SIS Project

Students Class:

```
package Lab_Project;

import java.util.ArrayList;

//model
public class Students {

            String name;
            int id;
            String major;
            String Username;
            String Password;
            ArrayList<Grade> grades = new ArrayList<Grade>();
            Grade grade;
            ArrayList<Courses> courses = new ArrayList<Courses>();


            public Students(String name, int id, String major, String username, String password, Grade grade) {
                  this.name = name;
                  this.id = id;
                  this.major = major;
                  Username = username;
                  Password = password;
                  this.grade = grade;
            }

            public Students() {
                  this.Username = "Hicham";
                  this.name = "Hicham";
                  this.id = 8998;
                  this.major = "CS";
                  this.Password = "0000";
                  ArrayList<Courses> courses_ = new ArrayList<Courses>();
                  for (int i = 0; i < 3; i++) {
                  }
                  this.grades = new ArrayList<Grade>();
                  this.courses = courses_;

            }

            public String getUsername() {
                  return Username;
            }
            public void setUsername(String username) {
                  Username = username;
            }
            public String getName() {
                  return name;
            }
            public void setName(String name) {
                  this.name = name;
```

```
        }

        public void setId(int id) {
            this.id = id;
        }
        public int getId() {
            return id;
        }

        public Grade getGrade() {
            return grade;
        }
        public void setGrade(Grade grade) {
            this.grade = grade;
        }

        public String getMajor() {
            return major;
        }
        public void setMajor(String major) {
            this.major = major;
        }

        public void setCourses(ArrayList<Courses> courses) {
            this.courses = courses;
        }
        public ArrayList<Courses> getCourses() {
            return courses;
        }

        protected String getPassword() {
            return Password;
        }
        public void setPassword(String password) {
            Password = password;
        }

        public boolean CheckUserPass(String User, String Pass) {
            if(this.Username==User && this.Password==Pass) return true;
            else return false;
        }
        public ArrayList<Grade> getGrades() {
            return grades;
        }
        public void setGrades(ArrayList<Grade> grades) {
            this.grades = grades;
        }




}
```

Courses Class:

```
package Lab_Project;



import java.util.ArrayList;

public class Courses implements Cloneable{
        String name;
        int Credits;
        String courseCode;
        String department;
        Instructor instructor;
        ArrayList<Students> students;
        ArrayList<Grade> grades;
        Grade grade;


public Courses() {
        this.name = "hoop";
        this.Credits = 69;
        this.courseCode = Credits+name;
        this.department = "snoop";
        this.instructor = null;

        ArrayList<Students> students_ = new ArrayList<Students>();

        this.students = students_;
}

public Courses(Courses course) {
        this.name = course.getName();
        this.Credits = course.getCredits();
        this.courseCode = course.getCourseCode();
        this.department = course.getDepartment ();

        this.instructor = course.getInstructor ();
}


        public Courses(String name, int number, String department, Instructor instructor) {
                this.name = name;
                this.Credits = number;
        this.courseCode = number+name;
        this.department = department;
        this.instructor = instructor;
}

protected ArrayList<Grade> getGrades() {
        return grades;
}

protected void setGrades(ArrayList<Grade> student_grades) {
        this.grades = student_grades;
}

public String getName() {
        return name;
}

public void setName(String name) {
        this.name = name;
}

public int getCredits() {
        return Credits;
}

public void setCredits(int number) {
        this.Credits = number;
}

public String getDepartment() {
        return department;
}

public void setDepartment(String department) {
        this.department = department;
}

public Grade getGrade() {
        return grade;
}

public void setGrade(Grade grade) {
        this.grade = grade;
}

public ArrayList<Students> getStudents() {
        return students;
}

public Instructor getInstructor() {
        return instructor;
}

public void setInstructor(Instructor instructor) {
        this.instructor = instructor;
    }



    public void setStudents(ArrayList<Students> students) {
        this.students = students;
    }

    public String getCourseCode() {
        return courseCode;
    }

    public void setCourseCode(String courseCode) {
        this.courseCode = courseCode;
    }

    @Override
    protected Object clone() throws CloneNotSupportedException {



        return (Courses)super.clone();
    }








}
```

Login View Class:

```
package Lab_Project;


import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
import java.awt.GridLayout;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.util.ArrayList;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
//View
public class LoginView implements ActionListener {

		//Image image = new Image(null);
				JFrame frame = new JFrame();
				JButton Login = new JButton("Login");
				JButton Reset = new JButton("Reset");

				JTextField userIDField = new JTextField();
				JPasswordField userPasswordField = new JPasswordField();
				JLabel userIDLabel = new JLabel("Username:");
				JLabel userPasswordLabel = new JLabel("Password:");
				JLabel messageLabel = new JLabel();
				static String Username;
				static String Password;

				ArrayList<Students> students;
				ArrayList<Instructor> instructors;
				ArrayList<Administrator> admin;
				ArrayList<Courses> courses;
		public LoginView(ArrayList<Students> students, ArrayList<Instructor> instructors,ArrayList<Administrator> admin_
,ArrayList<Courses> courses) {

				this.instructors = instructors;
				this.students = students;
				this.courses = courses;
				this.admin=admin_;
```

```
//                      try {
//                              frame.setContentPane(new JLabel(new
ImageIcon(ImageIO.read(getClass().getResource("pattern.png")))));
//                      } catch (IOException e) {
//                              // TODO Auto-generated catch block
//                              e.printStackTrace();
//                      }
                        frame.getContentPane().setBackground(new Color(105, 102, 103));


                        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                        frame.setBounds(0,0,600, 600);

                        frame.getContentPane().setLayout(null);
                        frame.setVisible(true);
                        userIDLabel.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));

                        userIDLabel.setBounds(50,100,75,25);
                        userPasswordLabel.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
                        userPasswordLabel.setBounds(50,150,75,25);
                        userIDField.setBounds(125,100,200,25);
                        userPasswordField.setBounds(125,150,200,25);
                        messageLabel.setBounds(125,250,250,35);
                        messageLabel.setFont(new Font(null,Font.ITALIC,25));
                        Login.setBounds(125,200,100,25);
                        Login.addActionListener(this);
                        Reset.setBounds(225,200,100,25);
                        Reset.addActionListener(this);
                        Login.setFocusable(false);
                        Reset.setFocusable(false);

                        frame.getContentPane().add(userIDLabel);
                        frame.getContentPane().add(userPasswordLabel);
                        frame.getContentPane().add(messageLabel);
                        frame.getContentPane().add(userPasswordField);
                        frame.getContentPane().add(userIDField);
                        frame.getContentPane().add(Login);
                        frame.getContentPane().add(Reset);

                        JLabel lblNewLabel = new JLabel("Student Information System");
                        lblNewLabel.setFont(new Font("Baskerville Old Face", Font.BOLD, 30));
                        lblNewLabel.setBounds(10, 11, 450, 46);
                        frame.getContentPane().add(lblNewLabel);




                }

                @Override
                public void actionPerformed(ActionEvent e) {
                        boolean found= false;
                        if(e.getSource()==Reset) {//if reset button is clicked resets text fields
                                userIDField.setText("");
                                userPasswordField.setText("");
                        }
```

```
        if(e.getSource()==Login) {//if login button clicked checks if username and password matches
            String userID = userIDField.getText();
            String userPassword = String.valueOf(userPasswordField.getPassword());

            for (int i = 0; i < instructors.size(); i++) {
                if (instructors.get(i).getUsername().equals(userID) &&
instructors.get(i).getPassword().equals(userPassword)) {
                new InstructorView(instructors.get(i), courses, instructors, students,admin);
                found = true;
          frame.dispose();
                }
   // when input credentials match with existing instructor credentials, new instructor frame created, login frame
disposed
            }

            if (found == false) { //if not found in instructors
                for (int i = 0; i < students.size(); i++) {
                    if (students.get(i).getUsername().equals(userID) &&
students.get(i).getPassword().equals(userPassword)){
                        new StudentView(students.get(i), students, courses,
instructors,admin); // next GUI....
                        System.out.println("Student view");
                        found = true;
                        frame.dispose();



                    }


            }


            for (int i = 0; i < admin.size(); i++) {
                if (admin.get(i).getUsername().equals(userID) &&
admin.get(i).getPassword().equals(userPassword)) {
                    new AdministratorView(students,instructors,courses,admin);
                     found = true;
                    frame.dispose();
                }

   // when input credentials match with existing instructor credentials, new instructor frame created, login frame
disposed
            }

            if (found == false) {
                JOptionPane.showMessageDialog(null, "Wrong Password / Username");
    userIDField.setText("");
            userPasswordField.setText("");
        }



        }

    }


}
```

```
}
```

Main Class:

```
package Lab_Project;

import java.util.ArrayList;

public class Main {

        public static void main(String[] args) {
                ArrayList<Students> students = new ArrayList<Students>();
                ArrayList<Instructor> instructors = new ArrayList<Instructor>();
                ArrayList<Administrator> admin = new ArrayList<Administrator>();
                ArrayList<Courses> courses = new ArrayList<Courses>();




                //(String name, int id, String major, String username, String password, Grade grade)
                students.add(new Students("Jawad", 88888, "CS", "Jawad", "1234", new Grade()));
                students.add(new Students("Ali", 88726, "CVE", "Ali", "1111", new Grade()));
                students.add(new Students("Rayan", 89059, "COE", "Rayan", "0000", new Grade()));
                students.add(new Students("Tameem", 88948, "CS", "Tameem", "0000", new Grade()));
                students.add(new Students("Amjad", 89059, "CS", "Amjad", "0000", new Grade()));

                //        public Instructor(String name, int id, String username, String password, String dept)
                instructors.add(new Instructor("Hicham", 80988, "Hicham", "0000", "CMP"));
                instructors.add(new Instructor("Michel", 12345, "Michel", "0000", "CMP"));
                instructors.add(new Instructor("Sofian", 99898, "Sofian", "1234", "CHM"));

                //Administrator(String Username,String Password)
                admin.add(new Administrator("Admin", "Admin"));
                LoginView Login = new LoginView(students, instructors, admin,courses);



        }

        public static Students GetInfo() {
                Students student=new Students();
                student.setName("jawad");
                student.setId(87996);
                student.setPassword("jawad123");
                student.setUsername("JawadAdas");

                return student;
        }



}
```

Grades Class:

```
package Lab_Project;

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


public class Grade {
    String letter;
    double grade;

    public Grade() {
        this.letter = "";
        this.grade = 0;
    }

    public Grade(double grade) {
        this.grade = grade;
    }

    public String getLetter() {
        return letter;
    }

    public void setLetter(String letter) {
        this.letter = letter;
    }

    public double getGrade() {
        return grade;
    }

    public void setGrade(double grade) {
        this.grade = grade;
    }

    // ex1 b

    public void roundGrade() {
        int intPart = (int) this.getGrade();
        double decimal = this.getGrade() - intPart; // or use Tokenizer and Wrapper
        if (decimal>=0.5) {
                this.setGrade(Math.ceil(this.getGrade()));
        }else {
                this.setGrade( Math.floor(this.getGrade()));
        }


    }
    public void computeGrade() {
        this.roundGrade();
        if (grade <=59.00){
                this.setLetter("F");
        }
        else if (grade<=69.00) {
                this.setLetter("D");
        }
        else if (grade<=79.00) {
                this.setLetter("C");
        }
        else if (grade <=89.00){
                this.setLetter("B");
        }
        else if (grade<=100.00) {
                this.setLetter("A");
        }

    }
```

```
        @Override
        public String toString() {
                return "LetterGrade=" + letter + ", NumericValue=" + grade ;
        }



}
```

