EAST WEST UNIVERSITY

Course Code: CSE110

Course Title: Object Oriented Programming

Section: 4

Project on "Student Information System"

Team Members

Sheikh Sarafat Hossain [2022-3-60-109]

Zarif Syndeed Sikder [2022-3-60-205]

Rijia Parveen Raya [2022-3-60-192]

SUBMITTED TO

Tanni Mittra [TM]

Senior Lecturer, Department of Computer Science and Engineering

East West University

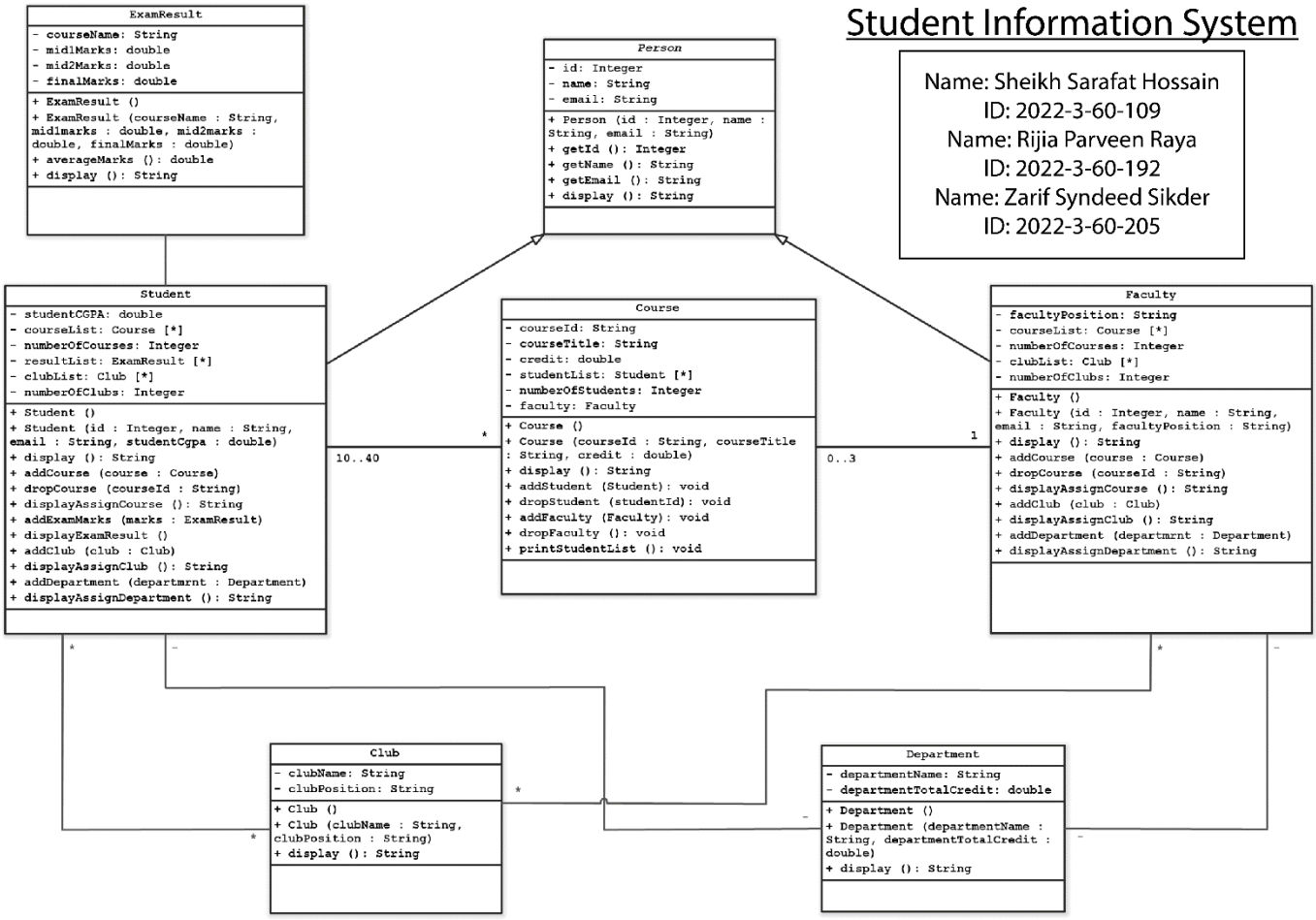
Date of Submission: 17th September, 2023

Table of Contents

1. Introduction

This report discusses the object-oriented design of a **Student Information System (SIS)** implemented in Java. The system includes several classes: **Person**, **Student**, **Faculty**, **Course**, **Department**, **Club**, and **ExamResult**. These classes are organized to represent the entities and relationships within an educational institution.

2. UML Diagram



3. Description of Classes

1. Person Class

The **Person** class is an abstract class which serves as the parent class for both **Student** and **Faculty** class. It contains private attributes for **id**, **name**, and **email**, as well as getter and setter methods for these attributes. The **display()** method is provided to display person-related information.

```
abstract class Person {


    private int id;

    private String name, email;


    public Person() {

    }


    public Person(int id, String name, String email) {

        this.id = id;

        this.name = name;

        this.email = email;

    }


    public int getId() {

        return id;

    }


    public void setId(int id) {

        this.id = id;

    }
```

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

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

    public String getEmail() {
        return email;
    }

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

    public void display() {
        System.out.println("");
    }

}
```

2. Student Class

The **Student** class extends the **Person** class and represents a student in an educational institution. It includes attributes such as **studentCGPA**, **courseList**, **resultList**, **clubList**, and **departmentList**. These attributes are used to store student-specific information, including their CGPA, enrolled courses, exam results, club memberships, and department associations. The class provides methods for adding and dropping courses, displaying course assignments, adding exam results, adding and displaying clubs, and adding and displaying departments.

