

Title

Create a Java console program to manage students.

Background Context

Write a program to manage information of student. The program implements terminology of Object Oriented Programming (OOP) paradigm. OOP is one of the best choosing ways to design software program.

In this assignment, we will use **ArrayList** to store list of student. In fact, ArrayList is popular used to manipulate with data. ArrayList provides some useful methods, such as: add(), remove(), sort() ., etc.

Program Specifications

A student information consists of ID, Student Name, Semester, Course Name (There are **only three courses**: Java, .Net, C/C++). The program allows use to create list of student, update/delete student information. On the other hand, use can search student(s) and sort result by student name

1. Main Screen as below:

```
WELCOME TO STUDENT MANAGEMENT
```

1. Create
2. Find and Sort
3. Update/Delete
4. Report
5. Exit

```
Please choose:
```

2. Function details:

There are 4 functions in Student Management Screen, as bellow:

1. **Create**: user creates student by inputting information from keyboard. Use has to create at least 10 students, if number of students greater than 10, the program shows a message: **Do you want to order now (Y/N)?** Choose Y to continue, N to return main screen.
2. **Find/Sort**: Find and list student (by name) and sort by name, it should display: Student name, semester and Course Name. User inputs student name or a part of student name.
3. **Update/Delete**: The program allows user find a student by ID. After finding a student by Id, a question is displayed (Do you want to update (U) or delete (D) student. If user chooses U, the program allows user updating. Choose D for deleting student.
4. **Report**: write a function to display student(s) with total Courses of this student, as: Student name, Course and Total of Course, for example:

Student name	Course
Nguyen Van A	Java
Nguyen Van A	Java
Nguyen Van B	.Net
Nguyen Van B	Java

The report as below:

```
+-----+-----+-----+-----+
| No. | Student name | Course | Total of course |
+-----+-----+-----+-----+
| 1 | Nguyen Van A | Java | 2 |
```

	2		Nguyen Van B		.Net		1	
	3		Nguyen Van B		Java		1	
+-----+-----+-----+-----+								

Technical Requirements

1. Using Object-Oriented programming style
2. Use only core Java functions and classes.

Guidelines

Slot	Task	Description
1	- Code Design - Create student	Student should create a Student class with attributes: id, studentName, semester and courseName
2	- Find and sort student	Should Collections.sort() and overwrite compare() method in Comparator interface
3	- Update/Delete student	
4	- Report	
5	- Review program	Mentor reviews student’s program.