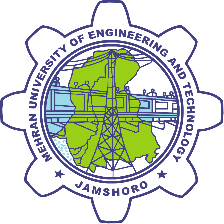
****

Subject: Object Oriented Programming

20CS020

20CS034

20CS042

Submitted To: Sir Jamsher

Submitted By:

Department: Computer Systems

Student Database Project

INTRODUCTION:

In this project, we have created a program based on object oriented programming language that is java, using object oriented law and techniques, this program creates an application based on swing and awt libraries of java, and asks the user to enter the details of the student one by one, and also displays all the attributes of the student on the side log as if in case of a mistake so to be edited using (EDIT) button, or can be deleted by pressing the (DELETE) button, after entering all the attributes hit (ADD) button and at the end hit (SAVE) button to save the info of the students, this will create a .txt file in the parent directory, containing all the attributes of all the students saved.

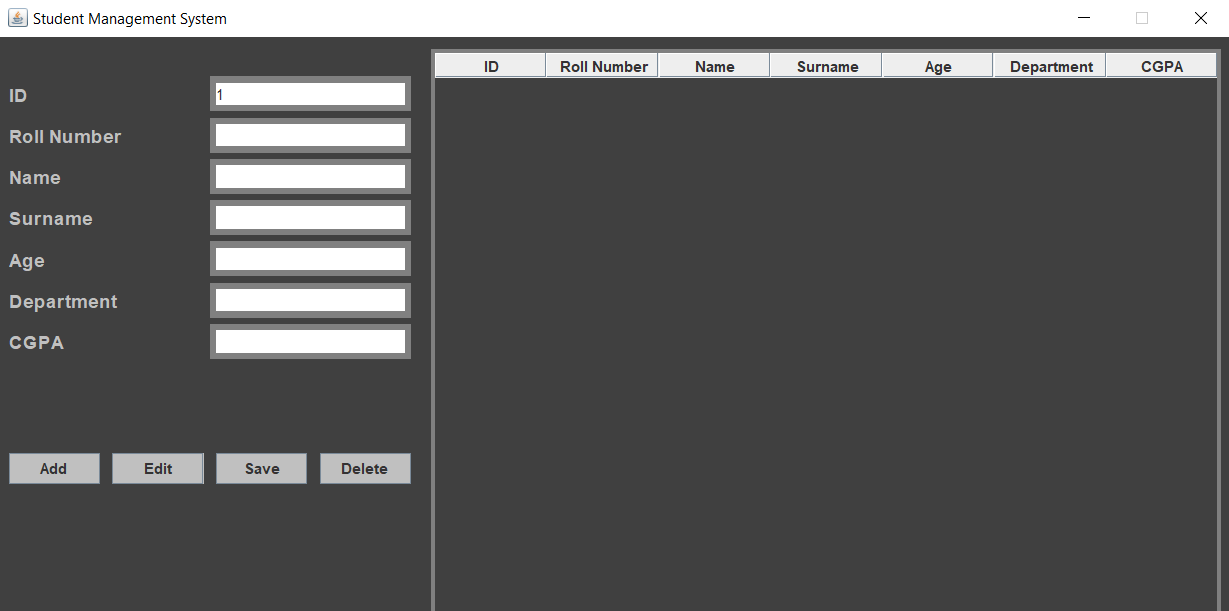
PROJECT DESCRIPTION:

In this project we have used the following libraries of java programming language:

* java.swing.\*
* java.awt.\*
* java.io.File

PROGRAM WORKING:

In this program application we have 10 classes including the (Main) class, and a header file/class (BUTTONS).



