

KING SAUD UNIVERSITY COLLEGE OF COMPUTER AND INFORMATION SCIENCES Computer Science Department		
CSC 111:Java Programming 1	Mini-project Course	1st Semester 1436-1437

Program Tasks Overview

You need to write a program for a system to manage the students in the CSC111 course. The program should enable a course clerk to complete the following tasks:

- Add a student to the set of students in the course.
- Remove a student from the set of students in the course.
- Given an academic advisor name, return the total number of students for that advisor.
- Find and print the information about a student given the student ID.
- Find information about the lowest, average, highest grade in the course.
- Print all the students.
- Print all the students with the same grade (such as A,B,C,D).
- Count the total number of students in the course.

Assumptions:

- Student ID is unique; no two students can have the same ID.
- Student Email should be built from the user ID.
- Student Grade should be calculated in the program.
- The number of students should not exceed 25 students in a course.
- The menu should repeat as long as the user needs.
- There is a counter for the number of students, totalStudents that will be incremented whenever a student is added successfully.

Required Classes

For your project, you must implement and use the following classes:

- I. Student.
- II. Course.

Student Class

A student class is identified by a 9 digit number that represents the student ID, ID, the name of the student, name, student email, email, student score, score, student grade, grade, and an academic advisor name, academicAdvisor.

For each student you should have the following services:

- **Student:** A constructor that initializes new student with these initial values:
 - ID: 435000000
 - name: -
 - email: -
 - score: 0.0
 - grade: -
 - academicAdvisor: -
- **SetStudent(ID, name, score, grade, email, academicAdvisor):** That sets the values for: ID, name, email, score, grade, academicAdvisor.
- **Getter methods (one for each):** That returns the values of: ID, name, email, score, grade, academicAdvisor
- **printStudentInfo():** That prints all the information of an existing student.

Course Class

A course class represents the course management system application, which is used to test your program. It should be able to keep track of up to 25 students. The course management system should maintain the following information:

- totalStudents: The total number of the registered students (less than or equal 25).
 - courseStudents: A collection of 25 student.
-

The methods' prototypes and behaviors are defined as follows (all public):

- **main:** the main method should contains a menu to ask the course clerk what he/she would like to do, as follows:
 1. Add a student
 2. Remove a student
 3. Find and print the information of a student

4. List all students
5. List students for a given grade
6. Calculate the lowest, average, highest grade in the course
7. Total number of students for a given academic advisor
8. Total number of students

- **addStudent**: Adds a student to the list of students in the course. You should create a new student and add it to the end of the `courseStudents[]`. This method should return true if the add operation was completed successfully, and false otherwise. The student is successfully added if his/her ID is correct, and the student is not already in `courseStudents[]`. Also, the method should not add the student information if the lists are full. The method header is described as below:

`Boolean addStudent(int ID, String name, double score, String email, char grade, String academicAdvisor)`

Hint: you can use `findStudent` inside this method (as described below).

- **findStudent**: that searches for a student. The search key is the student ID. The method returns the index of id in the list, or -1 if it is not found. The method header is described as below:

`int findStudent(String ID)`

- **removeStudent**: That replaces information of a student with asterisk (*) and numerical information with -1 given his id. The method should check if the student id already exists before removing his information. The method returns true if the operation is completed successfully and false otherwise. The method header is described as below:

`boolean removeStudent(String ID)`

- **isValidID()**: That returns true if the entered ID is correct and false otherwise. The length of id should be 9 and should contain only digits.

`boolean isValidID(String id)`

- **findStudentInfo**: That prints information (name, Id, score, grade, email and advisor) of an existing student given her id. The method header is described as below:

void findStudentInfo(String id)

- **printAll:** That prints all the students in the course. Nothing will be printed if there are no students.

void printAll()

- **printAll:** That prints names of all students with same given grade. Nothing will be printed if there is no students got that specific grade.

void printAll(char grade)

- **calculateGrade:** That finds and prints the lowest, highest and the average scores in the course.

void calculateGrade()

- **numberOfStudentsAdvisor:** That Returns the total number of students who are supervised by a specific academic advisor.

int NumStudentAdvisor(String advisor)

Requirements

- **You must define the classes Student and Course.**
- **Same as all programming assignments:**
 - Use meaningful variable name and good indentation.
 - You must avoid code duplication, by calling appropriate methods (rather than cutting and pasting code).
 - Don't change the names of any class, attribute or method.
 - Choose suitable types for your data.
 - Use clear documentation.
 - Use standards for naming constants and variables.
 - Validate user input.
 - Alert user when the list is full or empty.