InstructorView.java

```
package Lab_Project;


import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Date;
import java.util.InputMismatchException;
import java.util.Scanner;
import java.util.StringTokenizer;

import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JPopupMenu;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

public class InstructorView{

        JFrame frame = new JFrame();
    JPanel p2 = new JPanel();

    Date dt = new Date();
    String Semester=null;

    Instructor instructor;
    ArrayList<Courses> courses;
    ArrayList<Students> students;
    ArrayList<Instructor> instructors;

    //panel for changing password
    JPanel passpanel = new JPanel(new GridLayout(3,2));
    JPanel namepanel = new JPanel(new GridLayout(1,2));

        JLabel oldpasslabel = new JLabel("Enter Old Password:");
        JLabel newpass1label = new JLabel("Enter New Password:");
        JLabel newpass2label = new JLabel("Renter New Password:");

        JPasswordField oldpassField = new JPasswordField( );
        JPasswordField newpass1Field = new JPasswordField();
        JPasswordField newpass2Field = new JPasswordField();

        JButton confirmpassbutton = new JButton("Change password");

        JLabel newnamelabel = new JLabel("Enter New Name:");
        JTextField newnameField = new JTextField();

        //Used in  Change Grade Event Handeler
        Students tmpStd;
        Grade tmpGrd;

        ArrayList<Administrator> admin;

   public InstructorView(Instructor instructor_, ArrayList<Courses> courses_, ArrayList<Instructor>
_instructors, ArrayList<Students> _students,ArrayList<Administrator> admin_) {

                this.instructor = instructor_;
                this.courses = courses_;
                this.admin=admin_;
                this.instructors = _instructors;
                this.students = _students;
                frame.getContentPane().setBackground(new Color(105, 102, 103));
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.setSize(420,420);
                frame.setBounds(700, 250, 420, 420);
                frame.getContentPane().setLayout(null);
                frame.setVisible(true);

                //Creating a Label and setting its colors, font and bounds
                JLabel welcome = new JLabel("Instructor Page");
                welcome.setForeground(new Color(237, 222, 222));
                welcome.setFont(new Font("Bookman Old Style", Font.PLAIN, 19));
                welcome.setBounds(135, 11, 200, 35);

                //Creating MenuBar, Menus, and MenuItems. And naming them
                JMenuBar Menubar = new JMenuBar();
                JMenu Courses = new JMenu("Manage Courses");
                JMenu Personal = new JMenu("Personal Information");

                final JMenuItem AddCourse = new JMenuItem("Add Courses");
                final JMenuItem SaveCourseData = new JMenuItem("Save Records");
                JMenuItem ChangeUser = new JMenuItem("Change Username");
                JMenuItem ChangePass = new JMenuItem("Change Password");
                JMenuItem Logout = new JMenuItem("Logout");
                JMenuItem ChangeGrade = new JMenuItem("Change Grade");
                JMenuItem ReadCourseFile = new JMenuItem("Read Course File");


                //Changing Colors, fonts, and font colors of all menus, menu items, and the menu bar

                Menubar.setBackground(new Color(105, 102, 103));

                Courses.setFont(new Font("Bookman Old Style", Font.PLAIN, 13));
                Courses.setForeground(new Color(237, 222, 222));

                Logout.setFont(new Font("Bookman Old Style", Font.PLAIN, 13));
                Logout.setForeground(new Color(237, 222, 222));
                Logout.setBackground(new Color(105, 102, 103));

                Personal.setFont(new Font("Bookman Old Style", Font.PLAIN, 13));
                Personal.setForeground(new Color(237, 222, 222));

                AddCourse.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
                AddCourse.setForeground(new Color(237, 222, 222));
                AddCourse.setBackground(new Color(172, 83, 83));
```

```
        SaveCourseData.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
        SaveCourseData.setForeground(new Color(237, 222, 222));
        SaveCourseData.setBackground(new Color(172, 83, 83));

        ChangeUser.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
        ChangeUser.setForeground(new Color(237, 222, 222));
        ChangeUser.setBackground(new Color(172, 83, 83));

        ChangePass.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
        ChangePass.setForeground(new Color(237, 222, 222));
        ChangePass.setBackground(new Color(172, 83, 83));

        ChangeGrade.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
        ChangeGrade.setForeground(new Color(237, 222, 222));
        ChangeGrade.setBackground(new Color(172, 83, 83));

        ReadCourseFile.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
        ReadCourseFile.setForeground(new Color(237, 222, 222));
        ReadCourseFile.setBackground(new Color(172, 83, 83));


        //Adding Menu bar, Menus and Menu Items
        Courses.add(SaveCourseData);
        Courses.add(AddCourse);
        Courses.add(ChangeGrade);
        Courses.add(ReadCourseFile);
        Personal.add(ChangeUser);
        Personal.add(ChangePass);
        Menubar.add(Personal);
        Menubar.add(Courses);
        Menubar.add(Logout);


        frame.setJMenuBar(Menubar);
        frame.getContentPane().add(welcome);

        JButton changepass = new JButton("Change Password");
        JButton changename = new JButton("Change Name");
        JButton CourseData = new JButton("Course Data");
    JButton DisplayRecordsBtn = new JButton("Display Records");

    JLabel lblSelfserviceSystem = new JLabel("Self-Service System:");
        JLabel lblPersonalInformation = new JLabel("Personal Information:");
        lblSelfserviceSystem.setFont(new Font("Bookman Old Style", Font.PLAIN, 16));
        lblPersonalInformation.setFont(new Font("Bookman Old Style", Font.PLAIN, 16));

        lblSelfserviceSystem.setForeground(new Color(237, 222, 222));
        lblPersonalInformation.setForeground(new Color(237, 222, 222));

        lblPersonalInformation.setBounds(200, 75, 194, 29);
        lblSelfserviceSystem.setBounds(10, 75, 166, 29);

        frame.getContentPane().add(lblPersonalInformation);
        frame.getContentPane().add(lblSelfserviceSystem);

    DisplayRecordsBtn.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
    DisplayRecordsBtn.setForeground(new Color(237, 222, 222));
    DisplayRecordsBtn.setBackground(new Color(172, 83, 83));
    DisplayRecordsBtn.setBounds(10, 115, 156, 33);

    CourseData.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
    CourseData.setForeground(new Color(237, 222, 222));
    CourseData.setBackground(new Color(172, 83, 83));
    CourseData.setBounds(10, 165, 156, 33);


    changename.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
    changename.setForeground(new Color(237, 222, 222));
    changename.setBackground(new Color(172, 83, 83));
    changename.setBounds(200, 115, 156, 33);

    changepass.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
    changepass.setForeground(new Color(237, 222, 222));
    changepass.setBackground(new Color(172, 83, 83));
    changepass.setBounds(200, 165, 156, 33);

    frame.getContentPane().add(DisplayRecordsBtn);
        frame.getContentPane().add(CourseData);
        frame.getContentPane().add(changename);
        frame.getContentPane().add(changepass);
frame.setVisible(true);



if(dt.~~getMonth~~()+1>=1 && dt.~~getMonth~~()<=5) Semester ="Spring";
        if(dt.~~getMonth~~()+1>=9 && dt.~~getMonth~~()<=12) Semester ="Fall";
        if(dt.~~getMonth~~()+1>=6 && dt.~~getMonth~~()<=8) Semester ="Summer";

        ReadCourseFile.addActionListener(new ActionListener() {
```

```
                        public void actionPerformed(ActionEvent e) {
                                JFileChooser LoadCourse = new JFileChooser();
                                LoadCourse.setApproveButtonText("Select File");
                                LoadCourse.setAcceptAllFileFilterUsed(false);
                                int choice = LoadCourse.showOpenDialog(AddCourse);
                                if (choice == JFileChooser.APPROVE_OPTION) {

                                        File selectedFile = LoadCourse.getSelectedFile();
                                        System.out.println("Selected file: " +
selectedFile.getAbsolutePath());
             //FileReader reader = null;
                try{
                        Scanner in = new Scanner(selectedFile);

                        String data = "";

                        while(in.hasNextLine() != false) {

                        data = data + in.nextLine() + "\n";

                        }

                        //Dividing the entire text into lines
                        String tb_data[] = data.split("\n");
                                String row[][] = new String[tb_data.length][6];

                                //Column Data
                        String[] row1 = tb_data[0].split(";");

                        System.out.println(tb_data[0]);

                        for (int i = 0; i < 6 ; i++)
                        {
                        System.out.print(row1[i]);

                        }

                        //The block of text is saved into a 2d array
                        for (int i = 1; i < tb_data.length; i++) {
                                StringTokenizer st = new StringTokenizer(tb_data[i], ";");

                                int j = 0;

                                while (st.hasMoreTokens()) {
                                        row[i-1][j] = st.nextToken();
                                System.out.print(row[i-1][j]+" ");
```

```
                    j++;
                }
            System.out.println();

            }

            DefaultTableModel tablemodel = new DefaultTableModel(row,row1);



            JTable table = new JTable(tablemodel);
            JScrollPane pane = new JScrollPane(table);
            JPanel panel = new JPanel();
            panel.add(pane);

            JOptionPane.showOptionDialog(null, panel, "Course File",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                        {}, null);

                in.close();

        } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
        }

            }
            }
            });



    Logout.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                frame.dispose();

                new LoginView(students, instructors, admin,courses);

            }
            });



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

                JPanel CoursesPanel = new JPanel(new BorderLayout());
```

```
                              JLabel lbl = new JLabel("Which course would you like to change the
grades of?");

                              JPanel savecoursesPanel = new JPanel(new
GridLayout(instructor.getCourses().size(),1));

                              for (int i = 0; i < instructor.getCourses().size();i++) {
                                   JButton courseButton = new
JButton(instructor.getCourses().get(i).getName() + instructor.getCourses().get(i).getCourseCode());

                                   final Courses course = instructor.getCourses().get(i);
                                   savecoursesPanel.add(courseButton);


                                   courseButton.addActionListener(new ActionListener() {
                                   public void actionPerformed(ActionEvent e) {
                                        JPanel studentsPanel = new JPanel(new
GridLayout(course.getStudents().size(),1));
                                        JLabel selectStudent = new JLabel("Select the
Student you would like to change the grade for");

                                        studentsPanel.add(selectStudent);
                                        for (int i = 0; i < course.getStudents().size(); i++) {

                                             for (int j = 0; j <
course.getStudents().get(i).getCourses().size(); j++) {
                                                  if
(course.getStudents().get(i).getCourses().get(j) == course) {
                                                       tmpStd =
course.getStudents().get(i);
                                                       tmpGrd =
tmpStd.getGrades().get(j);
                                                       JButton studentButton = new
JButton(tmpStd.getName() + " Grade: " +
                                                  tmpStd.getGrades().get(j).getGrade());



studentButton.addActionListener(new ActionListener() {
                                                       public void
actionPerformed(ActionEvent e) {
                                                            JPanel
changegradePanel = new JPanel(new GridLayout(2,2));
```

```
                                                                              JLabel currentGrade =
new JLabel("Current Grade: " + tmpGrd.getGrade());
                                                                              JLabel newGrade =
new JLabel("Enter New Grade");
                                                                              JTextField
newGradeField = new JTextField();


changegradePanel.add(currentGrade);
                                                                      changegradePanel.add(new
JLabel());

changegradePanel.add(newGrade);

changegradePanel.add(newGradeField);



                                                                int a =
JOptionPane.showOptionDialog(null, changegradePanel, "Change Grade of " + tmpStd.getId(),
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                                              {"Confirm", "Cancel"},
null);

                                                                      if (a ==
JOptionPane.OK_OPTION) {
                                                                              try {

tmpGrd.setGrade(Double.parseDouble(newGradeField.getText()));
                                                                              }

catch(NumberFormatException e1) {

JOptionPane.showMessageDialog(null, "Enter a valid
number","Error",JOptionPane.INFORMATION_MESSAGE);


                                                                              }
                                                                      }
                                                                              }

                                                                              }
                                                                              );



studentsPanel.add(studentButton);
```

```
                                    }
                                }
                            }
                            JOptionPane.showOptionDialog(null,
studentsPanel, "Change Grade", JOptionPane.DEFAULT_OPTION,
JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                    {}, null);
                        }
                        });
                }
                CoursesPanel.add(lbl,BorderLayout.NORTH);
                CoursesPanel.add(savecoursesPanel,BorderLayout.SOUTH);


                JOptionPane.showOptionDialog(null, CoursesPanel, "Change
Grades", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                        {}, null);

            }
            });


  //EventHandler for Saving Course Data menu item that saves the selected course information
into a text file
        SaveCourseData.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                JPanel CourseDataPanel = new JPanel(new BorderLayout());


                JLabel lbl = new JLabel("Choose a Course to Save");

                JPanel coursesPanel = new JPanel(new
GridLayout(instructor.getCourses().size(),1));

                for (int i = 0; i < instructor.getCourses().size();i++) {
                JButton courseButton = new
JButton(instructor.getCourses().get(i).getName() + instructor.getCourses().get(i).getCourseCode());

                final Courses course = instructor.getCourses().get(i);
```

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

                                    JFileChooser DataFile = new JFileChooser(".");
                        int choice = DataFile.showSaveDialog(SaveCourseData);//Selects file
                        if (choice == JFileChooser.APPROVE_OPTION) {
                          File selectedFile = DataFile.getSelectedFile();
                           System.out.println("Selected file: " + selectedFile.getAbsolutePath());

                             try{
                               FileWriter writer = new FileWriter(selectedFile);//Creates FileWriter to
display records
                               writer.write("Name;"+instructor.getName() + ";ID; "+instructor.getId() +
";Department;" + instructor.getDept() + ";\n");//adds name to the text file
                               writer.write("Term;Fall2022;\n");//add ID to the text file

writer.write("Course:;"+course.getCourseCode()+";"+course.getName()+";\n");
                               writer.write("ID;Name;Grade"+";\n");


                               if(course.getStudents()!=null) {
                                     for(int j=0; j < course.getStudents().size(); j++) {//this for loop adds
all courses registered by instructor one by one
                                     writer.write(course.getStudents().get(j).getId() + " ; " +
course.getStudents().get(j).getName() + " ; " + course.getStudents().get(j).getGrade().getGrade());

                                     }
                               }
                               writer.close();
                             } catch (IOException e1) {
                               e1.printStackTrace();
                             }
                        }



                                    }
                                    });

                                    coursesPanel.add(courseButton);
                                    }



                                    CourseDataPanel.add(lbl,BorderLayout.NORTH);
                                    CourseDataPanel.add(coursesPanel,BorderLayout.SOUTH);
```

```
                                        JOptionPane.showOptionDialog(null, CourseDataPanel, "Save
Course", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                        {}, null);

                                }
                        });

        //Creates and displays PopupMenu on click
        CourseData.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ev) {

                        JPanel panel = new JPanel(new GridLayout(instructor.getCourses().size()+1,1));
                        panel.add(new JLabel("Choose a Course to Display"));
                        //Adding menu items to the popup
                        if(!(instructor.getCourses()==null)) {
                                for (int i = 0; i < instructor.getCourses().size();i++) {
                          JButton courseButton = new JButton(instructor.getCourses().get(i).getName() + " "
+ instructor.getCourses().get(i).getCourseCode());

                          final Courses course = instructor.getCourses().get(i);

                        //Display course data on menu item click
                        courseButton.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent ev) {

                                DefaultTableModel tableModel = new DefaultTableModel(new
String[]{"Name", instructor.getName(), "ID", String.valueOf(instructor.getId()), "Department",
instructor.getDept()},0);
                                JTable courseTable = new JTable(tableModel);

                                JScrollPane tablePane = new JScrollPane(courseTable);
                                JPanel tablePanel = new JPanel(new GridLayout(1,1,0,0));

                                tablePanel.add(tablePane);
                                tableModel.addRow(new String[] {"Term", "Fall 2022","","","",""});
                                tableModel.addRow(new String[]
{"Course",course.getCourseCode(),course.getName(),"","",""});

                                tableModel.addRow(new String[] {"ID","Name","Grade","","",""});

                                if (course.getStudents() != null) {
                                for (int j = 0; j < course.getStudents().size(); j ++) {

                                        for(int c = 0; c < course.getStudents().get(j).getCourses().size(); c++)
{
```

```
                                        if (course.getStudents().get(j).getCourses().get(c) == course)
                            tableModel.addRow(new String[]
{String.valueOf(course.getStudents().get(j).getId()),course.getStudents().get(j).getName(),String.valu
eOf(course.getStudents().get(j).getGrades().get(c).getGrade()),"","",""});
                                  }
                            }
                            }



                            JOptionPane.showOptionDialog(null, tablePanel, "Course Data",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                        {}, null);


                      }
                });


                         panel.add(courseButton);
                            }
                      }
                      JOptionPane.showOptionDialog(null, panel, "Course Data",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                  {}, null);


                }
                });


    //Display the Records for the instructor on a table using JOptionPane
    DisplayRecordsBtn.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ev) {

                      DefaultTableModel tableModel = new DefaultTableModel(new String[]{"Name",
instructor.getName(), "ID", String.valueOf(instructor.getId()), "Department", instructor.getDept()},0);
                      JTable instructorTable = new JTable(tableModel);

                      JScrollPane tablePane = new JScrollPane(instructorTable);
                      JPanel tablePanel = new JPanel(new GridLayout(1,1,0,0));


                      tableModel.addRow(new String[] {"Term", "Fall 2022"});
                      tableModel.addRow(new String[] {"Courses"});
                      tableModel.addRow(new String[] {"Name", "Number"});
                      if(!(instructor.getCourses()==null)) {
                            for (int i = 0; i < instructor.getCourses().size(); i++) {