Main Class:

import javax.swing.\*;  
import java.awt.\*;  
  
public class Main {  
  
 public static void main(String[] args) {  
  
 Students students = new Students();  
  
 MyFrame frame = new MyFrame();  
 int width = frame.getWidth();  
 int height = frame.getHeight();  
  
 JPanel panel = new JPanel();  
 Table table = new Table(width, height, students);  
 Form form = new Form(width, height, students, table);  
  
 panel.add(form, BorderLayout.*WEST*);  
 panel.add(table, BorderLayout.*EAST*);  
 table.setBackground(Color.*darkGray*);  
 form.setBackground(Color.*darkGray*);  
 panel.setBackground(Color.*darkGray*);  
 frame.add(panel);  
 frame.setVisible(true);  
 }  
}

(The Main class is the executable class in our project, it is the Principal class)

Frame Class:

import javax.swing.\*;  
  
public class MyFrame extends JFrame {  
  
 private int width;  
 private int height;  
 private String title;  
  
 public MyFrame() {  
  
 this.width = 1000;  
 this.height = 500;  
 this.title = "Student Management System";  
  
 this.setSize(this.width, this.height);  
 this.setResizable(false);  
 this.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 this.setLocationRelativeTo(null);  
 this.setTitle(this.title);  
 }  
}

This block of code creates a Frame on the execution of the program through which a user will interact with the application.

Form class:

import javax.swing.\*;  
import javax.swing.border.BevelBorder;  
import javax.swing.border.EtchedBorder;  
import javax.swing.border.TitledBorder;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class Form extends JPanel implements ActionListener {  
  
 private int width;  
 private int height;  
 private int fieldContainerHeight;  
 private int buttonContainerHeight;  
 private Students students;  
 private Table table;  
 private JPanel fieldContainer;  
 private JPanel buttonContainer;  
 private JLabel id;  
 private JTextField idText;  
 private JLabel rollNumber;  
 private JTextField rollNumberText;  
 private JLabel name;  
 private JTextField nameText;  
 private JLabel surname;  
 private JTextField surnameText;  
 private JLabel age;  
 private JTextField ageText;  
 private JLabel department;  
 private JTextField departmentText;  
 private JLabel cgpa;  
 private JTextField cgpaText;  
 private AddButton addButton;  
 private EditButton editButton;  
 private SaveButton saveButton;  
 private DeleteButton deleteButton;  
  
 public Form(int width, int height, Students students, Table table) {  
 this.width = width / 3;  
 this.height = height;  
 this.fieldContainerHeight = 300;  
 this.buttonContainerHeight = 25;  
 this.setPreferredSize(new Dimension(this.width, this.height));  
  
 Font font = new Font(Font.*SANS\_SERIF*, Font.*BOLD*, 15);  
  
 this.students = students;  
 this.table = table;  
  
 this.fieldContainer = new JPanel();  
 this.fieldContainer.setLayout(new GridLayout(9, 2, 0, 5));  
 this.fieldContainer.setPreferredSize(new Dimension(this.width - 10, this.fieldContainerHeight));  
 this.fieldContainer.setBackground(Color.*darkGray*);  
 this.buttonContainer = new JPanel();  
 this.buttonContainer.setLayout(new GridLayout(1, 4, 10, 0));  
 this.buttonContainer.setPreferredSize(new Dimension(this.width - 10, this.buttonContainerHeight));  
 this.buttonContainer.setBackground(Color.*darkGray*);  
  
 this.id = new JLabel("ID");  
 id.setFont(font);  
 this.id.setForeground(Color.*LIGHT\_GRAY*);  
 this.idText = new JTextField();  
 this.idText.setBorder(BorderFactory.*createLineBorder*(Color.*gray*, 5));  
 this.setUpIdTextField();  
  
 this.rollNumber = new JLabel("Roll Number");  
 rollNumber.setFont(font);  
 this.rollNumber.setForeground(Color.*LIGHT\_GRAY*);  
 this.rollNumberText = new JTextField();  
 this.rollNumberText.setBorder(BorderFactory.*createLineBorder*(Color.*gray*, 5));  
  
