



---

# FINAL SEMESTER PROJECT

---

BY:  
ABDULLAH(FA22-BCT-004)  
SUBMITTED TO:  
MAAM SANEHA AMIR



JANUARY 15, 2024  
COMSATS UNIVERSITY  
ISLAMABAD

## ADMIN CLASS:

```
package com.project;

import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.ArrayList;

public class Admin extends Person implements Serializable {
    private String position;
    private Student s1;
    private Attendance a1;
    private String password;

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public Student getS1() {
        return s1;
    }

    public void setS1(Student s1) {
        this.s1 = s1;
    }

    public Attendance getA1() {
```

```

        return a1;
    }

    public void setA1(Attendance a1) {
        this.a1 = a1;
    }

    public Admin() {
        this.position = null;
    }

    public Admin(String position) {
        this.position = position;
    }

    public Admin(String personId, String name, String dateOfBirth, String
contactInfo, String address, String email,
        String position, String password) {
        super(personId, name, dateOfBirth, contactInfo, address, email);
        this.position = position;
        this.password = password;
    }

    public String getPosition() {
        return position;
    }

    public void setPosition(String position) {
        this.position = position;
    }

    @Override
    public String toString() {
        return super.toString() + " Position: " + position;
    }

    public void addStudent(Student s2) {
        s1 = new Student();
        ArrayList<Student> list = s1.loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if
(list.get(i).getRegistration_number().equalsIgnoreCase(s2.getRegistration_number(
))) {
                return;
            }
        }
    }

```

```

    }
    s1.saveDataToFile(s2);
}

public void removeStudent(String reg_num) {
    s1 = new Student();
    ArrayList<Student> list = s1.loadDataFromFile();
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i).getRegistration_number().equalsIgnoreCase(reg_num)) {
            list.remove(i);
            break;
        }
    }
    try {
        File f = new File(
            "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Students.ser");
        try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(f))) {
            for (int i = 0; i < list.size(); i++) {
                oos.writeObject(list.get(i));
            }
        }
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }
}

public boolean searchFromFile(String reg_num) {
    s1 = new Student();
    ArrayList<Student> list = s1.loadDataFromFile();
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i).getRegistration_number().equals(reg_num)) {
            return true; // Found a match, no need to continue searching
        }
    }
    return false; // No match found in the entire list
}

// public void addAttendance(Attendance a2, String reg_num) {
// ArrayList<Attendance> list = a1.loadDataFromFile();
// for (Attendance existingAttendance : list) {
// if (existingAttendance.getS1().getRegistration_number().equals(reg_num)) {
// existingAttendance.saveDataToFile(a2);

```

```

// break;
// }
// }
// }

public void addAttendance(String reg_num, String attStatus, String cID) {
    s1 = new Student();
    a1 = new Attendance();
    ArrayList<Student> list = s1.loadDataFromFile();
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i) != null) {
            if
(list.get(i).getRegistration_number().equalsIgnoreCase(reg_num)) {
                a1.saveDataToFile(new Attendance(attStatus, reg_num, cID));
                break;
            }
        }
    }
}

public void addGrade(String reg_num, String marks, String cID) {
    s1 = new Student();
    Grade g1 = new Grade();
    ArrayList<Student> list = s1.loadDataFromFile();
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i) != null) {
            if
(list.get(i).getRegistration_number().equalsIgnoreCase(reg_num)) {
                g1.saveDataToFile(new Grade(marks, reg_num, cID));
                break;
            }
        }
    }
}

public boolean login(String a, String b) {
    ArrayList<Admin> list = loadDataFromFile();
    boolean flag = false;

    for (int i = 0; i < list.size(); i++) {
        if (list.get(i) != null) {
            if (a.equalsIgnoreCase(list.get(i).getEmail()) &&
b.equals(list.get(i).getPassword())) {

```

```

        flag = true;
        break;
    }
}

return flag;
}

public void saveDataToFile(Admin a1) {
    try {
        File f = new File(
            "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Admins.ser");
        ObjectOutputStream oos;
        if (f.exists()) {
            oos = new MyObjectOutputStream(new FileOutputStream(f, true));
        } else {
            oos = new ObjectOutputStream(new FileOutputStream(f));
        }
        oos.writeObject(a1);
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }
}

public ArrayList<Admin> loadDataFromFile() {
    ArrayList<Admin> list = new ArrayList<Admin>();
    try {
        try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(
            "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Admins.ser"))) {
            while (true) {
                Admin s1 = (Admin) ois.readObject();
                list.add(s1);
                System.out.println(s1.toString());
            }
        }
    } catch (ClassNotFoundException e) {
        System.out.println(e.getMessage());
    } catch (EOFException e) {
        return list;
    }
}

```

```

        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
        return null;
    }

    public void addFeesUpdate(Fees f1, String reg_num) {
        s1 = new Student();
        f1 = new Fees();
        ArrayList<Student> list = s1.loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i) != null) {
                if (list.get(i).getRegistration_number().equals(reg_num)) {
                    f1.saveDataToFile(f1);
                    break;
                }
            }
        }
    }

    // public ArrayList<Course> same(String courseName){
    //     Course c1 = new Course();
    //     ArrayList<Course>list = c1.loadDataFromFile();
    //     ArrayList<Course>alist = new ArrayList<Course>();
    //     for(int i = 0; i<list.size(); i++){
    //         if (!list.get(i).getCourseName().equals(null)) {
    //             if (list.get(i).getCourseName().equals(courseName)) {
    //                 return list.get(i).getReg_num();
    //             }
    //         }
    //     }
    //     return alist;
    // }

package com.project;

import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;

```

```
import java.io.Serializable;
import java.util.ArrayList;

public class Admin extends Person implements Serializable {
    private String position;
    private Student s1;
    private Attendance a1;
    private String password;

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public Student getS1() {
        return s1;
    }

    public void setS1(Student s1) {
        this.s1 = s1;
    }

    public Attendance getA1() {
        return a1;
    }

    public void setA1(Attendance a1) {
        this.a1 = a1;
    }

    public Admin() {
        this.position = null;
    }
}
```



```

    public Admin(String position) {
        this.position = position;
    }

    public Admin(String personId, String name, String dateOfBirth, String
contactInfo, String address, String email,
        String position, String password) {
        super(personId, name, dateOfBirth, contactInfo, address, email);
        this.position = position;
        this.password = password;
    }

    public String getPosition() {
        return position;
    }

    public void setPosition(String position) {
        this.position = position;
    }

    @Override
    public String toString() {
        return super.toString() + " Position: " + position;
    }

    public void addStudent(Student s2) {
        s1 = new Student();
        ArrayList<Student> list = s1.loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if
(list.get(i).getRegistration_number().equalsIgnoreCase(s2.getRegistration_number(
))) {
                return;
            }
        }
        s1.saveDataToFile(s2);
    }

    public void removeStudent(String reg_num) {
        s1 = new Student();
        ArrayList<Student> list = s1.loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i).getRegistration_number().equalsIgnoreCase(reg_num)) {
                list.remove(i);
                break;
            }
        }
    }

```

```

    }
}
try {
    File f = new File(
        "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Students.ser");
    try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(f))) {
        for (int i = 0; i < list.size(); i++) {
            oos.writeObject(list.get(i));
        }
    }
} catch (IOException e) {
    System.out.println(e.getMessage());
}
}

public boolean searchFromFile(String reg_num) {
    s1 = new Student();
    ArrayList<Student> list = s1.loadDataFromFile();
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i).getRegistration_number().equals(reg_num)) {
            return true; // Found a match, no need to continue searching
        }
    }
    return false; // No match found in the entire list
}

// public void addAttendance(Attendance a2, String reg_num) {
// ArrayList<Attendance> list = a1.loadDataFromFile();
// for (Attendance existingAttendance : list) {
// if (existingAttendance.getS1().getRegistration_number().equals(reg_num)) {
// existingAttendance.saveDataToFile(a2);
// break;
// }
// }
// }

public void addAttendance(String reg_num, String attStatus, String cID) {
    s1 = new Student();
    a1 = new Attendance();
    ArrayList<Student> list = s1.loadDataFromFile();
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i) != null) {