                            tableModel.addRow(new String[] {instructor.getCourses().get(i).getName(),
instructor.getCourses().get(i).getCourseCode()});
```

```
                        }
                }
                tablePanel.add(tablePane);
                JFrame f = new JFrame();


                JOptionPane.showOptionDialog(null, tablePanel, "Instructor Data",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                {}, null);



        }
        });

  //ActionListener for Instructor Change password
  changepass.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ev) {


                        passpanel.add(oldpasslabel);
                        passpanel.add(oldpassField);
                        passpanel.add(newpass1label);
                        passpanel.add(newpass1Field);
                        passpanel.add(newpass2label);
                        passpanel.add(newpass2Field);

                        //Pop up pane without any icon or option dialogue, just the contents of
passpanel
                        int a = JOptionPane.showOptionDialog(null, passpanel, "Change Password",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                        {"Confirm", "Cancel"}, null);


                        if (a == JOptionPane.OK_OPTION) {
                                //Compare current password with entered old password and compare
new passwords for double entry data verification
                                if
(String.valueOf(oldpassField.getPassword()).equals(instructor.getPassword())
                                &&
String.valueOf(newpass1Field.getPassword()).equals(String.valueOf(newpass2Field.getPassword())
))
                {
                        instructor.setPassword(String.valueOf(newpass1Field.getPassword()));
                        JOptionPane.showMessageDialog(null, "Password Changed Successfully");
                        System.out.println(instructor.getPassword());
                }
                else {
                        JOptionPane.showMessageDialog(null, "Incorrect Old password / New
Passwords");
                }

                        }

        }

        });


  //Event Handler for changename Button
  changename.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent ev) {

                namepanel.add(newnamelabel);
                namepanel.add(newnameField);

                int a = JOptionPane.showOptionDialog(null, namepanel, "Change Name",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                {"Confirm", "Cancel"}, null);

                if (a == JOptionPane.OK_OPTION) {
                        if(newnameField.getText().equals("")) //Check if name is entered
                                JOptionPane.showMessageDialog(null, "Please Enter a Name");
                        else {
                        instructor.setName(newnameField.getText());
                        JOptionPane.showMessageDialog(null, "Name Changed Successfully");
                }

                }

                }

        });

 //EventHandler for Change Username Menu that will let Instructor change their usernames
                ChangeUser.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {

                                namepanel.add(newnamelabel);
                                namepanel.add(newnameField);
                int a = JOptionPane.showOptionDialog(null, namepanel, "Change
Username", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                    {"Confirm", "Cancel"}, null);

                if (a == JOptionPane.OK_OPTION) {
                    if(newnameField.getText().equals("")) //Check if name is
entered
                        JOptionPane.showMessageDialog(null, "Please Enter
new Username");
                    else {
                    instructor.setUsername(newnameField.getText());
                    JOptionPane.showMessageDialog(null, "Username Changed
Successfully");
                    }

                }
            }
        });




        //EventHandler for Change Password menu item that will let Instructor change their
passwords
        ChangePass.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                //setting up the pop up panel when Change password is clicked
                passpanel.add(oldpasslabel);
                passpanel.add(oldpassField);
                passpanel.add(newpass1label);
                passpanel.add(newpass1Field);
                passpanel.add(newpass2label);
                passpanel.add(newpass2Field);

                //Pop up pane without any icon or option dialogue, just the contents of
passpanel
                int a = JOptionPane.showOptionDialog(null, passpanel, "Change
Password", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                {"Confirm", "Cancel"}, null);


                if (a == JOptionPane.OK_OPTION) {
                    //Compare current password with entered old password and
compare new passwords for double entry data verification
                    if
(String.valueOf(oldpassField.getPassword()).equals(instructor.getPassword())
                                                    &&
(String.valueOf(newpass1Field.getPassword()).equals(String.valueOf(newpass2Field.getPassword()))
))
                        {

instructor.setPassword(String.valueOf(newpass1Field.getPassword()));
                                JOptionPane.showMessageDialog(null, "Password Changed
Successfully");
                                System.out.println(instructor.getPassword());

                        }
                        else {
                                JOptionPane.showMessageDialog(null, "Incorrect Old password /
New Passwords");
                        }

                                }
                        }
                });


                //Event handler for AddCourse menuItem for the instructor to add a new course
                AddCourse.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {

                                JPanel CoursePanel= new JPanel(new GridLayout(3,2));

                                JLabel CourseNumberLabel = new JLabel("Enter Course Number");
                                JTextField CourseNumberField = new JTextField();

                                JLabel CourseNameLabel = new JLabel("Enter Course Name");
                                JTextField CourseNameField = new JTextField();

                                JLabel CourseCreditsLabel = new JLabel("Enter Course Credits");
                                JTextField CourseCreditsField = new JTextField();


                                CoursePanel.add(CourseNumberLabel);
                                CoursePanel.add(CourseNumberField);
                                CoursePanel.add(CourseNameLabel);
                                CoursePanel.add(CourseNameField);
                                CoursePanel.add(CourseCreditsLabel);
                                CoursePanel.add(CourseCreditsField);

                                int a = JOptionPane.showOptionDialog(null, CoursePanel, "Add New
Course", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
```

```
                              {"Confirm", "Cancel"}, null);


                              if (a == JOptionPane.OK_OPTION) {

                                   Courses tmp = new Courses();
                                   try {

tmp.setCredits(Integer.parseInt(CourseCreditsField.getText()));
                                   tmp.setDepartment(instructor.getDept());
                                   tmp.setName(CourseNameField.getText());
                                   tmp.setInstructor(instructor);

tmp.setCourseCode(instructor.getDept()+CourseNumberField.getText());

                                   if (instructor.getCourses().size() < 3 &&
!(instructor.getCourses()==null)) {
                                   courses.add(tmp);
                                   instructor.getCourses().add(tmp);
                                   }
                                   else {

JOptionPane.showMessageDialog(null, "Only a Maximum of 3 Courses can be Added");
                                   }
                                   }
                                   catch(NumberFormatException e2) {

JOptionPane.showMessageDialog(null, "Invalid Data Entries");

                                   }

                              }


               }
               });

}
}
```

Student View Class

```
package Lab_Project;


import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;

import java.awt.Font;
import java.awt.GridLayout;

import javax.swing.JButton;
import javax.swing.JFileChooser;

import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
import java.awt.event.ActionEvent;
import java.awt.Color;

public class StudentView extends JFrame {

        JFrame studentsFrame = new JFrame();


        JPanel passpanel = new JPanel(new GridLayout(3,2));
  JPanel namepanel = new JPanel(new GridLayout(1,2));

        JLabel oldpasslabel = new JLabel("Enter Old Password:");
        JLabel newpass1label = new JLabel("Enter New Password:");
        JLabel newpass2label = new JLabel("Renter New Password:");

        JPasswordField oldpassField = new JPasswordField();
        JPasswordField newpass1Field = new JPasswordField();
        JPasswordField newpass2Field = new JPasswordField();

        JButton confirmpassbutton = new JButton("Change password");

        JLabel newnamelabel = new JLabel("Enter New Name:");
        JTextField newnameField = new JTextField();

        Students student;
        //Used in add course event handler
        Courses course;


        ArrayList<Students> students;
        ArrayList<Courses> courses;
        ArrayList<Instructor> instructors;
        ArrayList<Administrator> admin;

        public StudentView() {}//default constructor

        public StudentView(Students student1, ArrayList<Students> student_, ArrayList<Courses> courses_,
ArrayList<Instructor> instructors_, ArrayList<Administrator> admin_) {
                student=student1;// assigning student to student1 which is passed by constructor

                //to call LoginView constuctor for Logout Button
                this.students = student_;
                this.courses = courses_;
                this.instructors = instructors_;
                this.admin=admin_;
                //Setting up the frame
                studentsFrame.getContentPane().setBackground(new Color(105, 102, 103));
                studentsFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                studentsFrame.setSize(420,420);
                studentsFrame.setBounds(700, 250, 420, 420);
                studentsFrame.getContentPane().setLayout(null);
                studentsFrame.setVisible(true);
```

```
//Creating a Label and setting its colors, font and bounds
JLabel welcome = new JLabel("Students Page");
welcome.setForeground(new Color(237, 222, 222));
welcome.setFont(new Font("Bookman Old Style", Font.PLAIN, 19));
welcome.setBounds(140, 11, 140, 35);

//Creating MenuBar, Menus, and MenuItems. And naming them
JMenuBar Menubar = new JMenuBar();
JMenu Records = new JMenu("Student Records");
JMenu Courses = new JMenu("Manage Courses");
JMenu Personal = new JMenu("Personal Information");
final JMenuItem AddCourse = new JMenuItem("Add Courses");
JMenuItem DisplayRecords = new JMenuItem("Display Records");
final JMenuItem SaveRecords = new JMenuItem("Save Records");
JMenuItem ChangeUser = new JMenuItem("Change Username");
JMenuItem ChangePass = new JMenuItem("Change Password");
JMenuItem Logout = new JMenuItem("Logout");



//Changin Colors, fonts, and font colors of all menus, menu items, and the menu bar

Menubar.setBackground(new Color(105, 102, 103));

Courses.setFont(new Font("Bookman Old Style", Font.PLAIN, 13));
Courses.setForeground(new Color(237, 222, 222));

Records.setFont(new Font("Bookman Old Style", Font.PLAIN, 13));
Records.setForeground(new Color(237, 222, 222));

Personal.setFont(new Font("Bookman Old Style", Font.PLAIN, 13));
Personal.setForeground(new Color(237, 222, 222));

Logout.setFont(new Font("Bookman Old Style", Font.PLAIN, 13));
Logout.setForeground(new Color(237, 222, 222));


AddCourse.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
AddCourse.setForeground(new Color(237, 222, 222));
AddCourse.setBackground(new Color(172, 83, 83));

DisplayRecords.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
DisplayRecords.setForeground(new Color(237, 222, 222));
DisplayRecords.setBackground(new Color(172, 83, 83));

SaveRecords.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
SaveRecords.setForeground(new Color(237, 222, 222));
SaveRecords.setBackground(new Color(172, 83, 83));

ChangeUser.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
ChangeUser.setForeground(new Color(237, 222, 222));
ChangeUser.setBackground(new Color(172, 83, 83));
```

```
                    ChangePass.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
                    ChangePass.setForeground(new Color(237, 222, 222));
                    ChangePass.setBackground(new Color(172, 83, 83));

                    //Adding Menu bar, Menus and Menu Items
                    Records.add(DisplayRecords);
                    Records.add(SaveRecords);
                    Courses.add(AddCourse);
                    Personal.add(ChangeUser);
                    Personal.add(ChangePass);
                    Menubar.add(Personal);
                    Menubar.add(Records);
                    Menubar.add(Courses);
                    Menubar.add(Logout);


                    studentsFrame.setJMenuBar(Menubar);
                    studentsFrame.getContentPane().add(welcome);


                      Logout.addActionListener(new ActionListener() {
                                        public void actionPerformed(ActionEvent e) {
                                                  studentsFrame.dispose();

                                                  new LoginView(students, instructors, admin,courses);

                                        }
                                        });

                    //EventHandeling for Add Courses Menu Item
                    AddCourse.addActionListener(new ActionListener() {
                              public void actionPerformed(ActionEvent e) {

                                        JPanel panel = new JPanel(new GridLayout(courses.size()+1,1));
                    panel.add(new JLabel("Choose a Course to Register"));
                    //Adding menu items to the popup
                    if(!(courses==null)) {
                              for (int i = 0; i < courses.size();i++) {
                      JButton courseButton = new JButton(courses.get(i).getName() + " " +
courses.get(i).getCourseCode());

                      final Courses course = courses.get(i);

                    //Display course data on menu item click
                    courseButton.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent ev) {
                              boolean registered = false;

                              //Check if student is already registered
                              for (int k = 0; k < student.getCourses().size(); k++)
                                        if (student.getCourses().get(k) == course) {
                                        JOptionPane.showMessageDialog(null, "Course Already
Registered!","Error",JOptionPane.INFORMATION_MESSAGE);
                                         registered = true;
```

```
                                                }

                                        if (student.getCourses().size() < 5 && registered == false &&
course.getStudents().size()<20) {
                                                                        student.getCourses().add(course);

                                                student.getGrades().add(new Grade());
                                                course.getStudents().add(student);
                                        }
                                        else if(student.getCourses().size() >=5)
                                        {
                                                JOptionPane.showMessageDialog(null, "Cannot Register in more than 5
Courses!","Error",JOptionPane.INFORMATION_MESSAGE);

                                        }
                                        else if(course.getStudents().size()>=20) {
                                                JOptionPane.showMessageDialog(null, "Course
Full","Error",JOptionPane.INFORMATION_MESSAGE);

                                        }

                                }
                        });

                              panel.add(courseButton);
                                        }
                                }
                                JOptionPane.showOptionDialog(null, panel, "Course Data", JOptionPane.DEFAULT_OPTION,
JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                {}, null);

                                        }
                                });


                                //EventHandler for Display Records menu item
                                DisplayRecords.addActionListener(new ActionListener() {
                                        public void actionPerformed(ActionEvent e) {
                                                DisplayData view = new DisplayData(student);//if Display Record is chosen from
menu DisplayData Constructor is called
                                        }
                                });


                                //EventHandler for Save Records menu item that saves student information into a text file
                                SaveRecords.addActionListener(new ActionListener() {
                                        public void actionPerformed(ActionEvent e) {
                                                JFileChooser DataFile = new JFileChooser(".");
                 int choice = DataFile.showSaveDialog(SaveRecords);//Selects file
                 if (choice == JFileChooser.APPROVE_OPTION) {
                     File selectedFile = DataFile.getSelectedFile();
                       System.out.println("Selected file: " + selectedFile.getAbsolutePath());

                           try{
                              FileWriter writer = new FileWriter(selectedFile);//Creates FileWriter to display records
                              writer.write("Name: "+student.getName()+"\n");//adds name to the text file
                              writer.write("ID: "+student.getId()+"\n");//add ID to the text file
                              writer.write("Major: "+student.getMajor()+"\n");//add Major to text file
                              writer.write("Courses: "+"\n");
                              for(int i=0;i<student.getCourses().size();i++) {//this for loop adds all courses registered by student
one by one
                                        writer.write("Course "+(i+1)+":\tName:
"+student.getCourses().get(i).getName()+"\tCourse
Code:"+student.getCourses().get(i).getCourseCode()+"\tCredits:"+student.getCourses().get(i).getCredits()+"\tGrade:"
+student.getCourses().get(i).getGrade().getGrade()+"\n");
                              }