```
class Student extends Person {


    private double studentCGPA;

    private ArrayList<Course> courseList = new ArrayList<Course>();

    private int numberOfCourses = 0;

    private ArrayList<ExamResult> resultList = new ArrayList<ExamResult>();

    private ArrayList<Club> clubList = new ArrayList<Club>();

    private int numberOfClubs = 0;

    private ArrayList<Department> departmentList = new
ArrayList<Department>();


    public ArrayList<Department> getDepartmentList() {

        return departmentList;

    }


    public void setDepartmentList(ArrayList<Department> departmentList) {

        this.departmentList = departmentList;

    }


    public Student() {

    }
```

```
public Student(int id, String name, String email, double studentCGPA) {
    super(id, name, email);
    this.studentCGPA = studentCGPA;
}

public double getStudentCGPA() {
    return studentCGPA;
}

public void setStudentCGPA(double studentCGPA) {
    this.studentCGPA = studentCGPA;
}

public ArrayList<Course> getCourseList() {
    return courseList;
}

public void setCourseList(ArrayList<Course> courseList) {
    this.courseList = courseList;
}

public int getNumberOfCourses() {
    return numberOfCourses;
}

public void setNumberOfCourse(int numberOfCourses) {
    this.numberOfCourses = numberOfCourses;
```

```
}

public ArrayList<ExamResult> getResultList() {
    return resultList;
}

public void setResultList(ArrayList<ExamResult> resultList) {
    this.resultList = resultList;
}

public ArrayList<Club> getClubList() {
    return clubList;
}

public void setClubList(ArrayList<Club> clubList) {
    this.clubList = clubList;
}

public int getNumberOfClubs() {
    return numberOfClubs;
}

public void setNumberOfClubs(int numberOfClubs) {
    this.numberOfClubs = numberOfClubs;
}

@Override
public void display() {
```

```
        System.out.println("Student Id: " + getId());
        System.out.println("Student Name: " + getName());
        System.out.println("Student Email: " + getEmail());
        System.out.println("Student CGPA: " + getStudentCGPA());

    }

    public void displayAssignCourse(){
        for(Course c: courseList){
            c.display();
        }
    }

    public void addCourse(Course course) {
        courseList.add(course);
        numberOfCourses++;
    }

    public void dropCourse(String coursedrop) {
        int i;
        for (i = 0; i < courseList.size(); i++) {
            if (courseList.get(i).getCourseId().equals(coursedrop)) {
                courseList.remove(i);
                numberOfCourses--;
                break;
            }
        }
    }
```

```
public void addExamMarks(ExamResult er){
    resultList.add(er);
}

public void displayAssignExamResult(){
    for(ExamResult er: resultList){
        er.display();
    }
}

public void addClub(Club c) {
    clubList.add(c);
    numberOfClubs++;

}

public void displayAssignClub() {
    for (Club c : clubList) {
        c.display();
    }
}

public void addDepartment(Department d ){
    departmentList.add(d);
}

public void displayAssignDepartment() {
```

```
        for (Department d : departmentList) {
            d.display();
        }
    }


}
```

3. Faculty Class

The **Faculty** class also extends the **Person** class and represents a faculty member in the institution. It includes attributes for **facultyPosition**, **CourseList**, **clubList**, and **departmentList**. Faculty members can teach courses, be part of clubs, and be associated with departments. The class provides methods for adding and dropping courses, displaying course assignments, adding and displaying clubs, and adding and displaying departments.

```
class Faculty extends Person {


    private String facultyPosition;

    private ArrayList<Course> CourseList = new ArrayList<Course>();

    private int numberOfCourses = 0;

    private ArrayList<Club> clubList = new ArrayList<Club>();

    private ArrayList<Department> departmentList = new
ArrayList<Department>();

    private int numberOfClubs = 0;


    public Faculty() {

    }


    public Faculty(int id, String name, String email, String facultyPosition)
{

        super(id, name, email);

        this.facultyPosition = facultyPosition;

    }


    public String getFacultyPosition() {

        return facultyPosition;

    }
```

```
public ArrayList<Department> getDepartmentList() {
    return departmentList;
}

public void setDepartmentList(ArrayList<Department> departmentList) {
    this.departmentList = departmentList;
}

public void setFacultyPosition(String facultyPosition) {
    this.facultyPosition = facultyPosition;
}

public ArrayList<Course> getCourseList() {
    return CourseList;
}

public void setCourseList(ArrayList<Course> CourseList) {
    this.CourseList = CourseList;
}

public int getNumberOfCourses() {
    return numberOfCourses;
}

public void setNumberOfCourse(int numberOfCourses) {
    this.numberOfCourses = numberOfCourses;
}
```

```
public ArrayList<Club> getClubList() {
    return clubList;
}

public void setClubList(ArrayList<Club> clubList) {
    this.clubList = clubList;
}

public int getNumberOfClubs() {
    return numberOfClubs;
}

public void setNumberOfClubs(int numberOfClubs) {
    this.numberOfClubs = numberOfClubs;
}



@Override
public void display() {
    System.out.println("Faculty Id: " + getId());
    System.out.println("Faculty Name: " + getName());
    System.out.println("Faculty Email: " + getEmail());
    System.out.println("Faculty Position: " + facultyPosition);



}
```

```
public void addCourse(Course c) {
    if (numberOfCourses <= 3) {
        CourseList.add(c);
        numberOfCourses++;
    } else {
        System.out.println("Course capacity reached maximum number.\n"
                + "Cant add more");
    }
}

public void dropCourse(String courseId) {
    for (Course c : CourseList) {
        if (c.getCourseId() == courseId) {
            CourseList.remove(c);
            numberOfCourses--;
        }
    }
}

public void displayAssignCourse() {
    for (Course c : CourseList) {
        c.display();
    }
}

public void addClub(Club c) {

    clubList.add(c);
```

```
        numberOfClubs++;


    }


    public void dropClub(String clubName) {
        for (Club c : clubList) {
            if (c.getClubName() == clubName) {
                clubList.remove(c);
                numberOfClubs++;
            }
        }
    }
    public void displayAssignClub() {
        for (Club c : clubList) {
            c.display();
        }
    }


    public void addDepartment(Department d ){
        departmentList.add(d);
    }
    public void displayAssignDepartment() {
        for (Department d : departmentList) {
            d.display();
        }
    }


}
```

4. Course Class

The **Course** class represents a course offered within the institution. It includes attributes for **courseId**, **courseTitle**, **credit**, and a list of enrolled students. This class also allows faculty assignments to courses and provides methods for adding and dropping students from courses, as well as displaying course information and the list of enrolled students.