```

```

        if
(list.get(i).getRegistration_number().equalsIgnoreCase(reg_num)) {
            a1.saveDataToFile(new Attendance(attStatus, reg_num, cID));
            break;
        }

    }

}

}

}

public void addGrade(String reg_num, String marks, String cID) {
    s1 = new Student();
    Grade g1 = new Grade();
    ArrayList<Student> list = s1.loadDataFromFile();
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i) != null) {
            if
(list.get(i).getRegistration_number().equalsIgnoreCase(reg_num)) {
                g1.saveDataToFile(new Grade(marks, reg_num, cID));
                break;
            }
        }
    }
}

}

}

public boolean login(String a, String b) {
    ArrayList<Admin> list = loadDataFromFile();
    boolean flag = false;

    for (int i = 0; i < list.size(); i++) {
        if (list.get(i) != null) {
            if (a.equalsIgnoreCase(list.get(i).getEmail()) &&
b.equals(list.get(i).getPassword())) {
                flag = true;
                break;
            }
        }
    }

    return flag;
}

public void saveDataToFile(Admin a1) {
    try {

```

```

        File f = new File(
            "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Admins.ser");
        ObjectOutputStream oos;
        if (f.exists()) {
            oos = new ObjectOutputStream(new FileOutputStream(f, true));
        } else {
            oos = new ObjectOutputStream(new FileOutputStream(f));
        }
        oos.writeObject(a1);
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }
}

public ArrayList<Admin> loadDataFromFile() {
    ArrayList<Admin> list = new ArrayList<Admin>();
    try {
        try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(
            "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Admins.ser"))) {
            while (true) {
                Admin s1 = (Admin) ois.readObject();
                list.add(s1);
                System.out.println(s1.toString());
            }
        } catch (ClassNotFoundException e) {
            System.out.println(e.getMessage());
        } catch (EOFException e) {
            return list;
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
        return null;
    }
}

public void addFeesUpdate(Fees f1, String reg_num) {
    s1 = new Student();
    f1 = new Fees();
    ArrayList<Student> list = s1.loadDataFromFile();
}

```

```

        for (int i = 0; i < list.size(); i++) {
            if (list.get(i) != null) {
                if (list.get(i).getRegistration_number().equals(reg_num)) {
                    f1.saveDataToFile(f1);
                    break;
                }
            }
        }
    }
}

// public ArrayList<Course> same(String courseName){
//     Course c1 = new Course();
//     ArrayList<Course>list = c1.loadDataFromFile();
//     ArrayList<Course>alist = new ArrayList<Course>();
//     for(int i = 0; i<list.size(); i++){
//         if (!list.get(i).getCourseName().equals(null)) {
//             if (list.get(i).getCourseName().equals(courseName)) {
//                 return list.get(i).getReg_num();
//             }
//         }
//     }
//     return alist;
// }
}

```

```

package com.project;

```

```

import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.ArrayList;

```

```

public class Admin extends Person implements Serializable {
    private String position;
    private Student s1;
    private Attendance a1;
    private String password;

    public String getEmail() {
        return email;
    }
}

```

```

    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public Student getS1() {
        return s1;
    }

    public void setS1(Student s1) {
        this.s1 = s1;
    }

    public Attendance getA1() {
        return a1;
    }

    public void setA1(Attendance a1) {
        this.a1 = a1;
    }

    public Admin() {
        this.position = null;
    }

    public Admin(String position) {
        this.position = position;
    }

    public Admin(String personId, String name, String dateOfBirth, String
contactInfo, String address, String email,
        String position, String password) {
        super(personId, name, dateOfBirth, contactInfo, address, email);
        this.position = position;
        this.password = password;
    }

```

```

    public String getPosition() {
        return position;
    }

    public void setPosition(String position) {
        this.position = position;
    }

    @Override
    public String toString() {
        return super.toString() + " Position: " + position;
    }

    public void addStudent(Student s2) {
        s1 = new Student();
        ArrayList<Student> list = s1.loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if
(list.get(i).getRegistration_number().equalsIgnoreCase(s2.getRegistration_number(
))) {
                return;
            }
        }
        s1.saveDataToFile(s2);
    }

    public void removeStudent(String reg_num) {
        s1 = new Student();
        ArrayList<Student> list = s1.loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i).getRegistration_number().equalsIgnoreCase(reg_num)) {
                list.remove(i);
                break;
            }
        }
        try {
            File f = new File(
                "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Students.ser");
            try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(f))) {
                for (int i = 0; i < list.size(); i++) {
                    oos.writeObject(list.get(i));
                }
            }
        }
    }

```

```

    }
    }
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }
}

public boolean searchFromFile(String reg_num) {
    s1 = new Student();
    ArrayList<Student> list = s1.loadDataFromFile();
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i).getRegistration_number().equals(reg_num)) {
            return true; // Found a match, no need to continue searching
        }
    }
    return false; // No match found in the entire list
}

// public void addAttendance(Attendance a2, String reg_num) {
// ArrayList<Attendance> list = a1.loadDataFromFile();
// for (Attendance existingAttendance : list) {
// if (existingAttendance.getS1().getRegistration_number().equals(reg_num)) {
// existingAttendance.saveDataToFile(a2);
// break;
// }
// }
// }

public void addAttendance(String reg_num, String attStatus, String cID) {
    s1 = new Student();
    a1 = new Attendance();
    ArrayList<Student> list = s1.loadDataFromFile();
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i) != null) {
            if
(list.get(i).getRegistration_number().equalsIgnoreCase(reg_num)) {
                a1.saveDataToFile(new Attendance(attStatus, reg_num, cID));
                break;
            }
        }
    }
}

}

public void addGrade(String reg_num, String marks, String cID) {

```



```

        s1 = new Student();
        Grade g1 = new Grade();
        ArrayList<Student> list = s1.loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i) != null) {
                if
(list.get(i).getRegistration_number().equalsIgnoreCase(reg_num)) {
                    g1.saveDataToFile(new Grade(marks, reg_num, cID));
                    break;
                }
            }
        }
    }

    public boolean login(String a, String b) {
        ArrayList<Admin> list = loadDataFromFile();
        boolean flag = false;

        for (int i = 0; i < list.size(); i++) {
            if (list.get(i) != null) {
                if (a.equalsIgnoreCase(list.get(i).getEmail()) &&
b.equals(list.get(i).getPassword())) {
                    flag = true;
                    break;
                }
            }
        }

        return flag;
    }

    public void saveDataToFile(Admin a1) {
        try {
            File f = new File(
                "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Admins.ser");
            ObjectOutputStream oos;
            if (f.exists()) {
                oos = new MyObjectOutputStream(new FileOutputStream(f, true));
            } else {
                oos = new ObjectOutputStream(new FileOutputStream(f));
            }
            oos.writeObject(a1);
        }
    }

```

```

        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }

    public ArrayList<Admin> loadDataFromFile() {
        ArrayList<Admin> list = new ArrayList<Admin>();
        try {
            try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(
                "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Admins.ser"))) {
                while (true) {
                    Admin s1 = (Admin) ois.readObject();
                    list.add(s1);
                    System.out.println(s1.toString());
                }
            }
        } catch (ClassNotFoundException e) {
            System.out.println(e.getMessage());
        } catch (EOFException e) {
            return list;
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
        return null;
    }

    public void addFeesUpdate(Fees f1, String reg_num) {
        s1 = new Student();
        f1 = new Fees();
        ArrayList<Student> list = s1.loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i) != null) {
                if (list.get(i).getRegistration_number().equals(reg_num)) {
                    f1.saveDataToFile(f1);
                    break;
                }
            }
        }
    }
}

// public ArrayList<Course> same(String courseName){
//     Course c1 = new Course();

```

```
//      ArrayList<Course>list = c1.loadDataFromFile();
//      ArrayList<Course>alist = new ArrayList<Course>();
//      for(int i = 0; i<list.size(); i++){
//          if (!list.get(i).getCourseName().equals(null)) {
//              if (list.get(i).getCourseName().equals(courseName)) {
//                  return list.get(i).getReg_num();
//              }
//          }
//      }
//      return alist;
// }

}
```