                              writer.close();
                         } catch (IOException e1) {
                              e1.printStackTrace();
                         }
               }
                                        }
                         });


                         //EventHandler for Change Username Menu that will let students change their usernames
                         ChangeUser.addActionListener(new ActionListener() {
                                   public void actionPerformed(ActionEvent e) {
                                             namepanel.add(newnamelabel);
                                             namepanel.add(newnameField);

                                             int a = JOptionPane.showOptionDialog(null, namepanel, "Change Username",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                                 {"Confirm", "Cancel"}, null);

                                             if (a == JOptionPane.OK_OPTION) {
                                                       if(newnameField.getText().equals("")) //Check if name is entered
                                                                 JOptionPane.showMessageDialog(null, "Please Enter new
Username");
                                                       else {
                                                       student.setUsername(newnameField.getText());
                                                       JOptionPane.showMessageDialog(null, "Username Changed
Successfully");
                                                       }

                                             }
                                   }
                         });




                         //EventHandler for Change Password menu item that will let students change their passwords
                         ChangePass.addActionListener(new ActionListener() {
                                   public void actionPerformed(ActionEvent e) {
                                             //setting up the pop up panel when Change password is clicked
                                             passpanel.add(oldpasslabel);
                                             passpanel.add(oldpassField);
                                             passpanel.add(newpass1label);
                                             passpanel.add(newpass1Field);
                                             passpanel.add(newpass2label);
                                             passpanel.add(newpass2Field);

                                             //Pop up pane without any icon or option dialogue, just the contents of passpanel
                                             int a = JOptionPane.showOptionDialog(null, passpanel, "Change Password",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                             {"Confirm", "Cancel"}, null);


                                             if (a == JOptionPane.OK_OPTION) {
                                                       //Compare current password with entered old password and compare
new passwords for double entry data verification
                                                       if
(String.valueOf(oldpassField.getPassword()).equals(student.getPassword())
                                                       &&
String.valueOf(newpass1Field.getPassword()).equals(String.valueOf(newpass2Field.getPassword())))
                              {
                                             student.setPassword(String.valueOf(newpass1Field.getPassword()));
                                             JOptionPane.showMessageDialog(null, "Password Changed Successfully");
                                             System.out.println(student.getPassword());

                              }
                              else {
                                             JOptionPane.showMessageDialog(null, "Incorrect Old password / New
Passwords");
                              }

                                             }
                              }
                         });


                         //Creating Buttons
                         final JButton AddCoursesButton = new JButton("Add Courses");
                         JButton DisplayData = new JButton("Display Data");
                         final JButton SaveData = new JButton("Save Student Data");//Save student data
                         JButton ChangeUsername = new JButton("Change Username");
                         JButton ChangePassword = new JButton("Change Password");

                         //Setting the colors and fonts for buttons
                         AddCoursesButton.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
                         AddCoursesButton.setForeground(new Color(237, 222, 222));
                         AddCoursesButton.setBackground(new Color(172, 83, 83));



                         DisplayData.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
                         DisplayData.setForeground(new Color(237, 222, 222));
                         DisplayData.setBackground(new Color(172, 83, 83));

                         SaveData.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
```

```
			SaveData.setForeground(new Color(237, 222, 222));
			SaveData.setBackground(new Color(172, 83, 83));

			ChangeUsername.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
			ChangeUsername.setForeground(new Color(237, 222, 222));
			ChangeUsername.setBackground(new Color(172, 83, 83));

			ChangePassword.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
			ChangePassword.setForeground(new Color(237, 222, 222));
			ChangePassword.setBackground(new Color(172, 83, 83));

			//Adding Buttons into frame and setting their bounds
			AddCoursesButton.setBounds(10, 115, 156, 33);
			DisplayData.setBounds(10, 165, 156, 33);
			SaveData.setBounds(10, 215, 156, 33);
			ChangeUsername.setBounds(200, 115, 156, 33);
			ChangePassword.setBounds(200, 165, 156, 33);

			studentsFrame.getContentPane().add(ChangePassword);
			studentsFrame.getContentPane().add(AddCoursesButton);
			studentsFrame.getContentPane().add(DisplayData);
			studentsFrame.getContentPane().add(SaveData);
			studentsFrame.getContentPane().add(ChangeUsername);

			AddCoursesButton.setFocusable(false);
			DisplayData.setFocusable(false);
			SaveData.setFocusable(false);
			ChangeUsername.setFocusable(false);
			ChangePassword.setFocusable(false);


			//Creating Label and setting its color, font, and bounds
			JLabel lblSelfserviceSystem = new JLabel("Self-Service System:");
			JLabel lblPersonalInformation = new JLabel("Personal Information:");

			lblSelfserviceSystem.setFont(new Font("Bookman Old Style", Font.PLAIN, 16));
			lblPersonalInformation.setFont(new Font("Bookman Old Style", Font.PLAIN, 16));

			lblSelfserviceSystem.setForeground(new Color(237, 222, 222));
			lblPersonalInformation.setForeground(new Color(237, 222, 222));

			lblPersonalInformation.setBounds(200, 75, 194, 29);
			lblSelfserviceSystem.setBounds(10, 75, 166, 29);

			studentsFrame.getContentPane().add(lblPersonalInformation);
			studentsFrame.getContentPane().add(lblSelfserviceSystem);

//			//EventHandler when add course button is clicked
//			AddCoursesButton.addActionListener(new ActionListener() {//let uses add courses through a  text
file that is formatted CourseCode;CourseName;CourseCredits;
//				public void actionPerformed(ActionEvent e) {
//					if((student.getCourses().size())==5) {// If student try to register more than 5
courses it shows a pop up message
```

```
//								JOptionPane.showMessageDialog(null, "Cannot Register in more than 5
Courses!","Error",JOptionPane.INFORMATION_MESSAGE);
//							}
//							else{//if student has less than 5 courses then he can register
//								JFileChooser LoadCourses = new JFileChooser();
//								LoadCourses.setApproveButtonText("Select File");
//								LoadCourses.setAcceptAllFileFilterUsed(false);
//								int choice = LoadCourses.showOpenDialog(AddCoursesButton);
//								if (choice == JFileChooser.APPROVE_OPTION) {
//									File selectedFile = LoadCourses.getSelectedFile();
//										System.out.println("Selected file: " +
selectedFile.getAbsolutePath());
//				//FileReader reader = null;
//						try{
//							Scanner in = new Scanner(selectedFile);
//							in.useDelimiter(";");
//
//							// loop that takes Courses information from text file
//							while(in.hasNext() && student.getCourses().size()<5) {//keeps adding courses
until end of file or student courses exceeds 5
//								Courses temp=new Courses();//temp Courses object to store file info in
//								temp.setCourseCode(in.next());//sets coursecode first
//								temp.setName(in.next());//sets course name second
//								temp.setCredits(Integer.parseInt(in.next()));// sets course credits last
//								temp.setGrade(new Grade(0));//sets grade to 0, only instructor is
allowed to change grade
//								student.courses.add(temp);//adds temp object into the students courses
ArrayList
//							}
//						} catch (IOException e1) {
//							e1.printStackTrace();
//						}
//								}
//							}
//						}});
//

				AddCoursesButton.addActionListener(new ActionListener() {//let uses add courses through a  text
file that is formatted CourseCode;CourseName;CourseCredits;
						public void actionPerformed(ActionEvent e) {

				JPanel panel = new JPanel(new GridLayout(courses.size()+1,1));
				panel.add(new JLabel("Choose a Course to Register"));
				//Adding menu items to the popup
				if(!(courses==null)) {
						for (int i = 0; i < courses.size();i++) {
				  JButton courseButton = new JButton(courses.get(i).getName() + " " +
courses.get(i).getCourseCode());

				  final Courses course = courses.get(i);

				//Display course data on menu item click
				courseButton.addActionListener(new ActionListener() {
				public void actionPerformed(ActionEvent ev) {
```

```
                                        boolean registered = false;

                                        //Check if student is already registered
                                        for (int k = 0; k < student.getCourses().size(); k++)
                                                if (student.getCourses().get(k) == course) {
                                                JOptionPane.showMessageDialog(null, "Course Already
Registered!","Error",JOptionPane.INFORMATION_MESSAGE);
                                                 registered = true;
                                                }

                                        if (student.getCourses().size() < 5 && registered == false) {
                                                                        student.getCourses().add(course);

                                                student.getGrades().add(new Grade());
                                                course.getStudents().add(student);
                                        }
                                        else if(student.getCourses().size() >=5)
                                        {
                                                JOptionPane.showMessageDialog(null, "Cannot Register in more than 5
Courses!","Error",JOptionPane.INFORMATION_MESSAGE);

                                        }

                                }
                        });

                                  panel.add(courseButton);
                                        }
                                }
                                JOptionPane.showOptionDialog(null, panel, "Course Data", JOptionPane.DEFAULT_OPTION,
JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                {}, null);
                                        }
                                        });




                                //Event HAndler for The button Display Record
                                DisplayData.addActionListener(new ActionListener() {
                                        public void actionPerformed(ActionEvent e) {
                                                DisplayData view = new DisplayData(student);//if Display Record button is
clicked DisplayData Constructor is called

                                        }
                                });

                                //EventHandler for Save Records button
                                SaveData.addActionListener(new ActionListener() {
                                        public void actionPerformed(ActionEvent e) {

                                                JFileChooser DataFile = new JFileChooser(".");
                         int choice = DataFile.showSaveDialog(SaveData);
                         if (choice == JFileChooser.APPROVE_OPTION) {
```

```
                    File selectedFile = DataFile.getSelectedFile();
                      System.out.println("Selected file: " + selectedFile.getAbsolutePath());

                          try{
                               FileWriter writer = new FileWriter(selectedFile);
                               writer.write("Name: "+student.getName()+"\n");
                               writer.write("ID: "+student.getId()+"\n");
                               writer.write("Major: "+student.getMajor()+"\n");
                               writer.write("Courses: "+"\n");

                               for(int i=0;i<student.getCourses().size();i++) {

                                                        writer.write("Course "+(i+1)+":\tName:
"+student.getCourses().get(i).getName()+"\tCourse
Code:"+student.getCourses().get(i).getCourseCode()+"\tCredits:"+student.getCourses().get(i).getCredits()+"\tGrade:"
+student.getGrades().get(i).getGrade());

                               }

                               writer.close();
                          } catch (IOException e1) {
                              e1.printStackTrace();
                          }
                }
                                        }
                             });



                             //Event Handler for Change Username button
                             ChangeUsername.addActionListener(new ActionListener() {
                                       public void actionPerformed(ActionEvent e) {
                                                 //setting up pop up panel for changing username
                                                 namepanel.add(newnamelabel);
                                                 namepanel.add(newnameField);

                                                 int a = JOptionPane.showOptionDialog(null, namepanel, "Change Username",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                                     {"Confirm", "Cancel"}, null);

                                                 if (a == JOptionPane.OK_OPTION) {
                                                           if(newnameField.getText().equals("")) //Check if name is entered
                                                                     JOptionPane.showMessageDialog(null, "Please Enter new
Username");
                                                           else {
                                                           student.setUsername(newnameField.getText());
                                                           JOptionPane.showMessageDialog(null, "Username Changed
Successfully");
                                                           }

                                                 }
                                       }
                             });
```

```
        //EventHandler for Changing Password button
        ChangePassword.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                //setting up a pop up panel to change password
                passpanel.add(oldpasslabel);
                passpanel.add(oldpassField);
                passpanel.add(newpass1label);
                passpanel.add(newpass1Field);
                passpanel.add(newpass2label);
                passpanel.add(newpass2Field);

                //Pop up pane without any icon or option dialogue, just the contents of passpanel
                int a = JOptionPane.showOptionDialog(null, passpanel, "Change Password",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                {"Confirm", "Cancel"}, null);


                if (a == JOptionPane.OK_OPTION) {
                    //Compare current password with entered old password and compare
new passwords for double entry data verification
                    if
(String.valueOf(oldpassField.getPassword()).equals(student.getPassword())
                    &&
String.valueOf(newpass1Field.getPassword()).equals(String.valueOf(newpass2Field.getPassword())))
            {
                student.setPassword(String.valueOf(newpass1Field.getPassword()));
                JOptionPane.showMessageDialog(null, "Password Changed Successfully");
                System.out.println(student.getPassword());

            }
            else {
                JOptionPane.showMessageDialog(null, "Incorrect Old Password / New
Passwords Not Matching");
            }

                }
            }
        });

    }
}
```

Instructor.java

```
package Lab_Project;

import java.util.ArrayList;

public class Instructor {

	String name;
	int id;
	String Username;
	String Password;
	String Dept;
	ArrayList<Courses> courses = new ArrayList<Courses>();

	public Instructor() {
		this.name = "amyo";
		this.id = 0000;
		Username = "amyo";
		Password = "0000";
		Dept = "CMP";
		for (int i = 0; i < 1; i++)
		this.courses.add(new Courses());
	}


	public Instructor(String name, int id, String username, String password, String dept) {
		this.name = name;
		this.id = id;
		Username = username;
		Password = password;
		Dept = dept;
	}

	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getUsername() {
		return Username;
	}
	public void setUsername(String username) {
		Username = username;
	}
	public String getPassword() {
		return Password;
	}
	public void setPassword(String password) {
		Password = password;
	}
	public String getDept() {
		return Dept;
	}
	public void setDept(String dept) {
		Dept = dept;
	}
	public ArrayList<Courses> getCourses() {
		return courses;
	}
	public void setCourses(ArrayList<Courses> courses) {
		this.courses = courses;
	}

	@Override
	public String toString() {
		return "Instructor [name=" + name + ", id=" + id + ", Username=" + Username + ", Password=" +
Password
				+ ", Dept=" + Dept + ", courses=" + courses + "]";
	}



}


}
```

—---

```
package Lab_Project;

public class Administrator {

	String Name;
	String Username;
	String Password;


	Administrator(){
		this.Username = "Amjad";
		this.Password = "2003";

	}

	Administrator(String Username,String Password){
		this.Username=Username;
		this.Password=Password;
	}

	public String getUsername() {
		return Username;
	}
	public void setUsername(String username) {
		Username = username;
	}
	public String getPassword() {
		return Password;
	}
	public void setPassword(String password) {
		Password = password;
	}

	public String getName() {
		return Name;
	}

	public void setName(String name) {
		Name = name;
	}
}
```

—---

Display Data

```
package Lab_Project;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import java.util.Date;
public class DisplayData extends JFrame {


