

PRO192- LAB 6

Develop School management OOP application with following components and requirements

1. Create MVC console application in NetBean:

- a. Define class Student with following attributes and methods:
 - + ID, Name, Average
 - + Constructors, get/set(), toString()Define class School with following attributes and methods:
 - + List of Student to store data for School
 - + **Read text file to load data for School– sch.txt is provided**
 - + Search Student by different criteria
 - + Print any List of Student
 - + Add new Student from console
 - + **Write data from List of Student to text file has same directory with sch.txt file**
- b. Develop SchoolManagement class for application
 - + Application menu for application include: Display all Students, Student Search, Add new Student, Student statistical
 - Hint: An abstract Menu class is provided for use in the application. Override method execute()
 - + Method for searching Student with searching menu
 - + Method for sort Student with sort menu

2. Technical requirements:

- + Create package **model** for Student and School class
- + Create package **view** for Menu and SchoolManagement class.
- + Perform data validation for attributes when the user enters incorrect data
- + User interface as sample images below

School Management System

- ```

1.List all students
2.Search student
3.Add new student
4.Sorting display
5.Exit

```

Enter selection..1

List of Students

```

Student{ID=S01, Name=Jonh Horstman, aver=7.8}
Student{ID=S02, Name=John Smith, aver=8.5}
Student{ID=S03, Name=Cays Horstman, aver=9.6}
Student{ID=S04, Name=David Beckham, aver=8.0}

```

Total : 4 students.

**When choose search Student:**

Enter selection..2

Student Searching

- ```
-----  
1.Find by StudentID  
2.Find by Name  
-----
```

Enter selection..2

Enter Name :Smith

List of Students

```
-----  
Student{ID=S02, Name=John Smith, aver=8.5}  
-----
```

Total : 1 students.

Student Searching

- ```

1.Find by StudentID
2.Find by Name

```

Enter selection..

**When choose sort Student:**

```

Enter selection..4
Student Sorting

1.Sort by StudentID
2.Sort by Name
3.Sort average point in descending

Enter selection..3
List of Students

Student{ID=S03, Name=Cays Horstman, aver=9.6}
Student{ID=S02, Name=John Smith, aver=8.5}
Student{ID=S04, Name=David Beckham, aver=8.0}
Student{ID=S01, Name=Jonh Horstman, aver=7.8}

Total : 4 students.
```

**Notice: Student must strictly follow all requirements as describe above for marking,  
Any modification the requirement detail is not allowed and will get no mark!**