Assignment 1 Object oriented programming

IMPORTANT: This assignment must be done individually

Read Section 1 to understand the programming requirements, Section 2 to understand the programming tasks that you need to carry out and Section 2 to know what you need to submit as the result.

1. Description

In this assignment, you will create a Java program named **CourseMan** for a simple course management system. The purpose of this program is to allow the students of a faculty of a university to enrol into pre-defined course modules each semester, and to allow the administrative staff of the faculty to enter the module marks for each enrolment and to compute the final grade from these marks.

Students have the following attributes: id, name, dob (the abbreviation for date of birth), address, email. id is a unique student identifier which is generated automatically by the system using the formula: the letter "S" followed by a number, which is auto-incremented from the current year. For example, the first student to be registered into the program in the year 2015 will have the id of S2015. The second student of that year has the id of 2016, and so on.

The attributes of modules are: code, name, semester (the integral number of a course semester), and credits (the number of credits). The module code is a unique module identifier which is generated by the system using the formula: the letter "M" followed by the next auto-incremented integer starting from a base, which is calculated by multiplying the semester attribute value by one hundred. For example, the first module with the semester attribute value of 1 will have the code of M101, the second module will have the code of M102 and so on. Similarly, the first module with the semester attribute of 2 will have the code of M201, etc.

There are two types of modules that the program needs to support. The first type is compulsory module, which inherits all the attributes of module. The second type is elective module, which has one extra attribute called department name. This attribute is used to record the name of the department that runs the module (students are allowed to take elective subjects in another department). Note that, every module instance must be either compulsory or elective. There are no other types of module.

An enrolment records a fact that a student has registered interest to study a specific module in a given semester. It also holds data about the internal mark, examination mark, and the final grade that the student has obtained in the module Thus, enrolments will have the following attributes: student (the student instance who initiated the enrolment), module (the module instance), internal mark (floating), examination mark (floating), and final grade. Final grade is a single character which must be one of the followings: "E" (excellent), "G" (good), "P" (pass), and "F" (failed). Further, the final grade is generated by the program from an aggregated mark, which is computed from the internal and examination marks

using the formula: aggregated mark = 0.4 * internal mark + 0.6 * examination mark.

2. Task requirements

1. You must create a top-level package named *courseman1* for the program.
2. Specify and implement suitable Java classes for the stated program requirements.
3. Specify and implement an *EnrolmentManager* class that is responsible for managing the enrolment objects. This class must have the following operations:
   (a) *addEnrolment(Enrolment)* – adds a new enrolment
   (b) *getEnrolment(Student,Module): Enrolment* – returns the enrolment of the specified student and module
   (c) *setMarks(Student, Module, double, double)* – records the internal and examination marks (in that order) into the enrolment of the specified module of the given student
   (d) *report(): String* – generates an initial report of all the current enrolments (without marks) in the program
   (e) *reportAssessment(): String* – generates an assessment report of all the enrolments together with their internal, exam marks and final grade
   (f) *sort()* – sorts the current enrolments in the descending order of the student identifier

Note: you are recommended to use a variant of the the quick sort algorithm that operates on a *Comparable* array of objects provided by the *Arrays·java* library file attached to this assignment

4. Specify and implement the *CourseManDemo* class, which is the main program class. This class has a *main* method that performs the following tasks:
   (a) Initialise at least 5 modules (3 compulsory and 2 elective) and 5 students
   (b) Initialise and use *EnrolmentManager* to create 10 enrolments (containing a mixture of compulsory and elective modules)
   (c) Print a non-sorted initial report of the enrolments on the standard output
   (d) Print a sorted initial report of the enrolments on the standard output
   (e) Initialise and enter marks for 5 enrolments
   (f) Print a (sorted) assessment report of the enrolments on the standard output
   (g) End the program

3. Submission requirements

You must submit a single zip file containing the *courseman1* package to the portal by the due date. The zip file name must be of the form *a1_Sid·zip*, where *Sid* is your student identifier (the remaining bits of the file name must not be changed!). For example, if your student id is 1512345678 then your zip file

must be named *a1_1512345678.zip*.

IMPORTANT: failure to name the file as shown will result in no marks being given!

NO PLAGIARISM, strictly penalty of marks deduction will be applied!