## STUDENT CLASS:

```
package com.project;

import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.ArrayList;

public class Student extends Person implements Serializable {
    private String registration_number;
    private Course c1;
    private Grade g1;
    private String password;
    private Fees f1;
    private Attendance a1;

    public Student(String registration_number, Course c1, Grade g1, String
password, Fees f1, Attendance a1) {
        this.registration_number = registration_number;
        this.c1 = c1;
        this.g1 = g1;
        this.password = password;
        this.f1 = f1;
        this.a1 = a1;
    }

    public Student(String personId, String name, String dateOfBirth, String
contactInfo, String address, String email,
        String registration_number, Course c1, Grade g1, String password,
Fees f1, Attendance a1) {
        super(personId, name, dateOfBirth, contactInfo, address, email);
        this.registration_number = registration_number;
        this.c1 = c1;
        this.g1 = g1;
        this.password = password;
        this.f1 = f1;
        this.a1 = a1;
    }
}
```

```

public Student() {
    super(null, null, null, null, null);
    this.registration_number = null;
    this.c1 = null;
    this.g1 = null;
    this.password = null;
    this.f1 = null;
    this.a1 = null;
}

public Student(String registration_number, Course c1, Grade g1, String
password, Fees f1) {
    this.registration_number = registration_number;
    this.c1 = c1;
    this.g1 = g1;
    this.password = password;
    this.f1 = f1;
}

public Student(String personId, String name, String dateOfBirth, String
contactInfo, String address, String email,
    String registration_number, Course c1, Grade g1, String password,
Fees f1) {
    super(personId, name, dateOfBirth, contactInfo, address, email);
    this.registration_number = registration_number;
    this.c1 = c1;
    this.g1 = g1;
    this.password = password;
    this.f1 = f1;
}

public Student(String registration_number, Course c1, Grade g1, String
password) {
    this.registration_number = registration_number;
    this.c1 = c1;
    this.g1 = g1;
    this.password = password;
}

public Student(String personId, String name, String dateOfBirth, String
contactInfo, String address, String email,
    String registration_number, Course c1, Grade g1, String password) {
    super(personId, name, dateOfBirth, contactInfo, address, email);
    this.registration_number = registration_number;
    this.c1 = c1;
}

```

```

        this.g1 = g1;
        this.password = password;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public Student(String registration_number, Course c1, Grade g1) {
        this.registration_number = registration_number;
        this.c1 = c1;
        this.g1 = g1;
    }

    public Student(String personId, String name, String dateOfBirth, String
contactInfo, String address, String email,
        String registration_number, Course c1, Grade g1) {
        super(personId, name, dateOfBirth, contactInfo, address, email);
        this.registration_number = registration_number;
        this.c1 = c1;
        this.g1 = g1;
    }

    public Student(String registration_number, Course c1) {
        this.registration_number = registration_number;
        this.c1 = c1;
    }

    public Student(String personId, String name, String dateOfBirth, String
contactInfo, String address, String email,
        String registration_number, Course c1) {
        super(personId, name, dateOfBirth, contactInfo, address, email);
        this.registration_number = registration_number;
        this.c1 = c1;
    }

    // public Student() {
    // this.registration_number = null;
    // this.c1 = null;
    // this.g1 = null;
    // this.password = null;

```

```

// this.f1 = null;
// this.a1 = null;

// }

public Student(String registration_number) {
    this.registration_number = registration_number;
}

public Student(String personId, String name, String dateOfBirth, String
contactInfo, String address, String email,
    String registration_number, String Password) {
    super(personId, name, dateOfBirth, contactInfo, address, email);
    this.registration_number = registration_number;
    this.password = Password;
}

public String getRegistration_number() {
    return registration_number;
}

public void setRegistration_number(String registration_number) {
    this.registration_number = registration_number;
}

@Override
public String toString() {
    return super.toString() + "Registration Number: " + registration_number +
"\n";
}

public void displayFees(Student s1) {
    Fees f1 = new Fees();
    ArrayList<Fees> list = f1.loadDataFromFile();
    for (int i = 0; i < list.size(); i++) {
        if (!list.get(i).getStatus().equals(null)) {
            if (this.registration_number.equals(s1.registration_number)) {
                System.out.println(list.get(i).toString());
            }
        }
    }
}

public String displayCoursesEnrolled(String reg_num) {

```

```

        Course c1 = new Course();
        ArrayList<Course> list = c1.loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if (!list.get(i).getReg_num().equals(null))
                if (list.get(i).getReg_num().equals(reg_num)) {
                    return list.get(i).toString();
                }
        }
        return "No Data to Display";
    }

    public void updateProfileEmail(String reg_num, String email) {
        ArrayList<Student> list = loadDataFromFile();
        boolean studentFound = false;

        for (int i = 0; i < list.size(); i++) {
            if (list.get(i).getRegistration_number().equalsIgnoreCase(reg_num)) {
                list.get(i).setEmail(email);
                studentFound = true;
                break;
            }
        }

        if (!studentFound) {
            System.out.println("Student not found!");
            return;
        }

        try {
            File f = new File(
                "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Students.ser");
            ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(f));

            for (int i = 0; i < list.size(); i++) {
                oos.writeObject(list.get(i));
            }

            oos.close();
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
}

```



```

// public void enrollInCourse(String reg_num, String CourseID) {
// ArrayList<Course> list = c1.loadDataFromFile();
// boolean courseFound = false;

// for (int i = 0; i < list.size(); i++) {
// if (list.get(i)..equals(reg_num) &&
// list.get(i).getCourseID().equalsIgnoreCase(CourseID)) {
// setC1(c1); // Make sure setC1 method is correctly implemented
// courseFound = true;
// break;
// }
// }

// if (!courseFound) {
// System.out.println("Course doesn't Exist!");
// return;
// }
// try {
// File f = new File(
// "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab Work\\Final
// Semester
//
// Project\\studentportalmanagementsystem\\src\\main\\resources\\Students.ser");
// ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(f));

// for (int i = 0; i < list.size(); i++) {
// oos.writeObject(list.get(i));
// }

// oos.close();
// } catch (IOException e) {
// System.out.println(e.getMessage());
// }
// }

public void displayGrade(Student s1) {
    ArrayList<Grade> list = g1.loadDataFromFile();
    for (int i = 0; i < list.size(); i++) {
        if
(this.registration_number.equalsIgnoreCase(s1.registration_number)) {
            System.out.println(list.get(i).getGradeValue());
        }
    }
}
}

```

```

public void addCourse(Course c2) {
    ArrayList<Course> list = c1.loadDataFromFile();
    for (int i = 0; i < list.size(); i++) {
        if (!list.get(i).getCourseID().equalsIgnoreCase(c2.getCourseID())) {
            c1.saveDataToFile(c2);
        } else {
            System.out.println("Course already exists!");
            return;
        }
    }
}

public void removeCourse(String reg_num, String cID) {
    ArrayList<Student> list = loadDataFromFile();
    boolean courseFound = false;

    for (int i = 0; i < list.size(); i++) {
        if (list.get(i).getRegistration_number().equals(reg_num)) {

            if (list.get(i).c1.getCourseID().equals(cID)) {
                courseFound = true;
                list.remove(i);
            }
        }
    }

    if (!courseFound) {
        System.out.println("Course doesn't Exist!");
        return;
    }
    try {
        File f = new File(
            "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Students.ser");
        try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(f))) {
            for (int j = 0; j < list.size(); j++) {
                oos.writeObject(list.get(j));
            }
        }
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }
}

```

```

    }

    public void saveDataToFile(Student s1) {
        try {
            File f = new File(
                "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Students.ser");
            ObjectOutputStream oos;
            if (f.exists()) {
                oos = new MyObjectOutputStream(new FileOutputStream(f, true));
            } else {
                oos = new ObjectOutputStream(new FileOutputStream(f));
            }
            oos.writeObject(s1);
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }

    public ArrayList<Student> loadDataFromFile() {
        ArrayList<Student> list = new ArrayList<Student>();
        try {
            try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(
                "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Students.ser"))) {
                while (true) {
                    Student s1 = (Student) ois.readObject();
                    list.add(s1);
                    System.out.println(s1.toString());
                }
            }
        } catch (ClassNotFoundException e) {
            System.out.println(e.getMessage());
        } catch (EOFException e) {
            return list;
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
        return null;
    }
}

