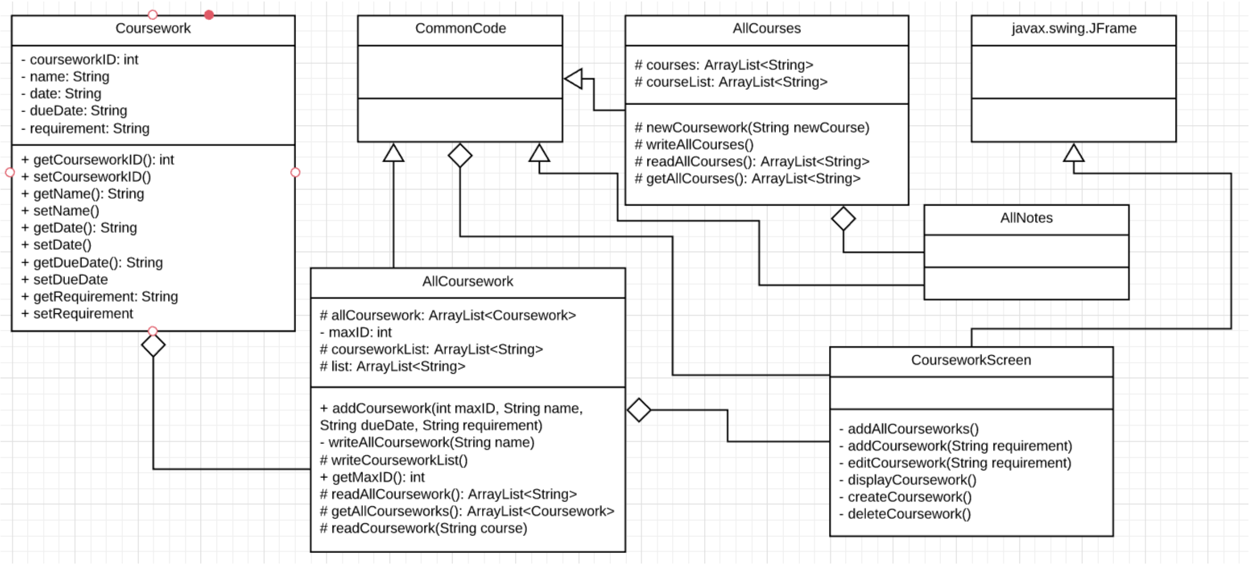
CourseworkScreen: JFrame form used to add and display all the courseworks.

Main: where the program begins. Main is a JFrame form used to create and display Notes. Using the same MVC structure, but is this case the view is now called initComponents();

Notes are listed in a text file called Course List, and saved within a file folder with the name the user choses e.g.(You create a Course called Maths and want to add some notes, the notes will be saved in a text file called Maths, inside a file folder called Maths). Courseworks are listed in a file called Coursework List and every coursework is saved in a file folder called Coursework.

2 CLASS DIAGRAM

2.1 THE CLASS DIAGRAM


Coursework
- courseworkID: int
- name: String
- date: String
- dueDate: String
- requirement: String
+ getCourseworkID(): int
+ setCourseworkID()
+ getName(): String
+ setName()
+ getDate(): String
+ setDate()
+ getDueDate(): String
+ setDueDate
+ getRequirement: String
+ setRequirement
CommonCode
AllCourses
# courses: ArrayList<String>
# courseList: ArrayList<String>
# newCoursework(String newCourse)
# writeAllCourses()
# readAllCourses(): ArrayList<String>
# getAllCourses(): ArrayList<String>
javax.swing.JFrame
AllNotes
AllCoursework
# allCoursework: ArrayList<Coursework>
- maxID: int
# courseworkList: ArrayList<String>
# list: ArrayList<String>
+ addCoursework(int maxID, String name, String dueDate, String requirement)
- writeAllCoursework(String name)
# writeCourseworkList()
+ getMaxID(): int
# readAllCoursework(): ArrayList<String>
# getAllCourseworks(): ArrayList<Coursework>
# readCoursework(String course)
CourseworkScreen
- addAllCourseworks()
- addCoursework(String requirement)
- editCoursework(String requirement)
- displayCoursework()
- createCoursework()
- deleteCoursework()

2.2 A Brief Explanation of the Diagram

The diagram above describes the Coursework relationship with the other classes. Coursework is an object an I have defined that a Coursework has some private attributes (ID, name, date, dueDate and requirements). AllCoursework has an instance of Coursework (an ArrayList of Coursework) which holds all the courseworks. AllCoursework is the class responsible for adding, writing and reading Courseworks, and extends CommonCode to be able to do that. Coursework has more than one constructor, in case the user does not want to set a due date for example. ID, name and requirements however are mandatory to create a note.

CourseworkScreen in the meantime is the class responsible for presenting the existent courseworks to the user and perhaps collecting all the data needed to create another Coursework it the user wishes to do so. If the user adds a new Coursework then CourseworkScreen will collect and transfer all the date to AllCoursework and AllCoursework will be the one actually saving and loading the files. CourseworkScreenhas gets the list back and displays the information to the user. CourseworkScreen also has an instance of CommonCode and extends all the methods and properties from javax.swing.JFrame class.

AllCourses is there to make sure we control all the Courses we want to take notes from. AllCourses controls, saves, edits, deletes and generates a list of Courses that can be selected in the Main class (first screen when the program is running). Main will receive this list of Courses and display into the combo box menu. Because Notes has an attribute field called course, AllNotes is now able to get the selected course from the combo box and display only the notes that have the matching course as a course attribute.

Because some the classes are being instantiated in another class most of the methods are protected (can only be accessed by the class or its subclasses).

3 THE CODE FOR THE EXTRA CLASSES CREATED

3.1 ALLCOURSES.JAVA

```
package coursework;

/**
 *
 * @author Willian Cieslak
 */
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.io.Writer;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class AllCourses extends CommonCode {

    protected ArrayList<String> courses = new ArrayList<>();
    protected ArrayList<String> courseList = new ArrayList<>();

    AllCourses() {
    }

    //Creates a new file and text file using the same name.
    protected void newCourseFile(String newCourse) {

        String file = appDir + fileSeparator + newCourse;
        String path = file + fileSeparator + newCourse + ".txt";

        //Verifies if the directory exists. If not then a new one is created
        File directory = new File(file);
        if (!directory.exists()) {
            directory.mkdir();

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

            try (Writer writer = new BufferedWriter(new OutputStreamWriter(
                    new FileOutputStream(path), "utf-8"))) {
            } catch (UnsupportedEncodingException ex) {
                Logger.getLogger(AllCourses.class.getName()).log(Level.SEVERE, null, ex);
            } catch (IOException ex) {
                Logger.getLogger(AllCourses.class.getName()).log(Level.SEVERE, null, ex);
            }
        } else { //If the folder already exists, display an error message.
            JPanel panel = new JPanel();
            JOptionPane.showMessageDialog(null, "Course name already exists", "Error",
JOptionPane.ERROR_MESSAGE);
            System.out.println("File already exists");
            courseList.remove(courseList.size() - 1);
        }
```

```
}

//List all the courses in the txt file "Course List"
protected void writeAllCourses() {

    String path = appDir + fileSeparator + "Course List.txt";

    for (String s : courses) {
        String tmp = s.toString() + "\t";
        courseList.add(tmp);

    }

    try {
        writeTextFile(path, courseList);
    } catch (IOException ex) {
        System.out.println("Problem! " + path);
    }
}

//Read all the courses in "Course List"
protected ArrayList<String> readAllCourses() throws FileNotFoundException, IOException {

    ArrayList<String> readCourses = new ArrayList<>();

    readCourses = readTextFile(appDir + fileSeparator + "Course List.txt");

    String tmp = "";
    for (int i = 0; i < readCourses.size(); i++) {
            courseList.add(readCourses.get(i));
        }

        return courseList;
    }

    //Returns all the courses listed in the file "Course List"
    protected ArrayList<String> getAllCourses() {
        return courseList;
    }

}
```

3.2 AllCoursework.java

```
package coursework;


import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;


/**
 *
 * @author Willian Cieslak
 */
public class AllCoursework extends CommonCode {


   public AllCoursework() {


   }


   protected ArrayList<Coursework> allCoursework = new ArrayList<>();
   private int maxID = 0;
   protected ArrayList<String> courseworkList = new ArrayList<>();
   protected ArrayList<String> list = new ArrayList<>();


   //Requirements to create a Coursework
   public void addCoursework(int maxID, String name, String dueDate, String requirement) {


       String date = getDateAndTime();
       Coursework myCoursework = new Coursework(maxID, name, date, dueDate, requirement);
       allCoursework.add(myCoursework);
       writeAllCoursework(name);
```

```
    }


    //Creates a file inside the folder "Courseworks" with the same name choosen by the user
    private void writeAllCoursework(String name) {
        String path = appDir + fileSeparator + "Courseworks" + fileSeparator + name.trim() +
".txt";
        ArrayList<String> writeNote = new ArrayList<>();


        for (Coursework c : allCoursework) {
            String tmp = c.getCourseworkID() + "\t";
            tmp += c.getName() + "\t";
            tmp += c.getDate() + "\t";
            tmp += c.getDueDate() + "\t";
            tmp += c.getRequirement() + "\t";
            writeNote.add(tmp);


            System.out.println("ID: " + c.getCourseworkID());
            System.out.println("Name: " + c.getName());
            System.out.println("Date: " + c.getDate());
            System.out.println("Due date: " + c.getDueDate());
            System.out.println("Req: " + c.getRequirement());


        }
        try {
            writeTextFile(path, writeNote);
        } catch (IOException ex) {
            System.out.println("Problem! " + path);
        }
    }
    //Adds the Coursework name to the list "Coursework List"
    protected void writeCourseworkList() {

        String path = appDir + fileSeparator + "Coursework List.txt";
        //ArrayList<String> writeNote = new ArrayList<>();

        for (String s : courseworkList) {
            String tmp = s.toString() + "\t";
            list.add(tmp);

        }

        try {
            writeTextFile(path, list);
        } catch (IOException ex) {
            System.out.println("Problem! " + path);
        }

    }

    public final int getMaxID() {
        maxID++;
        return maxID;
    }

    //Reads all the courses from the list in "Coursework List"
    protected ArrayList<String> readAllCourseworks() throws FileNotFoundException,
IOException {

        ArrayList<String> readCourses = new ArrayList<>();
        readCourses = readTextFile(appDir + fileSeparator + "Coursework List.txt");

        String tmp = "";
        for (int i = 0; i < readCourses.size(); i++) {
            courseworkList.add(readCourses.get(i));
        }

        return courseworkList;
    }

    //Returns all the couseworks
    protected ArrayList<Coursework> getAllCourseworks() {
        return allCoursework;
    }

    //Reads the coursework from the selected Coursework name
    protected void readCoursework(String course) {

        try {

            ArrayList<String> readNotes = new ArrayList<>();

            readNotes = readTextFile(appDir + fileSeparator + "Courseworks" + fileSeparator + course.trim() + ".txt");
            System.out.println(readNotes.get(0));

            if (!"File not found".equals(readNotes.get(0))) {
                allCoursework.clear();
                for (String str : readNotes) {
                    String[] tmp = str.split("\t");

                    int nid = Integer.parseInt(tmp[0]);
                    Coursework c = new Coursework(nid, tmp[1], tmp[2], tmp[3], tmp[4]);
                    allCoursework.add(c);

                    if (nid > maxID) {
                        maxID = nid;
                    }
                }
            }
            maxID++;

            if ("File not found".equals(readNotes.get(0))) {
                allCoursework.clear();
            }

        } catch (Exception e) {
            System.out.println("Could not load the file properly");
        }
    }

}
```

3.3 COURSEWORK.JAVA

```
package coursework;

/**
 *
 * @author Willian Cieslak
 */
public class Coursework {

    private int courseworkID = 0;
    private String name = "";
    private String date = "";
    private String requirement = "";
    private String dueDate = "";

    public Coursework() {

    }

    public Coursework(int max, String name, String date, String dueDate, String requirement) {
        setCourseworkID(max);
        setName(name);
        setDate(date);
        setDueDate(dueDate);
        setRequirement(requirement);
    }

    public Coursework(int max, String name, String dueDate, String requirement) {
        setCourseworkID(max);
    setName(name);
    //setDate(date);
    setDueDate(dueDate);
    setRequirement(requirement);
}

public Coursework(int max, String name, String requirement) {
    setCourseworkID(max);
    setName(name);
    setRequirement(requirement);
}

/**
 * @return the courseworkID
 */
public int getCourseworkID() {
    return courseworkID;
}

/**
 * @param courseworkID the courseworkID to set
 */
public void setCourseworkID(int courseworkID) {
    this.courseworkID = courseworkID;
}

/**
 * @return the name
 */
```

```
public String getName() {
    return name;
}

/**
 * @param name the name to set
 */
public void setName(String name) {
    this.name = name;
}

/**
 * @return the date
 */
public String getDate() {
    return date;
}

/**
 * @param date the date to set
 */
public void setDate(String date) {
    this.date = date;
}

/**
 * @return the requirement
 */
public String getRequirement() {
        return requirement;
    }

    /**
     * @param requirement the requirement to set
     */
    public void setRequirement(String requirement) {
        this.requirement = requirement;
    }

    /**
     * @return the dueDate
     */
    public String getDueDate() {
        return dueDate;
    }

    /**
     * @param dueDate the dueDate to set
     */
    public void setDueDate(String dueDate) {
        this.dueDate = dueDate;
    }

}
```

