[**PTS** HYPERLINK "http://10.22.127.127/"  **HYPERLINK "http://10.22.127.127/"Practice Tracking System** HYPERLINK "http://10.22.127.127/"](http://10.22.127.127/)   [Toggle navigation](http://10.22.127.127/Practice/AssignmentDetails/158#)

* [Grade Book](http://10.22.127.127/Student/GradeTable)
* [Class Tracking](http://10.22.127.127/ClassTracking)
* [Assignment](http://10.22.127.127/Assignment)
* [Assignments](http://10.22.127.127/Practice)
* [Ranking](http://10.22.127.127/Practice/Ranking)
* [Reports](http://10.22.127.127/Practice/AssignmentDetails/158#)
* [Summary](http://10.22.127.127/Report/Summary)
* [Export Report](http://10.22.127.127/Report/ProcessExportReport)
* [Assignment](http://10.22.127.127/Report/Assignment)
* [Student Rank](http://10.22.127.127/Report/Rank)
* [Student Progress](http://10.22.127.127/Report/StudentProgress)
* [Export Data For SMS](http://10.22.127.127/Practice/AssignmentDetails/158#)
* [Student In System](http://10.22.127.127/Report/StudentInSystem)
* [Detail Student Mark](http://10.22.127.127/Report/DetailStudentMark)
* [Administration](http://10.22.127.127/Practice/AssignmentDetails/158#)
* [Batch](http://10.22.127.127/Batch)
* [Term](http://10.22.127.127/Term)
* [Class](http://10.22.127.127/Class)
* [User](http://10.22.127.127/User)
* [Group](http://10.22.127.127/Group)
* [VDI](http://10.22.127.127/VDI)
* [Setting](http://10.22.127.127/Practice/AssignmentDetails/158#)
* [General Settings](http://10.22.127.127/GeneralSetting)
* [Rank Settings](http://10.22.127.127/RankSetting)
* [Audit Logs](http://10.22.127.127/Audit)

View Assignment Create a Java console program to manage students.

* [Home](http://10.22.127.127/)
* [Practice](http://10.22.127.127/Practice)
* View Assignment

[Back to List](http://10.22.127.127/Practice)   [Template](http://10.22.127.127/Assignment/DownloadSubmissionTemplate)

Assignment Detail

|  |  |  |  |
| --- | --- | --- | --- |
| Assignment Name | Create a Java console program to manage students. | Assignment Code | J1.L.P0021 |
| Status | Approved | Category | Long Assignment |
| Topic | Java OOP manual grading | Short Description | Create a Java console program to manage students. |
| Background | 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. | | |
| Description |  | | |
| 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     * **Main Screen as below:**     **WELCOME TO STUDENT MANAGEMENT**   * **Create** * **Find and Sort** * **Update/Delete** * **Report** * **Exit**   **(Please choose 1 to Create, 2 to Find and Sort, 3 to Update/Delete, 4 to Report and 5 to Exit program).**   * **Function details:**   There are 4 functions in Student Management Screen, as bellow:   * **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 continue (Y/N)? Choose Y to continue, N to return main screen. * **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. * **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. * **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:  Nguyen Van A | Java | 2  Nguyen Van B | Java | 1  Nguyen Van C | Java | 1 | | |
| Student Instruction | * Using Object-Oriented programming style * 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 |  | | | |
| Manual Grading? | Yes | Slots | 3 |
| Line of Code | 175 | Difficulty | 3 |
| Comment |  | | |
| Attached File(s) |  | | |

**Version** 1.1.0

**Copyright © 2015** [**FPT University**](http://www.fpt.edu.vn/)**.** All rights reserved.