Objective-Create a CLI-based CRUD system for managing students records

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

class Student {
    int id;
    String name;
    double marks;

    public Student(int id, String name, double marks) {
        this.id = id;
        this.name = name;
        this.marks = marks;
    }

    public void display() {
        System.out.println("ID: " + id + ", Name: " + name + ", Marks: " +
marks);
    }
}

class StudentManager {
    static ArrayList<Student> studentList = new ArrayList<>();
    static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        int choice;

        do {
            System.out.println("\n--- Student Management ---");
            System.out.println("1. Add Student");
            System.out.println("2. View Students");
            System.out.println("3. Update Student");
            System.out.println("4. Delete Student");
            System.out.println("5. Exit");
            System.out.print("Choose option: ");
            choice = scanner.nextInt();
            scanner.nextLine();  // consume newline

            switch (choice) {
                case 1 -> addStudent();
                case 2 -> viewStudents();
                case 3 -> updateStudent();
                case 4 -> deleteStudent();
```

```
            case 5 -> System.out.println("Exiting program...");
            default -> System.out.println("Invalid option!");
        }
    } while (choice != 5);
}

static void addStudent() {
    System.out.print("Enter ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    System.out.print("Enter Name: ");
    String name = scanner.nextLine();
    System.out.print("Enter Marks: ");
    double marks = scanner.nextDouble();
    studentList.add(new Student(id, name, marks));
    System.out.println("Student added successfully.");
}

static void viewStudents() {
    if (studentList.isEmpty()) {
        System.out.println("No students found.");
        return;
    }
    System.out.println("\nStudent List:");
    for (Student s : studentList) {
        s.display();
    }
}

static void updateStudent() {
    System.out.print("Enter ID to update: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    for (Student s : studentList) {
        if (s.id == id) {
            System.out.print("Enter new name: ");
            s.name = scanner.nextLine();
            System.out.print("Enter new marks: ");
            s.marks = scanner.nextDouble();
            System.out.println("Student updated.");
            return;
        }
    }
    System.out.println("Student ID not found.");
}

static void deleteStudent() {
    System.out.print("Enter ID to delete
```

```
");
        int id = scanner.nextInt();
        for (Student s : studentList) {
            if (s.id == id) {
                studentList.remove(s);
                System.out.println("Student deleted.");
                return;
            }
        }
        System.out.println("Student ID not found.");
    }
}
```

OUTPUT

--- Student Management ---

1. Add Student

2. View Students

3. Update Student

4. Delete Student

5. Exit

Choose option: 1

Enter ID: 245

Enter Name: SAURABH

Enter Marks: 70

Student added successfully.

--- Student Management ---

1. Add Student

2. View Students

3. Update Student

4. Delete Student

5. Exit

Choose option: 2

Student List:

ID: 245, Name: SAURABH, Marks: 70.0

--- Student Management ---

1. Add Student

2. View Students

3. Update Student

4. Delete Student

5. Exit

Choose option: 3

Enter ID to update: 245

Enter new name: KULDEEP

Enter new marks: 60

Student updated.

--- Student Management ---

1. Add Student

2. View Students

3. Update Student

4. Delete Student

5. Exit

Choose option: 4

Enter ID to delete: 245

Student deleted.

--- Student Management ---

1. Add Student

2. View Students

3. Update Student

4. Delete Student

5. Exit

Choose option: 5

Exiting program...

Process finished with exit code 0