3.4 CourseworkScreen.java

```
package coursework;

import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

/**
 *
 * @author Willian Cieslak
 */
public class CourseworkScreen extends javax.swing.JFrame {

    private AllCoursework allCourseworks = new AllCoursework();
    private CommonCode cc = new CommonCode();

    /**
     * Creates new form CourseworkScreen
     */
    public CourseworkScreen() {
        try {
            //Load initial components, list all the courses and display the
            //coursework from the coursework that is selected on the combobox
            initComponents();
            allCourseworks.readAllCourseworks();
            System.out.println("Coursework list: " + allCourseworks.courseworkList);
            addAllCourseworks();
            allCourseworks.readCoursework(cbCourseworks.getSelectedItem().toString());
            System.out.println(cbCourseworks.getSelectedItem().toString());
        } catch (IOException ex) {
            Logger.getLogger(CourseworkScreen.class.getName()).log(Level.SEVERE, null, ex);
        }

        displayCoursework();

    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jInternalFrame1 = new javax.swing.JInternalFrame();
        jDesktopPane1 = new javax.swing.JDesktopPane();
        lblCourseworkName = new javax.swing.JLabel();
        lblName = new javax.swing.JLabel();
        jtfCourseworkName = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jtaRequirements = new javax.swing.JTextArea();
        jtfDueDate = new javax.swing.JTextField();
        btnCreateCoursework = new javax.swing.JButton();
        lblName1 = new javax.swing.JLabel();
        jtfDisplayCourseworkName = new javax.swing.JTextField();
        lblDuedate2 = new javax.swing.JLabel();
        jtfDisplayDueDate = new javax.swing.JTextField();
        lblRequiremets2 = new javax.swing.JLabel();
        jScrollPane2 = new javax.swing.JScrollPane();
        jtaDisplayRequirements = new javax.swing.JTextArea();
        btnEdit = new javax.swing.JButton();
        btnSave = new javax.swing.JButton();
        cbCourseworks = new javax.swing.JComboBox<>();
        jMenuBar1 = new javax.swing.JMenuBar();
        jMenuFile = new javax.swing.JMenu();
        jMenuAddCoursework = new javax.swing.JMenuItem();
        jMenuDeleteCoursework = new javax.swing.JMenuItem();
        jSeparator1 = new javax.swing.JPopupMenu.Separator();
        jMenuExit = new javax.swing.JMenuItem();

        jInternalFrame1.setVisible(true);

        javax.swing.GroupLayout jInternalFrame1Layout = new
javax.swing.GroupLayout(jInternalFrame1.getContentPane());
        jInternalFrame1.getContentPane().setLayout(jInternalFrame1Layout);
        jInternalFrame1Layout.setHorizontalGroup(

jInternalFrame1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 0, Short.MAX_VALUE)
        );
        jInternalFrame1Layout.setVerticalGroup(

jInternalFrame1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGap(0, 0, Short.MAX_VALUE)
);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

lblCourseworkName.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
lblCourseworkName.setText("Coursework:");

lblName.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
lblName.setText("Coursework Name:");

jtfCourseworkName.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel1.setText("Due date:");

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel2.setText("Requirements: ");

jtaRequirements.setColumns(20);
jtaRequirements.setFont(new java.awt.Font("Monospaced", 0, 18)); // NOI18N
jtaRequirements.setRows(5);
jScrollPane1.setViewportView(jtaRequirements);

jtfDueDate.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

btnCreateCoursework.setText("Save Coursework");
btnCreateCoursework.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCreateCourseworkActionPerformed(evt);
    }
});

lblName1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
lblName1.setText("Coursework Name:");

jtfDisplayCourseworkName.setEditable(false);
jtfDisplayCourseworkName.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

lblDuedate2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
lblDuedate2.setText("Due date:");

jtfDisplayDueDate.setEditable(false);
jtfDisplayDueDate.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

lblRequiremets2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
lblRequiremets2.setText("Requirements: ");

jtaDisplayRequirements.setEditable(false);
jtaDisplayRequirements.setColumns(20);
jtaDisplayRequirements.setFont(new java.awt.Font("Monospaced", 0, 18)); // NOI18N
jtaDisplayRequirements.setRows(5);
jScrollPane2.setViewportView(jtaDisplayRequirements);

btnEdit.setText("Edit");
btnEdit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnEditActionPerformed(evt);
```

```
      }
    });

    btnSave.setText("Save");
    btnSave.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSaveActionPerformed(evt);
      }
    });

    cbCourseworks.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    cbCourseworks.addItemListener(new java.awt.event.ItemListener() {
      public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cbCourseworksItemStateChanged(evt);
      }
    });

    jDesktopPane1.setLayer(lblCourseworkName,
javax.swing.JLayeredPane.DEFAULT_LAYER);
    jDesktopPane1.setLayer(lblName, javax.swing.JLayeredPane.DEFAULT_LAYER);
    jDesktopPane1.setLayer(jtfCourseworkName,
javax.swing.JLayeredPane.DEFAULT_LAYER);
    jDesktopPane1.setLayer(jLabel1, javax.swing.JLayeredPane.DEFAULT_LAYER);
    jDesktopPane1.setLayer(jLabel2, javax.swing.JLayeredPane.DEFAULT_LAYER);
    jDesktopPane1.setLayer(jScrollPane1, javax.swing.JLayeredPane.DEFAULT_LAYER);
    jDesktopPane1.setLayer(jtfDueDate, javax.swing.JLayeredPane.DEFAULT_LAYER);
    jDesktopPane1.setLayer(btnCreateCoursework,
javax.swing.JLayeredPane.DEFAULT_LAYER);
    jDesktopPane1.setLayer(lblName1, javax.swing.JLayeredPane.DEFAULT_LAYER);
    jDesktopPane1.setLayer(jtfDisplayCourseworkName,
javax.swing.JLayeredPane.DEFAULT_LAYER);
```

```
        jDesktopPane1.setLayer(lblDuedate2, javax.swing.JLayeredPane.DEFAULT_LAYER);
        jDesktopPane1.setLayer(jtfDisplayDueDate,
javax.swing.JLayeredPane.DEFAULT_LAYER);
        jDesktopPane1.setLayer(lblRequiremets2,
javax.swing.JLayeredPane.DEFAULT_LAYER);
        jDesktopPane1.setLayer(jScrollPane2, javax.swing.JLayeredPane.DEFAULT_LAYER);
        jDesktopPane1.setLayer(btnEdit, javax.swing.JLayeredPane.DEFAULT_LAYER);
        jDesktopPane1.setLayer(btnSave, javax.swing.JLayeredPane.DEFAULT_LAYER);
        jDesktopPane1.setLayer(cbCourseworks, javax.swing.JLayeredPane.DEFAULT_LAYER);

        javax.swing.GroupLayout jDesktopPane1Layout = new
javax.swing.GroupLayout(jDesktopPane1);
        jDesktopPane1.setLayout(jDesktopPane1Layout);
        jDesktopPane1Layout.setHorizontalGroup(

jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jDesktopPane1Layout.createSequentialGroup()
                .addGap(115, 115, 115)
                .addGroup(jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alig
nment.LEADING, false)
                    .addComponent(lblCourseworkName, javax.swing.GroupLayout.DEFAULT_SIZE,
224, Short.MAX_VALUE)
                    .addComponent(cbCourseworks, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 109,
Short.MAX_VALUE)
                .addGroup(jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alig
nment.LEADING, false)
                    .addComponent(btnCreateCoursework,  javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jLabel1)
                    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 172,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                    .addComponent(lblName)
                    .addComponent(jtfCourseworkName)
                    .addComponent(jtfDueDate)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 450, Short.MAX_VALUE))
                .addGap(56, 56, 56)
                .addGroup(jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(lblName1)
                    .addComponent(lblRequiremets2, javax.swing.GroupLayout.PREFERRED_SIZE, 172, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(lblDuedate2)
                    .addGroup(jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                        .addGroup(jDesktopPane1Layout.createSequentialGroup()
                            .addComponent(btnEdit)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(btnSave))
                        .addComponent(jtfDisplayDueDate, javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jScrollPane2, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT_SIZE, 450, Short.MAX_VALUE)
                        .addComponent(jtfDisplayCourseworkName, javax.swing.GroupLayout.Alignment.LEADING)))
                .addGap(335, 335, 335))
        );
        jDesktopPane1Layout.setVerticalGroup(
jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jDesktopPane1Layout.createSequentialGroup()
                .addGap(16, 16, 16)
                .addGroup(jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(lblName)
                    .addComponent(lblName1))
                .addGap(13, 13, 13)
                .addGroup(jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jtfCourseworkName, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jtfDisplayCourseworkName, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(22, 22, 22)
                .addGroup(jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(lblDuedate2))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jtfDueDate, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jtfDisplayDueDate, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 45, Short.MAX_VALUE)
                .addGroup(jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(lblRequiremets2))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                  .addGroup(jDesktopPane1Layout.createSequentialGroup()
                    .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 365, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addGroup(jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                      .addComponent(btnEdit)
                      .addComponent(btnSave)))
                  .addGroup(jDesktopPane1Layout.createSequentialGroup()
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 365, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(btnCreateCoursework)))
                .addGap(22, 22, 22))
              .addGroup(jDesktopPane1Layout.createSequentialGroup()
                .addGap(33, 33, 33)
                .addComponent(lblCourseworkName)
                .addGap(18, 18, 18)
                .addComponent(cbCourseworks, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            );

            jMenuFile.setText("File");

            jMenuAddCoursework.setText("Add Coursework");
            jMenuAddCoursework.addActionListener(new java.awt.event.ActionListener() {
              public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuAddCourseworkActionPerformed(evt);
              }
```

```
});
jMenuFile.add(jMenuAddCoursework);

jMenuDeleteCoursework.setText("Delete Coursework");
jMenuDeleteCoursework.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuDeleteCourseworkActionPerformed(evt);
    }
});
jMenuFile.add(jMenuDeleteCoursework);
jMenuFile.add(jSeparator1);

jMenuExit.setText("Exit");
jMenuExit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuExitActionPerformed(evt);
    }
});
jMenuFile.add(jMenuExit);

jMenuBar1.add(jMenuFile);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
                .addComponent(jDesktopPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jDesktopPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        );

        pack();
    }// </editor-fold>

    private void jMenuAddCourseworkActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        createCoursework();
    }

    private void jMenuDeleteCourseworkActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        deleteCoursework();
    }

    private void jMenuExitActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        this.dispose();
    }

    private void btnCreateCourseworkActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
```

```
    createCoursework();
}

private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jtaDisplayRequirements.setEditable(true);
}

private void cbCourseworksItemStateChanged(java.awt.event.ItemEvent evt) {

    System.out.println(cbCourseworks.getSelectedItem().toString());
    displayCoursework();
    allCourseworks.readCoursework(cbCourseworks.getSelectedItem().toString());
}

private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String requirements = jtaDisplayRequirements.getText().toString();
    // jtaDisplayRequirements.setText("");

    allCourseworks.allCoursework.clear();

    editCoursework(requirements);
    jtaDisplayRequirements.setEditable(false);

    JOptionPane.showMessageDialog(null, "File Saved!");
}

/**
```

```
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;

                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(CourseworkScreen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(CourseworkScreen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(CourseworkScreen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(CourseworkScreen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        //</editor-fold>
        //</editor-fold>
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new CourseworkScreen().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnCreateCoursework;
    private javax.swing.JButton btnEdit;
    private javax.swing.JButton btnSave;
    private javax.swing.JComboBox<String> cbCourseworks;
    private javax.swing.JDesktopPane jDesktopPane1;
    private javax.swing.JInternalFrame jInternalFrame1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JMenuItem jMenuAddCoursework;
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JMenuItem jMenuDeleteCoursework;
    private javax.swing.JMenuItem jMenuExit;
```

```
private javax.swing.JMenu jMenuFile;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JPopupMenu.Separator jSeparator1;
private javax.swing.JTextArea jtaDisplayRequirements;
private javax.swing.JTextArea jtaRequirements;
private javax.swing.JTextField jtfCourseworkName;
private javax.swing.JTextField jtfDisplayCourseworkName;
private javax.swing.JTextField jtfDisplayDueDate;
private javax.swing.JTextField jtfDueDate;
private javax.swing.JLabel lblCourseworkName;
private javax.swing.JLabel lblDuedate2;
private javax.swing.JLabel lblName;
private javax.swing.JLabel lblName1;
private javax.swing.JLabel lblRequiremets2;
// End of variables declaration

private void addAllCourseworks() {

    for (String crse : allCourseworks.courseworkList) {
        cbCourseworks.addItem(crse);
    }
}

private void addCoursework(String requirement) {

    int id = allCourseworks.getMaxID();
    String name = jtfCourseworkName.getText();
    String dueDate = jtfDueDate.getText();
    allCourseworks.addCoursework(id, name, dueDate, requirement);

}

private void editCoursework(String requirement) {

    int id = allCourseworks.getMaxID();
    String name = jtfDisplayCourseworkName.getText();
    String dueDate = jtfDisplayDueDate.getText();

    allCourseworks.allCoursework.clear();

    allCourseworks.addCoursework(id, name, dueDate, requirement);

}

private void displayCoursework() {

    String txtNotes = "";

    String courseworkName = "";
    String dueDate = "";
    String requirements = "";

    for (Coursework c : allCourseworks.getAllCourseworks()) {
        courseworkName = c.getName();
        dueDate = c.getDueDate();
        requirements = c.getRequirement();
        }

        System.out.println("Coursework name: " + courseworkName);

        jtfDisplayCourseworkName.setText(courseworkName);
        jtfDisplayDueDate.setText(dueDate);
        jtaDisplayRequirements.setText(requirements);

    }

    private void createCoursework() {
        String newCoursework = jtfCourseworkName.getText();
        String requirements = jtaRequirements.getText();

        if (newCoursework.equals("")) {
            JPanel panel = new JPanel();
            JOptionPane.showMessageDialog(null, "Course name cannot be blank", "Error",
JOptionPane.ERROR_MESSAGE);
        } else {
            allCourseworks.courseworkList.add(newCoursework);
            addCoursework(requirements);
            cbCourseworks.addItem(newCoursework);
            allCourseworks.writeCourseworkList();

            jtfCourseworkName.setText("");
            jtfDueDate.setText("");
            jtaRequirements.setText("");
        }
    }
    private void deleteCoursework() {

        //to be implemented

    }

}
```