        Date dt = new Date();
        JFrame frame = new JFrame();
        private JTable table;
        String Semester=null;
        Object Data [][];

        public DisplayData(Students student) {

                //Here getMonth() method checks for the current month and depending on
that it sets semester to either Spring,Fall , or Summer
                if(dt.~~getMonth~~()+1>=1 && dt.~~getMonth~~()<=5) Semester ="Spring";
                if(dt.~~getMonth~~()+1>=9 && dt.~~getMonth~~()<=12) Semester ="Fall";
                if(dt.~~getMonth~~()+1>=6 && dt.~~getMonth~~()<=8) Semester ="Summer";

                frame.setBounds(100, 100, 800, 300);
                frame.setVisible(true);
                frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                frame.setVisible(true);

                JScrollPane scrollPane = new JScrollPane();
                frame.getContentPane().add(scrollPane, BorderLayout.CENTER);

                Data= new Object[student.getCourses().size()+2][6];
                Data[0][0]="Semester";
                Data[0][1]= Semester;
                Data[1][0]= "Courses";
                Data[1][1]= "Name";
                Data[1][2]= "Number";
                Data[1][3]= "Credits";
                Data[1][4]= "Grade";

                for(int i=0;i<student.getCourses().size();i++) {
                        Data[i+2][0]= Integer.toString(i+1);
                        Data[i+2][1]=student.getCourses().get(i).getName();
                        Data[i+2][2]=student.getCourses().get(i).getCourseCode();
```

```
            Data[i+2][3]= student.getCourses().get(i).getCredits();

            Data[i+2][4]=student.getGrades().get(i).getGrade();
            Data[i+2][5]=null;

        }

        table = new JTable();
        table.setModel(new DefaultTableModel(
//          new Object[][] {
//              {"Semester", Semester , null, null, null, null},
//              {"Courses", "Name", "Number", "Credits", "Grade", null},
//              {1, student.getCourses().get(0).getName(),
student.getCourses().get(0).getCourseCode(), student.getCourses().get(0).getCredits(),
student.getCourses().get(0).getGrade().getGrade(), null},
//              {2, student.getCourses().get(1).getName(),
student.getCourses().get(1).getCourseCode(), student.getCourses().get(1).getCredits(),
student.getCourses().get(1).getGrade().getGrade(), null},
//              {3, student.getCourses().get(2).getName(),
student.getCourses().get(2).getCourseCode(), student.getCourses().get(2).getCredits(),
student.getCourses().get(2).getGrade().getGrade(), null},
//              {4, student.getCourses().get(3).getName(),
student.getCourses().get(3).getCourseCode(), student.getCourses().get(3).getCredits(),
student.getCourses().get(3).getGrade().getGrade(), null},
//              {5, student.getCourses().get(4).getName(),
student.getCourses().get(4).getCourseCode(), student.getCourses().get(4).getCredits(),
student.getCourses().get(4).getGrade().getGrade(), null},
//
//          }
                Data,
            new String[] {
                "Name:", student.getName(), "ID:",
Integer.toString(student.getId()), "Major:", student.getMajor()
            }
        ));
        scrollPane.setViewportView(table);

        frame.add(scrollPane);
        }

}
```

Administrator View

```
package Lab_Project;

import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Frame;
import java.awt.GridLayout;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JPopupMenu;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;

import java.awt.Color;
import java.awt.Container;

import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JMenuBar;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;

import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.StringTokenizer;
import java.awt.event.ActionEvent;

public class AdministratorView extends JFrame {



        private JPanel contentPane;
        JLabel Choose = new JLabel();
        JTextField IDChosen = new JTextField();
        JPanel ChooseID = new JPanel(new GridLayout(1,2));
        int ID;

        JPanel namepanel = new JPanel(new GridLayout(1,2));
        JLabel newnamelabel = new JLabel("Enter New Name:");
        JTextField newnameField = new JTextField();
        boolean notfound=true;

        JPanel passpanel = new JPanel(new GridLayout(3,2));
        JLabel oldpasslabel = new JLabel("Enter Old Password:");
        JLabel newpass1label = new JLabel("Enter New Password:");
        JLabel newpass2label = new JLabel("Renter New Password:");
        JPasswordField oldpassField = new JPasswordField();
        JPasswordField newpass1Field = new JPasswordField();
        JPasswordField newpass2Field = new JPasswordField();
        JButton confirmpassbutton = new JButton("Change password");


        ArrayList<Students> students;
        ArrayList<Instructor> instructors;
        ArrayList<Courses> courses;
        ArrayList<Administrator> admin;



        public AdministratorView(ArrayList<Students> students_, ArrayList<Instructor> instructors_,
ArrayList<Courses> courses_, ArrayList<Administrator> admin_) {

                this.students = students_;
                this.instructors = instructors_;
                this.courses = courses_;
                this.admin = admin_;


                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setBounds(700, 250, 420, 420);
                contentPane = new JPanel();
                contentPane.setBackground(new Color(105, 102, 103));
                contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
                this.getContentPane().setBackground(new Color(105, 102, 103));
```

```
getContentPane().setLayout(null);
final Container container = getContentPane();
JLabel welcome = new JLabel("Admin Page");
welcome.setForeground(new Color(237, 222, 222));
welcome.setFont(new Font("Bookman Old Style", Font.PLAIN, 19));
welcome.setBounds(140, 11, 140, 35);
getContentPane().add(welcome);


//Creating MenuBar
JMenuBar menuBar = new JMenuBar();
menuBar.setBackground(new Color(105, 102, 103));
setJMenuBar(menuBar);

//Creating and setting up Students Menu and its items
JMenu Students = new JMenu("Students");
Students.setFont(new Font("Bookman Old Style", Font.PLAIN, 13));
Students.setForeground(new Color(237, 222, 222));
menuBar.add(Students);

JMenuItem DisplayStudentData = new JMenuItem("Display Student Records");
DisplayStudentData.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
DisplayStudentData.setForeground(new Color(237, 222, 222));
DisplayStudentData.setBackground(new Color(172, 83, 83));
Students.add(DisplayStudentData);

final JMenuItem SaveStudentData = new JMenuItem("Sava Student Records");
SaveStudentData.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
SaveStudentData.setForeground(new Color(237, 222, 222));
SaveStudentData.setBackground(new Color(172, 83, 83));
Students.add(SaveStudentData);

JMenuItem ChangeStudentUser = new JMenuItem("Change a Student's Username");
ChangeStudentUser.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
ChangeStudentUser.setForeground(new Color(237, 222, 222));
ChangeStudentUser.setBackground(new Color(172, 83, 83));
Students.add(ChangeStudentUser);

JMenuItem ChangeStudentPass = new JMenuItem("Change a Student's Password");
ChangeStudentPass.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
ChangeStudentPass.setForeground(new Color(237, 222, 222));
ChangeStudentPass.setBackground(new Color(172, 83, 83));
Students.add(ChangeStudentPass);


//Creating and setting up Instructors Menu and its items
JMenu Instructors = new JMenu("Instructors");
Instructors.setFont(new Font("Bookman Old Style", Font.PLAIN, 13));
Instructors.setForeground(new Color(237, 222, 222));
```

```
			menuBar.add(Instructors);

			JMenuItem DisplayInstructorData = new JMenuItem("Display Instructor Records");

			DisplayInstructorData.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
			DisplayInstructorData.setForeground(new Color(237, 222, 222));
			DisplayInstructorData.setBackground(new Color(172, 83, 83));
			Instructors.add(DisplayInstructorData);

			JMenuItem DisplayCourseData = new JMenuItem("Dsiplay Instructor's Course Records");

			DisplayCourseData.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
			DisplayCourseData.setForeground(new Color(237, 222, 222));
			DisplayCourseData.setBackground(new Color(172, 83, 83));
			Instructors.add(DisplayCourseData);

			final JMenuItem SaveCourseData = new JMenuItem("Save Instructor's Course
Records");

			SaveCourseData.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
			SaveCourseData.setForeground(new Color(237, 222, 222));
			SaveCourseData.setBackground(new Color(172, 83, 83));
			Instructors.add(SaveCourseData);

			JMenuItem ChangeInstructorPass = new JMenuItem("Change an Instructor's
Password");

			ChangeInstructorPass.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
			ChangeInstructorPass.setForeground(new Color(237, 222, 222));
			ChangeInstructorPass.setBackground(new Color(172, 83, 83));
			Instructors.add(ChangeInstructorPass);

			JMenuItem ChangeInstructorUser = new JMenuItem("Change an Instructor's
Username");

			ChangeInstructorUser.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
			ChangeInstructorUser.setForeground(new Color(237, 222, 222));
			ChangeInstructorUser.setBackground(new Color(172, 83, 83));
			Instructors.add(ChangeInstructorUser);

			final JMenuItem ReadCourseDataFile = new JMenuItem("Read Course Data File");

			ReadCourseDataFile.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
			ReadCourseDataFile.setForeground(new Color(237, 222, 222));
			ReadCourseDataFile.setBackground(new Color(172, 83, 83));
			Instructors.add(ReadCourseDataFile);

			JMenuItem Logout = new JMenuItem("Logout");
			Logout.setFont(new Font("Bookman Old Style", Font.PLAIN, 13));
Logout.setForeground(new Color(237, 222, 222));
menuBar.add(Logout);









//Setting Labels above buttons
JLabel ManageStudents = new JLabel("Manage Students:");
JLabel ManaeInstructors= new JLabel("Manage Instructors:");

ManageStudents.setFont(new Font("Bookman Old Style", Font.PLAIN, 16));
ManaeInstructors.setFont(new Font("Bookman Old Style", Font.PLAIN, 16));

ManageStudents.setForeground(new Color(237, 222, 222));
ManaeInstructors.setForeground(new Color(237, 222, 222));

ManaeInstructors.setBounds(200, 75, 194, 29);
ManageStudents.setBounds(10, 75, 166, 29);

this.getContentPane().add(ManaeInstructors);
this.getContentPane().add(ManageStudents);

//Creating Student Buttons and setting colors, fonts, bounds
JButton DisplayStudentDataBtn = new JButton("Display Student Data");


DisplayStudentDataBtn.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
DisplayStudentDataBtn.setForeground(new Color(237, 222, 222));
DisplayStudentDataBtn.setBackground(new Color(172, 83, 83));
DisplayStudentDataBtn.setBounds(10, 115, 166, 33);
this.getContentPane().add(DisplayStudentDataBtn);
DisplayStudentDataBtn.setFocusable(false);

final JButton SaveStudentDataBtn = new JButton("Save Student Data");


SaveStudentDataBtn.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
SaveStudentDataBtn.setForeground(new Color(237, 222, 222));
SaveStudentDataBtn.setBackground(new Color(172, 83, 83));
SaveStudentDataBtn.setBounds(10, 165, 166, 33);
this.getContentPane().add(SaveStudentDataBtn);
SaveStudentDataBtn.setFocusable(false);
```

```
JButton ChangeStudentUserBtn = new JButton("Change Username");


ChangeStudentUserBtn.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
ChangeStudentUserBtn.setForeground(new Color(237, 222, 222));
ChangeStudentUserBtn.setBackground(new Color(172, 83, 83));
ChangeStudentUserBtn.setBounds(10, 215, 166, 33);
this.getContentPane().add(ChangeStudentUserBtn);
ChangeStudentUserBtn.setFocusable(false);

JButton ChangeStudentPassBtn = new JButton("Change Password");


ChangeStudentPassBtn.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
ChangeStudentPassBtn.setForeground(new Color(237, 222, 222));
ChangeStudentPassBtn.setBackground(new Color(172, 83, 83));
ChangeStudentPassBtn.setBounds(10, 265, 166, 33);
this.getContentPane().add(ChangeStudentPassBtn);
ChangeStudentPassBtn.setFocusable(false);

//Creating Instructor Buttons and setting colors, fonts, bounds
JButton DisplayInstructorDataBtn = new JButton("Display Instructor Data");


DisplayInstructorDataBtn.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
DisplayInstructorDataBtn.setForeground(new Color(237, 222, 222));
DisplayInstructorDataBtn.setBackground(new Color(172, 83, 83));
DisplayInstructorDataBtn.setBounds(200, 115, 200, 33);
this.getContentPane().add(DisplayInstructorDataBtn);
DisplayInstructorDataBtn.setFocusable(false);

JButton DisplayCourseDataBtn = new JButton("Display Courses Data");

DisplayCourseDataBtn.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
DisplayCourseDataBtn.setForeground(new Color(237, 222, 222));
DisplayCourseDataBtn.setBackground(new Color(172, 83, 83));
DisplayCourseDataBtn.setBounds(200, 165, 200, 33);
this.getContentPane().add(DisplayCourseDataBtn);
DisplayCourseDataBtn.setFocusable(false);

JButton SaveCourseDataBtn = new JButton("Save Course Data");

SaveCourseDataBtn.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
SaveCourseDataBtn.setForeground(new Color(237, 222, 222));
SaveCourseDataBtn.setBackground(new Color(172, 83, 83));
SaveCourseDataBtn.setBounds(200, 215, 200, 33);
this.getContentPane().add(SaveCourseDataBtn);
SaveCourseDataBtn.setFocusable(false);
```

```
		JButton ChangeInstructorUserBtn = new JButton("Change Username");

		ChangeInstructorUserBtn.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
		ChangeInstructorUserBtn.setForeground(new Color(237, 222, 222));
		ChangeInstructorUserBtn.setBackground(new Color(172, 83, 83));
		ChangeInstructorUserBtn.setBounds(200, 265, 200, 33);
		this.getContentPane().add(ChangeInstructorUserBtn);
		ChangeInstructorUserBtn.setFocusable(false);

		JButton ChangeInstructorPassBtn = new JButton("Change Password");

		ChangeInstructorPassBtn.setFont(new Font("Bookman Old Style", Font.PLAIN, 12));
		ChangeInstructorPassBtn.setForeground(new Color(237, 222, 222));
		ChangeInstructorPassBtn.setBackground(new Color(172, 83, 83));
		ChangeInstructorPassBtn.setBounds(200, 315, 200, 33);
		this.getContentPane().add(ChangeInstructorPassBtn);
		ChangeInstructorPassBtn.setFocusable(false);

		//Logout menu item Event Handler
		Logout.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {

				//disposes frames in AdminView
				Frame[] frames = AdministratorView.getFrames();
				for (int i = 0; i < frames.length; i++) {
					frames[i].dispose();
				}
				new LoginView(students, instructors, admin,courses);

			}});