```

```
public Course getC1() {
    return c1;
}

public void setC1(Course c1) {
    this.c1 = c1;
}

public Grade getG1() {
    return g1;
}

public void setG1(Grade g1) {
    this.g1 = g1;
}

public Fees getF1() {
    return f1;
}

public void setF1(Fees f1) {
    this.f1 = f1;
}

public Attendance getA1() {
    return a1;
}

public void setA1(Attendance a1) {
    this.a1 = a1;
}

public boolean login(String a, String b) {
    ArrayList<Student> list = loadDataFromFile();
    boolean flag = false;

    for (int i = 0; i < list.size(); i++) {
        if (list.get(i) != null) {
            if (a.equalsIgnoreCase(list.get(i).getRegistration_number()) &&
b.equals(list.get(i).getPassword())) {
                flag = true;
                break;
            }
        }
    }
}
```

```

        return flag;
    }

    public boolean displayStudent(String reg_num) {
        ArrayList<Student> list = loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i).getRegistration_number().equals(reg_num)) {
                return true;
            }
        }
        return false;
    }

    public String DisplayStudent(String reg_num) {
        ArrayList<Student> list = loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i).getRegistration_number().equals(reg_num)) {
                return list.get(i).toString();
            }
        }
        return "No Data to display";
    }

    public String DisplayFeesStatus(String reg_num) {
        ArrayList<Student> list = loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i).getRegistration_number().equals(reg_num)) {
                return list.get(i).f1.getStatus();
            }
        }
        return "No Data to display";
    }

    public String DisplayGradeStatus(String reg_num) {
        Grade g1 = new Grade();
        ArrayList<Grade> list = g1.loadDataFromFile();
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i).getReg_num().equals(reg_num)) {
                if (!list.get(i).getGradeValue().equals(null)) {
                    return list.get(i).getGradeValue();
                }
            }
        }
    }

```

```

        }
    }

    }
    return "No Data to display";
}

public void addCourse(String reg_num, String cID) {
    Course c1 = new Course();
    ArrayList<Course> list = c1.loadDataFromFile();
    boolean courseFound = false;
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i).getCourseID() != null &&
list.get(i).getCourseID().equals(cID)) {
            courseFound = true;
            break;
        }
    }

    if (!courseFound) {
        System.out.println("Course doesn't Exist!");
        return;
    } else {
        c1.saveDataToFile(new Course(cID, reg_num));
    }

    try {
        File f = new File(
            "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Courses.ser");
        ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(f));

        for (int i = 0; i < list.size(); i++) {
            oos.writeObject(list.get(i));
        }
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }
}

public String displayAttendance(String reg_num) {
    Attendance a1 = new Attendance();
    ArrayList<Attendance> list = a1.loadDataFromFile();

```

```
        for (int i = 0; i < list.size(); i++) {  
            if (!list.get(i).getAttendanceStatus().equals(null))  
                if (list.get(i).getReg_num().equals(reg_num)) {  
                    return list.get(i).toString();  
                }  
        }  
        return "No Data to Display";  
    }  
}
```

## ATTENDANCE CLASS:

```
package com.project;

import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.ArrayList;

public class Attendance implements Serializable {
    private Course c1;
    private Date d1;
    private String attendanceStatus;
    private String reg_num;
    private String cID;

    public Attendance(String att, String reg, String c) {
        this.attendanceStatus = att;
        this.reg_num = reg;
        this.cID = c;
    }

    public Attendance(Course c1, Date d1, String attendanceStatus, String
reg_num, Student s1) {
        this.c1 = c1;
        this.d1 = d1;
        this.attendanceStatus = attendanceStatus;
        this.reg_num = reg_num;
        this.s1 = s1;
    }

    public String getReg_num() {
        return reg_num;
    }

    public void setReg_num(String reg_num) {
        this.reg_num = reg_num;
    }

    private Student s1;
```



```

public Attendance(String cID, String attend) {
    new Course(cID);
    this.attendanceStatus = attend;
}

public Attendance() {
    this.attendanceStatus = null;
}

public Attendance(Course c1, Date d1, String attendanceStatus, Student s1) {
    this.c1 = c1;
    this.d1 = d1;
    this.attendanceStatus = attendanceStatus;
    this.s1 = s1;
}

public Attendance(Course c1, Date d1, String attendanceStatus) {
    this.c1 = c1;
    this.d1 = d1;
    this.attendanceStatus = attendanceStatus;
}

public Attendance(Course c1, String attendanceStatus) {
    this.c1 = c1;
    this.attendanceStatus = attendanceStatus;
}

public Attendance(Course c1) {
    this.c1 = c1;
}

public Course getC1() {
    return c1;
}

public void setC1(Course c1) {
    this.c1 = c1;
}

@Override
// public String toString() {
//     return "Course Info: " + c1.toString() + "\n" + "Date: " +
// d1.toString() + "\n" + " Attendance Status: "
//         + attendanceStatus + "\n";

```

```

// }
public String toString() {
    return "Reg Num " + reg_num + "\n" + "Course ID " + cID + "\n" + "
Attendance Status: "
        + attendanceStatus + "\n";
}

public void saveDataToFile(Attendance a1) {
    try {
        File f = new File(
            "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Attendances.ser");
        ObjectOutputStream oos;
        if (f.exists()) {
            oos = new MyObjectOutputStream(new FileOutputStream(f, true));
        } else {
            oos = new ObjectOutputStream(new FileOutputStream(f));
        }
        oos.writeObject(a1);
        oos.close();
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }
}

public ArrayList<Attendance> loadDataFromFile() {
    ArrayList<Attendance> list = new ArrayList<Attendance>();
    try {
        try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(
            "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Attendances.ser")))
        {
            while (true) {
                Attendance a1 = (Attendance) ois.readObject();
                list.add(a1);
                System.out.println(a1.toString());
            }
        }
    } catch (ClassNotFoundException e) {
        System.out.println(e.getMessage());
    } catch (EOFException e) {
        return list;
    }
}

```

```

        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
        return null;
    }

    public String getAttendanceStatus() {
        return attendanceStatus;
    }

    public void setAttendanceStatus(String attendanceStatus) {
        this.attendanceStatus = attendanceStatus;
    }

    public Date getD1() {
        return d1;
    }

    public void setD1(Date d1) {
        this.d1 = d1;
    }

    public Student getS1() {
        return s1;
    }

    public void setS1(Student s1) {
        this.s1 = s1;
    }

    public String getcID() {
        return cID;
    }

    public void setcID(String cID) {
        this.cID = cID;
    }

    // public void addAttendance(String CourseID, String reg_num, String
    // attendance_stat) {
    // ArrayList<Attendance> list = loadDataFromFile();
    // Admin a2 = new Admin();
    // for (int i = 0; i < list.size(); i++) {
    // if (list.get(i).s1.getRegistration_number().equalsIgnoreCase(reg_num)) {

```

```
// Attendance a1 = new Attendance(CourseID, attendance_stat);  
// a2.addAttendance(a1, reg_num);  
// }  
// }  
// }  
}
```

## DATE CLASS:

```
package com.project;

import java.io.Serializable;

class Date implements Serializable {
    private int day;
    private int month;
    private int year;

    public Date() {
        this.day = 0;
        this.month = 0;
        this.year = 0;
    }

    public Date(int d, int m, int y) {
        this.day = d;
        this.month = m;
        this.year = y;
    }

    public Date(Date d) {
        this.day = d.day;
        this.month = d.month;
        this.year = d.year;
    }

    public void setDay(int day) {
        this.day = day;
    }

    public void setMonth(int month) {
        this.month = month;
    }

    public void setYear(int year) {
        this.year = year;
    }

    public int getDay() {
        return day;
    }
}
```

```
public int getMonth() {  
    return month;  
}  
  
public int getYear() {  
    return year;  
}  
  
public void Display() {  
    System.out.println(day + " - " + month + " - " + year);  
}  
  
public String toString() {  
    return "Date [day=" + day + ", month=" + month + ", year=" + year + "];"  
}  
  
}
```