 this.name = new JLabel("Name");  
 name.setFont(font);  
 this.name.setForeground(Color.*LIGHT\_GRAY*);  
 this.nameText = new JTextField();  
 this.nameText.setBorder(BorderFactory.*createLineBorder*(Color.*gray*, 5));  
  
 this.surname = new JLabel("Surname");  
 surname.setFont(font);  
 this.surname.setForeground(Color.*LIGHT\_GRAY*);  
 this.surnameText = new JTextField();  
 this.surnameText.setBorder(BorderFactory.*createLineBorder*(Color.*gray*, 5));  
  
 this.age = new JLabel("Age");  
 age.setFont(font);  
 this.age.setForeground(Color.*LIGHT\_GRAY*);  
 this.ageText = new JTextField();  
 this.ageText.setBorder(BorderFactory.*createLineBorder*(Color.*gray*, 5));  
  
 this.department = new JLabel("Department");  
 department.setFont(font);  
 this.department.setForeground(Color.*LIGHT\_GRAY*);  
 this.departmentText = new JTextField();  
 this.departmentText.setBorder(BorderFactory.*createLineBorder*(Color.*gray*, 5));  
  
 this.cgpa = new JLabel("CGPA");  
 cgpa.setFont(font);  
 this.cgpa.setForeground(Color.*LIGHT\_GRAY*);  
 this.cgpaText = new JTextField();  
 this.cgpaText.setBorder(BorderFactory.*createLineBorder*(Color.*gray*, 5));  
  
 this.addButton = new AddButton("Add", this.students);  
 this.addButton.setBackground(Color.*lightGray*);  
 this.addButton.addActionListener(this);  
  
 this.editButton = new EditButton("Edit");  
 this.editButton.setBackground(Color.*lightGray*);  
 this.editButton.addActionListener(this);  
  
 this.saveButton = new SaveButton("Save");  
 this.saveButton.setBackground(Color.*lightGray*);  
 this.saveButton.addActionListener(this);  
  
 this.deleteButton = new DeleteButton("Delete");  
 this.deleteButton.setBackground(Color.*lightGray*);  
 this.deleteButton.addActionListener(this);  
  
 this.fieldContainer.add(this.id);  
 this.fieldContainer.add(this.idText);  
 this.fieldContainer.add(this.rollNumber);  
 this.fieldContainer.add(this.rollNumberText);  
 this.fieldContainer.add(this.name);  
 this.fieldContainer.add(this.nameText);  
 this.fieldContainer.add(this.surname);  
 this.fieldContainer.add(this.surnameText);  
 this.fieldContainer.add(this.age);  
 this.fieldContainer.add(this.ageText);  
 this.fieldContainer.add(this.department);  
 this.fieldContainer.add(this.departmentText);  
 this.fieldContainer.add(this.cgpa);  
 this.fieldContainer.add(this.cgpaText);  
  
 this.add(this.fieldContainer);  
  
 this.buttonContainer.add(this.addButton, BorderLayout.*CENTER*);  
 this.buttonContainer.add(this.editButton);  
 this.buttonContainer.add(this.saveButton);  
 this.buttonContainer.add(this.deleteButton);  
 this.add(this.buttonContainer);  
 }  
  
 private void setUpIdTextField() {  
  
 if (this.students.getStudents().size() > 0) {  
 this.idText.setText(Integer.*toString*(this.students.getStudents().get(this.students.getStudents().size() - 1).getId() + 1));  
 }  
 else {  
 this.idText.setText("1");  
 }  
 this.idText.setCaretPosition(this.idText.getText().length());  
 }  
  
 private boolean validateForm(String id, String rollNumber, String name, String surname, String age, String department, String cpga) {  
  
 boolean isValid = true;  
  
 if (id.trim().equals("") || id.trim().equals(" ") || !this.isNumber(id)) {  
 isValid = false;  
 }  
  
 if(rollNumber.trim().equals("") || rollNumber.trim().equals(" ")) {  
 isValid = false;  
 }  
  
