
University Management System

Design and implement a program to manage university courses, teachers, and students. All the classes must belong to the module **University**. For testing, run **main.swift** file in *Xcode*.

R1 - University

The user interface (not part of this assignment) interacts through the class **University**, whose constructor receives as argument the name of the university.

The name of the university can be retrieved through function **getName()**.

It is possible to assign the name of the Rector by means of the function **setRector()** that accepts as arguments the first and last name of the Rector.

The function **getRector()** returns the first and last name of the Rector concatenated and separated by a blank space (" ").

R2 - Students

It is possible to enter information concerning a new student through the method **enroll()** of class **University**, that receives as arguments first and last name of the student; the method returns the ID number that has been assigned to the student.

ID numbers are assigned progressively in each university starting from number 10000.

To retrieve the information about a student we can use the function **student()** that receives as an argument the ID number and returns a string made up of ID, first, and last name separated by blank spaces.

R3 - Courses

To define a new course we can use the function **activate()** that receives as arguments the title of the course and the name of the responsible teacher. The method returns an integer that corresponds to the course code. Course codes are assigned progressively starting from 10.

To retrieve the information about a course we can use the function **course()** that accepts the course's code and returns a string containing code, title, and teacher separated by blank spaces.

R4 - Course attendace

Students that wish to attend a course must be registered through the function **register()** that accepts as arguments the ID of the student and the course's code.

To get the list of students attending a course we can use the function **listAttendees()** that accepts the code of the course and returns a string containing the list of attendees.

The students appear one per row (rows are terminated by a new-line character ' \n ') and each row is formatted as describe in requirement **R2**.

Given the ID of a student, it is possible to get the list of courses she is attending through the function **studyPlan()**. The courses are described one per row and formatted as described in requirement **R3**.

Based on a work at <http://softeng.polito.it/courses/02JEY/> by Marco Torchiano licensed under a Creative Commons Attribution 4.0 International License. Files modified and adapted by [Luca Mezzatesta](#) ([HalfDeloper group](#)).

To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.