## FEES CLASS:

```
package com.project;

import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.ArrayList;

public class Fees implements Serializable {
    private String status;
    private String total_fees;
    private String reg_num;

    public Fees(String status, String total_fees, String reg_num) {
        this.status = status;
        this.total_fees = total_fees;
        this.reg_num = reg_num;
    }

    public Fees() {
        this.status = null;
        this.total_fees = null;
        new Student();
    }

    public Fees(String status, String total_fees) {
        this.status = status;
        this.total_fees = total_fees;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public String getTotal_fees() {
```

```

        return total_fees;
    }

    public void setTotal_fees(String total_fees) {
        this.total_fees = total_fees;
    }

    @Override
    public String toString() {
        return " Status: " + status + " Total Fees:" + total_fees + "\n" + "
Registration Number: " + reg_num;
    }

    public void saveDataToFile(Fees f1) {
        try {
            File f = new File(
                "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Fees.ser");
            ObjectOutputStream oos;
            if (f.exists()) {
                oos = new MyObjectOutputStream(new FileOutputStream(f, true));
            } else {
                oos = new ObjectOutputStream(new FileOutputStream(f));
            }
            oos.writeObject(f1);
            oos.close();
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }

    public ArrayList<Fees> loadDataFromFile() {
        ArrayList<Fees> list = new ArrayList<Fees>();
        try {
            try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(
                "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Fees.ser"))) {
                while (true) {
                    Fees f1 = (Fees) ois.readObject();
                    list.add(f1);
                    System.out.println(f1.toString());
                }
            }
        }
    }

```



```
    }  
    } catch (ClassNotFoundException e) {  
        System.out.println(e.getMessage());  
    } catch (EOFException e) {  
        return list;  
    } catch (IOException e) {  
        System.out.println(e.getMessage());  
    }  
    return null;  
  
}  
  
public String getreg_num() {  
    return reg_num;  
}  
  
public void setreg_num(String reg_num) {  
    this.reg_num = reg_num;  
}  
  
}
```

## GRADE CLASS:

```
package com.project;

import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.ArrayList;

public class Fees implements Serializable {
    private String status;
    private String total_fees;
    private String reg_num;

    public Fees(String status, String total_fees, String reg_num) {
        this.status = status;
        this.total_fees = total_fees;
        this.reg_num = reg_num;
    }

    public Fees() {
        this.status = null;
        this.total_fees = null;
        new Student();
    }

    public Fees(String status, String total_fees) {
        this.status = status;
        this.total_fees = total_fees;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public String getTotal_fees() {
```

```

        return total_fees;
    }

    public void setTotal_fees(String total_fees) {
        this.total_fees = total_fees;
    }

    @Override
    public String toString() {
        return " Status: " + status + " Total Fees:" + total_fees + "\n" + "
Registration Number: " + reg_num;
    }

    public void saveDataToFile(Fees f1) {
        try {
            File f = new File(
                "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Fees.ser");
            ObjectOutputStream oos;
            if (f.exists()) {
                oos = new MyObjectOutputStream(new FileOutputStream(f, true));
            } else {
                oos = new ObjectOutputStream(new FileOutputStream(f));
            }
            oos.writeObject(f1);
            oos.close();
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }

    public ArrayList<Fees> loadDataFromFile() {
        ArrayList<Fees> list = new ArrayList<Fees>();
        try {
            try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(
                "D:\\University Stuff\\Abdullah University-3\\OOP\\Lab\\Lab
Work\\Final Semester
Project\\studentportalmanagementsystem\\src\\main\\resources\\Fees.ser"))) {
                while (true) {
                    Fees f1 = (Fees) ois.readObject();
                    list.add(f1);
                    System.out.println(f1.toString());
                }
            }
        }
    }

```

```
    }  
    } catch (ClassNotFoundException e) {  
        System.out.println(e.getMessage());  
    } catch (EOFException e) {  
        return list;  
    } catch (IOException e) {  
        System.out.println(e.getMessage());  
    }  
    return null;  
  
}  
  
public String getreg_num() {  
    return reg_num;  
}  
  
public void setreg_num(String reg_num) {  
    this.reg_num = reg_num;  
}  
  
}
```

## PERSON CLASS:

```
package com.project;

import java.io.Serializable;

public class Person implements Serializable {
    protected String personId;
    protected String name;
    protected String dateOfBirth;
    protected String contactInfo;
    protected String address;
    protected String email;

    public Person() {
        this.personId = null;
        this.name = null;
        this.dateOfBirth = null;
        this.contactInfo = null;
        this.address = null;
        this.email = null;
    }

    public Person(String personId, String name, String dateOfBirth, String
contactInfo, String address, String email) {
        this.personId = personId;
        this.name = name;
        this.dateOfBirth = dateOfBirth;
        this.contactInfo = contactInfo;
        this.address = address;
        this.email = email;
    }

    public String getPersonId() {
        return personId;
    }

    public void setPersonId(String personId) {
        this.personId = personId;
    }

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

```

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

public String getDateOfBirth() {
    return dateOfBirth;
}

public void setDateOfBirth(String dateOfBirth) {
    this.dateOfBirth = dateOfBirth;
}

public String getContactInfo() {
    return contactInfo;
}

public void setContactInfo(String contactInfo) {
    this.contactInfo = contactInfo;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

public String getEmail() {
    return email;
}

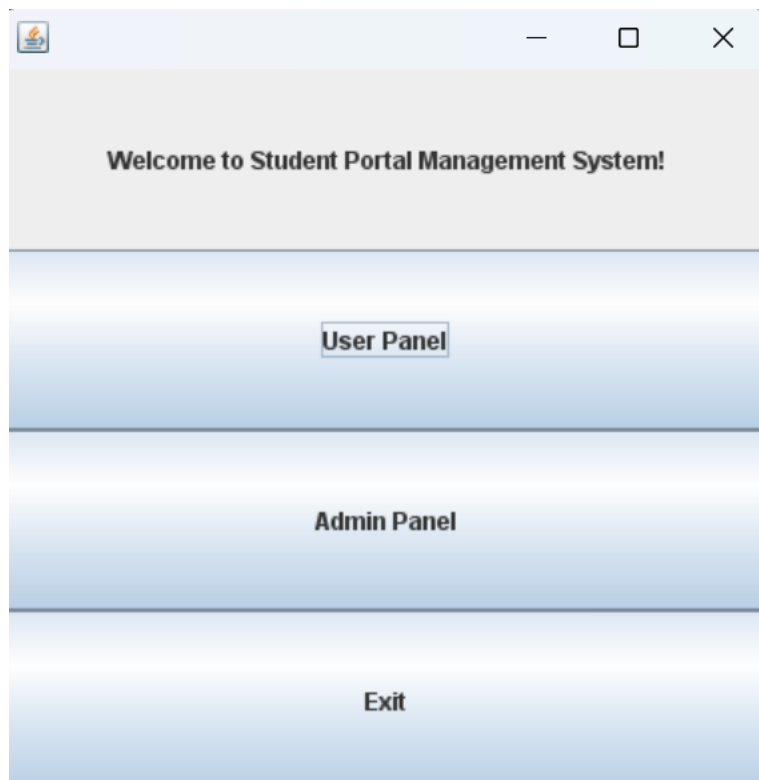
public void setEmail(String email) {
    this.email = email;
}