```
public class Course {


    private String courseId, courseTitle;

    private double credit;

    private ArrayList<Student> studentList = new ArrayList<Student>();

    private ArrayList<ExamResult> resultList = new ArrayList<ExamResult>();

    private int numberOfStudent = 0;

    private Faculty faculty;


    public Course() {

    }


    public Course(String courseId, String courseTitle, double credit) {

        this.courseId = courseId;

        this.courseTitle = courseTitle;

        this.credit = credit;

    }


    public String getCourseId() {

        return courseId;

    }
```

```
public void setCourseId(String courseId) {
    this.courseId = courseId;
}

public String getCourseTitle() {
    return courseTitle;
}

public void setCourseTitle(String courseTitle) {
    this.courseTitle = courseTitle;
}

public double getCredit() {
    return credit;
}

public void setCredit(double credit) {
    this.credit = credit;
}

public ArrayList<Student> getStudentList() {
    return studentList;
}

public void setStudentList(ArrayList<Student> studentList) {
    this.studentList = studentList;
}
```

```
public int getNumberOfStudent() {
    return numberOfStudent;
}

public Faculty getFaculty() {
    return faculty;
}

public void setFaculty(Faculty faculty) {
    this.faculty = faculty;
}

public void display() {
    System.out.println("Course Id: " + courseId);
    System.out.println("Course Title : " + courseTitle);
    System.out.println("Course Credit : " + credit);
}

public void addStudent(Student s) {
    if (numberOfStudent <= 40) {
        studentList.add(s);
        numberOfStudent++;
    } else {
        System.out.println("Student capacity reached maximum number.\n"
                + "Cant add more");
    }
}
```

```
public void dropStudent(int studentId) {
    int i;
    for (i = 0; i < studentList.size(); i++) {
        if (studentList.get(i).getId() == studentId){
            studentList.remove(i);
            numberOfStudent--;
            break;
        }

    }

}

public void addMarks(ExamResult e){
    resultList.add(e);
}

public void printMarks(){
    for(ExamResult i : resultList){
        i.display();
    }
}

public void addFaculty(Faculty faculty) {
    this.faculty = faculty;
}

public void dropFaculty() {
```

```
        this.faculty = null;
    }

    public void printStudentList(){
        for(Student s: studentList){
            s.display();
        }
    }

}
```

5. Department Class

The **Department** class represents academic departments within the institution. It includes attributes for **departmentName** and **departmentTotalCredit**. This class provides methods for displaying department information.

```
public class Department {

    private String departmentName;
    private double departmentTotalCredit;

    public Department() {
    }

    public Department(String departmentName, double departmentTotalCredit) {
        this.departmentName = departmentName;
        this.departmentTotalCredit = departmentTotalCredit;
    }

    public String getDepartmentName() {
        return departmentName;
    }

    public void setDepartmentName(String departmentName) {
        this.departmentName = departmentName;
```

```
    }

    public double getDepartmentTotalCredit() {
        return departmentTotalCredit;
    }

    public void setDepartmentTotalCredit(double departmentTotalCredit) {
        this.departmentTotalCredit = departmentTotalCredit;
    }

    public void display() {
        System.out.println("Department Name: " + departmentName);
        System.out.println("Department Total Credit: " + departmentTotalCredit);


    }

}
```

6. Club Class

The **Club** class represents clubs or organizations within the institution. It includes attributes for **clubName** and **clubPosition**. This class provides methods for displaying club information.

```
public class Club {

    private String clubName, clubPosition;

    public Club() {
    }

    public Club(String clubName, String clubPosition) {
        this.clubName = clubName;
        this.clubPosition = clubPosition;
    }

    public String getClubName() {
        return clubName;
    }

    public void setClubName(String clubName) {

        this.clubName = clubName;
    }

    public String getClubPosition() {
        return clubPosition;
```

```
    }

    public void setClubPosition(String clubPosition) {

        this.clubPosition = clubPosition;
    }

    public void display() {
        System.out.println("Club Name: " + clubName);
        System.out.println("Club Position: " + clubPosition);
    }

}
```

7. ExamResult Class

The **ExamResult** class represents the exam results for courses. It includes attributes for **courseName**, **mid1Marks**, **mid2Marks**, and **finalMarks**. This class calculates and displays the average marks for a course.

```
public class ExamResult {


    private String courseName;

    private double mid1Marks, mid2Marks, finalMarks;

    private ArrayList<Course> CourseList = new ArrayList<Course>();

    public ExamResult() {

    }


    public ExamResult(String courseName, double mid1Marks, double mid2Marks,
double finalMarks) {

        this.courseName = courseName;

        this.mid1Marks = mid1Marks;

        this.mid2Marks = mid2Marks;

        this.finalMarks = finalMarks;

    }


    public String getCourseName() {

        return courseName;

    }


    public void setCourseName(String courseName) {

        this.courseName = courseName;
```

```
}

public double getMid1Marks() {
    return mid1Marks;
}

public void setMid1Marks(double mid1Marks) {
    this.mid1Marks = mid1Marks;
}

public double getMid2Marks() {
    return mid2Marks;
}

public void setMid2Marks(double mid2Marks) {
    this.mid2Marks = mid2Marks;
}

public double getFinalMarks() {
    return finalMarks;
}

public void setFinalMarks(double finalMarks) {
    this.finalMarks = finalMarks;
}

public double averageMarks(){
    return (mid1Marks+mid2Marks+finalMarks) / 3;
```

```
    }
    public void addCourse(Course e){
        CourseList.add(e);
    }

    public void courseDisplay(){
        for(Course i: CourseList){
            i.display();
        }
    }

    public void display() {
        System.out.println("Mid-Term-1 marks: " + mid1Marks);
        System.out.println("Mid-Term-2 marks: " + mid2Marks);
        System.out.println("Final-Term marks: " + finalMarks);

    }

}
```

4. Main Class

The main class of the **Student Information System (SIS)** program serves as the entry point and control center for managing student information and related data. It utilizes a menu-driven approach, where users can choose from a range of options to perform actions within the system. The key functionalities provided by the main class include:

1. Student Management

- Users can add, drop, update, search, and print student information.
- The system allows for the addition of students' personal details, including ID, name, email, and CGPA.
- Students can be assigned to courses, clubs, and departments.
- Faculty members and administrators can search for specific students and retrieve their information.
- A comprehensive print function displays all student details along with their course, club, and department assignments.

2. Faculty Management

- Similar to student management, faculty management offers options to add, drop, update, search, and print faculty information.
- Faculty members' details, such as ID, name, email, and position, can be added and updated.
- Faculty members can be assigned to courses, clubs, and departments.
- Faculty members can be searched by ID, and their details can be retrieved.
- A print function displays all faculty details along with their course, club, and department assignments.

3. Course Management

- Users can add, drop, update, search, and print course information.
- Course details, including ID, title, and credit, can be added and updated.
- Faculty members can be assigned to teach courses.
- Enrolled students can be managed by adding or dropping them from courses.

- Course details can be retrieved by specifying the course ID.
- A print function displays all course details, including assigned faculty members and enrolled students.

4. Club Management

- Club information can be added, updated, searched, and printed.
- Clubs' details, such as name and position, can be added or updated.
- Faculty members and administrators can search for clubs by name.
- A print function displays all club details.

5. Department Management

- Department information can be added, updated, searched, and printed.
- Department details, including name and total departmental credit, can be added and updated.
- Faculty members can be assigned to departments.
- Courses can be associated with departments.
- Department information can be retrieved by specifying the department name.
- A print function displays all department details, including assigned faculty members and associated courses.

6. Update and Search Functionalities

- Users can update and search for information related to students, faculty, courses, clubs, and departments.
- The update functionality allows for modifications to existing data, while the search functionality retrieves specific information from the system.

7. Exit

- Users have the option to exit the program, ending their interaction with the SIS.