3.5 MAIN.JAVA

```
package coursework;


import java.io.IOException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;


/**
 *
 * @author Willian Cieslak
 */
public class Main extends javax.swing.JFrame {


    CommonCode cc = new CommonCode();
    ArrayList<String> note = new ArrayList<>();
    public ArrayList<String> courses = new ArrayList<>();
    String crse = "";
    AllNotes allNotes = new AllNotes(this);
    AllCourses allCourses = new AllCourses();
    String newCourse = "";


    public Main() {


        model();
        initComponents();
        controller();
```

```
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jDesktopPane1 = new javax.swing.JDesktopPane();
    jDialog1 = new javax.swing.JDialog();
    jDialog2 = new javax.swing.JDialog();
    jDesktopPane2 = new javax.swing.JDesktopPane();
    jScrollPane2 = new javax.swing.JScrollPane();
    jtaNewNote = new javax.swing.JTextArea();
    jScrollPane3 = new javax.swing.JScrollPane();
    jtaAllNotes = new javax.swing.JTextArea();
    cbCourses = new javax.swing.JComboBox<>();
    btnNewNote = new javax.swing.JButton();
    btnSearch = new javax.swing.JButton();
    jtfSearch = new javax.swing.JTextField();
    btnAddCourse = new javax.swing.JButton();
    jlNotes = new javax.swing.JLabel();
    jtfAddCourse = new javax.swing.JTextField();
    lblNewNotes = new javax.swing.JLabel();
    jMenuBar1 = new javax.swing.JMenuBar();
```

```
        jMenu1 = new javax.swing.JMenu();
        jMenuSave = new javax.swing.JMenuItem();
        jSeparator1 = new javax.swing.JPopupMenu.Separator();
        jMenuDeleteCourse = new javax.swing.JMenuItem();
        jMenuCoursework = new javax.swing.JMenuItem();
        jSeparator2 = new javax.swing.JPopupMenu.Separator();
        jMenuExit = new javax.swing.JMenuItem();
        jMenu2 = new javax.swing.JMenu();
        jMenuItem1 = new javax.swing.JMenuItem();

        javax.swing.GroupLayout jDesktopPane1Layout = new 
javax.swing.GroupLayout(jDesktopPane1);
        jDesktopPane1.setLayout(jDesktopPane1Layout);
        jDesktopPane1Layout.setHorizontalGroup(

jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 100, Short.MAX_VALUE)
        );
        jDesktopPane1Layout.setVerticalGroup(

jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 100, Short.MAX_VALUE)
        );

        javax.swing.GroupLayout jDialog1Layout = new 
javax.swing.GroupLayout(jDialog1.getContentPane());
        jDialog1.getContentPane().setLayout(jDialog1Layout);
        jDialog1Layout.setHorizontalGroup(
            jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 400, Short.MAX_VALUE)
        );
        jDialog1Layout.setVerticalGroup(
            jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 300, Short.MAX_VALUE)
        );

        setTitle("Coursework - Willian Cieslak  // 000990795 - wc7377c");

        javax.swing.GroupLayout jDialog2Layout = new javax.swing.GroupLayout(jDialog2.getContentPane());
        jDialog2.getContentPane().setLayout(jDialog2Layout);
        jDialog2Layout.setHorizontalGroup(
            jDialog2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 400, Short.MAX_VALUE)
        );
        jDialog2Layout.setVerticalGroup(
            jDialog2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 300, Short.MAX_VALUE)
        );

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jtaNewNote.setColumns(20);
        jtaNewNote.setFont(new java.awt.Font("Monospaced", 0, 18)); // NOI18N
        jtaNewNote.setRows(5);
        jScrollPane2.setViewportView(jtaNewNote);

        jtaAllNotes.setEditable(false);
        jtaAllNotes.setColumns(20);
        jtaAllNotes.setFont(new java.awt.Font("Monospaced", 0, 18)); // NOI18N
        jtaAllNotes.setRows(5);
```

```
jScrollPane3.setViewportView(jtaAllNotes);

cbCourses.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
cbCourses.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cbCoursesItemStateChanged(evt);
    }
});
cbCourses.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cbCoursesActionPerformed(evt);
    }
});

btnNewNote.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
btnNewNote.setText("New Note");
btnNewNote.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnNewNoteActionPerformed(evt);
    }
});

btnSearch.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
btnSearch.setText("Search");
btnSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSearchActionPerformed(evt);
    }
});
```

```
jtfSearch.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jtfSearch.setToolTipText("");
jtfSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jtfSearchActionPerformed(evt);
    }
});

btnAddCourse.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
btnAddCourse.setText("Add Course");
btnAddCourse.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAddCourseActionPerformed(evt);
    }
});

jlNotes.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
jlNotes.setText("NOTES:");

jtfAddCourse.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

lblNewNotes.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
lblNewNotes.setText("New Notes:");

jDesktopPane2.setLayer(jScrollPane2, javax.swing.JLayeredPane.DEFAULT_LAYER);
jDesktopPane2.setLayer(jScrollPane3, javax.swing.JLayeredPane.DEFAULT_LAYER);
jDesktopPane2.setLayer(cbCourses, javax.swing.JLayeredPane.DEFAULT_LAYER);
jDesktopPane2.setLayer(btnNewNote, javax.swing.JLayeredPane.DEFAULT_LAYER);
```

```
        jDesktopPane2.setLayer(btnSearch, javax.swing.JLayeredPane.DEFAULT_LAYER);
        jDesktopPane2.setLayer(jtfSearch, javax.swing.JLayeredPane.DEFAULT_LAYER);
        jDesktopPane2.setLayer(btnAddCourse, javax.swing.JLayeredPane.DEFAULT_LAYER);
        jDesktopPane2.setLayer(jlNotes, javax.swing.JLayeredPane.DEFAULT_LAYER);
        jDesktopPane2.setLayer(jtfAddCourse, javax.swing.JLayeredPane.DEFAULT_LAYER);
        jDesktopPane2.setLayer(lblNewNotes, javax.swing.JLayeredPane.DEFAULT_LAYER);

        javax.swing.GroupLayout jDesktopPane2Layout = new javax.swing.GroupLayout(jDesktopPane2);
        jDesktopPane2.setLayout(jDesktopPane2Layout);
        jDesktopPane2Layout.setHorizontalGroup(

jDesktopPane2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(jDesktopPane2Layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jDesktopPane2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addComponent(btnNewNote, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
              .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 500, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(lblNewNotes))
            .addGap(18, 18, 18)
            .addGroup(jDesktopPane2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addGroup(jDesktopPane2Layout.createSequentialGroup()
                .addComponent(jlNotes, javax.swing.GroupLayout.PREFERRED_SIZE, 212, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(27, 27, 27)
                .addComponent(btnSearch)
                .addGap(18, 18, 18)
                .addComponent(jtfSearch, javax.swing.GroupLayout.PREFERRED_SIZE, 160, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(btnAddCourse, javax.swing.GroupLayout.PREFERRED_SIZE, 111, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jtfAddCourse, javax.swing.GroupLayout.PREFERRED_SIZE, 160, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(cbCourses, javax.swing.GroupLayout.PREFERRED_SIZE, 131, javax.swing.GroupLayout.PREFERRED_SIZE))
              .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 932, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap())
        );
        jDesktopPane2Layout.setVerticalGroup(

jDesktopPane2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(jDesktopPane2Layout.createSequentialGroup()
            .addGap(16, 16, 16)
            .addGroup(jDesktopPane2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(btnSearch)
              .addComponent(jtfSearch, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(btnAddCourse)
              .addComponent(cbCourses, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jtfAddCourse, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jlNotes)
              .addComponent(lblNewNotes))
```

```
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(jDesktopPane2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jDesktopPane2Layout.createSequentialGroup()
                        .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 620, Short.MAX_VALUE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(btnNewNote))
                    .addComponent(jScrollPane3))
                .addContainerGap())
        );

        //for (String crse : allCourses.courseList) {
            //              cbCourses.addItem(crse);
            //        }

        jMenu1.setText("File");

        jMenuSave.setText("Save File");
        jMenuSave.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuSaveActionPerformed(evt);
            }
        });
        jMenu1.add(jMenuSave);
        jMenu1.add(jSeparator1);

        jMenuDeleteCourse.setText("Delete Course");
        jMenuDeleteCourse.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jMenuDeleteCourseActionPerformed(evt);
    }
});
jMenu1.add(jMenuDeleteCourse);

jMenuCoursework.setText("Coursework Screen");
jMenuCoursework.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuCourseworkActionPerformed(evt);
    }
});
jMenu1.add(jMenuCoursework);
jMenu1.add(jSeparator2);

jMenuExit.setText("Exit");
jMenuExit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuExitActionPerformed(evt);
    }
});
jMenu1.add(jMenuExit);

jMenuBar1.add(jMenu1);

jMenu2.setText("Edit");
jMenu2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenu2ActionPerformed(evt);
    }
```

```
        });

        jMenuItem1.setText("Edit text");
        jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem1ActionPerformed(evt);
            }
        });
        jMenu2.add(jMenuItem1);

        jMenuBar1.add(jMenu2);

        setJMenuBar(jMenuBar1);

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jDesktopPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jDesktopPane2)
        );

        pack();
    }// </editor-fold>
    private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String lyst = allNotes.searchAllNotesByKeyword("", 0, jtfSearch.getText());
        jtaAllNotes.setText(lyst);
    }

    private void btnAddCourseActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        newCourse = jtfAddCourse.getText().toString();

        if (newCourse.equals("")) {
            JPanel panel = new JPanel();
            JOptionPane.showMessageDialog(null, "Course name cannot be blank", "Error",
JOptionPane.ERROR_MESSAGE);
        } else {
            jtfAddCourse.setText("");
            allCourses.courses.add(newCourse);
            addCourse(newCourse);
            cbCourses.addItem(newCourse);

            //JOptionPane.showMessageDialog(null, "Course added!");
        }
    }

    private void btnNewNoteActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        addNote(jtaNewNote.getText());
        jtaNewNote.setText("");
```

```
}

private void cbCoursesActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jMenu2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:


}

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jtaAllNotes.setEditable(true);
}

private void jMenuSaveActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    String notes = jtaAllNotes.getText().toString();
    jtaAllNotes.setText("");

    allNotes.allNotes.clear();

    addNote(notes);
    jtaAllNotes.setEditable(false);

    JOptionPane.showMessageDialog(null, "File Saved!");
```

```
}

private void jtfSearchActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
}

private void cbCoursesItemStateChanged(java.awt.event.ItemEvent evt) {
   // TODO add your handling code here:
   if (cbCourses.equals(null)) {
      System.out.println("Problem!");
   } else {
      loadNotesFromCourse();
   }
}

private void jMenuExitActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
   System.exit(0);
}

private void jMenuCourseworkActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
   CourseworkScreen cs = new CourseworkScreen();
   cs.setVisible(true);
}

private void jMenuDeleteCourseActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
        int reply = JOptionPane.showConfirmDialog(null, "Are you sure you want to delete this course ?", "", JOptionPane.YES_NO_OPTION);
        if (reply == JOptionPane.YES_OPTION) {
            System.out.println(allCourses.courseList);


            allCourses.courseList.remove(cbCourses.getSelectedIndex());


            allCourses.writeAllCourses();
            //addAllCourses();
            JOptionPane.showMessageDialog(null, "Course deleted. Reset the program to update the changes");
        } else {
            JOptionPane.showMessageDialog(null, "No changes have been made");
        }
    }


    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
        }
    }
} catch (ClassNotFoundException ex) {
    java.util.logging.Logger.getLogger(Main.class
            .getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(Main.class
            .getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(Main.class
            .getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(Main.class
            .getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
```

```
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>

JOptionPane.showMessageDialog(null, "Willian Cieslak - wc7377c");

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Main().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton btnAddCourse;
private javax.swing.JButton btnNewNote;
private javax.swing.JButton btnSearch;
public javax.swing.JComboBox<String> cbCourses;
private javax.swing.JDesktopPane jDesktopPane1;
private javax.swing.JDesktopPane jDesktopPane2;
private javax.swing.JDialog jDialog1;
private javax.swing.JDialog jDialog2;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuCoursework;
private javax.swing.JMenuItem jMenuDeleteCourse;
private javax.swing.JMenuItem jMenuExit;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuSave;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JPopupMenu.Separator jSeparator1;
private javax.swing.JPopupMenu.Separator jSeparator2;
private javax.swing.JLabel jlNotes;
private javax.swing.JTextArea jtaAllNotes;
private javax.swing.JTextArea jtaNewNote;
private javax.swing.JTextField jtfAddCourse;
private javax.swing.JTextField jtfSearch;
private javax.swing.JLabel lblNewNotes;
// End of variables declaration

protected void model() {
    try {
        System.out.println(allCourses.readAllCourses());
    } catch (IOException ex) {
        Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
    }
}

protected void controller() {
    addAllCourses();
    addAllNotes();

}

private void addNote(String text) {
    allNotes.addNote(allNotes.getMaxID(), cbCourses.getSelectedItem().toString(), text);
    addAllNotes();
}

private void addAllNotes() {
    String txtNotes = "";

    for (Note n : allNotes.getAllNotes()) {
        txtNotes += n.getNote() + "\n";
    }
```

```
        jtaAllNotes.setText(txtNotes);
    }

    private void addCourse(String newCourse) {
        allCourses.newCourseFile(newCourse);
        allCourses.writeAllCourses();
    }

    private void addAllCourses() {

        for (String crse : allCourses.courseList) {
            cbCourses.addItem(crse);
        }

    }

    private void loadNotesFromCourse() {
        String course = cbCourses.getSelectedItem().toString();
        allNotes.readNotesFromCourse(course.trim());
        addAllNotes();
    }

}
```