		//Students Menu and Buttons Event Handling

			//Event handling for admin to display data for certain student through a menu
		DisplayStudentData.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				Choose.setText("Choose Student to display his Records");
				ChooseID.add(Choose);
				ChooseID.add(IDChosen);

				int a = JOptionPane.showOptionDialog(null, ChooseID, "Change
Username", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
						{"Confirm", "Cancel"}, null);


				if (a == JOptionPane.OK_OPTION) {
					if(IDChosen.getText().equals("")) //Check if name is entered
```

```
                                        JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Student");
                                else {
                                        try {
                                                ID=Integer.parseInt(IDChosen.getText());

                                        try {
                                                for(int i=0;i<students.size();i++) {
                                                        if(ID==students.get(i).getId()) {
                                                                DisplayData ViewData = new
DisplayData(students.get(i));

                                                                notfound=false;
                                                        }
                                                }

                                        }catch(Exception e1) {
                                                e1.printStackTrace();
                                        }
                                }catch(NumberFormatException e2){
                                        JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                }
                                        if(notfound)JOptionPane.showMessageDialog(null, "ID
not found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);


                                 notfound=true;

                                }
                        }
                        }});


                        //Event handling for admin to save student records through a menu
                SaveStudentData.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {

                                //asks user to enter id of student to display his data
                                Choose.setText("Choose Student to display his Records");
                                ChooseID.add(Choose);
                                ChooseID.add(IDChosen);

                                //pop up for the admin to choose id
                                int a = JOptionPane.showOptionDialog(null, ChooseID, "Student ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                        {"Confirm", "Cancel"}, null);
                                if (a == JOptionPane.OK_OPTION) {
                                if(IDChosen.getText().equals("")) //Check if name is entered
                                        JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Student");
                                else {
                                        try {
                                        ID=Integer.parseInt(IDChosen.getText());

                                        try {
                                                for(int i=0;i<students.size();i++) {
                                                        if(ID==students.get(i).getId()) {
                                                                notfound=false;
                                                                JFileChooser DataFile = new
JFileChooser(".");
                                                          int choice =
DataFile.showSaveDialog(SaveStudentData);
                                                          if (choice == JFileChooser.APPROVE_OPTION) {
                                                             File selectedFile = DataFile.getSelectedFile();
                                                              System.out.println("Selected file: " +
selectedFile.getAbsolutePath());

                                                try{
                                                  FileWriter writer = new FileWriter(selectedFile);
                                                  writer.write("Name:
"+students.get(i).getName()+"\n");
                                                  writer.write("ID: "+students.get(i).getId()+"\n");
                                                  writer.write("Major:
"+students.get(i).getMajor()+"\n");
                                                  writer.write("Courses: "+"\n");
                                                  for(int
j=0;j<students.get(i).getCourses().size();j++) {
                                                                          writer.write("Course "+(j+1)+":\tName:
"+students.get(i).getCourses().get(j).getName()+"\tCourse
Code:"+students.get(i).getCourses().get(j).getCourseCode()+"\tCredits:"+students.get(i).getCourses().get(
j).getCredits()+"\tGrade:"+students.get(i).getCourses().get(j).getGrade().getGrade()+"\n");
                                                  }

                                                  writer.close();
                                                } catch (IOException e1) {
                                                    e1.printStackTrace();
                                                }
                                                          }

                                                        }

//JOptionPane.showMessageDialog(null, "ID not found", "InfoBox: " + "Error",
JOptionPane.INFORMATION_MESSAGE);
                                                }
```

```
								}catch(Exception e1) {
									e1.printStackTrace();
								}
							}catch(NumberFormatException e2){
								JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

							}


								if(notfound)JOptionPane.showMessageDialog(null, "ID
not found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);
							}
					}
						notfound = true;
					}
				});

					//Event handling for admin to change student username through a menu
				ChangeStudentUser.addActionListener(new ActionListener() {
					public void actionPerformed(ActionEvent e) {
						Choose.setText("Choose Student to Change his username");
						ChooseID.add(Choose);
						ChooseID.add(IDChosen);

						int a = JOptionPane.showOptionDialog(null, ChooseID, "Student ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
								{"Confirm", "Cancel"}, null);


						if (a == JOptionPane.OK_OPTION) {
							if(IDChosen.getText().equals("")) //Check if name is entered
								JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Student");
							else {
								try {
									ID=Integer.parseInt(IDChosen.getText());


								try {
									for(int i=0;i<students.size();i++) {
										if(ID==students.get(i).getId()) {
											notfound=false;
											namepanel.add(newnamelabel);
											namepanel.add(newnameField);

											int a1 =
JOptionPane.showOptionDialog(null, namepanel, "Change Username",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
```

```
                                                                                    {"Confirm",
"Cancel"}, null);

                                                                        if (a1 ==
JOptionPane.OK_OPTION) {

if(newnameField.getText().equals("")) //Check if name is entered

JOptionPane.showMessageDialog(null, "Please Enter new Username");
                                                                              else {

students.get(i).setUsername(newnameField.getText());


JOptionPane.showMessageDialog(null, "Username Changed Successfully");
                                                                              }

                                                                        }
                                                                  }
                                                            }

                                                      }catch(Exception e1) {
                                                            e1.printStackTrace();
                                                      }
                                                }catch(NumberFormatException e2){
                                                      JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                                }
                                                }
                                          }
                                                if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                                          notfound=true;
                                    }
                              });


                                    //Event handling for admin to change student password through a menu
                              ChangeStudentPass.addActionListener(new ActionListener() {
                                    public void actionPerformed(ActionEvent e) {
                                          Choose.setText("Choose Student to Change his Password");
                                          ChooseID.add(Choose);
                                          ChooseID.add(IDChosen);

                                          int a = JOptionPane.showOptionDialog(null, ChooseID, "Student ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                      {"Confirm", "Cancel"}, null);
                                                if (a == JOptionPane.OK_OPTION) {
                                                        if(IDChosen.getText().equals("")) //Check if name is entered
                                                                JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Student");
                                                        else {
                                                                try {
                                                                        ID=Integer.parseInt(IDChosen.getText());


                                                                try {
                                                                        for(int i=0;i<students.size();i++) {
                                                                                if(ID==students.get(i).getId()) {
                                                                                        notfound=true;
                                                                                        //setting up a pop up panel to
change password
                                                                                        passpanel.add(oldpasslabel);
                                                                                        passpanel.add(oldpassField);
                                                                                        passpanel.add(newpass1label);
                                                                                        passpanel.add(newpass1Field);
                                                                                        passpanel.add(newpass2label);
                                                                                        passpanel.add(newpass2Field);

                                                                                        //Pop up pane without any icon
or option dialogue, just the contents of passpanel
                                                                                        int a1 =
JOptionPane.showOptionDialog(null, passpanel, "Change Password", JOptionPane.DEFAULT_OPTION,
JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                                                        {"Confirm", "Cancel"}, null);

                                                                                        if
(String.valueOf(oldpassField.getPassword()).equals(students.get(i).getPassword())
                                                                                        &&
String.valueOf(newpass1Field.getPassword()).equals(String.valueOf(newpass2Field.getPassword()))){

students.get(i).setPassword(String.valueOf(newpass1Field.getPassword()));
                                                                                JOptionPane.showMessageDialog(null,
"Password Changed Successfully");

System.out.println(students.get(i).getPassword());

                                                                        }
                                                                                                else {

JOptionPane.showMessageDialog(null, "Incorrect Old Password / New Passwords Not Matching");
                                                                                                }

                                                                                        }
```

```
						}
					}catch(Exception e1) {
					e1.printStackTrace();

				}
			}catch(NumberFormatException e2){
				JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

			}
				}


			if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);
		}
	  notfound=true;
	}

    });

	//Event Handling for Admin displaying student data through a button
    DisplayStudentDataBtn.addActionListener(new ActionListener() {
	public void actionPerformed(ActionEvent e) {
		Choose.setText("Choose Student to display his Records");
		ChooseID.add(Choose);
		ChooseID.add(IDChosen);

		int a = JOptionPane.showOptionDialog(null, ChooseID, "Change
Username", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
				{"Confirm", "Cancel"}, null);


		if (a == JOptionPane.OK_OPTION) {
			if(IDChosen.getText().equals("")) //Check if name is entered
				JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Student");
			else {
				try {
					ID=Integer.parseInt(IDChosen.getText());


				try {
					for(int i=0;i<students.size();i++) {
						if(ID==students.get(i).getId()) {
							DisplayData ViewData = new
DisplayData(students.get(i));

							notfound=false;
						}
```

```
                                                }

                                        }catch(Exception e1) {
                                                e1.printStackTrace();
                                        }
                                }catch(NumberFormatException e2){
                                        JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                }
                                        if(notfound) JOptionPane.showMessageDialog(null, "ID
not found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                                 notfound=true;

                                }
                        }
                        }
                });

                        //Event Handling for Admin to save Student data
                SaveStudentDataBtn.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {

                                //asks user to enter id of student to display his data
                                Choose.setText("Choose Student to display his Records");
                                ChooseID.add(Choose);
                                ChooseID.add(IDChosen);

                                //pop up for the admin to choose id
                                int a = JOptionPane.showOptionDialog(null, ChooseID, "Student ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                        {"Confirm", "Cancel"}, null);


                                if (a == JOptionPane.OK_OPTION) {
                                        if(IDChosen.getText().equals("")) //Check if name is entered
                                                JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Student");
                                        else {
                                                try {
                                                        ID=Integer.parseInt(IDChosen.getText());


                                                try {
                                                        for(int i=0;i<students.size();i++) {
                                                                if(ID==students.get(i).getId()) {
                                                                        notfound=false;
```

```
                                                                                    JFileChooser DataFile = new
JFileChooser(".");
                                                    int choice =
DataFile.showSaveDialog(SaveStudentDataBtn);
                                                    if (choice == JFileChooser.APPROVE_OPTION) {
                                                      File selectedFile = DataFile.getSelectedFile();
                                                        System.out.println("Selected file: " +
selectedFile.getAbsolutePath());

                                                          try{
                                                            FileWriter writer = new FileWriter(selectedFile);
                                                            writer.write("Name:
"+students.get(i).getName()+"\n");
                                                            writer.write("ID: "+students.get(i).getId()+"\n");
                                                            writer.write("Major:
"+students.get(i).getMajor()+"\n");
                                                            writer.write("Courses: "+"\n");
                                                            for(int
j=0;j<students.get(i).getCourses().size();j++) {
                                                                    writer.write("Course "+(j+1)+":\tName:
"+students.get(i).getCourses().get(j).getName()+"\tCourse
Code:"+students.get(i).getCourses().get(j).getCourseCode()+"\tCredits:"+students.get(i).getCourses().get(
j).getCredits()+"\tGrade:"+students.get(i).getCourses().get(j).getGrade().getGrade()+"\n");
                                                            }

                                                            writer.close();
                                                          } catch (IOException e1) {
                                                              e1.printStackTrace();
                                                          }
                                                    }

                                                                  }
                                                          }

                                                  }catch(Exception e1) {
                                                          e1.printStackTrace();
                                                  }
                                          }catch(NumberFormatException e2){
                                                  JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                          }
                                                  if(notfound) JOptionPane.showMessageDialog(null, "ID
not found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);


                                          }
                            }
                                          notfound=true;
```

```
					}
				});

					//Event Handling for admin changing username through a button
				ChangeStudentUserBtn.addActionListener(new ActionListener() {
					public void actionPerformed(ActionEvent e) {
						Choose.setText("Choose Student to Change his username");
						ChooseID.add(Choose);
						ChooseID.add(IDChosen);

						int a = JOptionPane.showOptionDialog(null, ChooseID, "Student ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
								{"Confirm", "Cancel"}, null);


						if (a == JOptionPane.OK_OPTION) {
							if(IDChosen.getText().equals("")) //Check if name is entered
								JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Student");
							else {
								try {
									ID=Integer.parseInt(IDChosen.getText());


								try {
									for(int i=0;i<students.size();i++) {
										if(ID==students.get(i).getId()) {
											notfound=false;
											namepanel.add(newnamelabel);
											namepanel.add(newnameField);

											int a1 =
JOptionPane.showOptionDialog(null, namepanel, "Change Username",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
													{"Confirm",
"Cancel"}, null);

											if (a1 ==
JOptionPane.OK_OPTION) {

if(newnameField.getText().equals("")) //Check if name is entered

JOptionPane.showMessageDialog(null, "Please Enter new Username");
												else {

students.get(i).setUsername(newnameField.getText());
JOptionPane.showMessageDialog(null, "Username Changed Successfully");
                                                                                                    }

                                                                                          }
                                                                                }
                                                                      }

                                                            }catch(Exception e1) {
                                                                      e1.printStackTrace();
                                                            }
                                                  }catch(NumberFormatException e2){
                                                            JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                                  }
                                                  }
                                        }
                                                  if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                                        notfound=true;
                              }
                    });

                              //Event Handling for Admin to change Student Password through a button
                    ChangeStudentPassBtn.addActionListener(new ActionListener() {
                              public void actionPerformed(ActionEvent e) {
                                        Choose.setText("Choose Student to Change his Password");
                                        ChooseID.add(Choose);
                                        ChooseID.add(IDChosen);

                                        int a = JOptionPane.showOptionDialog(null, ChooseID, "Student ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                            {"Confirm", "Cancel"}, null);


                                        if (a == JOptionPane.OK_OPTION) {
                                                  if(IDChosen.getText().equals("")) //Check if name is entered
                                                            JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Student");
                                                  else {
                                                            try {
                                                                      ID=Integer.parseInt(IDChosen.getText());


                                                            try {
                                                                      for(int i=0;i<students.size();i++) {
                                                                                if(ID==students.get(i).getId()) {
```

```
                                                                        notfound=false;
                                                                        //setting up a pop up panel to
change password
                                                                        passpanel.add(oldpasslabel);
                                                                        passpanel.add(oldpassField);
                                                                        passpanel.add(newpass1label);
                                                                        passpanel.add(newpass1Field);
                                                                        passpanel.add(newpass2label);
                                                                        passpanel.add(newpass2Field);

                                                                        //Pop up pane without any icon
or option dialogue, just the contents of passpanel
                                                                        int a1 =
JOptionPane.showOptionDialog(null, passpanel, "Change Password", JOptionPane.DEFAULT_OPTION,
JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                                        {"Confirm", "Cancel"}, null);

                                                                        if (a1 ==
JOptionPane.OK_OPTION) {
                                                                                if
(String.valueOf(oldpassField.getPassword()).equals(students.get(i).getPassword())
                                                                                &&
String.valueOf(newpass1Field.getPassword()).equals(String.valueOf(newpass2Field.getPassword()))){

students.get(i).setPassword(String.valueOf(newpass1Field.getPassword()));

