**Assignment Report**

The purpose of the assignment was to design and implement an application to register and manage courses, students and faculty members across the branches. Our team is composed of 6 students : Thomas Culino (23174), Foucauld Bonnefont (23178), Victor Cazaux (23166), Theo Underwood (23163), Nicolas Goncalves (23167), Benjamin Toubiana (23206). Our code can be found on Github with the following link : <https://github.com/BenjaminTOUD/dorset-college-oop-group-2>. We code everything ourselves and did not took anything from internet. We also made a video showing the features of our application. You can watch it by clicking on the following link : <https://youtu.be/iWUmaspXlg8>.

Our project is composed of different classes forming a console application. We assume that no student or any user can be deleted from the system as well as notes and attendance. We will now present a little description of each class.

Explanation of classes and attributes

Class User :

The User class is an abstract class. It has several attributes such as :

* Int ID
* String name
* String password
* String adress
* String inscrptionDate
* String phoneNumber

This class also regroups different methods :

* Login which checks if the password entered is correct
* ChangePassword which enables to change the password
* ChangeAdress which enables to change the adress
* ToString which describe briefly the user

Class Student

It’s a public class inheriting from the class User. It creates an object Student with all his information such as his ID, his address, etc. The attributes are:

* List<Course> courses
* Int year
* List<int> payment
* List<List<int>> notes
* List<bool> attendance

This class includes several methods such as:

* RegisterCourse which allows the student to register for courses
* ToStringAttendance which displays his attendance of courses
* ToStringNotes which displays his notes
* ToStringPayment which displays his payments
* ToStringCourses which displays the courses the student registered for

Class Admin

The Admin class inherits from the User class. It represents the user who have all the rights. It does not have any attribute but we can find several methods such as :

* CreateCourse which create a new Course and write it on the database (a .txt file)
* CreateTT which creates and displays a timetable from the courses followed by a student. Each course is randomly placed in the timetable.
* TrackPayment which displays a list of all payments made and the total amount spent
* AddStudent which creates a new Student and write it on the database (.txt file)
* ToString which returns a description of the admin.

Class FacultyMember

The FacultyMember Class inherits from the User Class. There is only one attribute which is a List<Course> courses. This class includes several methods such as :

* GradeAStudent(Student student) : void method taking a student as parameter. Method used to add a note (double) in the student’s grades list.
* AttendThere(Student student) : void method taking a student as parameter. Method used to write « present » in the student’s attendance list.
* AttendAbsent(Student student) : void method taking a student as parameter. Method used to write « absent » in the student’s attendance list.
* CreateExam(string name, double duration, Faculty member prof, string room, List<student> class) : void used to create an object Exam with the information given in parameters.
* EditCoursePlan(Course course) : Void method taking a course as parameter. Method used to edit an existing courseplan from a specific course.

Class Tutor

The class inherits from the FacultyMember class (which already inherits from the user abstract class).

It allows to create a tutor, a privileged faculty member who can access personal student details. It accesses students by the list of students named “clas” in each courses that the tutor can teach. It has only one attribute :

* List<Course> courseList

The two methods of this class are :

* IsStudent, which allows us to check if a student is in a course of the tutor or not
* StudentDetail, which prints the personal details of a student if he is in a class of the tutor

Class Course

The Course class is used to create the following object: a course, an exam or a project. The exam and project class inherit from the course class. The attributes are:

* String name
* Double duration
* String coursePlan
* FacultyMember prof
* String room
* List<Student> clas

The only method is the toStringCoursePlan which returns a description of the course.

Class Exam :

The Exam class inherits from the Course class. It’s used to create exams and does not have any attribute.

Class Program :

The program class has numerous methods such as :

* IsRegistered which returns an array of length 2. The first index shows if the ID is registered and the second one indicate the type of user (Student, Admin, …)
* CheckPassword which checks if the password is good
* CheckCourse which checks if the course exists in the database
* CheckStudent which checks if the student appear in the database
* CheckSTeacher which checks if the teacher is in the database
* FileToArray which converts database lines into an array with all the informations
* ListToStudent which converts the array from the previous method to a Student
* ListToAdmin which converts the array into an object from the Admin class
* ListToTeach which converts the array into an object from the FacultyMember class
* ListToTut which converts the array into a Tutor
* ListToCourse which converts the array into a Course
* ListToExam which converts the array into objects of the Exam class
* SaveEverything which overwrites our database files with every object created
* The main which enables us to display and run our application

There are also several methods used to have a prettier application in the console such as VGC which displays text with the good amount of spaces.

Ultimately, our application is able to manage exams, assignments , results (Class FacultyMember and the method is GradeAStudent), attendance (Class FacultyMember and the method is AttendThere), course plans, the timetable (Class Admin and the method is CreateTT) and the fees paid (Class Admin and the method is TrackPayment).

What could have we done better ? :

We could have developped more our code by using the project class which is not used currently. By doing so, we would have gotten a more complete application. If we had more time, we would have upgraded this class by managing the creation of any project and its description. In order to have a better application, we could have tried to manage the Academic Calendar. Indeed, as it is now, our application only manage Time Table but not the whole Academic Year. Managing it would have added, for example, the date of every assignment.