4 WHITE BOX TESTING

4.1 TESTING FOR CLASS …

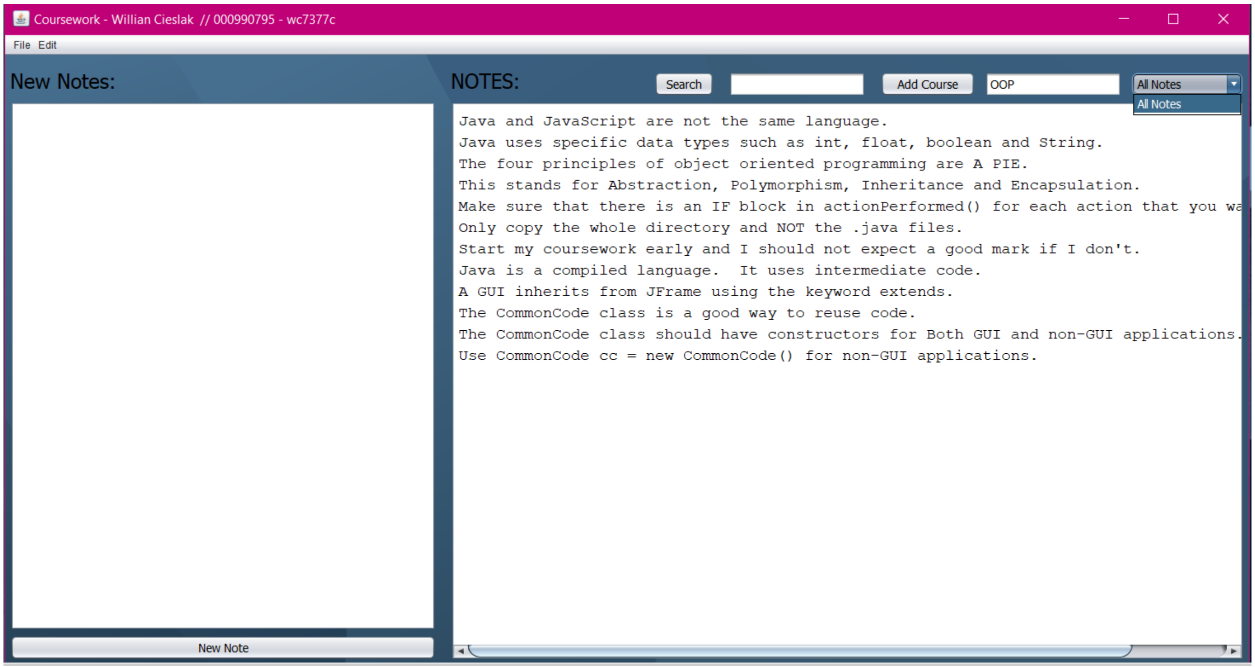
4.2 TESTING FOR CLASS …

4.3 TESTING FOR CLASS …

4.4 TESTING FOR CLASS …

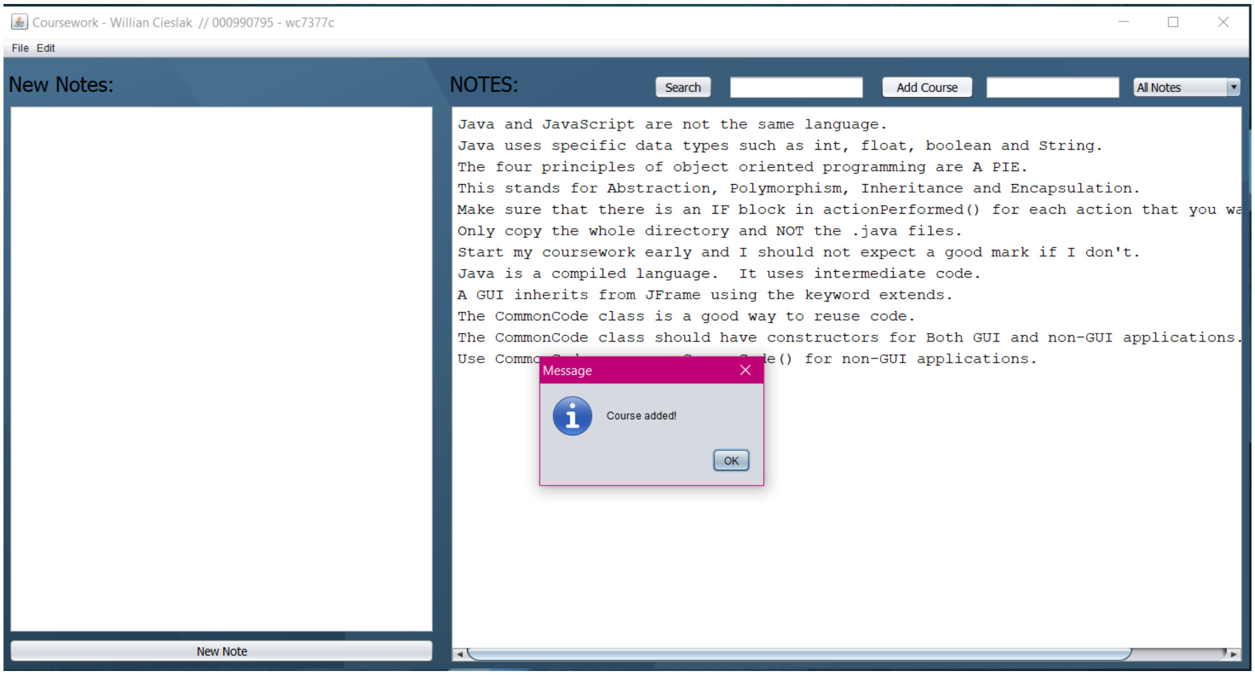
5 Screen Shots of the Program Working

5.1 Screen 1 – Adding a Course



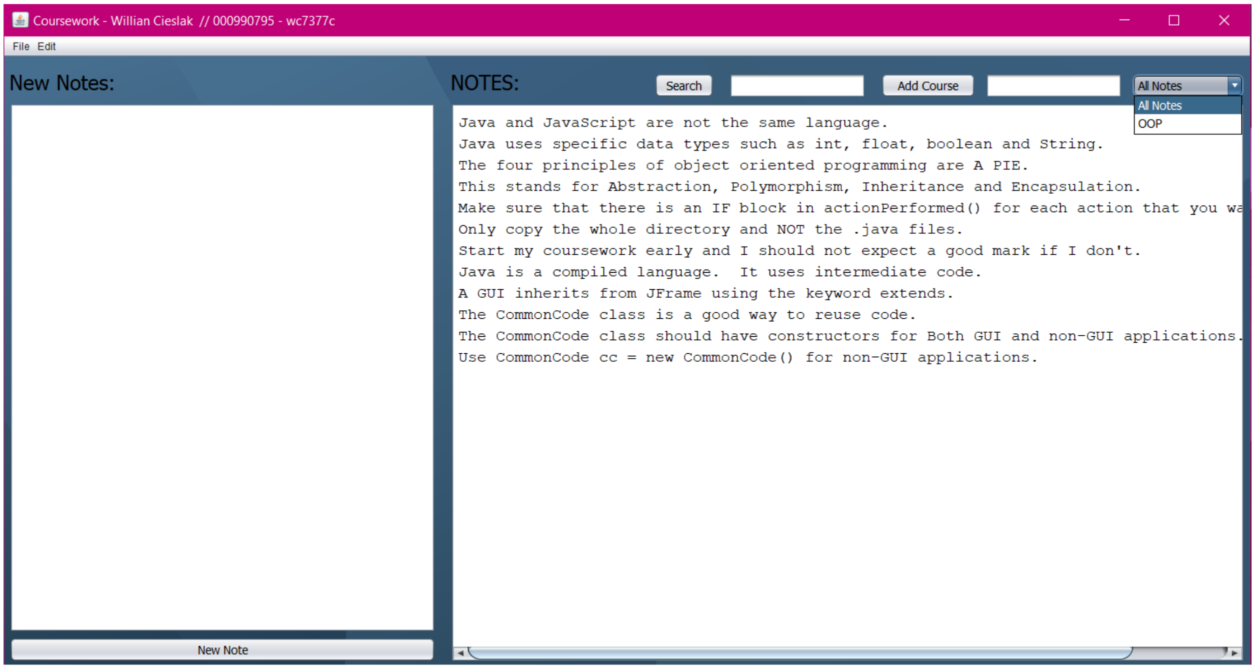
OOP

```
Java and JavaScript are not the same language.
Java uses specific data types such as int, float, boolean and String.
The four principles of object oriented programming are A PIE.
This stands for Abstraction, Polymorphism, Inheritance and Encapsulation.
Make sure that there is an IF block in actionPerformed() for each action that you wa
Only copy the whole directory and NOT the .java files.
Start my coursework early and I should not expect a good mark if I don't.
Java is a compiled language.  It uses intermediate code.
A GUI inherits from JFrame using the keyword extends.
The CommonCode class is a good way to reuse code.
The CommonCode class should have constructors for Both GUI and non-GUI applications.
Use CommonCode cc = new CommonCode() for non-GUI applications.
```



```
Java and JavaScript are not the same language.
Java uses specific data types such as int, float, boolean and String.
The four principles of object oriented programming are A PIE.
This stands for Abstraction, Polymorphism, Inheritance and Encapsulation.
Make sure that there is an IF block in actionPerformed() for each action that you wa
Only copy the whole directory and NOT the .java files.
Start my coursework early and I should not expect a good mark if I don't.
Java is a compiled language.  It uses intermediate code.
A GUI inherits from JFrame using the keyword extends.
The CommonCode class is a good way to reuse code.
The CommonCode class should have constructors for Both GUI and non-GUI applications.
Use Commo                  e() for non-GUI applications.
```

Course added!

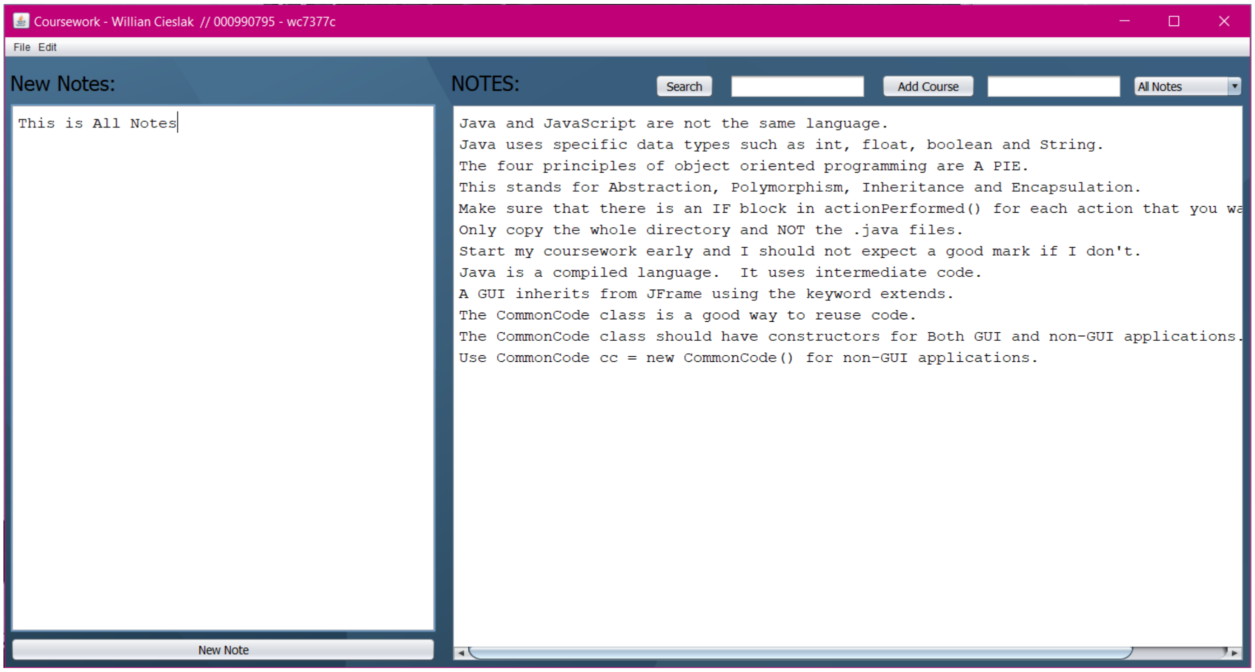


```
Java and JavaScript are not the same language.
Java uses specific data types such as int, float, boolean and String.
The four principles of object oriented programming are A PIE.
This stands for Abstraction, Polymorphism, Inheritance and Encapsulation.
Make sure that there is an IF block in actionPerformed() for each action that you wa
Only copy the whole directory and NOT the .java files.
Start my coursework early and I should not expect a good mark if I don't.
Java is a compiled language.  It uses intermediate code.
A GUI inherits from JFrame using the keyword extends.
The CommonCode class is a good way to reuse code.
The CommonCode class should have constructors for Both GUI and non-GUI applications.
Use CommonCode cc = new CommonCode() for non-GUI applications.
```