```
import java.util.Objects;
import java.util.Scanner;
import java.util.ArrayList;

public class MainClass {

    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);

        try {
            ArrayList<Student> studentList = new ArrayList<Student>();
            ArrayList<Faculty> facultyList = new ArrayList<Faculty>();
            ArrayList<Course> CourseList = new ArrayList<Course>();
            ArrayList<Club> clubList = new ArrayList<Club>();
            ArrayList<ExamResult> resultList = new ArrayList<ExamResult>();
            ArrayList<Department> departmentList = new
ArrayList<Department>();

            boolean res = true;
            while (res) {
                System.out.println("\t\t\t\t----------------------------");
                System.out.println("\t\t\t\t Student Information System");
                System.out.println("\t\t\t\t----------------------------");
                System.out.println("");
                System.out.println("a. Student Panel: ");
                System.out.println("b. Admin & Faculty Panel: ");
                System.out.println("0. Exit: ");
                char mainfunction = sc.next().charAt(0);
```

```
                switch (mainfunction) {
                    //case a started
                    case 'a': {
                        boolean stp = true;
                        while (stp) {
                            System.out.println("\t\t\t\t-------------");
                            System.out.println("\t\t\t\tStudent Panel");
                            System.out.println("\t\t\t\t-------------");
                            System.out.println("a. Add: "); //done
                            System.out.println("b. Drop: "); //done
                            System.out.println("c. Search: "); //done
                            System.out.println("d. Print: ");//done
                            System.out.println("e. Back: ");//done
                            char studentpanel = sc.next().charAt(0);
                            switch (studentpanel) {
                                case 'a': { //add
                                    boolean sp = true;
                                    while (sp) {
                                        System.out.println("a. Add a Student:
"); //done
                                        System.out.println("b. Add Student to
Course: "); //done
                                        System.out.println("c. Add Student to
Club: "); //done
                                        System.out.println("d. Add Student to
Department: "); //done
                                        System.out.println("e. Back: ");
//done

                                        char student = sc.next().charAt(0);
```

```
                                        switch (student) {

                                            case 'a': {
                                                System.out.print("Enter
Student ID: ");
                                                int sid = sc.nextInt();
                                                sc.nextLine();
                                                System.out.print("Enter
Student Name: ");
                                                String sname = sc.nextLine();
                                                System.out.print("Enter
Student Email:");
                                                String semail =
sc.nextLine();
                                                System.out.print("Enter
Student CGPA: ");
                                                double scgpa =
sc.nextDouble();
                                                studentList.add(new
Student(sid, sname, semail, scgpa));
                                                System.out.println("Student
Added Successfully!");
                                                break;
                                            }
                                            case 'b': {
                                                System.out.print("Enter
Student ID to add course: ");
                                                int id = sc.nextInt();
                                                sc.nextLine();
                                                for (int ii = 0; ii <
studentList.size(); ii++) {
                                                    if
(studentList.get(ii).getId() == id) {
```

```
System.out.print("Enter Course ID: ");
                                                        String cid =
sc.nextLine();
                                                        for (int jj = 0; jj <
CourseList.size(); jj++) {
                                                            if
(Objects.equals(CourseList.get(jj).getCourseId(), cid)) {
studentList.get(ii).addCourse(CourseList.get(jj));
CourseList.get(jj).addStudent(studentList.get(ii));
System.out.println("Course added successfully");
                                                                break;
                                                            }
                                                        }
                                                        break;
                                                    }
                                                }
                                                break;
                                            }

                                            case 'c': {
                                                System.out.print("Enter
Student ID to add club: ");
                                                int sid = sc.nextInt();
                                                sc.nextLine();
                                                for (int ii = 0; ii <
studentList.size(); ii++) {
                                                    if
(studentList.get(ii).getId() == sid) {
```

```
System.out.print("Enter Club Name: ");
                                                    String cname =
sc.nextLine();
                                                    for (int jj = 0; jj <
clubList.size(); jj++) {
                                                        if
(Objects.equals(clubList.get(jj).getClubName(), cname)) {
studentList.get(ii).addClub(clubList.get(jj));
System.out.println("Club added successfully");
                                                            break;
                                                        }
                                                    }
                                                    break;
                                                }
                                            }
                                            break;
                                        }

                                        case 'd': {
                                            System.out.print("Enter
Student ID to add Department: ");
                                            int sid = sc.nextInt();
                                            sc.nextLine();
                                            for (int ii = 0; ii <
studentList.size(); ii++) {
                                                if
(studentList.get(ii).getId() == sid) {
System.out.print("Enter Department Name: ");
```

```
                                                    String dname =
sc.nextLine();
                                                    for (int jj = 0; jj <
departmentList.size(); jj++) {
                                                        if
(Objects.equals(departmentList.get(jj).getDepartmentName(), dname)) {
studentList.get(ii).addDepartment(departmentList.get(jj));
System.out.println("Department added successfully");
                                                            break;
                                                        }
                                                    }
                                                    break;
                                                }
                                            }
                                            break;
                                        }
                                        case 'e': { //back
                                            sp = false;
                                            break;
                                        }
                                        default: {
System.out.println("Invalid");
                                            sp = true;
                                            stp = true;
                                            break;
                                        }
```