 if (name.trim().equals("") || name.trim().equals(" ")) {  
 isValid = false;  
 }  
 if (surname.trim().equals("") || surname.trim().equals(" ")) {  
 isValid = false;  
 }  
  
 if (age.trim().equals("") || age.trim().equals(" ") || !this.isNumber(age)) {  
 isValid = false;  
 }  
  
 if (department.trim().equals("") || department.trim().equals(" ")) {  
 isValid = false;  
 }  
  
 if (cpga.trim().equals("") || cpga.trim().equals(" ")) {  
 isValid = false;  
 }  
  
 return isValid;  
 }  
  
 private boolean isNumber(String input) {  
 // Checks if the passed input is a number to validate id and age  
  
 try {  
 Integer.*parseInt*(input);  
 return true;  
 }  
 catch (Exception e) {  
 return false;  
 }  
 }  
  
 private boolean checkIdExists(int id) {  
 // Checks if another student already has the given id number to maintain unique id numbers for each student  
  
 for (Student student : this.students.getStudents()) {  
 if (student.getId() == id) {  
 return true;  
 }  
 }  
 return false;  
 }  
  
 private void emptyForm() {  
 // Empties the form to make form more user-friendly  
  
 this.idText.setText("");  
 this.rollNumberText.setText("");  
 this.nameText.setText("");  
 this.surnameText.setText("");  
 this.ageText.setText("");  
 this.departmentText.setText("");  
 this.cgpaText.setText("");  
  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == this.addButton) {  
 String id = this.idText.getText();  
 String rollNumber = this.rollNumberText.getText();  
 String name = this.nameText.getText();  
 String surname = this.surnameText.getText();  
 String age = this.ageText.getText();  
 String department = this.departmentText.getText();  
 String cgpa = this.cgpaText.getText();  
  
 boolean formIsValid = this.validateForm(id, rollNumber, name, surname, age, department, cgpa);  
  
 if (formIsValid) {  
 boolean idExists = this.checkIdExists(Integer.*parseInt*(id));  
  
 if (!idExists) {  
 this.addButton.addStudent(this.table, id, rollNumber, name, surname, age, department, cgpa);  
 this.emptyForm();  
 this.setUpIdTextField();  
 JOptionPane.*showMessageDialog*(null, "Student Added Successfully", "Success", JOptionPane.*PLAIN\_MESSAGE*);  
 }  
 else {  
 JOptionPane.*showMessageDialog*(null, "ID Already Exists", "Error", JOptionPane.*ERROR\_MESSAGE*);  
 }  
 }  
 else {  
 JOptionPane.*showMessageDialog*(null, "Form Invalid!", "Error", JOptionPane.*ERROR\_MESSAGE*);  
 }  
 }  
 if (e.getSource() == this.saveButton) {  
 this.saveButton.saveData(this.students);  
 }  
 if (e.getSource() == this.editButton) {  
 this.editButton.edit(this.table.getTable(), this.students, this.idText, this.rollNumberText, this.nameText, this.surnameText, this.ageText, this.departmentText, this.cgpaText);  
 }  
 if (e.getSource() == this.deleteButton) {  
 this.deleteButton.delete(this.table.getTable(), this.students);  
 this.setUpIdTextField();  
 }  
 }  
}

This class is quit complicated in a sense the where should each element should be placed, the only work that this class does is creates margins of the frame and placed each element on its respected place that is hard coded by us the developers.