@Override
public String toString() {
    return "IDUni: " + personId + "\n" + "Name: " + name + "\n" + "Date of
Birth: " + dateOfBirth + "\n"
        + " Contact Info: "
        + contactInfo + "\n" + "Address: " + address + "\n" + "Email: " +
email + "\n";
}

```

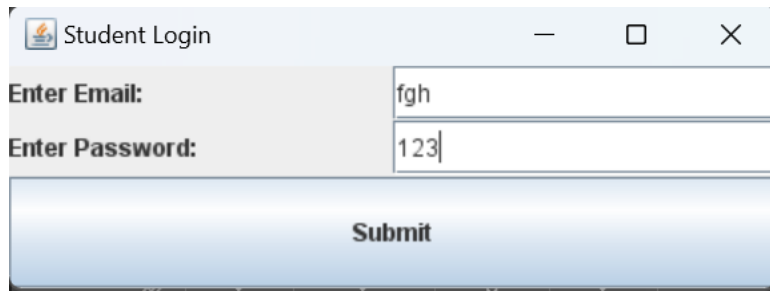
}

## MAIN PANEL:





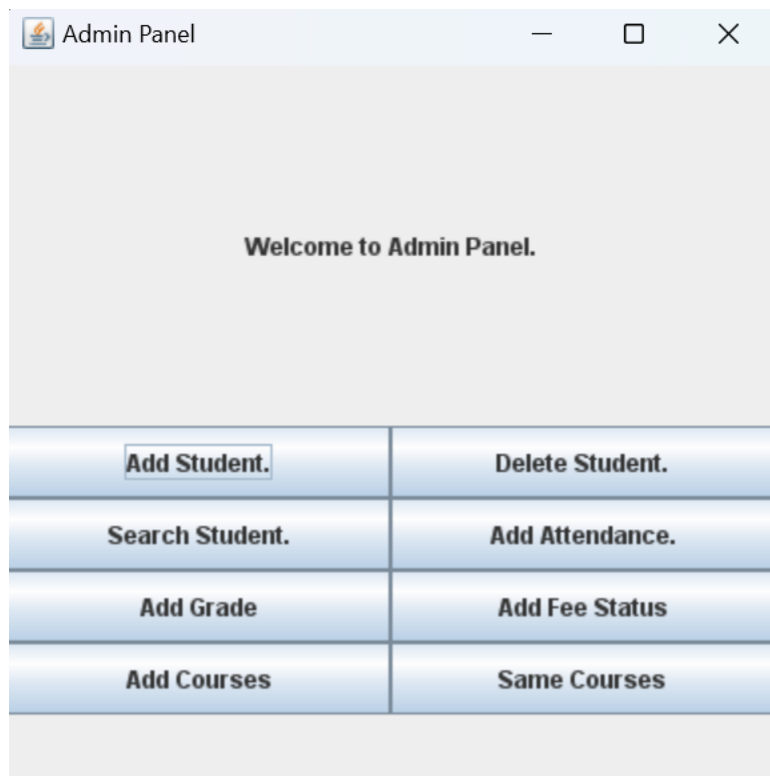
## LOGIN PANEL:



A screenshot of a web browser window titled "Student Login". The window contains two input fields: "Enter Email:" with the text "fgh" and "Enter Password:" with the text "123". Below the input fields is a blue "Submit" button.

Enter Email:	fgh
Enter Password:	123
<b>Submit</b>	

## ADMIN PANEL:



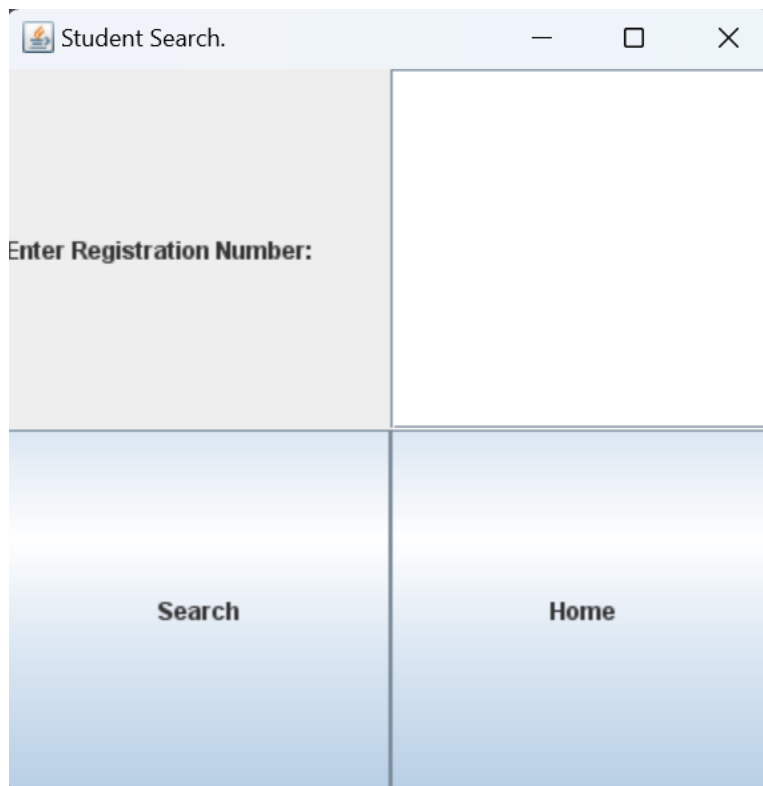
## STUDENT ENTRY:

Student Entry.			
Enter Name:	<input type="text"/>		
Enter Person ID:	<input type="text"/>		
Enter Date of Birth:	<input type="text"/>		
Enter Contact:	<input type="text"/>		
Enter Address:	<input type="text"/>		
Enter Email:	<input type="text"/>		
Enter Registration Number:	<input type="text"/>		
Create Password:	<input type="text"/>		
Submit	Exit	Display	Home

## STUDENT DELETE:

Student Delete.	
Enter Registration Number:	
Delete	Home

## SEARCH STUDENT:



A screenshot of a web application window titled "Student Search." The window has a light blue header bar with a small icon on the left and standard window controls (minimize, maximize, close) on the right. The main content area is divided into four quadrants by a thin blue border. The top-left quadrant has a light gray background and contains the text "Enter Registration Number:". The top-right quadrant is white and empty. The bottom-left quadrant has a light blue gradient background and contains the text "Search". The bottom-right quadrant has a light blue gradient background and contains the text "Home".

Enter Registration Number:	
Search	Home

## ADD ATTENDANCE:

Attendance Update.	
Enter Registration Number:	<input type="text"/>
Enter Course ID:	<input type="text"/>
Enter Attendance Status:	<input type="text"/>
Submit	Home

## ADD GRADE:

Grade Update.	
Enter Registration Number:	<input type="text"/>
Enter Course ID:	<input type="text"/>
Enter Grade Status:	<input type="text"/>
<input type="submit" value="Submit"/>	<input type="button" value="Home"/>

## FEES UPDATE:

Fees Status Update.	
Enter Registration Number:	<input type="text"/>
Enter Total Fees	<input type="text"/>
Enter Fees Status:	<input type="text"/>
Submit	Home



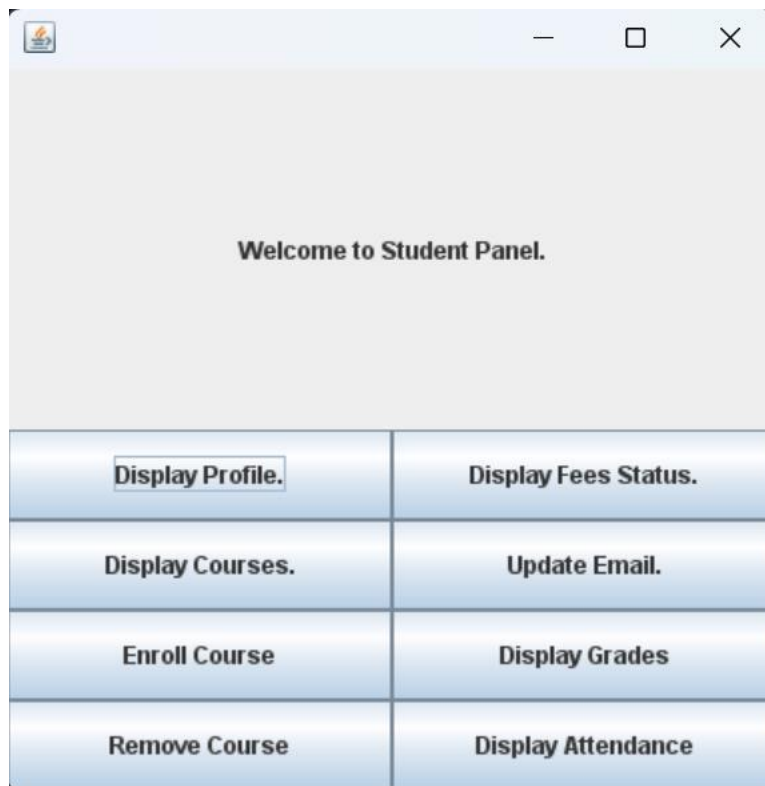