```
                    }


                } //add while end
                break;
            } // a case end


            case 'b': { //drop b case started
                boolean sp = true;
                while (sp) {
                    System.out.println("a. Drop Course:
");
                    System.out.println("b. Back: ");
                    char drop = sc.next().charAt(0);
                    switch (drop) {
                        case 'a': {
                            System.out.print("Enter
Course ID: ");
                            sc.nextLine();
                            String cid = sc.nextLine();
                            for (int i = 0; i <
CourseList.size(); i++) {
                                if
(Objects.equals(CourseList.get(i).getCourseId(), cid)) {
                                    for (int j = 0; j <
studentList.size(); j++) {

System.out.print("Enter Student ID: ");
                                        int sid =
sc.nextInt();
                                        if
(Objects.equals(studentList.get(j).getId(), sid)) {
```

```
CourseList.get(i).dropStudent(sid);

studentList.get(j).dropCourse(cid);

System.out.println("Dropped Successfully!");
                                                                    break;
                                                                }
                                                            }


                                                        }
                                                    }
                                                    break;
                                                }

                                                case 'b': {
                                                    sp = false;
                                                    break;
                                                }
                                                default: {
                                                    System.out.println("Invalid
Input Try Again!");
                                                    sp = true;
                                                    break;
                                                }
                                            }

                                        }
                                        break;
                                    }
```

```
                                case 'c': { //search


                                    boolean sp = true;
                                    while (sp) {
                                        System.out.println("a. Search Course:
");
                                        System.out.println("b. Search
Faculty: ");
                                        System.out.println("c. Search Club:
");
                                        System.out.println("d. Search
Department: ");
                                        System.out.println("e. Search Course
Result: ");
                                        System.out.println("f. Back: ");
                                        char ps = sc.next().charAt(0);
                                        switch (ps) {
                                            case 'a': {
                                                System.out.print("Enter
Course ID: ");
                                                sc.nextLine();
                                                String cid = sc.nextLine();


                                                for (Course i : CourseList) {
                                                    if
(Objects.equals(i.getCourseId(), cid)) {
                                                        i.display();
                                                        //
i.getFaculty().display();
                                                    } else {
```

```
System.out.println("Not Found!");
                                                        }
                                                    }

                                                    break;
                                                }

                                                case 'b': {
                                                    System.out.println("Enter
Faculty ID: ");
                                                    int fid = sc.nextInt();
                                                    for (Faculty i : facultyList)
{
                                                        if (fid == i.getId()) {
                                                            i.display();
                                                        } else {

System.out.println("Not Found!");
                                                        }
                                                    }

                                                    break;
                                                }
                                                case 'c': {
                                                    System.out.print("Enter Club
Name: ");
                                                    sc.nextLine();
                                                    String cname = sc.nextLine();
```

```
                                                for (Club i : clubList) {
                                                    if
(Objects.equals(i.getClubName(), cname)) {
                                                        i.display();
                                                        //
i.getFaculty().display();
                                                    } else {

System.out.println("Not Found!");
                                                    }
                                                }

                                                break;
                                            }
                                            case 'd': {
                                                System.out.print("Enter
Department name: ");
                                                sc.nextLine();
                                                String dname = sc.nextLine();

                                                for (Department i :
departmentList) {
                                                    if
(Objects.equals(i.getDepartmentName(), dname)) {
                                                        i.display();
                                                        //
i.getFaculty().display();
                                                    } else {

System.out.println("Not Found!");
                                                    }
                                                }
```

```
                    break;
                }
                case 'e': {
                    System.out.print("Enter
Course Name: ");
                    String cname = sc.nextLine();

                    for (Course i : CourseList) {
                        if
(cname.equals(i.getCourseId())) {
                            i.printMarks();
                        } else {

System.out.println("Not Found!");
                        }
                    }
                    break;
                }
                case 'f': {
                    sp = false;
                    break;
                }
                default: {
                    System.out.println("Invalid
Input Please Try Again!");
                    break;
                }
            }
```

```
                }

                break;
            }
            case 'd': { //print
                boolean sp = true;
                while (sp) {
                    System.out.println("a. Print Student
Information: ");
                    System.out.println("b. Print Student
Department: ");
                    System.out.println("c. Print Student
Courses: ");
                    System.out.println("d. Print Student
Courses Result: ");
                    System.out.println("e. Print Student
Club: ");
                    System.out.println("f. Back: ");
                    char ps = sc.next().charAt(0);
                    switch (ps) {
                        case 'a': {
                            for (Student i : studentList)
{
                                i.display();
                            }
                            break;
                        }
                        case 'b': {
                            for (Student i : studentList)
{
i.displayAssignDepartment();
```

```
        }
        break;
    }
    case 'c': {
        for (Student i : studentList)
{


            i.displayAssignCourse();


        }
        break;
    }

    case 'd': {
        for (Student i : studentList)
{


i.displayAssignExamResult();


        }
        break;
    }
    case 'e': {
        for (Student i : studentList)
{


            i.displayAssignClub();
        }
        break;
    }
```

```
                                        case 'f': {
                                            sp = false;
                                            break;
                                        }

                                        default: {
                                            System.out.println("Invalid
Input Please Try Again!");
                                            break;
                                        }
                                    }
                                }
                                break;
                            }

                            case 'e': { //back
                                stp = false;
                                break;
                            }

                            default: {
                                System.out.println("Invalid Input Please
Try Again!");
                                stp = true;
                                break;
                            }
                        }
                    } //student panel while loop ends
```

```
                    break;
                }
                //case a ended

                //case b started (Admin & Faculty Panel)
                case 'b': {
                    boolean stp = true;
                    while (stp) {
                        System.out.println("\t\t\t\t----------------------
");
                        System.out.println("\t\t\t\tAdmin & Faculty
Panel");
                        System.out.println("\t\t\t\t----------------------
");
                        System.out.println("a. Add: ");
                        System.out.println("b. Drop: ");
                        System.out.println("c. Update: ");
                        System.out.println("d. Search: ");
                        System.out.println("e. Print: ");
                        System.out.println("f. Back: ");
                        char studentpanel = sc.next().charAt(0);
                        switch (studentpanel) {
                            case 'a': { //add
                                boolean sp = true;
                                while (sp) {
                                    System.out.println("a. Add Faculty:
");
                                    System.out.println("b. Add Course:
");
                                    System.out.println("c. Add Club: ");
```