5.2 Screen 1 – A brief description

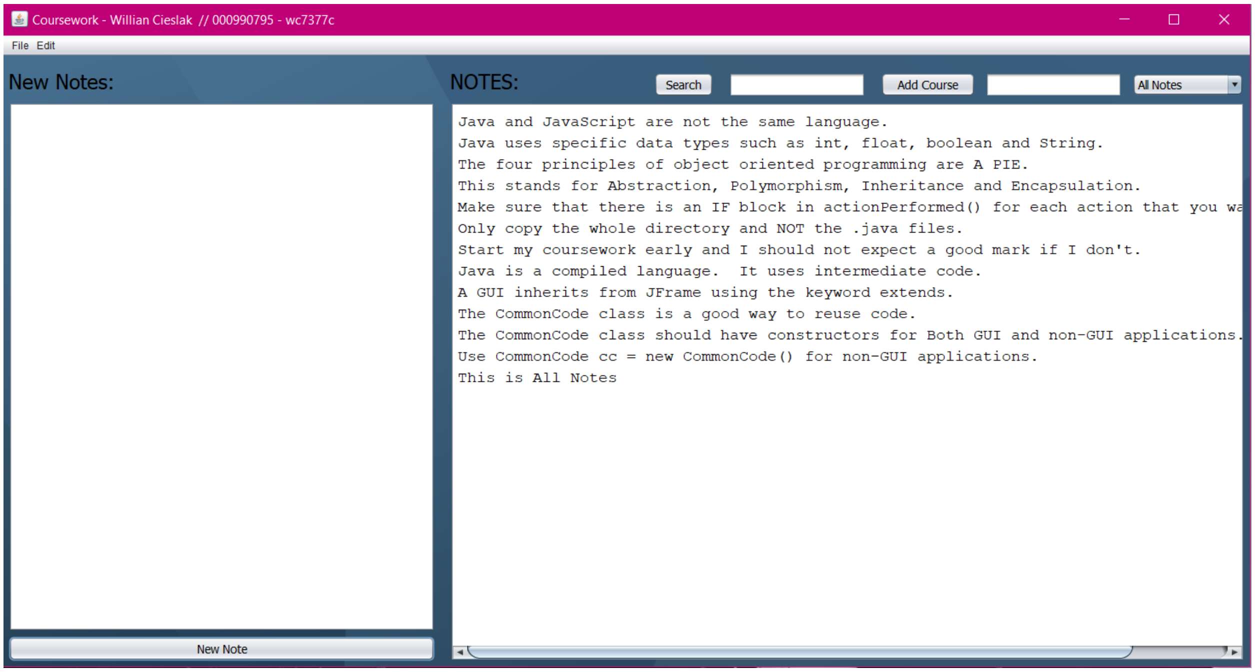
- Clicking on the button “Add Course” will add an item with the chosen name in the combo box on the right
- At this moment we only added a new Course to the list of courses, and OOP has no notes.

5.3 Screen 2 – Adding Notes

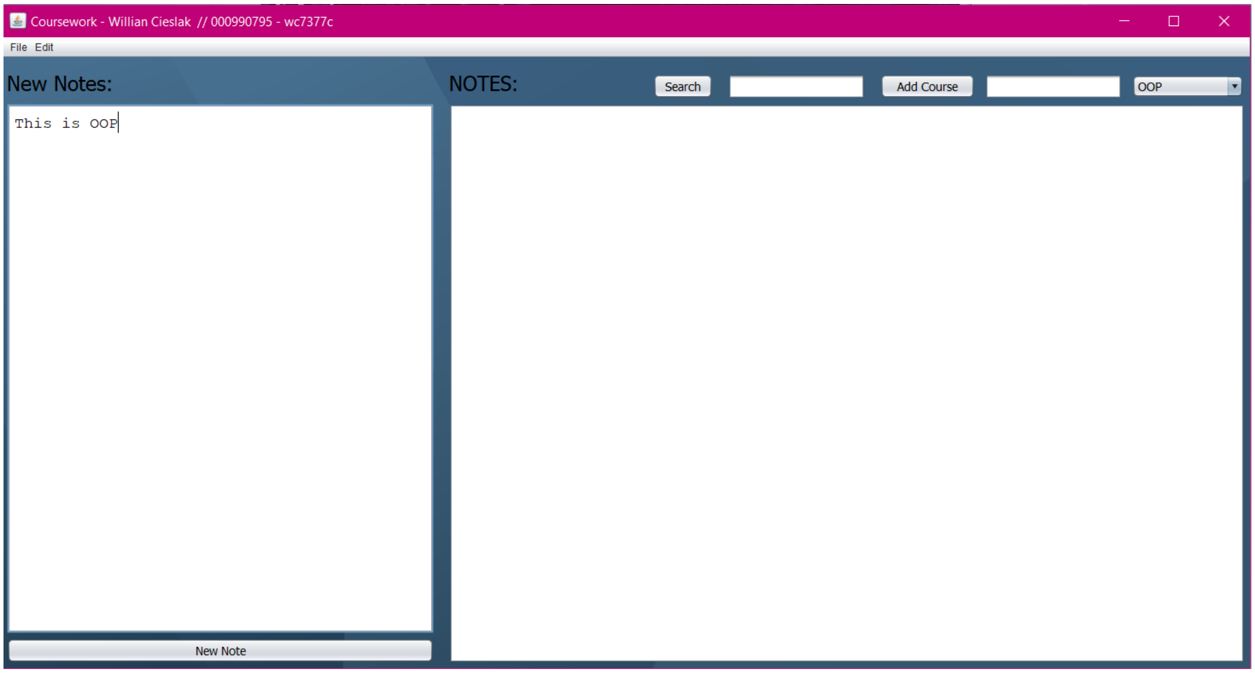


```
This is All Notes
```

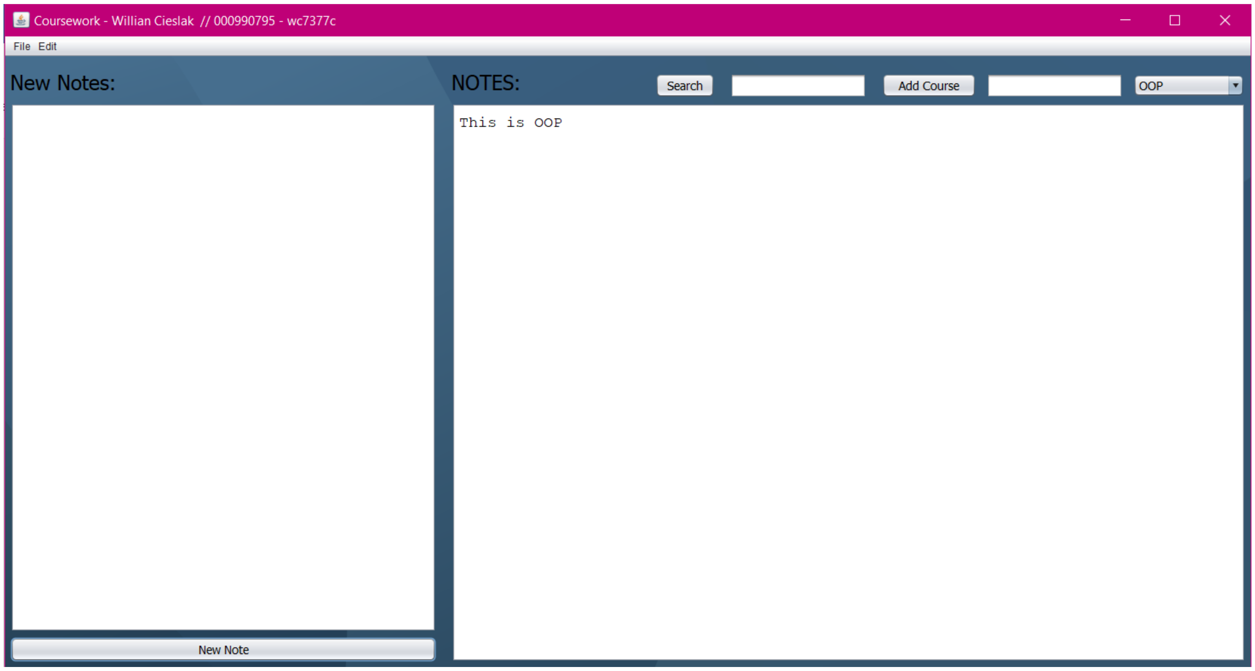
```
Java and JavaScript are not the same language.
Java uses specific data types such as int, float, boolean and String.
The four principles of object oriented programming are A PIE.
This stands for Abstraction, Polymorphism, Inheritance and Encapsulation.
Make sure that there is an IF block in actionPerformed() for each action that you wa
Only copy the whole directory and NOT the .java files.
Start my coursework early and I should not expect a good mark if I don't.
Java is a compiled language.  It uses intermediate code.
A GUI inherits from JFrame using the keyword extends.
The CommonCode class is a good way to reuse code.
The CommonCode class should have constructors for Both GUI and non-GUI applications.
Use CommonCode cc = new CommonCode() for non-GUI applications.
```



```
Java and JavaScript are not the same language.
Java uses specific data types such as int, float, boolean and String.
The four principles of object oriented programming are A PIE.
This stands for Abstraction, Polymorphism, Inheritance and Encapsulation.
Make sure that there is an IF block in actionPerformed() for each action that you wa
Only copy the whole directory and NOT the .java files.
Start my coursework early and I should not expect a good mark if I don't.
Java is a compiled language.  It uses intermediate code.
A GUI inherits from JFrame using the keyword extends.
The CommonCode class is a good way to reuse code.
The CommonCode class should have constructors for Both GUI and non-GUI applications.
Use CommonCode cc = new CommonCode() for non-GUI applications.
This is All Notes
```



This is OOP

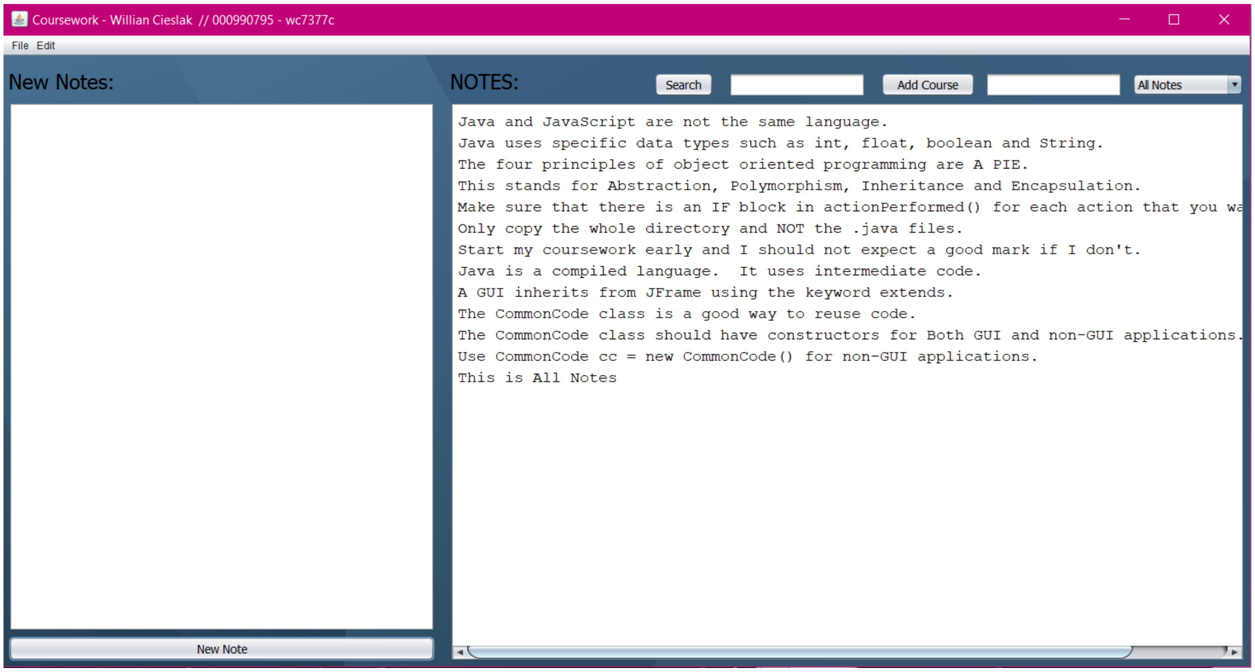


This is OOP

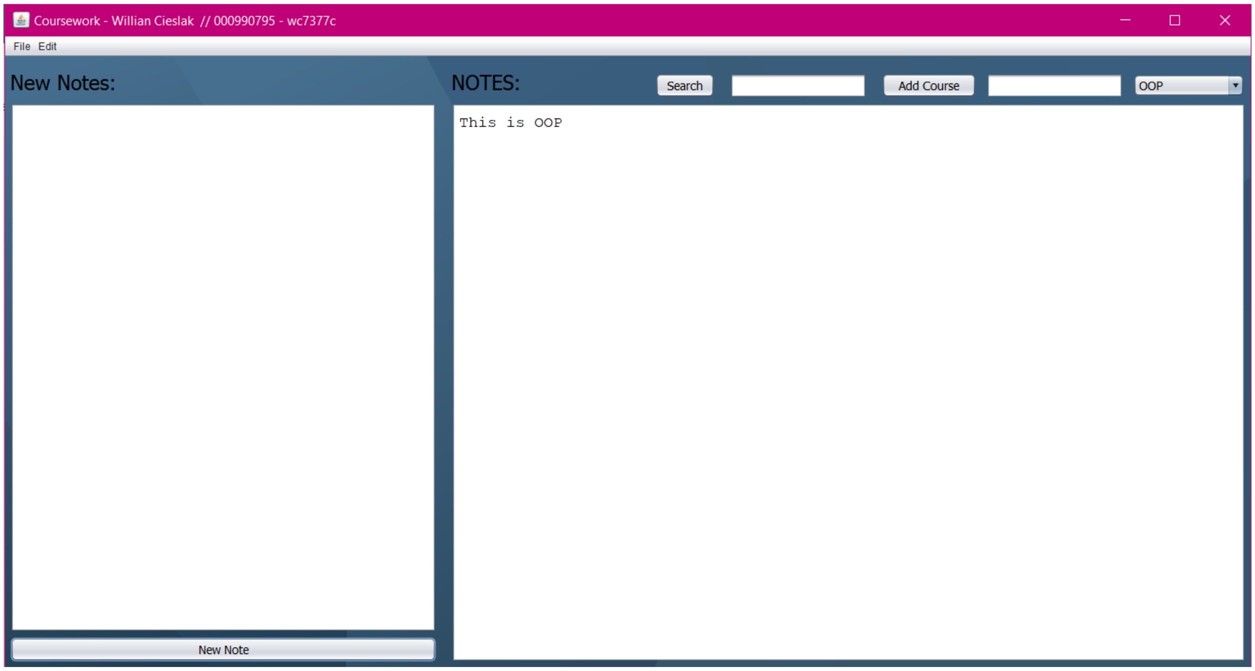
5.4 Screen 2 – A brief description

“New Note” creates a new note. The course selected on the combo box defines in which course your notes will be saved.