## COURSE UPDATE:

Courses Update.	
Enter Registration Number:	<input type="text"/>
Enter Course ID:	<input type="text"/>
Enter Course Name:	<input type="text"/>
<input type="submit" value="Submit"/>	<input type="button" value="Home"/>

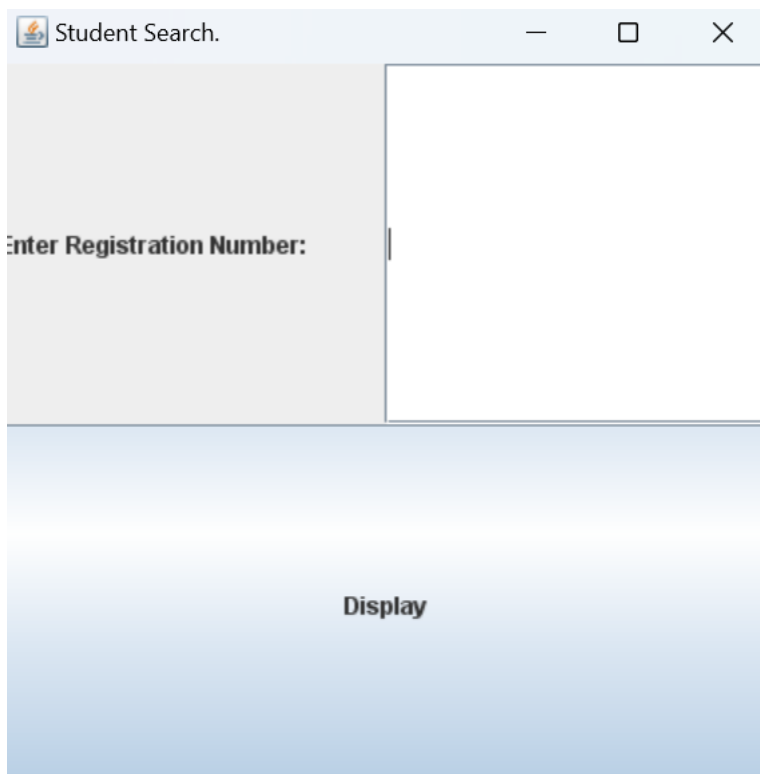
**COURSE SAME REGISTRATION DISPLAY:**

The image shows a window titled "Student Search." with standard window controls (minimize, maximize, close). The window is divided into two main sections. The top section has a light gray background on the left with the text "Enter Course Name:" and a white rectangular input field on the right. The bottom section is a single light blue rectangular area containing the word "Display" in the center, which serves as a button.

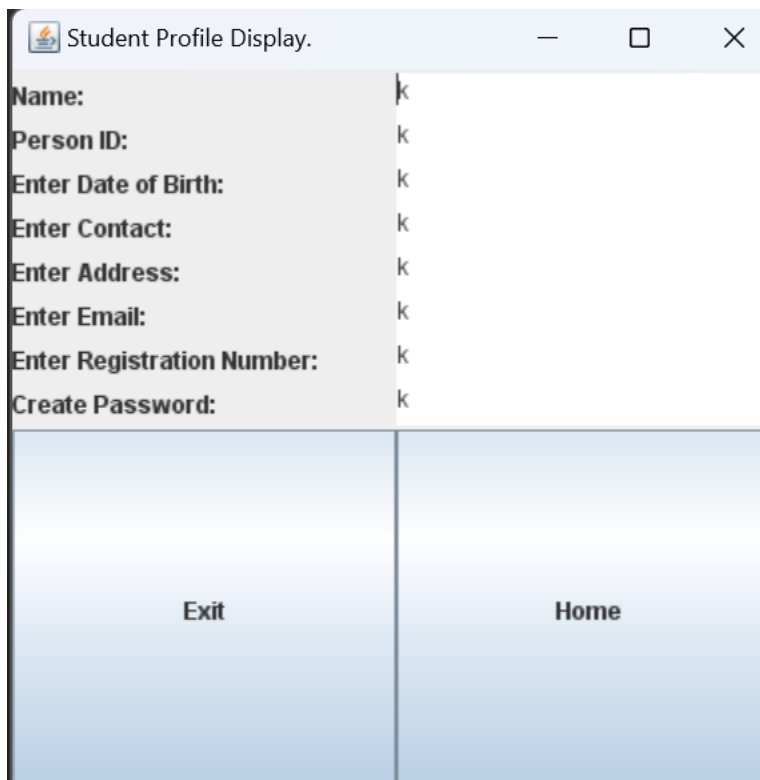
## STUDENT PANEL:



## PROFILE DISPLAY:



A window titled "Student Search." with a light blue header bar containing a small icon, the title, and standard window controls (minimize, maximize, close). The main area is divided into two sections. The top section has a light gray background on the left with the text "Enter Registration Number:" and a vertical text cursor, and a white rectangular input field on the right. The bottom section has a light blue gradient background with the word "Display" centered.



A window titled "Student Profile Display." with a light blue header bar containing a small icon, the title, and standard window controls. The main area is divided into two sections. The top section has a light gray background on the left with labels and a white input field on the right. The labels are: "Name:", "Person ID:", "Enter Date of Birth:", "Enter Contact:", "Enter Address:", "Enter Email:", "Enter Registration Number:", and "Create Password:". The input field contains the letter "k". The bottom section has a light blue gradient background and is divided into two buttons: "Exit" on the left and "Home" on the right.

**FEES STATUS DISPLAY:**

Student Fees.

Enter Registration Number:

Display

Student Search.

Fees Status:

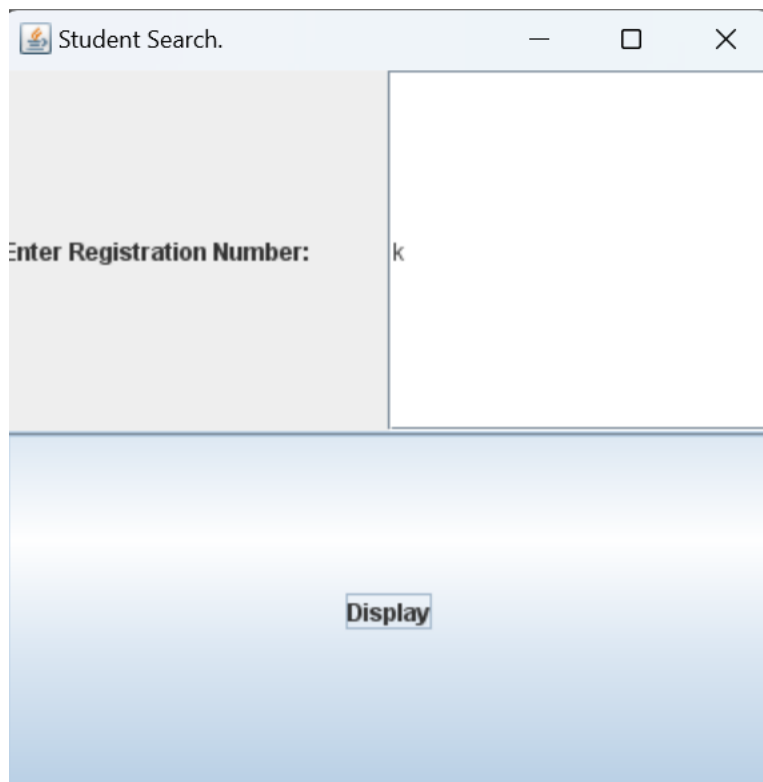
Paid

Total Fees:

9000

Home

## DISPLAY COURSES REGISTERED:

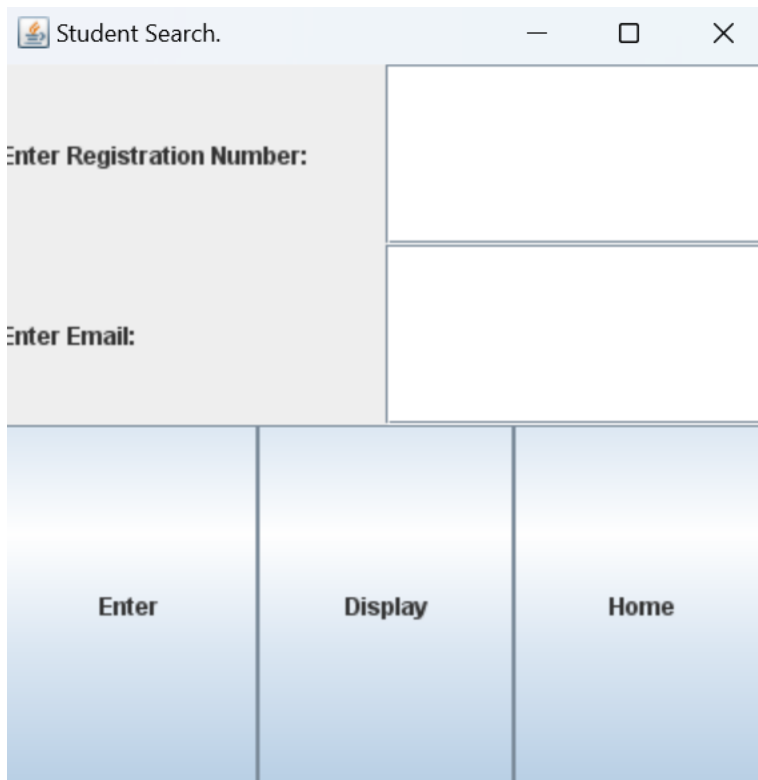


A screenshot of a Windows application window titled "Student Search." The window has a standard title bar with minimize, maximize, and close buttons. The main area is divided into two sections. The top section has a light gray background and contains the text "Enter Registration Number:" followed by a text input field containing the letter "k". The bottom section has a light blue gradient background and contains a single button labeled "Display".