```
                                        System.out.println("d. Add
Department: ");
                                        System.out.println("e. Add Faculty to
Course: ");
                                        System.out.println("f. Add Faculty to
Club: ");
                                        System.out.println("g. Add Faculty to
Department: ");
                                        System.out.println("h. Add Result to
Course: ");
                                        System.out.println("i. Back: ");
                                        char admin = sc.next().charAt(0);
                                        switch (admin) {
                                            case 'a': { //add faculty
                                                System.out.print("Enter
Faculty ID: ");
                                                int fid = sc.nextInt();
                                                sc.nextLine();
                                                System.out.print("Enter
Faculty Name: ");
                                                String fname = sc.nextLine();
                                                System.out.print("Enter
Faculty Email: ");
                                                String femail =
sc.nextLine();
                                                System.out.print("Enter
Faculty Position: ");
                                                String fpos = sc.nextLine();
                                                facultyList.add(new
Faculty(fid, fname, femail, fpos));
                                                System.out.println("Faculty
Added Successfully!");
                                                break;
```

```
                                                    }
                                                    case 'b': { //add course


                                                        sc.nextLine();
                                                        System.out.print("Enter
Course ID: ");
                                                        String cid = sc.nextLine();
                                                        System.out.print("Enter
Course Title: ");
                                                        String ct = sc.nextLine();
                                                        System.out.print("Enter
Course Credit: ");
                                                        double cc = sc.nextDouble();
                                                        sc.nextLine();
                                                        CourseList.add(new
Course(cid, ct, cc));
                                                        System.out.println("Course
Added Successfully!");
                                                        break;
                                                    }


                                                    case 'c': { //add club
                                                        System.out.print("Enter Club
Name: ");
                                                        sc.nextLine();
                                                        String cname = sc.nextLine();
                                                        System.out.print("Enter Club
Position: ");
                                                        String cp = sc.nextLine();
                                                        clubList.add(new Club(cname,
cp));
```

```
                                                System.out.println("Club
Added Successfully!");
                                                break;

                                            }

                                            case 'd': {
                                                System.out.print("Enter
Department Name: ");
                                                sc.nextLine();
                                                String dname = sc.nextLine();
                                                System.out.print("Enter total
Departmental Credit: ");
                                                Double dc = sc.nextDouble();
                                                departmentList.add(new
Department(dname, dc));
System.out.println("Department Added Successfully!");
                                                break;
                                            }

                                            case 'e': {
                                                System.out.print("Enter
Faculty ID to add Course:");
                                                int fid = sc.nextInt();
                                                sc.nextLine();

                                                for (int ii = 0; ii <
facultyList.size(); ii++) {
                                                    if
(facultyList.get(ii).getId() == fid) {
```

```
System.out.print("Enter Course ID: ");
                                                    String cid =
sc.nextLine();
                                                    for (int jj = 0; jj <
CourseList.size(); jj++) {
                                                        if
(Objects.equals(CourseList.get(jj).getCourseId(), cid)) {
facultyList.get(ii).addCourse(CourseList.get(jj));
CourseList.get(jj).addFaculty(facultyList.get(ii));
System.out.println("Course added successfully");
                                                            break;
                                                        }
                                                    }
                                                    break;
                                                }
                                            }
                                            break;
                                            // facultyList.
                                        }

                                        case 'f': {
                                            System.out.print("Enter
Faculty ID to add Club:");
                                            int fid = sc.nextInt();
                                            sc.nextLine();

                                            for (int ii = 0; ii <
facultyList.size(); ii++) {
```

```
                                                    if
(facultyList.get(ii).getId() == fid) {

System.out.print("Enter Club Name: ");

                                                        String cname =
sc.nextLine();

                                                        for (int jj = 0; jj <
clubList.size(); jj++) {

                                                            if
(Objects.equals(clubList.get(jj).getClubName(), cname)) {

facultyList.get(ii).addClub(clubList.get(jj));

System.out.println("Club added successfully");

                                                                break;

                                                            }

                                                        }

                                                        break;

                                                    }

                                                }


                                                case 'g': {

                                                    System.out.print("Enter
Faculty ID to add Department:");

                                                    int fid = sc.nextInt();

                                                    sc.nextLine();


                                                    for (int ii = 0; ii <
facultyList.size(); ii++) {
```

```
                                                    if
(facultyList.get(ii).getId() == fid) {

System.out.print("Enter Department Name: ");

                                                        String dname =
sc.nextLine();

                                                        for (int jj = 0; jj <
departmentList.size(); jj++) {

                                                            if
(Objects.equals(departmentList.get(jj).getDepartmentName(), dname)) {

facultyList.get(ii).addDepartment(departmentList.get(jj));

System.out.println("Department added successfully");

                                                                break;

                                                            }

                                                        }

                                                        break;

                                                    }

                                                }

                                                break;

                                            }


                                            case 'h': {//result

                                                System.out.print("Enter
Student ID: ");

                                                int sid = sc.nextInt();

                                                sc.nextLine();

                                                System.out.print("Enter
Course ID to add Exam Results:");

                                                String cid = sc.nextLine();
```

```
                                                System.out.println("Enter Mid
1 Result: ");
                                                double mid1 = sc.nextInt();
                                                System.out.println("Enter Mid
2 Result: ");
                                                double mid2 = sc.nextInt();
                                                System.out.println("Enter
Final Result: ");
                                                double finall = sc.nextInt();
                                                resultList.add(new
ExamResult(cid, mid1, mid2, finall));


                                                for (int j = 0; j <
resultList.size(); j++) {
                                                    if
(Objects.equals(resultList.get(j).getCourseName(), cid)) {
                                                        for (int i = 0; i <
CourseList.size(); i++) {
                                                            if
(Objects.equals(CourseList.get(i), cid)) {
                                                                for (int k =
0; k < studentList.size(); k++) {

CourseList.get(i).addMarks(resultList.get(j));

studentList.get(k).addExamMarks(resultList.get(j));
                                                                    break;
                                                                }


                                                            }
                                                        }
                                                    }
                                                }
```

```
                    break;
                }

                case 'i': { //back
                    sp = false;
                    break;
                }

                default: {
                    System.out.println("Invalid
Input");
                    sp = true;
                    stp = true;
                    break;
                }
            }

        } //add while end
        break;
    } // a case end
    case 'b': { //drop b case started
        boolean sp = true;
        while (sp) {
            System.out.println("a. Drop Student
From Course: ");
            System.out.println("b. Drop Faculty
From Course: ");
            System.out.println("c. Back: ");
```

