# Java Assignment 3

Om Varshney. AI ML B2. 21070126117

## Student.java

/\*

This is the student class that implements all getter, setter methods to deal with each student object.

\*/

public class Student {

    private String prn;

    private String name;

    private String dob;

    private int marks;

    public Student(String prn, String name, String dob, int marks) {

        this.prn = prn;

        this.name = name;

        this.dob = dob;

        this.marks = marks;

    }

    public String getPRN() {

        return prn;

    }

    public String getName() {

        return name;

    }

    public String getDOB() {

        return dob;

    }

    public int getMarks() {

        return marks;

    }

    public void setPRN(String prn) {

        this.prn = prn;

    }

    public void setName(String name) {

        this.name = name;

    }

    public void setDOB(String dob) {

        this.dob = dob;

    }

    public void setMarks(int marks) {

        this.marks = marks;

    }

    @Override

    public String toString() {

        return "PRN: " + prn + "\n" +

                "Name: " + name + "\n" +

                "Date of Birth: " + dob + "\n" +

                "Marks: " + marks;

    }

}

## Main.java

import java.util.\*;

/\*

For this assignment we had to implement the following

Create a student class with the capacity to store information like prn, name, DoB, marks etc.

Create an array of objects of Student class and perform operations like:

a. Add students,

b. Display,

c. Search (by prn, by name, by position),

d. Update/Edit

e. Delete.

\*/

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int choice, count = 0;

        Student[] students = new Student[10];

        do {

            System.out.println("\nMenu:");

            System.out.println("1. Add Student");

            System.out.println("2. Display Students");

            System.out.println("3. Search by PRN");

            System.out.println("4. Search by Name");

            System.out.println("5. Search by Position");

            System.out.println("6. Update/Edit");

            System.out.println("7. Delete");

            System.out.println("8. Exit");

            System.out.print("Enter your choice: ");

            choice = sc.nextInt();

            switch (choice) {

                case 1:

                    System.out.print("Enter PRN: ");

                    String prn = sc.next();

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

                    String name = sc.next();

                    System.out.print("Enter Date of Birth (dd/mm/yyyy): ");

                    String dob = sc.next();

                    System.out.print("Enter Marks: ");

                    int marks = sc.nextInt();

                    students[count] = new Student(prn, name, dob, marks);

                    count++;

                    break;

                case 2:

                    for (int i = 0; i < count; i++) {

                        System.out.println("Student " + (i + 1) + ":");

                        System.out.println(students[i]);

                    }

                    break;

                case 3:

                    System.out.print("Enter PRN to search: ");

                    String prnSearch = sc.next();

                    for (int i = 0; i < count; i++) {

                        if (students[i].getPRN().equals(prnSearch)) {

                            System.out.println("Student found at position " + (i + 1) + ":");

                            System.out.println(students[i]);

                            break;

                        }

                    }

                    break;

                case 4:

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

                    String nameSearch = sc.next();

                    for (int i = 0; i < count; i++) {

                        if (students[i].getName().equals(nameSearch)) {

                            System.out.println("Student found at position " + (i + 1) + ":");

                            System.out.println(students[i]);

                            break;

                        }

                    }

                    break;

                case 5:

                    System.out.print("Enter position to search: ");

                    int position = sc.nextInt();

                    if (position >= 1 && position <= count) {

                        System.out.println("Student at position " + position + ":");

                        System.out.println(students[position - 1]);

                    } else {

                        System.out.println("Invalid position.");

                    }

                    break;

                case 6:

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

                    int updatePosition = sc.nextInt();

                    if (updatePosition >= 1 && updatePosition <= count) {

                        System.out.println("Current details:");

                        System.out.println(students[updatePosition - 1]);

                        System.out.print("Enter new PRN: ");

                        String newPRN = sc.next();

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

                        String newName = sc.next();

                        System.out.print("Enter new Date of Birth (dd/mm/yyyy): ");

                        String newDOB = sc.next();

                        System.out.print("Enter new Marks: ");

                        int newMarks = sc.nextInt();

                        students[updatePosition - 1] = new Student(newPRN, newName, newDOB, newMarks);

                        System.out.println("Student details updated.");

                    } else {

                        System.out.println("Invalid position.");

                    }

                    break;

                case 7:

                    System.out.print("Enter position to delete: ");

                    int deletePosition = sc.nextInt();

                    if (deletePosition >= 1 && deletePosition <= count) {

                        for (int i = deletePosition - 1; i < count - 1; i++) {

                            students[i] = students[i + 1];