Table class:

import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
import java.awt.\*;  
  
public class Table extends JPanel {  
  
 private int width;  
 private int height;  
 private String[] columns = {"ID", "Roll Number", "Name", "Surname", "Age", "Department", "CGPA"};  
 private Students students;  
 private JTable table;  
 private DefaultTableModel tableModel;  
  
 public Table(int width, int height, Students students) {  
 this.width = (width / 3) \* 2;  
 this.height = height;  
 this.students = students;  
  
 this.setLayout(new FlowLayout());  
 this.table = new JTable() {  
 @Override // Overrides the method to make a cell selectable but not editable  
 public boolean isCellEditable(int row, int column) {  
 return false;  
 }  
 };  
 this.tableModel = (DefaultTableModel) table.getModel();  
 this.tableModel.setColumnIdentifiers(this.columns);  
 this.table.setPreferredScrollableViewportSize(new Dimension(this.width - 40, this.height));  
 this.table.setFillsViewportHeight(true);  
  
 this.loadData();  
  
 JScrollPane scrollPane = new JScrollPane(this.table);  
 scrollPane.setBorder(BorderFactory.*createLineBorder*(Color.*gray*, 3));  
  
 Font font = new Font(Font.*SANS\_SERIF*, Font.*BOLD*, 12);  
 table.setFont(font);  
 table.getTableHeader().setFont(font);  
 table.setBackground(Color.*darkGray*);  
 table.setForeground(Color.*LIGHT\_GRAY*);  
 table.setGridColor(Color.*LIGHT\_GRAY*);  
 this.add(scrollPane);  
 }  
  
 private void loadData() {  
 // Loads in the student data and displays it in the table  
  
 for (Object[] data : this.students.getData()) {  
 this.tableModel.addRow(data);  
 }  
 }  
  
 public void addRow(String id, String rollNumber, String name, String surname, String age, String department, String cgpa) {  
 // Adds a single row of student data to the table  
  
 Object[] data = {id, rollNumber, name, surname, age, department, cgpa};  
 this.tableModel.addRow(data);  
 }  
  
 public void insertRow(int index, String id, String rollNumber, String name, String surname, String age, String department, String cgpa)  
 {  
 Object[] data = {id, rollNumber, name, surname, age, department, cgpa};  
 this.tableModel.insertRow(index, data);  
 }  
  
 public JTable getTable() {  
 return table;  
 }  
}

This class is responsible for displaying of the attributes of the entered students on the side block of the frame, so as the user can have a better control, if wants to edit, delete or save and particular if not all students.

Student class:

public class Student {  
  
 private int id;  
 private String rollNumber;  
 private String name;  
 private String surname;  
 private int age;  
 private String department;  
 private String cgpa;  
  
 public Student() {  
 }  
  
 public Student(int id, String rollNumber, String name, String surname, int age, String department, String cgpa) {  
 this.id = id;  
 this.rollNumber = rollNumber;  
 this.name = name;  
 this.surname = surname;  
 this.age = age;  
 this.department = department;  
 this.cgpa = cgpa;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getRollNumber() {  
 return rollNumber;  
 }  
  
 public void setRollNumber(String rollNumber) {  
 this.rollNumber = rollNumber;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getSurname() {  
 return surname;  
 }  
  
 public void setSurname(String surname) {  
 this.surname = surname;  
 }  
  
 public int getAge() {  
 return age;  
 }  
  
 public void setAge(int age) {  
 this.age = age;  
 }  
  
 public String getDepartment() {  
 return department;  
 }  
  
 public void setDepartment(String department) {  
 this.department = department;  
 }  
  
 public String getCgpa() {  
 return cgpa;  
 }  
  
 public void setCgpa(String cgpa) {  
 this.cgpa = cgpa;  
 }  
  
 @Override  
 public String toString() {  
 return this.id + "," + this.rollNumber + ","+ this.name + "," + this.surname + "," + this.age + "," + this.department + "," + this.cgpa;  
 }  
}

This student class fetches and sets the private field of student, this holds the entered attributes of the students.

Button Source file:

import javax.swing.\*;  
  
abstract class Button extends JButton {  
  
 private String buttonText;  
  
 public Button(String text) {  
 this.buttonText = text;  
 this.setText(this.buttonText);  
 this.setFocusable(false);  
 }  
}

This block of code is further extended by all the button classes, this code just declares a button and is further adjusted by the respective qualities of the buttons.