```
                                        char drop = sc.next().charAt(0);
                                        switch (drop) {
                                            case 'a': {
                                                System.out.print("Enter
Course ID: ");
                                                sc.nextLine();
                                                String cid = sc.nextLine();
                                                for (int i = 0; i <
CourseList.size(); i++) {
                                                    if
(Objects.equals(CourseList.get(i).getCourseId(), cid)) {
                                                        for (int j = 0; j <
studentList.size(); j++) {

System.out.print("Enter Student ID: ");
                                                            int sid =
sc.nextInt();
                                                            if
(Objects.equals(studentList.get(j).getId(), sid)) {

CourseList.get(i).dropStudent(sid);

studentList.get(j).dropCourse(cid);

System.out.println("Dropped Successfully!");
                                                                break;
                                                            }
                                                        }

                                                    }
                                                }
                                                break;
```

```
                                        }

                                        case 'b': {
                                            System.out.print("Enter
Faculty ID: ");
                                            int fid = sc.nextInt();
                                            sc.nextLine();

                                            for (int i = 0; i <
facultyList.size(); i++) {
                                                if
(Objects.equals(facultyList.get(i).getId(), fid)) {
System.out.print("Enter Course ID: ");
                                                    String cid =
sc.nextLine();
                                                    for (int j = 0; j <
CourseList.size(); j++) {
                                                        if
(Objects.equals(CourseList.get(i).getCourseId(), cid)) {
CourseList.get(j).dropFaculty();
facultyList.get(i).dropCourse(cid);
System.out.println("Dropped Successfully!");
                                                            break;
                                                        } else {
System.out.println("Failed to Drop!");
                                                        }
                                                    }
```

```
                    }
                }
                break;

            }

            case 'c': {
                sp = false;
                break;
            }

            default: {
                System.out.println("Invalid
Input Try Again!");
                sp = true;
                break;
            }
        }

    }
    break;
}

case 'c': {

    boolean sp = true;
    while (sp) {

        System.out.println("\nUpdate Menu");
```

```
                                        System.out.println("a. Update Student
Info: ");
                                        System.out.println("b. Update Course
Info: ");
                                        System.out.println("c. Update Faculty
Info: ");
                                        System.out.println("d. Update Club
Info: ");
                                        System.out.println("e. Update
Department Info: ");
                                        System.out.println("f. Back: ");


                                        System.out.print("Enter your choice:
");
                                        char ucc = sc.next().charAt(0);


                                        switch (ucc) {


                                            case 'a': {


                                                System.out.print("Enter
Student ID to update: ");
                                                int studentId = sc.nextInt();
                                                sc.nextLine();


                                                for (int i = 0; i <
studentList.size(); i++) {
                                                    if
(studentList.get(i).getId() == studentId) {
System.out.print("Enter Student's new ID: ");
                                                        int sid =
sc.nextInt();
```

```
                                                    sc.nextLine();

studentList.get(i).setId(sid);

System.out.print("Enter Student's new Name: ");
                                                    String sname =
sc.nextLine();

studentList.get(i).setName(sname);

System.out.print("Enter Student's new Email: ");
                                                    String smail =
sc.nextLine();

studentList.get(i).setEmail(smail);

System.out.print("Enter Student's new CGPA: ");
                                                    Double sCGPA =
sc.nextDouble();

studentList.get(i).setStudentCGPA(sCGPA);

System.out.println("ID successfully updated for: " + sid);
                                                        break;
                                                    } else {

System.out.println("Not found");
                                                    }
                                                }
                                                break;
                                            }

                                            case 'b': {
```

```
                                                System.out.print("Enter
Course ID to update: ");
                                                String courseId =
sc.nextLine();


                                                for (int i = 0; i <
CourseList.size(); i++) {
                                                    if
(CourseList.get(i).getCourseId().equals(courseId)) {
System.out.print("Enter Course's new ID:");
                                                        String cid =
sc.nextLine();
CourseList.get(i).setCourseId(cid);
System.out.println("Enter Course's new Title: ");
                                                        String ct =
sc.nextLine();
CourseList.get(i).setCourseTitle(ct);
System.out.println("Enter Course's new Credit: ");
                                                        double cc =
sc.nextDouble();
CourseList.get(i).setCredit(cc);
System.out.println("ID successfully updated for: " + cid);
                                                        break;
                                                    } else {
System.out.println("Not found");
                                                    }
```

```
                }
                break;
            }

            case 'c': {

                System.out.print("Enter
Faculty ID to update: ");
                int facultyId = sc.nextInt();
                sc.nextLine();

                for (int i = 0; i <
facultyList.size(); i++) {
                    if
(facultyList.get(i).getId() == facultyId) {
System.out.print("Enter Faculty's new ID: ");
                        int fid =
sc.nextInt();
                        sc.nextLine();
facultyList.get(i).setId(fid);

System.out.print("Enter Faculty's new Name: ");
                        String fname =
sc.nextLine();
facultyList.get(i).setName(fname);
System.out.print("Enter Faculty's new Email: ");
                        String fmail =
sc.nextLine();
```

```
facultyList.get(i).setEmail(fmail);

System.out.print("Enter Faculty's new Position: ");

                                                    String fpos =
sc.nextLine();

facultyList.get(i).setFacultyPosition(fpos);

System.out.println("ID successfully updated for: " + fid);

                                                        break;

                                                    } else {

System.out.println("Not found");

                                                    }

                                                }

                                                break;

                                            }

                                            case 'd': {

                                                System.out.print("Enter Club
Name to update: ");

                                                String cn = sc.nextLine();


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

                                                    if
(clubList.get(i).getClubName().equals(cn)) {

System.out.print("Enter Club's new Name:");

                                                        String cname =
sc.nextLine();

clubList.get(i).setClubName(cname);
```

```
System.out.println("Club name successfully updated for: " + cname);
                                                        break;
                                                    } else {
System.out.println("Not found");
                                                    }
                                                }
                                                break;
                                            }
                                            case 'e': {
                                                System.out.print("Enter
Department Name to update: ");
                                                String dn = sc.nextLine();

                                                for (int i = 0; i <
departmentList.size(); i++) {
                                                    if
(departmentList.get(i).getDepartmentName().equals(dn)) {
System.out.print("Enter Department's new Name:");
                                                        String dname =
sc.nextLine();
departmentList.get(i).setDepartmentName(dname);
System.out.println("Department name successfully updated for: " + dname);
                                                        break;
                                                    } else {
System.out.println("Not found");
                                                    }
```

```
                    }
                    break;
                }
                case 'f': {
                    sp = false;
                    break;
                }

                default: {
                    sp = true;
                    break;
                }