                        }

                        count--;

                        System.out.println("Student deleted.");

                    } else {

                        System.out.println("Invalid position.");

                    }

                    break;

                case 8:

                    System.out.println("Exiting...");

                    break;

                default:

                    System.out.println("Invalid choice.");

            }

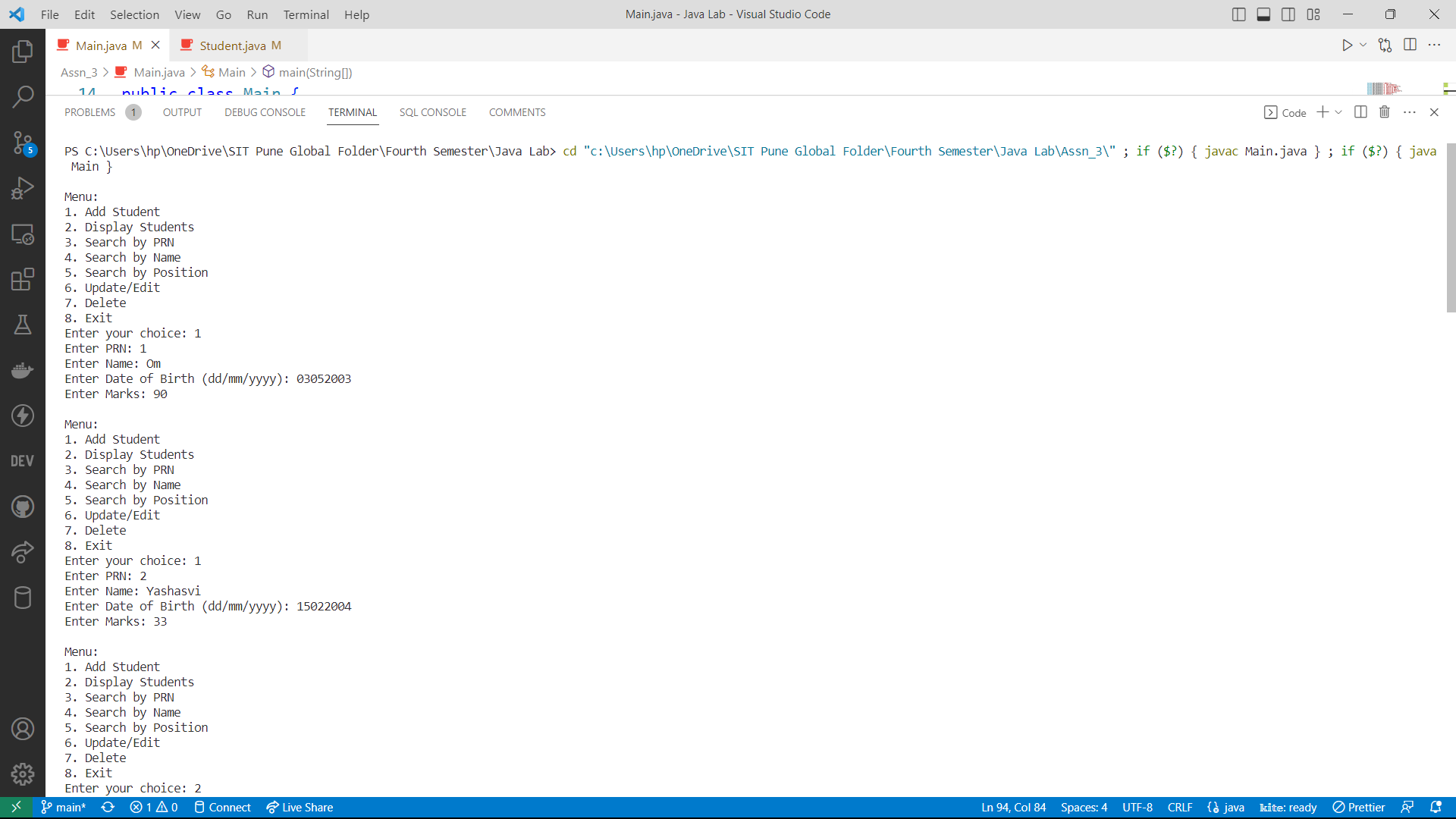
        } while (choice != 8);

        sc.close();

    }

}

## Outputs



Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

## GitHub Repository Link

https://github.com/om-varshney/Java-Lab/tree/main/Assn\_3