5.5 Screen 3 – Reading Saved Notes



```
Java and JavaScript are not the same language.
Java uses specific data types such as int, float, boolean and String.
The four principles of object oriented programming are A PIE.
This stands for Abstraction, Polymorphism, Inheritance and Encapsulation.
Make sure that there is an IF block in actionPerformed() for each action that you wa
Only copy the whole directory and NOT the .java files.
Start my coursework early and I should not expect a good mark if I don't.
Java is a compiled language.  It uses intermediate code.
A GUI inherits from JFrame using the keyword extends.
The CommonCode class is a good way to reuse code.
The CommonCode class should have constructors for Both GUI and non-GUI applications.
Use CommonCode cc = new CommonCode() for non-GUI applications.
This is All Notes
```

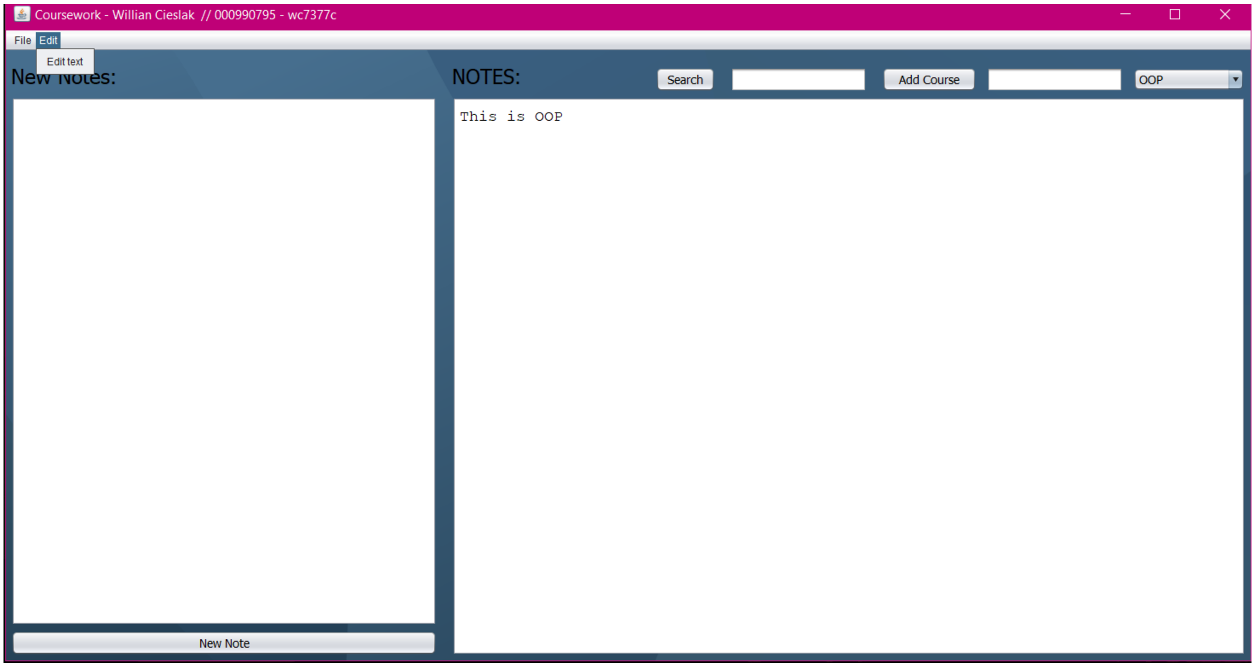


```
This is OOP
```

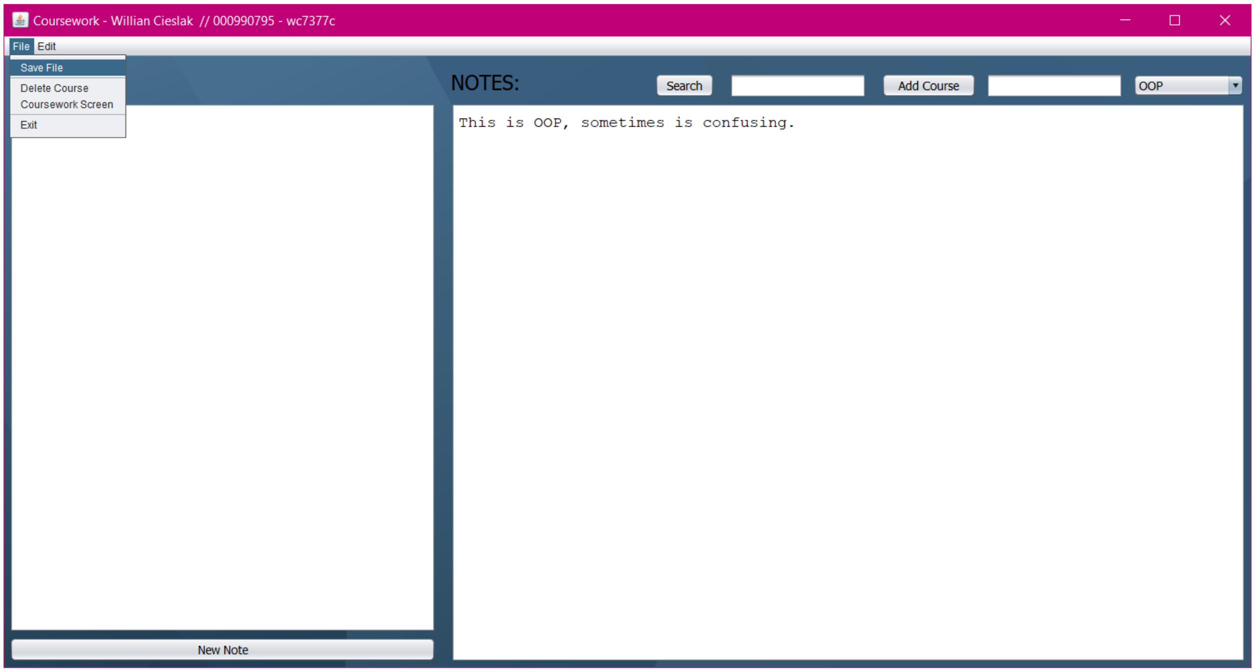
5.6 Screen 3 – A brief description

Changing the course on the combo box will load on text area on the right all the notes saved in the course. All Notes had some previous notes plus the one we just added, and OOP as a new course recently created has only one note.

5.7 Screen 4 – Editing Notes



This is OOP



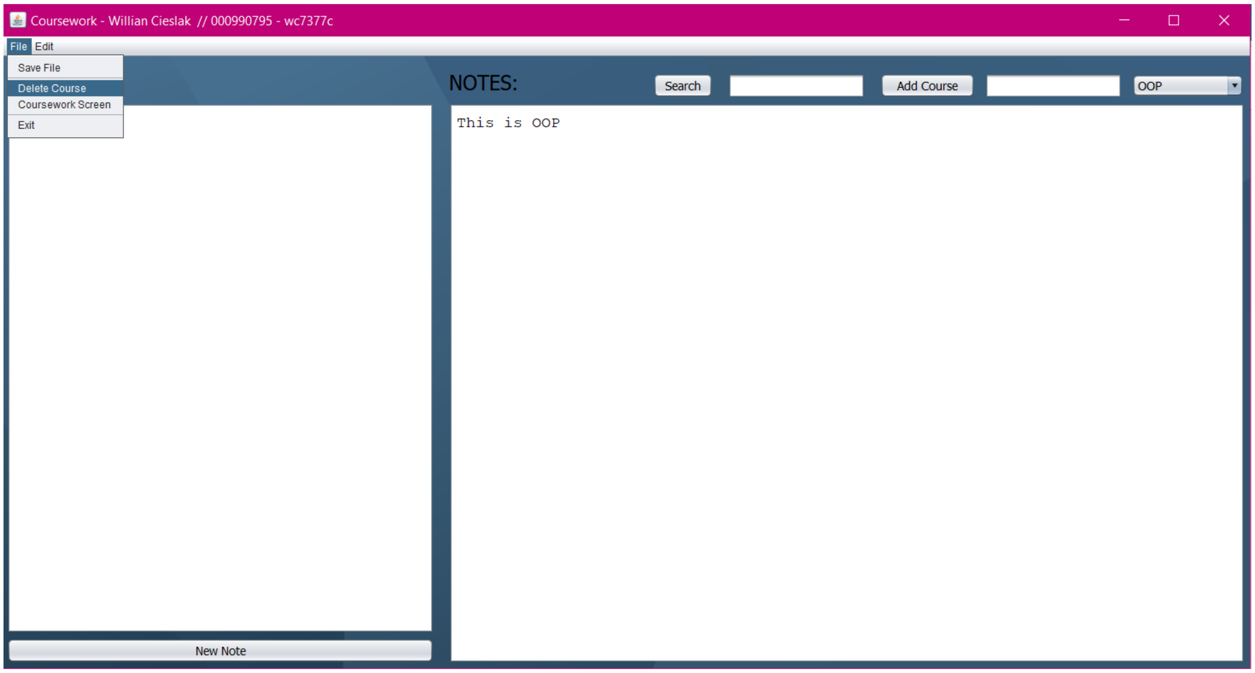
This is OOP, sometimes is confusing.

5.8 Screen 4 – A brief description

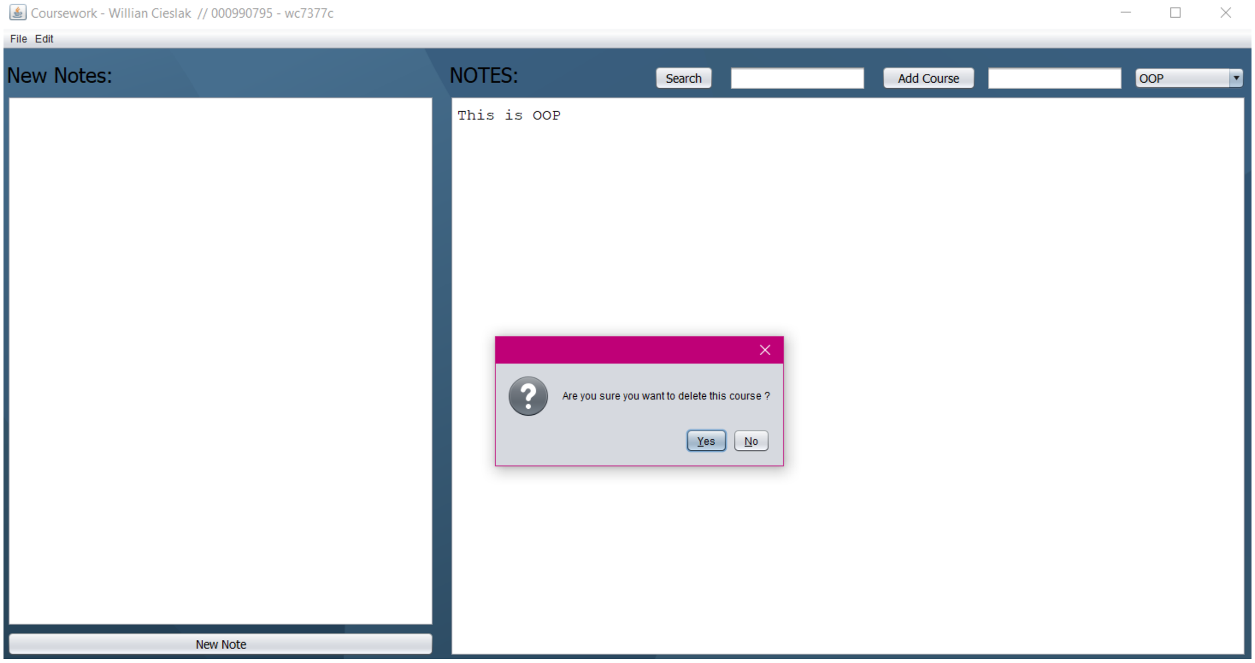
The field that displays notes is no editable. In order to edit notes you should go to Edit > Edit text, that will allow you to make the changes that you want. When you are done, click on File > Save file, and all your changes will be saved into the selected course.