JOptionPane.showMessageDialog(null, "Password Changed Successfully");

System.out.println(students.get(i).getPassword());

                                                                }
                                                                                        else {

JOptionPane.showMessageDialog(null, "Incorrect Old Password / New Passwords Not Matching");
                                                                                        }

                                                                        }
                                                                }
                                                                }
                                                        }catch(Exception e1) {
                                                        e1.printStackTrace();

                                                }
                                        }catch(NumberFormatException e2){
                                                JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                        }
```

```
                                                }

                                        }
                                        if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                          notfound=true;
                        }
                });


//_____________________________________________________________________________________
_______________________________________________________


                //Instructor Menu and Button Event Handling

                        //Event handling for Instructor Menus

                                //Event Handling so Admin can display Instructor data through the menu
                DisplayInstructorData.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                Choose.setText("Choose Instructor to Display their Data");
                                ChooseID.add(Choose);
                                ChooseID.add(IDChosen);

                                int a = JOptionPane.showOptionDialog(null, ChooseID, "Instructor ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                {"Confirm", "Cancel"}, null);


                                if (a == JOptionPane.OK_OPTION) {
                                        if(IDChosen.getText().equals("")) //Check if name is entered
                                                JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Instructor");
                                        else {
                                                try {
                                                        ID=Integer.parseInt(IDChosen.getText());


                                                try {
                                                        for(int i=0;i<instructors.size();i++) {
                                                                if(ID==instructors.get(i).getId()) {
                                                                        notfound=false;
                                                                        DefaultTableModel tableModel =
new DefaultTableModel(new String[]{"Name", instructors.get(i).getName(), "ID",
String.valueOf(instructors.get(i).getId()), "Department", instructors.get(i).getDept()},0);
                                                                JTable instructorTable = new
JTable(tableModel);
```

```
                                                                        JScrollPane tablePane = new
JScrollPane(instructorTable);
                                                                        JPanel tablePanel = new JPanel(new
GridLayout(1,1,0,0));


                                                                        tableModel.addRow(new String[]
{"Term", "Fall 2022"});
                                                                        tableModel.addRow(new String[]
{"Courses"});
                                                                        tableModel.addRow(new String[]
{"Name", "Number"});

if(!(instructors.get(i).getCourses()==null)) {
                                                                                for (int j = 0; j <
instructors.get(i).getCourses().size(); j++) {

                                                                                tableModel.addRow(new
String[] {instructors.get(i).getCourses().get(j).getName(),
instructors.get(i).getCourses().get(j).getCourseCode()});

                                                                                }
                                                                        }
                                                                        tablePanel.add(tablePane);


                                                                        JOptionPane.showOptionDialog(null,
tablePanel, "Instructor Data", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null,
new Object[]
                                                                                        {}, null);
                                                                        }
                                                                }

                                                        }catch(Exception e1) {
                                                                e1.printStackTrace();
                                                        }
                                                }catch(NumberFormatException e2){
                                                        JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                                }
                                                }
                                        }
                                                if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                                        notfound=true;
```

```
//
				}
			});
					//Event Handling so Admin can display Instructor course data through the
menu
			DisplayCourseData.addActionListener(new ActionListener() {
				public void actionPerformed(ActionEvent e) {

					Choose.setText("Choose Instructor to Display their Data");
					ChooseID.add(Choose);
					ChooseID.add(IDChosen);

					int a = JOptionPane.showOptionDialog(null, ChooseID, "Instructor ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
							{"Confirm", "Cancel"}, null);


					if (a == JOptionPane.OK_OPTION) {
						if(IDChosen.getText().equals("")) //Check if name is entered
							JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Instructor");
						else {
							try {
								ID=Integer.parseInt(IDChosen.getText());


							try {
								for(int i=0;i<instructors.size();i++) {
									if(ID==instructors.get(i).getId()) {
										notfound=false;
										final Instructor instructor =
instructors.get(i);

										JPanel panel = new JPanel(new
GridLayout(instructor.getCourses().size()+1,1));


								//Adding menu items to the popup
								if(!(instructor.getCourses()==null)) {
									for (int j = 0; j <
instructor.getCourses().size();j++) {
									  JButton courseItem = new
JButton(instructor.getCourses().get(j).getName() + " " + instructor.getCourses().get(j).getCourseCode());

									  final Courses course =
instructor.getCourses().get(i);

								//Display course data on menu item click
```

```
                                                        courseItem.addActionListener(new
ActionListener() {
                                                        public void actionPerformed(ActionEvent ev) {

                                                                DefaultTableModel tableModel = new
DefaultTableModel(new String[]{"Name", instructor.getName(), "ID", String.valueOf(instructor.getId()),
"Department", instructor.getDept()},0);
                                                                JTable courseTable = new
JTable(tableModel);

                                                                JScrollPane tablePane = new
JScrollPane(courseTable);
                                                                JPanel tablePanel = new JPanel(new
GridLayout(1,1,0,0));

                                                                tablePanel.add(tablePane);
                                                                tableModel.addRow(new String[]
{"Term", "Fall 2022","","","",""});
                                                                tableModel.addRow(new String[]
{"Course",course.getCourseCode(),course.getCourseCode(),course.getName(),"",""});

                                                                tableModel.addRow(new String[]
{"ID","Name","Grade","","",""});

                                                                if (course.getStudents() != null) {
                                                                for (int j = 0; j <
course.getStudents().size(); j ++) {
                                                                tableModel.addRow(new String[]
{String.valueOf(course.getStudents().get(j).getId()),course.getStudents().get(j).getName(),String.valueOf(
course.getStudents().get(j).getGrade().getGrade()),"","",""});
                                                                }
                                                                }



                                                                JOptionPane.showOptionDialog(null,
tablePanel, "Course Data", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new
Object[]
                                                                                {}, null);

                                                        }
                                                });

                                                          panel.add(courseItem);
                                                                }
                                                        }
                                                JOptionPane.showOptionDialog(null, panel, "Course
Data", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                        {}, null);
```

```
                                        }
                                    }

                                }catch(Exception e1) {
                                    e1.printStackTrace();
                                }
                            }catch(NumberFormatException e2){
                                JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                            }
                            }
                        }
                            if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                        notfound=true;


                    }
                });
                        //Event Handling so Admin can save Instructor course data through the
menu
                SaveCourseData.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {

                        Choose.setText("Choose Instructor to Change his username");
                        ChooseID.add(Choose);
                        ChooseID.add(IDChosen);

                        int a = JOptionPane.showOptionDialog(null, ChooseID, "Instructor ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                {"Confirm", "Cancel"}, null);


                        if (a == JOptionPane.OK_OPTION) {
                            if(IDChosen.getText().equals("")) //Check if name is entered
                                JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Instructor");
                            else {
                                try {
                                    ID=Integer.parseInt(IDChosen.getText());

                                try {
                                    for(int i=0;i<instructors.size();i++) {
                                        if(ID==instructors.get(i).getId()) {
                                            notfound=false;
                                            final Instructor instructor =
instructors.get(i);
```

```
                                                                                            JPanel CourseDataPanel = new
JPanel(new BorderLayout());


                                                                                    JLabel lbl = new JLabel("Choose a
Course to Save");

                                                                                    JPanel coursesPanel = new JPanel(new
GridLayout(instructor.getCourses().size(),1));

                                                                                    for (int j = 0; j <
instructor.getCourses().size();j++) {
                                                                                    JButton courseButton = new
JButton(instructor.getCourses().get(j).getName() + instructor.getCourses().get(j).getCourseCode());

                                                                                    final Courses course =
instructor.getCourses().get(j);

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

                                                                                    JFileChooser DataFile = new
JFileChooser(".");
                                                              int choice =
DataFile.showSaveDialog(SaveCourseData);//Selects file
                                                              if (choice == JFileChooser.APPROVE_OPTION) {
                                                                 File selectedFile = DataFile.getSelectedFile();
                                                                  System.out.println("Selected file: " +
selectedFile.getAbsolutePath());

                                                                    try{
                                                                        FileWriter writer = new
FileWriter(selectedFile);//Creates FileWriter to display records
                                                                        writer.write("Name; "+instructor.getName() + ";
ID ; "+instructor.getId() +  "; Department; " + instructor.getDept() + "\n");//adds name to the text file
                                                                        writer.write("Term; Fall2022;\n");//add ID to the
text file
                                                                        writer.write("Course: "+
course.getCourseCode() + "; " + course.getName()+"\n");
                                                                        writer.write("ID ; Name; Grade"+"\n");



                                                                        for(int j=0; j < course.getStudents().size(); j++)
{//this for loop adds all courses registered by instructor one by one
writer.write(course.getStudents().get(j).getId() + " ; " + course.getStudents().get(j).getName() + " ; " +
course.getStudents().get(j).getGrade().getGrade());
                                                                        }

                                                                        writer.close();
                                                                    } catch (IOException e1) {
                                                                        e1.printStackTrace();
                                                                    }
                                                            }



                                                                                }
                                                                                });

                                                                                coursesPanel.add(courseButton);
                                                                                }




CourseDataPanel.add(lbl,BorderLayout.NORTH);

CourseDataPanel.add(coursesPanel,BorderLayout.SOUTH);

                                                                                JOptionPane.showOptionDialog(null,
CourseDataPanel, "Save Course", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE,
null, new Object[]
                                                                                                {}, null);
                                                                                }
                                                                        }

                                                            }catch(Exception e1) {
                                                                        e1.printStackTrace();
                                                            }
                                                }catch(NumberFormatException e2){
                                                            JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                                }
                                                }
                                    }
                                                if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                                    notfound=true;


                        }
```

```
		});
				//Event Handling so Admin can Change Instructor's Username
		ChangeInstructorPass.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {

				Choose.setText("Choose Instructor to Change his Password");
				ChooseID.add(Choose);
				ChooseID.add(IDChosen);

				int a = JOptionPane.showOptionDialog(null, ChooseID, "Instructor ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
						{"Confirm", "Cancel"}, null);


				if (a == JOptionPane.OK_OPTION) {
					if(IDChosen.getText().equals("")) //Check if name is entered
						JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Instructor");
					else {
						try {
							ID=Integer.parseInt(IDChosen.getText());


						try {
							for(int i=0;i<instructors.size();i++) {
								if(ID==instructors.get(i).getId()) {
									notfound=false;
									//setting up a pop up panel to
change password
									passpanel.add(oldpasslabel);
									passpanel.add(oldpassField);
									passpanel.add(newpass1label);
									passpanel.add(newpass1Field);
									passpanel.add(newpass2label);
									passpanel.add(newpass2Field);

									//Pop up pane without any icon
or option dialogue, just the contents of passpanel
									int a1 =
JOptionPane.showOptionDialog(null, passpanel, "Change Password", JOptionPane.DEFAULT_OPTION,
JOptionPane.PLAIN_MESSAGE, null, new Object[]
									{"Confirm", "Cancel"}, null);
									if (a1 ==
JOptionPane.OK_OPTION) {
									if
(String.valueOf(oldpassField.getPassword()).equals(instructors.get(i).getPassword())
									&&
String.valueOf(newpass1Field.getPassword()).equals(String.valueOf(newpass2Field.getPassword()))){
```

```
instructors.get(i).setPassword(String.valueOf(newpass1Field.getPassword()));
                                                                                JOptionPane.showMessageDialog(null,
"Password Changed Successfully");

System.out.println(instructors.get(i).getPassword());

                                                                      }
                                                                                                    else {

JOptionPane.showMessageDialog(null, "Incorrect Old Password / New Passwords Not Matching");
                                                                                                    }

                                                                                          }
                                                                                }
                                                                                }
                                                                      }catch(Exception e1) {
                                                                      e1.printStackTrace();

                                                            }
                                                  }catch(NumberFormatException e2){
                                                            JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                                  }
                                                            }

                                                  }
                                                  if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                                notfound=true;


                              }
                    });
                                        //Event Handling so Admin can Change Instructor's Password
                    ChangeInstructorUser.addActionListener(new ActionListener() {
                              public void actionPerformed(ActionEvent e) {

                                        Choose.setText("Choose Instructor to Change his username");
                                        ChooseID.add(Choose);
                                        ChooseID.add(IDChosen);

                                        int a = JOptionPane.showOptionDialog(null, ChooseID, "Instructor ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                            {"Confirm", "Cancel"}, null);
                                        if (a == JOptionPane.OK_OPTION) {
                                                if(IDChosen.getText().equals("")) //Check if name is entered
                                                        JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Instructor");
                                                else {
                                                        try {
                                                                ID=Integer.parseInt(IDChosen.getText());


                                                        try {
                                                                for(int i=0;i<instructors.size();i++) {
                                                                        if(ID==instructors.get(i).getId()) {
                                                                                notfound=false;
                                                                                namepanel.add(newnamelabel);
                                                                                namepanel.add(newnameField);

                                                                                int a1 =
JOptionPane.showOptionDialog(null, namepanel, "Change Username",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                                                                {"Confirm",
"Cancel"}, null);

                                                                                if (a1 ==
JOptionPane.OK_OPTION) {

if(newnameField.getText().equals("")) //Check if name is entered

JOptionPane.showMessageDialog(null, "Please Enter new Username");
                                                                                        else {

instructors.get(i).setUsername(newnameField.getText());


JOptionPane.showMessageDialog(null, "Username Changed Successfully");
                                                                                        }

                                                                                }
                                                                        }
                                                                }

                                                        }catch(Exception e1) {
                                                                e1.printStackTrace();
                                                        }
                                                }catch(NumberFormatException e2){
                                                        JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                                }
                                                }
```

```
                        }

                                if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                        notfound=true;



                }
        });


                //Event Handling for Instructor's Management Buttons

                        //Event Handling so Admin can Display Instructor Data
        DisplayInstructorDataBtn.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {


                        Choose.setText("Choose Instructor to Display their Data");
                        ChooseID.add(Choose);
                        ChooseID.add(IDChosen);

                        int a = JOptionPane.showOptionDialog(null, ChooseID, "Instructor ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                        {"Confirm", "Cancel"}, null);


                        if (a == JOptionPane.OK_OPTION) {
                                if(IDChosen.getText().equals("")) //Check if name is entered
                                        JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Instructor");
                                else {
                                        try {
                                                ID=Integer.parseInt(IDChosen.getText());


                                        try {
                                                for(int i=0;i<instructors.size();i++) {
                                                        if(ID==instructors.get(i).getId()) {
                                                                notfound=false;
                                                                DefaultTableModel tableModel =
new DefaultTableModel(new String[]{"Name", instructors.get(i).getName(), "ID",
String.valueOf(instructors.get(i).getId()), "Department", instructors.get(i).getDept()},0);
                                                        JTable instructorTable = new
JTable(tableModel);
```