A screenshot of a Windows message dialog box titled "Message". It features a light blue title bar with a close button. The main area has a light gray background and contains an information icon (a lowercase 'i' in a circle) on the left. To the right of the icon, the following text is displayed: "CourseID: 1234", "Reg\_num: k", and "Course name: DAA". At the bottom center of the dialog is an "OK" button.

## EMAIL UPDATE:



A screenshot of a web application window titled "Student Search." The window has a light blue header bar with a small icon on the left and standard window controls (minimize, maximize, close) on the right. The main content area is divided into two columns. The left column has a light gray background and contains two labels: "Enter Registration Number:" and "Enter Email:". The right column has a white background and contains two empty text input fields, one corresponding to each label. Below the input fields, there is a row of three blue buttons with white text: "Enter", "Display", and "Home".

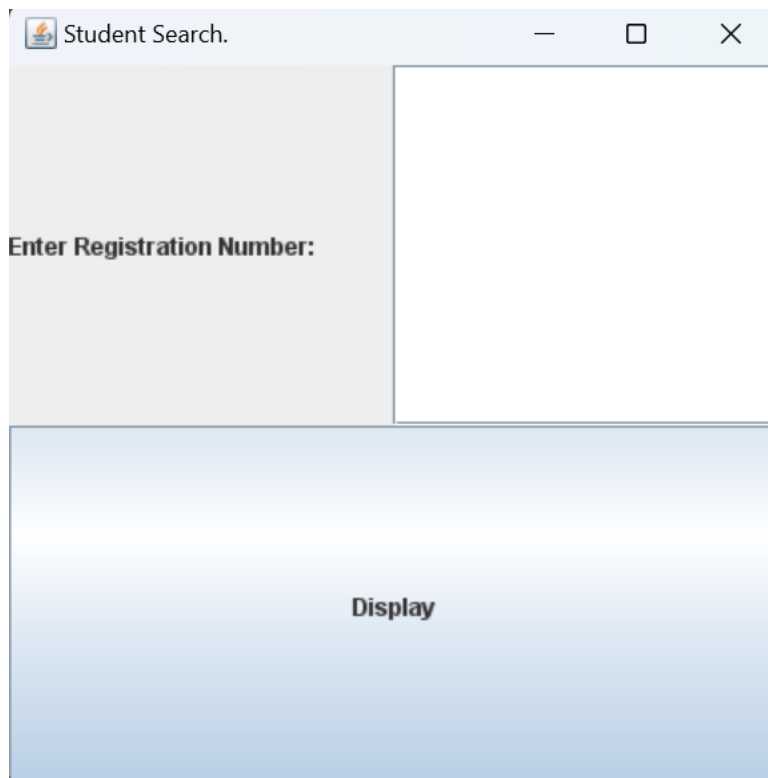
Student Search.		
Enter Registration Number:	<input type="text"/>	
Enter Email:	<input type="text"/>	
Enter	Display	Home

## ENROLL IN COURSE:

Courses Update.	
Enter Registration Number:	<input type="text"/>
Enter Course ID:	<input type="text"/>
Enter Course Name:	<input type="text"/>
<input type="submit" value="Submit"/>	<input type="button" value="Home"/>



## GRADE DISPLAY:



A window titled "Student Search." with standard minimize, maximize, and close buttons. The window is divided into three sections: a label section, an input section, and a button section.

Enter Registration Number:

Display

## REMOVE COURSE:

Student Search.

Enter Registration Number:

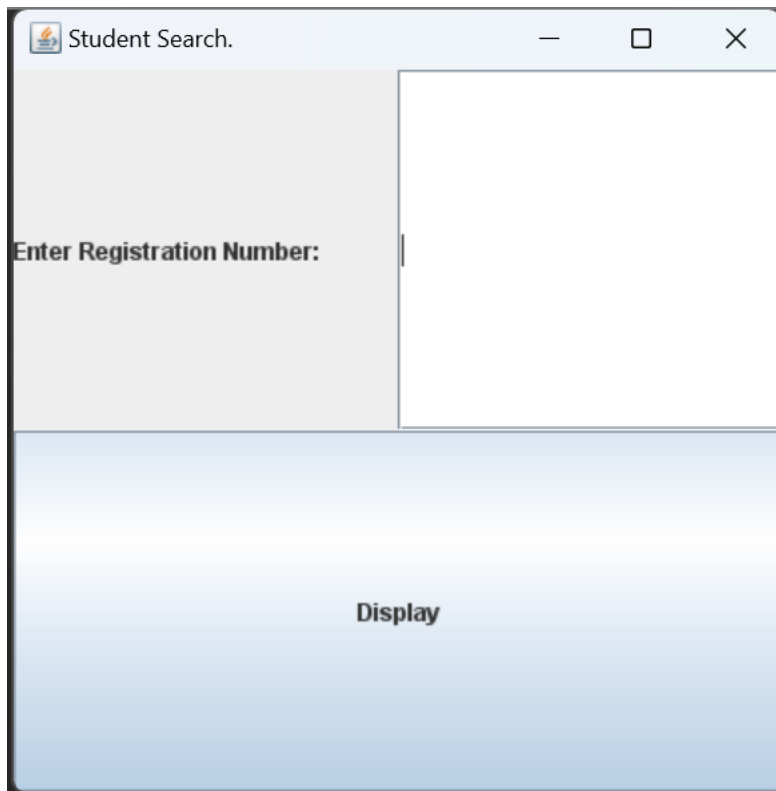
Enter CourseID:

Remove

Display

Home

### ATTENDANCE DISPLAY:



A screenshot of a software window titled "Student Search." with standard Windows window controls (minimize, maximize, close). The window is divided into three main sections. The top-left section is a light gray area containing the text "Enter Registration Number:". To its right is a white rectangular input field. The bottom section is a large light blue area with a gradient, containing the word "Display" centered in a bold black font.

Student Search.

Enter Registration Number:

Display