            }
        }

        break;
    }
    //case c end update end
    case 'd': { //search
        boolean sp = true;
        while (sp) {
            System.out.println("a. Search Course:
");
            System.out.println("b. Search
Faculties: ");
            System.out.println("c. Search Club:
");
            System.out.println("d. Search
Department: ");
```

```
                                        System.out.println("e. Search Course
Result: ");
                                        System.out.println("f. Search Whether
a faculty teaches a Course: ");
                                        System.out.println("h. Search Courses
taken by student: ");
                                        System.out.println("i. Back: ");
                                        char ps = sc.next().charAt(0);
                                        switch (ps) {
                                            case 'a': {
                                                System.out.print("Enter
Course ID: ");
                                                sc.nextLine();
                                                String cid = sc.nextLine();

                                                for (Course i : CourseList) {
                                                    if
(Objects.equals(i.getCourseId(), cid)) {
                                                        i.display();
                                                    } else {

System.out.println("Not Found!");
                                                    }
                                                }

                                                break;
                                            }

                                            case 'b': {
                                                System.out.println("Enter
Faculty ID: ");
```

```
                int fid = sc.nextInt();
                for (Faculty i : facultyList)
{
                    if (fid == i.getId()) {
                        i.display();
                    } else {

System.out.println("Not Found!");
                    }
                }

                break;
            }
            case 'c': {
                System.out.print("Enter Club
Name: ");
                sc.nextLine();
                String cname = sc.nextLine();

                for (Club i : clubList) {
                    if
(Objects.equals(i.getClubName(), cname)) {
                        i.display();
                    } else {

System.out.println("Not Found!");
                    }
                }

                break;
```

```
                                        }
                                        case 'd': {
                                            System.out.print("Enter
Department Name: ");
                                            sc.nextLine();
                                            String dname = sc.nextLine();

                                            for (Department i :
departmentList) {
                                                if
(Objects.equals(i.getDepartmentName(), dname)) {
                                                    i.display();
                                                } else {

System.out.println("Not Found!");
                                                }
                                            }

                                            break;
                                        }
                                        case 'e': { //exam result search
                                            System.out.println("Enter
Student ID: ");

                                            int sid = sc.nextInt();

                                            for (Student i : studentList)
{
                                                if
(Objects.equals(i.getId(), sid)) {

System.out.print("Enter Course ID: ");
```

```
                                                    String cid =
sc.nextLine();
                                                    for (Course j :
CourseList) {
                                                        if
(Objects.equals(j.getCourseId(), cid)) {
j.printMarks();
                                                            break;
                                                        }
                                                    }
                                                }
                                                break;
                                            }

                                            case 'f': {
                                                System.out.print("Enter
Faculty ID: ");
                                                int fid = sc.nextInt();
                                                sc.nextLine();
                                                for (Faculty i : facultyList)
{
                                                    if (fid == i.getId()) {
System.out.print("Enter Course ID: ");
                                                    String cid =
sc.nextLine();
                                                    for (Course c :
CourseList) {
```

```
                                                    if
(Objects.equals(c.getCourseId(), cid)) {
                                                        i.display();
                                                    }
                                                }

                                            } else {

System.out.println("Not Found!");
                                            }
                                        }
                                        break;
                                    }
                                    case 'h': {
                                        System.out.print("Enter
Course ID: ");
                                        sc.nextLine();
                                        String cid = sc.nextLine();
                                        for (Course c : CourseList) {
                                            if
(Objects.equals(c.getCourseId(), cid)) {

System.out.println("Enter Student ID: ");
                                                int sid =
sc.nextInt();
                                                for (Student s :
studentList) {
                                                    if (sid ==
s.getId()) {
                                                        s.display();

s.displayAssignCourse();
```

```
                                    }
                                }
                            }
                        }
                        break;
                    }

                    case 'i': {
                        sp = false;
                        break;
                    }
                    default: {
                        System.out.println("Invalid
Input Please Try Again!");
                        break;
                    }
                }

            }

            break;
        }
        case 'e': {
            //print

            boolean sp = true;
            while (sp) {
```

```
                                        System.out.println("a. Print Student
Details: ");
                                        System.out.println("b. Print Faculty
Details: ");
                                        System.out.println("c. Print Course
Details: ");
                                        System.out.println("d. Back: ");
                                        char ps = sc.next().charAt(0);
                                        switch (ps) {
                                            case 'a': {
                                                System.out.println("All
Students: ");
                                                for (int bb = 0; bb <
studentList.size(); bb++) {
                                                    System.out.println("(" +
bb + ")");
studentList.get(bb).display();
studentList.get(bb).displayAssignClub();
studentList.get(bb).displayAssignDepartment();
studentList.get(bb).displayAssignCourse();
                                                }
                                                break;
                                            }
                                            case 'b': {
                                                System.out.println("All
Faculty Members: ");
                                                for (int bb = 0; bb <
facultyList.size(); bb++) {
                                                    System.out.println("(" +
bb + ")");
```

```
facultyList.get(bb).display();

facultyList.get(bb).displayAssignCourse();

facultyList.get(bb).displayAssignClub();

facultyList.get(bb).displayAssignDepartment();


                                                    }

                                                    break;

                                                }

                                                case 'c': {

                                                    System.out.println("All
Courses :");

                                                    for (int bb = 0; bb <
CourseList.size(); bb++) {

                                                        System.out.println("(" +
bb + ")");

CourseList.get(bb).display();

CourseList.get(bb).getFaculty();

System.out.println(CourseList.get(bb).getNumberOfStudent());

CourseList.get(bb).getStudentList();

                                                    }

                                                    break;

                                                }


                                                case 'd': {

                                                    sp = false;
```

```
                                        break;
                                    }

                                    default: {
                                        System.out.println("Invalid
Input Please Try Again!");
                                        break;
                                    }
                                }
                            }
                            break;

                        }
                        case 'f': { //back
                            stp = false;
                            break;
                        }

                        default: {
                            System.out.println("Invalid Input Please
Try Again!");
                            stp = true;
                            break;
                        }
                    }
                } //admin panel while loop ends
                break;
            }
            //case b ended
```

```
                    //Case 0 started (Exit)
                    case '0': {
                        res = false;
                        break;
                    }
                    //Case 0 Ended

                    default: {
                        System.out.println("Invalid Input Try Again!");
                        res = true;
                        break;
                    }
                } // main switch ended

                //main switch started
            }
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

5. Project Flowchart