```
                                                                                JScrollPane tablePane = new
JScrollPane(instructorTable);
                                                                                JPanel tablePanel = new JPanel(new
GridLayout(1,1,0,0));


                                                                                tableModel.addRow(new String[]
{"Term", "Fall 2022"});
                                                                                tableModel.addRow(new String[]
{"Courses"});
                                                                                tableModel.addRow(new String[]
{"Name", "Number"});

if(!(instructors.get(i).getCourses()==null)) {
                                                                                        for (int j = 0; j <
instructors.get(i).getCourses().size(); j++) {

                                                                                        tableModel.addRow(new
String[] {instructors.get(i).getCourses().get(j).getName(),
instructors.get(i).getCourses().get(j).getCourseCode()});

                                                                                        }
                                                                                }
                                                                                tablePanel.add(tablePane);


                                                                                JOptionPane.showOptionDialog(null,
tablePanel, "Instructor Data", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null,
new Object[]
                                                                                                {}, null);
                                                                                }
                                                                        }

                                                                }catch(Exception e1) {
                                                                        e1.printStackTrace();
                                                                }
                                                        }catch(NumberFormatException e2){
                                                                JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                                        }
                                                        }
                                                }
                                                        if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                                                notfound=true;
```

```
                }
            });

                        //Event Handling so Admin can Display Course Data
            DisplayCourseDataBtn.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    Choose.setText("Choose Instructor to Display their Data");
                    ChooseID.add(Choose);
                    ChooseID.add(IDChosen);

                    int a = JOptionPane.showOptionDialog(null, ChooseID, "Instructor ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                            {"Confirm", "Cancel"}, null);


                    if (a == JOptionPane.OK_OPTION) {
                        if(IDChosen.getText().equals("")) //Check if name is entered
                            JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Instructor");
                        else {
                            try {
                                ID=Integer.parseInt(IDChosen.getText());


                            try {
                                for(int i=0;i<instructors.size();i++) {
                                    if(ID==instructors.get(i).getId()) {
                                        notfound=false;
                                        final Instructor instructor =
instructors.get(i);

                                        JPanel panel = new JPanel(new
GridLayout(instructor.getCourses().size()+1,1));


                                //Adding menu items to the popup
                                if(!(instructor.getCourses()==null)) {
                                    for (int j = 0; j <
instructor.getCourses().size();j++) {
                                  JButton courseItem = new
JButton(instructor.getCourses().get(j).getName() + " " + instructor.getCourses().get(j).getCourseCode());

                                  final Courses course =
instructor.getCourses().get(i);

                                //Display course data on menu item click
```

```
                                                            courseItem.addActionListener(new
ActionListener() {
                                                            public void actionPerformed(ActionEvent ev) {

                                                                    DefaultTableModel tableModel = new
DefaultTableModel(new String[]{"Name", instructor.getName(), "ID", String.valueOf(instructor.getId()),
"Department", instructor.getDept()},0);
                                                                    JTable courseTable = new
JTable(tableModel);

                                                                    JScrollPane tablePane = new
JScrollPane(courseTable);
                                                                    JPanel tablePanel = new JPanel(new
GridLayout(1,1,0,0));

                                                                    tablePanel.add(tablePane);
                                                                    tableModel.addRow(new String[]
{"Term", "Fall 2022","","","",""});
                                                                    tableModel.addRow(new String[]
{"Course",course.getCourseCode(),course.getName(),"",""});

                                                                    tableModel.addRow(new String[]
{"ID","Name","Grade","","",""});

                                                                    if (course.getStudents() != null) {
                                                                    for (int j = 0; j <
course.getStudents().size(); j ++) {
                                                                    tableModel.addRow(new String[]
{String.valueOf(course.getStudents().get(j).getId()),course.getStudents().get(j).getName(),String.valueOf(
course.getStudents().get(j).getGrade().getGrade()),"","",""});
                                                                    }
                                                                    }



                                                                    JOptionPane.showOptionDialog(null,
tablePanel, "Course Data", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new
Object[]
                                                                                    {}, null);

                                                            }
                                                    });

                                                              panel.add(courseItem);
                                                                    }
                                                            }
                                                    JOptionPane.showOptionDialog(null, panel, "Course
Data", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                            {}, null);
```

```
									}
								}

							}catch(Exception e1) {
								e1.printStackTrace();
							}
						}catch(NumberFormatException e2){
							JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

						}
						}
					}
						if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);


					notfound=true;



				}
			});

					//Event Handling so Admin can Save Course Data
			SaveCourseDataBtn.addActionListener(new ActionListener() {
				public void actionPerformed(ActionEvent e) {


					Choose.setText("Choose Instructor to Change his username");
					ChooseID.add(Choose);
					ChooseID.add(IDChosen);

					int a = JOptionPane.showOptionDialog(null, ChooseID, "Instructor ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
							{"Confirm", "Cancel"}, null);


					if (a == JOptionPane.OK_OPTION) {
						if(IDChosen.getText().equals("")) //Check if name is entered
							JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Instructor");
						else {
							try {
								ID=Integer.parseInt(IDChosen.getText());


							try {
								for(int i=0;i<instructors.size();i++) {
									if(ID==instructors.get(i).getId()) {
```

```
                                                                    notfound=false;
                                                                    final Instructor instructor =
instructors.get(i);
                                                                    JPanel CourseDataPanel = new
JPanel(new BorderLayout());


                                                              JLabel lbl = new JLabel("Choose a
Course to Save");

                                                              JPanel coursesPanel = new JPanel(new
GridLayout(instructor.getCourses().size(),1));

                                                              for (int j = 0; j <
instructor.getCourses().size();j++) {
                                                              JButton courseButton = new
JButton(instructor.getCourses().get(j).getName() + instructor.getCourses().get(j).getCourseCode());

                                                              final Courses course =
instructor.getCourses().get(j);

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

                                                              JFileChooser DataFile = new
JFileChooser(".");
                                                  int choice =
DataFile.showSaveDialog(SaveCourseData);//Selects file
                                                  if (choice == JFileChooser.APPROVE_OPTION) {
                                                    File selectedFile = DataFile.getSelectedFile();
                                                     System.out.println("Selected file: " +
selectedFile.getAbsolutePath());

                                                       try{
                                                         FileWriter writer = new
FileWriter(selectedFile);//Creates FileWriter to display records
                                                         writer.write("Name; "+instructor.getName() + ";
ID ; "+instructor.getId() +  "; Department; " + instructor.getDept() + "\n");//adds name to the text file
                                                         writer.write("Term; Fall2022;\n");//add ID to the
text file
                                                         writer.write("Course: "+
course.getCourseCode() + "; " + course.getName()+"\n");
                                                         writer.write("ID ; Name; Grade"+"\n");
```

```
                                                                   for(int j=0; j < course.getStudents().size(); j++)
{//this for loop adds all courses registered by instructor one by one

writer.write(course.getStudents().get(j).getId() + " ; " + course.getStudents().get(j).getName() + " ; " +
course.getStudents().get(j).getGrade().getGrade());
                                                                   }

                                                                   writer.close();
                                                               } catch (IOException e1) {
                                                                   e1.printStackTrace();
                                                               }
                                                        }



                                                                          }
                                                                          });

                                                                          coursesPanel.add(courseButton);
                                                                          }




CourseDataPanel.add(lbl,BorderLayout.NORTH);

CourseDataPanel.add(coursesPanel,BorderLayout.SOUTH);

                                                                          JOptionPane.showOptionDialog(null,
CourseDataPanel, "Save Course", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE,
null, new Object[]
                                                                                          {}, null);
                                                                          }
                                                                  }

                                                        }catch(Exception e1) {
                                                                e1.printStackTrace();
                                                        }
                                                }catch(NumberFormatException e2){
                                                        JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                                }
                                                }
                                        }
                                                if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                                        notfound=true;
```

```
                        }
                });

                                //Event Handling so Admin can Change Instructor's Username
                ChangeInstructorUserBtn.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {

                                Choose.setText("Choose Instructor to Change his username");
                                ChooseID.add(Choose);
                                ChooseID.add(IDChosen);

                                int a = JOptionPane.showOptionDialog(null, ChooseID, "Instructor ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                {"Confirm", "Cancel"}, null);


                                if (a == JOptionPane.OK_OPTION) {
                                        if(IDChosen.getText().equals("")) //Check if name is entered
                                                JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Instructor");
                                        else {
                                                try {
                                                        ID=Integer.parseInt(IDChosen.getText());


                                                try {
                                                        for(int i=0;i<instructors.size();i++) {
                                                                if(ID==instructors.get(i).getId()) {
                                                                        notfound=false;
                                                                        namepanel.add(newnamelabel);
                                                                        namepanel.add(newnameField);

                                                                        int a1 =
JOptionPane.showOptionDialog(null, namepanel, "Change Username",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                                                        {"Confirm",
"Cancel"}, null);

                                                                        if (a1 ==
JOptionPane.OK_OPTION) {

if(newnameField.getText().equals("")) //Check if name is entered

JOptionPane.showMessageDialog(null, "Please Enter new Username");
                                                                                else {
```

```
instructors.get(i).setUsername(newnameField.getText());


JOptionPane.showMessageDialog(null, "Username Changed Successfully");
                                                                                                    }

                                                                                          }
                                                                                }
                                                                      }

                                                            }catch(Exception e1) {
                                                                      e1.printStackTrace();
                                                            }
                                                  }catch(NumberFormatException e2){
                                                            JOptionPane.showMessageDialog(null, "Please Enter an
ID Number, not String");

                                                  }
                                                  }
                                        }
                                                  if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                                        notfound=true;



                              }
                    });

                                        //Event Handling so Admin can Change Instructor's Password
                    ChangeInstructorPassBtn.addActionListener(new ActionListener() {
                              public void actionPerformed(ActionEvent e) {

                                        Choose.setText("Choose Instructor to Change his Password");
                                        ChooseID.add(Choose);
                                        ChooseID.add(IDChosen);

                                        int a = JOptionPane.showOptionDialog(null, ChooseID, "Instructor ID",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                            {"Confirm", "Cancel"}, null);


                                        if (a == JOptionPane.OK_OPTION) {
                                                  if(IDChosen.getText().equals("")) //Check if name is entered
                                                            JOptionPane.showMessageDialog(null, "Please Enter
the ID of the Instructor");
                                                  else {
```

```
                                        try {
                                                ID=Integer.parseInt(IDChosen.getText());



                                        try {
                                                for(int i=0;i<instructors.size();i++) {
                                                        if(ID==instructors.get(i).getId()) {
                                                                notfound=false;
                                                                //setting up a pop up panel to
change password
                                                                passpanel.add(oldpasslabel);
                                                                passpanel.add(oldpassField);
                                                                passpanel.add(newpass1label);
                                                                passpanel.add(newpass1Field);
                                                                passpanel.add(newpass2label);
                                                                passpanel.add(newpass2Field);

                                                                //Pop up pane without any icon
or option dialogue, just the contents of passpanel
                                                                int a1 =
JOptionPane.showOptionDialog(null, passpanel, "Change Password", JOptionPane.DEFAULT_OPTION,
JOptionPane.PLAIN_MESSAGE, null, new Object[]
                                                                {"Confirm", "Cancel"}, null);
                                                                if (a1 ==
JOptionPane.OK_OPTION) {
                                                                if
(String.valueOf(oldpassField.getPassword()).equals(instructors.get(i).getPassword())
                                                                &&
String.valueOf(newpass1Field.getPassword()).equals(String.valueOf(newpass2Field.getPassword()))){

instructors.get(i).setPassword(String.valueOf(newpass1Field.getPassword()));
                                                        JOptionPane.showMessageDialog(null,
"Password Changed Successfully");

System.out.println(instructors.get(i).getPassword());

                                                }
                                                                        else {

JOptionPane.showMessageDialog(null, "Incorrect Old Password / New Passwords Not Matching");
                                                                        }

                                                                }
                                                        }
                                                        }
                                                }catch(Exception e1) {
                                                e1.printStackTrace();
                                        }
                                        }catch(NumberFormatException e2){
                                                JOptionPane.showMessageDialog(null, "Please
Enter an ID Number, not String");

                                        }
                                        }

                                }
                                if(notfound) JOptionPane.showMessageDialog(null, "ID not
found", "InfoBox: " + "Error", JOptionPane.INFORMATION_MESSAGE);

                                notfound=true;

                        }
                });


                //Read data file saved by Instructor
                ReadCourseDataFile.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {


                                        JFileChooser LoadCourse = new JFileChooser();
                                        LoadCourse.setApproveButtonText("Select File");
                                        LoadCourse.setAcceptAllFileFilterUsed(false);
                                        int choice =
LoadCourse.showOpenDialog(ReadCourseDataFile);
                                        if (choice == JFileChooser.APPROVE_OPTION) {

                                                File selectedFile =
LoadCourse.getSelectedFile();
                                                System.out.println("Selected file: " +
selectedFile.getAbsolutePath());
                                  //FileReader reader = null;
                                     try{
                                Scanner in = new Scanner(selectedFile);

                                String data = "";

                                while(in.hasNextLine() != false) {

                                data = data + in.nextLine() + "\n";

                                }

                                //Dividing the entire text into lines
                                String tb_data[] = data.split("\n");
                                        String row[][] = new String[tb_data.length][6];
```

```
				//Column Data
			String[] row1 = tb_data[0].split(";");

			System.out.println(tb_data[0]);

			for (int i = 0; i < 6 ; i++)
			{
			System.out.print(row1[i]);

			}

			//The block of text is saved into a 2d array
			for (int i = 1; i < tb_data.length; i++) {
				StringTokenizer st = new StringTokenizer(tb_data[i], ";");

				int j = 0;

				while (st.hasMoreTokens()) {
					row[i-1][j] = st.nextToken();
				System.out.print(row[i-1][j]+" ");

					j++;
				}
			System.out.println();

			}

			DefaultTableModel tablemodel = new
DefaultTableModel(row,row1);



			JTable table = new JTable(tablemodel);
			JScrollPane pane = new JScrollPane(table);
			JPanel panel = new JPanel();
			panel.add(pane);

			JOptionPane.showOptionDialog(null, panel, "Course File",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[]
								{}, null);

				in.close();

		 } catch (IOException e1) {
		// TODO Auto-generated catch block
		e1.printStackTrace();
		 }
```

```
                              }

                    }
               });



               this.setVisible(true);
          }




     }
```