Note: If you have multiple notes, Editing the file will erase all previous Notes ID and save everything under a new id for all the notes. e.g. (You have 3 notes with the ID 1, 2 and 3. Any changes made to the notes will save notes 1, 2 and 3 under a single note ID 4).

5.9 Screen 5 – Deleting a Course

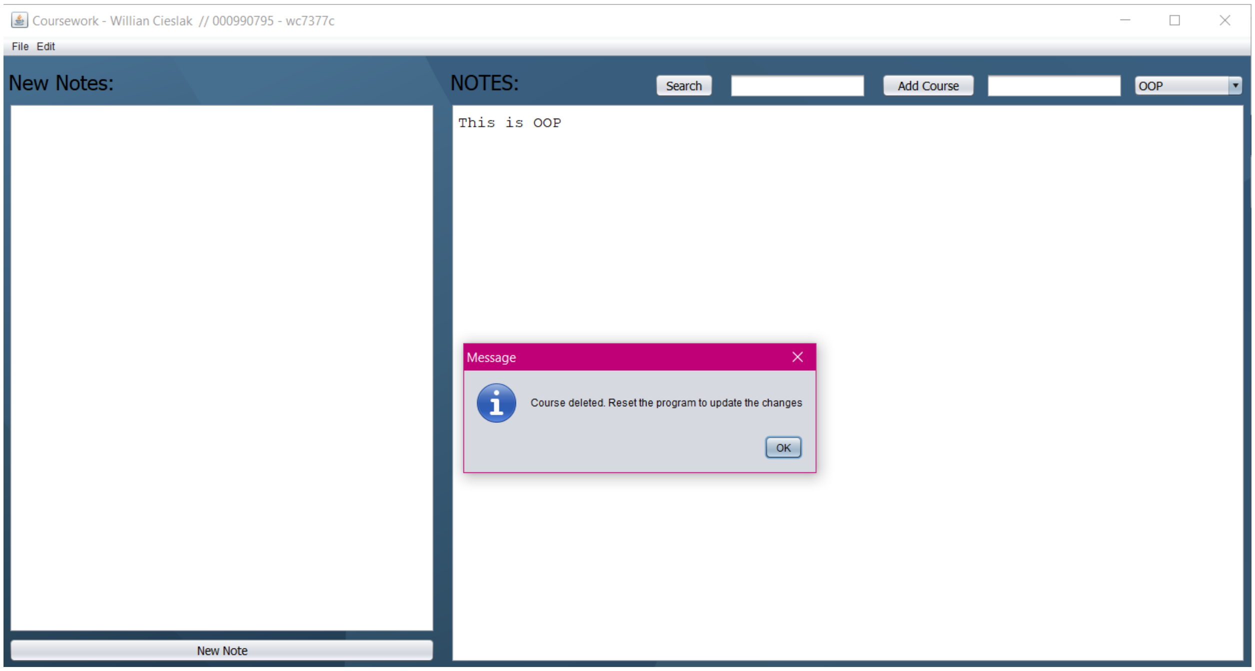


This is OOP



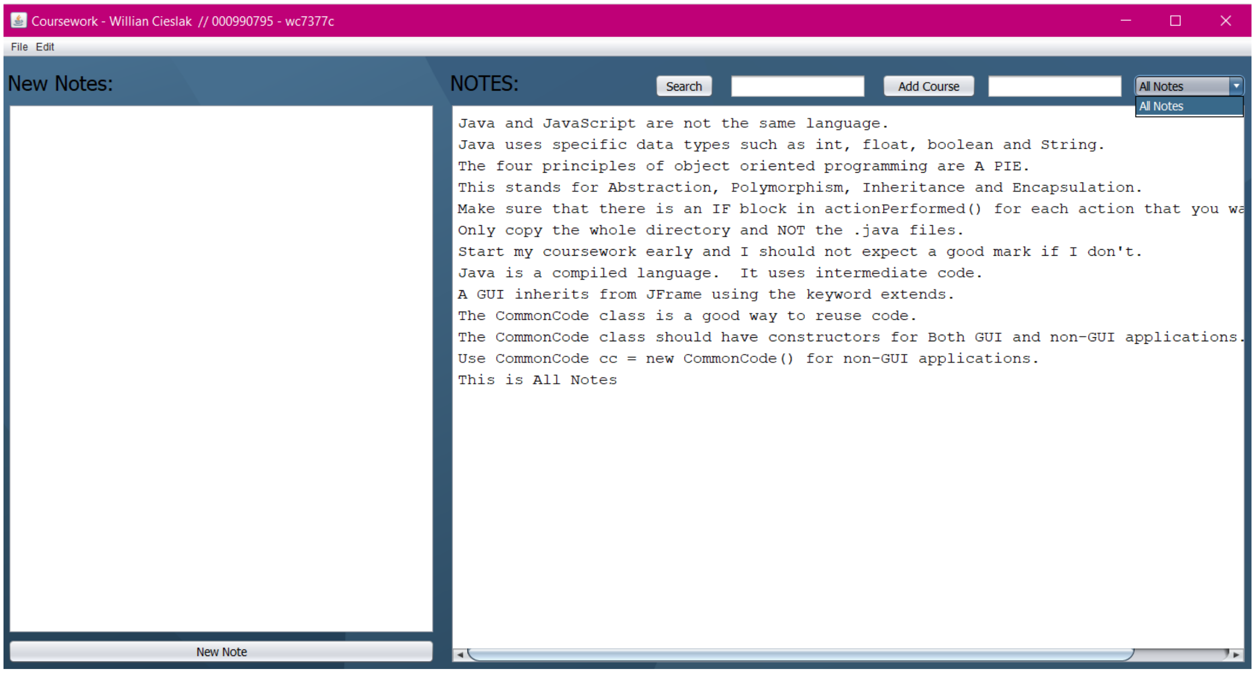
This is OOP

Are you sure you want to delete this course ?



```
This is OOP
```

Course deleted. Reset the program to update the changes

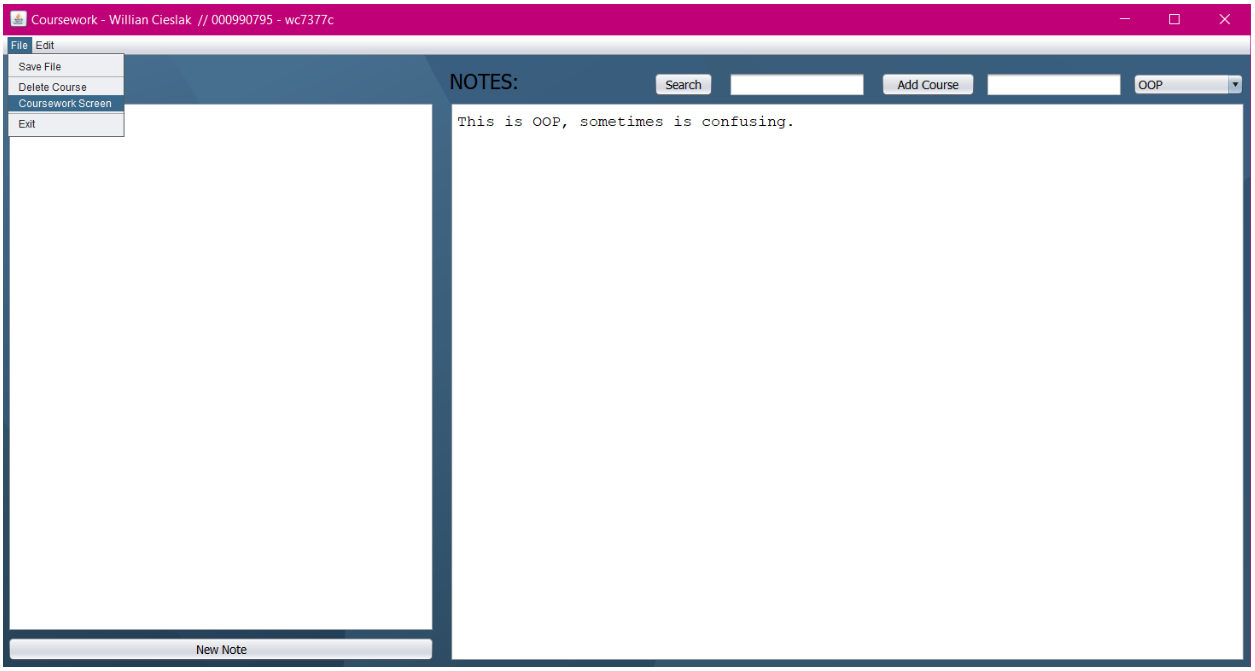


```
Java and JavaScript are not the same language.
Java uses specific data types such as int, float, boolean and String.
The four principles of object oriented programming are A PIE.
This stands for Abstraction, Polymorphism, Inheritance and Encapsulation.
Make sure that there is an IF block in actionPerformed() for each action that you wa
Only copy the whole directory and NOT the .java files.
Start my coursework early and I should not expect a good mark if I don't.
Java is a compiled language.  It uses intermediate code.
A GUI inherits from JFrame using the keyword extends.
The CommonCode class is a good way to reuse code.
The CommonCode class should have constructors for Both GUI and non-GUI applications.
Use CommonCode cc = new CommonCode() for non-GUI applications.
This is All Notes
```

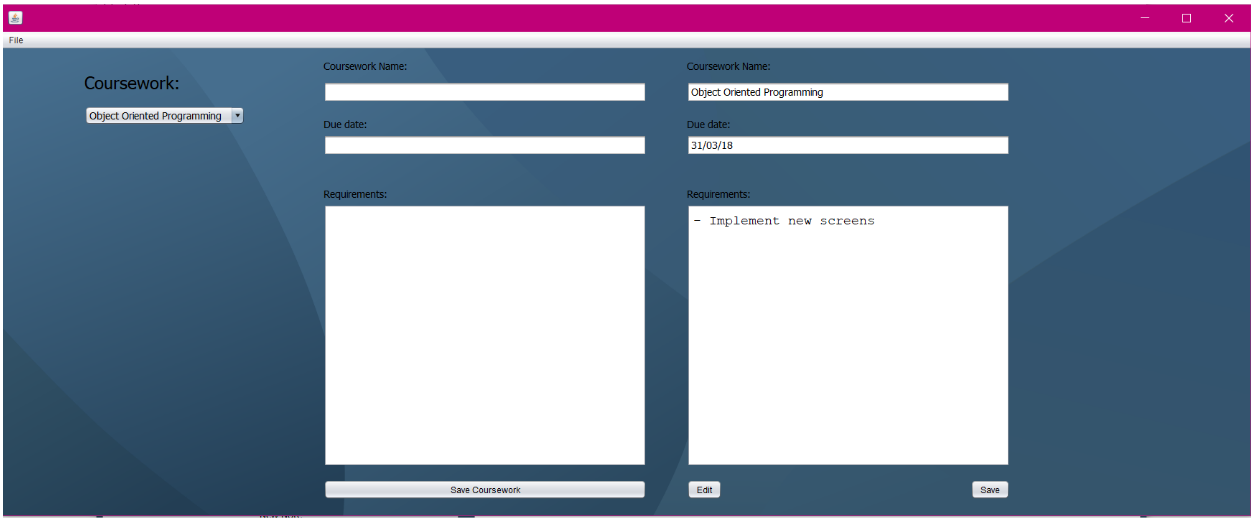
5.10 Screen 5 – A brief description

File > Delete Course will delete the selected course on the combo box menu. Reset the program will be needed to update the list of courses.

5.11 Screen 6 – Opening Coursework Screen



```
This is OOP, sometimes is confusing.
```



Object Oriented Programming

Coursework Name:

Due date:

Requirements:

Coursework Name: Object Oriented Programming

Due date: 31/03/18

Requirements:

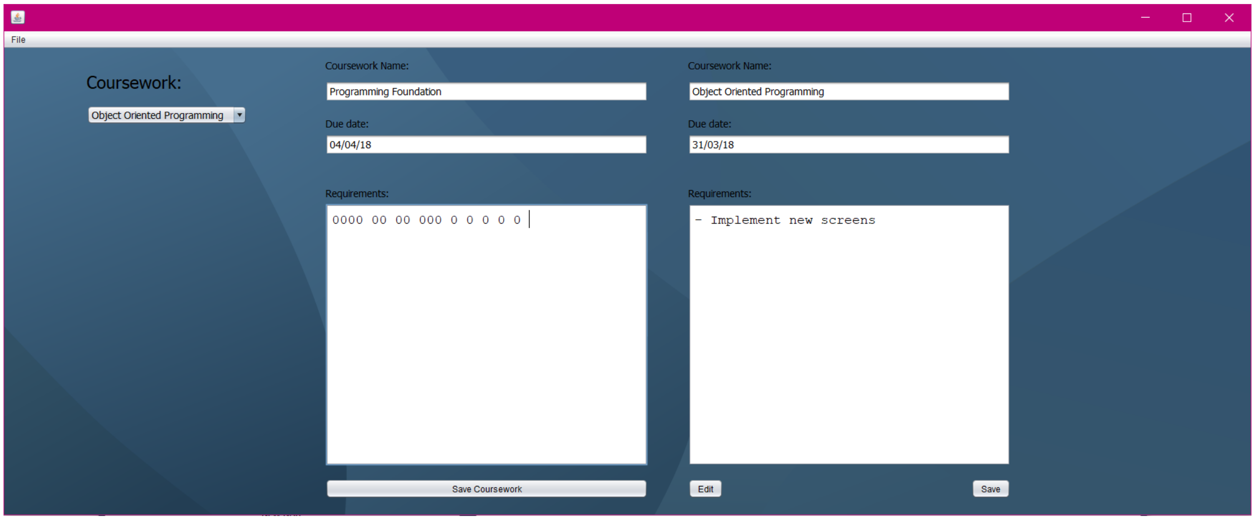
```
- Implement new screens
```

5.12 SCREEN 6 – A BRIEF DESCRIPTION

File > Coursework Screen will open the Coursework Screen, where all your courseworks are crated and displayed.

As it happens from the notes, the coursework selected on the combo box will load all the details on the right side of the screen.

5.13 Screen 7 – Adding a Coursework



Coursework: Object Oriented Programming

Coursework Name: Programming Foundation

Due date: 04/04/18

Requirements:

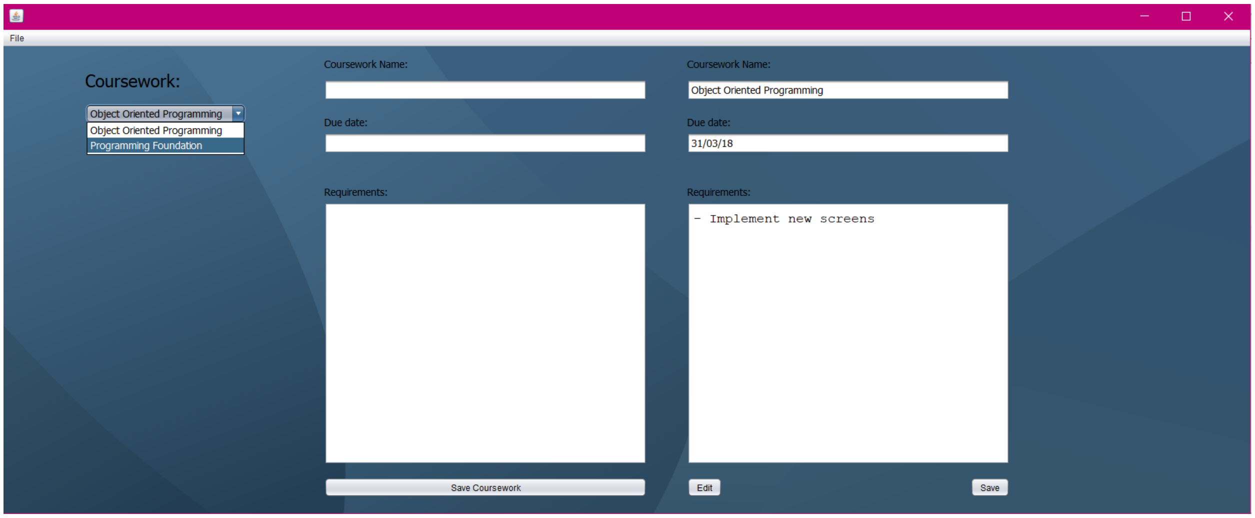
```
0000 00 00 000 0 0 0 0 0
```

Coursework Name: Object Oriented Programming

Due date: 31/03/18

Requirements:

```
- Implement new screens
```



Coursework: Object Oriented Programming

- Object Oriented Programming
- Programming Foundation

Coursework Name:

Due date:

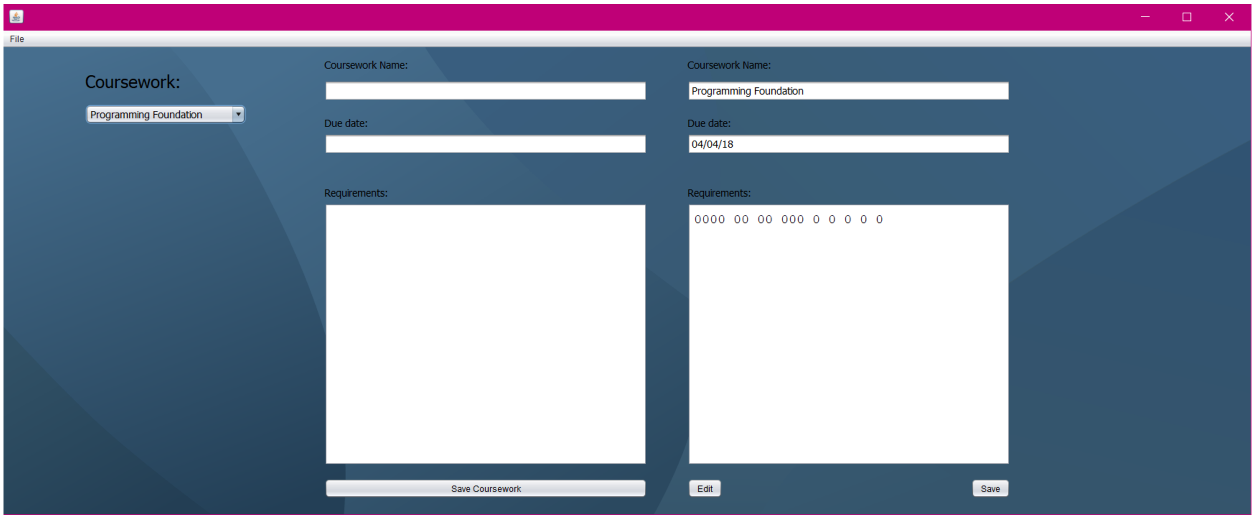
Requirements:

Coursework Name: Object Oriented Programming

Due date: 31/03/18

Requirements:

```
- Implement new screens
```



Coursework: Programming Foundation

Coursework Name:

Due date:

Requirements:

Coursework Name: Programming Foundation

Due date: 04/04/18

Requirements:

```
0000 00 00 000 0 0 0 0 0
```

5.14 Screen 7 – A brief description

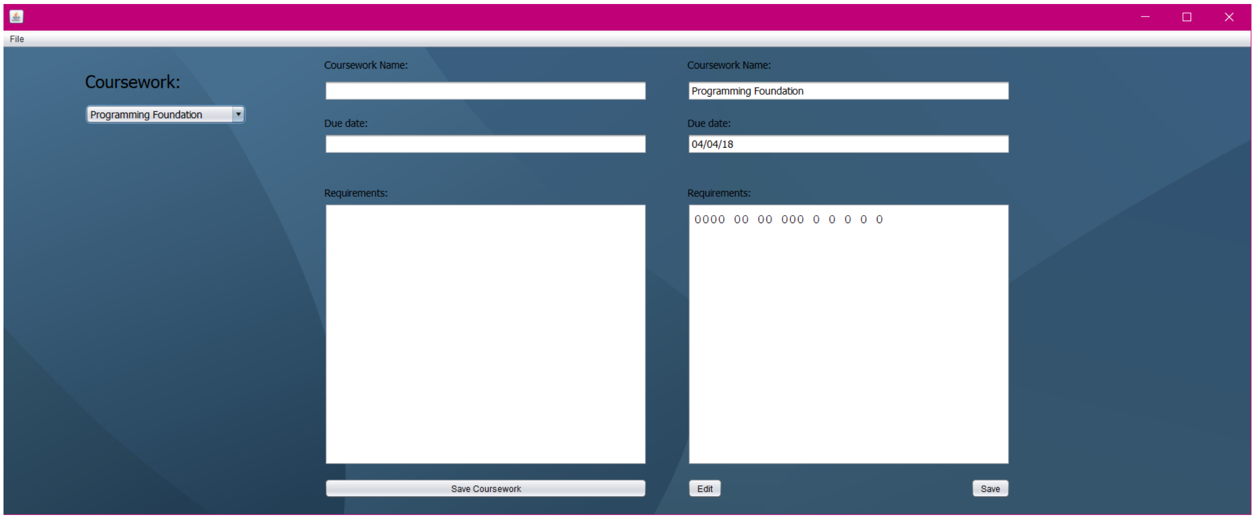
Filling out all the details in the middle and then clicking in “Save coursework” will add a new coursework to the coursework list on the combo box menu. Selecting another item on the combo box will display the details from that particular coursework.

File > Add Coursework performs the same functions that Save Coursework button.

Deleting a coursework is not implemented.

Unfortunately, the only way to save multiple requirements for the coursework is if you type everything in the same line. Typing on different lines will still save the requirements in the coursework text file but the program will only load the first line.

5.15 Screen 8 – Editing a Coursework



Coursework: Programming Foundation

Coursework Name:

Due date:

Requirements:

Coursework Name: Programming Foundation

Due date: 04/04/18

Requirements: 0000 00 00 000 0 0 0 0 0

5.16 Screen 8 – A brief description

All the fields on the right side are not editable. By clicking on the "Edit" button it will make the field requirements now editable. Once all the changes are done click "Save" and your changes will be saved.

6 THE EVALUATION

6.1 WHAT WENT WELL?

I believe I could visualize the problem and what I had to do to make the program work. That is one of the reason of why I chose to create another way of creating the view(), and because I have done a little bit of java before I was able to understand most of the problems I faced and came up with a (perhaps not the best) solution, that allowed me to keep moving forward. It might not be the proper way of doing it as I am aware but even with some mistakes I managed handled those problems.

6.2 WHAT WENT LESS WELL?

I spent an awful amount of time trying to find the perfect solution, things that require more experience and time. After searching on the internet and trying lots of things I realizes that it wouldn't be possible to progress the way I imagined, so I had to improvise, make it simpler, reduce expectations and make the basic work. I jumped straight to the code without having a design to follow, later I also had to spend some more time solving logical problems and removing/adapting redundant methods inside the program. Still, I didn't make the best use of OOP because I had to "copy" some code. Abstraction wasn't implemented as well because it would cause a big chaos having to modify some classes after the program was working.

6.3 WHAT WAS LEARNED?

To develop the new features, I was required to understand what the original code was doing. Also, as the problems started to occur I had to debug my code and try to spot the errors. Not only now I am able to understand more about OOP, but it makes easier from me to read other people's code. I have managed to solve and overcome most of the problems all by myself by trying, testing, debugging and searching on the internet. Now, I have a better understanding of how classes are connected with each other and how to define their behavior. Plus, encapsulation, inheritance and polymorphism made more sense when I was trying to develop it by myself.

6.4 HOW WOULD A SIMILAR TASK BE COMPLETED DIFFERENTLY?

Designing diagrams and the behavior of each classes before coding would deliver a more elegant solution, because it will give you a clear idea of how the program should behave and how you are required to code each class. With everything defined also I believe you are less likely to make mistakes if you manage to design the proper OOP way. For this coursework, one simple change I would make is create Course as an object.

6.5 HOW COULD THE COURSE BE IMPROVED?

The course could be improved by giving more challenges during the week. I know that didn’t happen for me, but I am aware that most people didn’t bother to complete de tutorial because all the code would be given afterwards. Also, the start point of the coursework is the given code. And because that is not ours is a lot harder to understand what is happening.

7 SELF-GRADING

Please assess yourself objectively for each section shown below and then enter the total mark you expect to get.

Viva 1 (3)

Not attended or no work demonstrated – 0
Work demonstrated was not up to the standard expected – 1
Work demonstrated was up to the standard expected – 2
Work demonstrated exceeded the standard expected – 3

Viva 2 (3)

Not attended or no work demonstrated – 0
Work demonstrated was not up to the standard expected – 1
Work demonstrated was up to the standard expected – 2
Work demonstrated exceeded the standard expected – 3

Viva 3 (4)

Not attended or no work demonstrated – 0
Work demonstrated was not up to the standard expected – 1
Work demonstrated was up to the standard expected – 2
Work demonstrated exceeded the standard expected – 3 or 4

For this section I think I got : 7 out of 10

Professional programming style (10)

Coding (up to 5)

Indenting has not been used – 0
Indenting has been used occasionally – 1
Indenting has been used, but not regularly – 2
Indenting has been used regularly – 3 to 4
All code has been indented correctly – 5

Naming of variables, procedure and classes (up to 5)

Professional naming has not been used – 0
Professional naming has been used occasionally – 1
Professional naming has been used, but not regularly – 2
Professional naming has been used regularly – 3 to 4
All items have been named professionally correctly – 5

For this section I think I got : 17 out of 10

Use of OOP techniques (50)

Abstraction (20)

No extra classes or objects have been created – 0
Classes and objects have been created superficially – 1 to 7
Classes and objects have been created and used correctly – 8 to 13
New and useful classes and objects have been created – 14 to 17
The use of classes and objects exceeds the specification – 18 to 20

Encapsulation (10)

No encapsulation has been used – 0
Class variables have been encapsulated superficially – 1 to 3
Class variables have been encapsulated correctly – 4 to 6
The use of encapsulation exceeds the specification – 7 to 10

Inheritance (10)

No inheritance has been used – 0
Classes have been inherited superficially – 1 to 3
Classes have been inherited correctly – 4 to 6
The use of inheritance exceeds the specification – 7 to 10

Polymorphism (10)

No polymorphism has been used – 0
A procedure has been polymorphised – 1 to 3
A procedure has been polymorphised and used appropriately – 4 to 6
The use of polymorphism exceeds the specification – 4 to 10

For this section I think I got : 25 out of 50

Testing (10)

Testing has not been demonstrated in the documentation – 0
Little white box testing has been documented – 1 to 3
White box testing has been documented for all the coursework – 4 to 6
White box testing has been documented for the whole program – 7

For this section I think I got : 5 out of 10

Evaluation (10)

No evaluation was shown in the documentation – 0
The evaluation shows a lack of thought – 1 to 3
The evaluation shows thought – 4 or 5
The evaluation shows clearly demonstrates increased awareness – 6 or 7

For this section I think I got : 7 out of 10

Documentation (10)

The documentation cannot be understood on first reading – 0
The documentation is readable, but a section(s) are missing – 1 to 3
The documentation is complete – 4 to 6
The documentation is complete and of a high standard – 7

For this section I think I got : 5 out of 10

I think my overall mark would be : 70 out of 100