Introduction

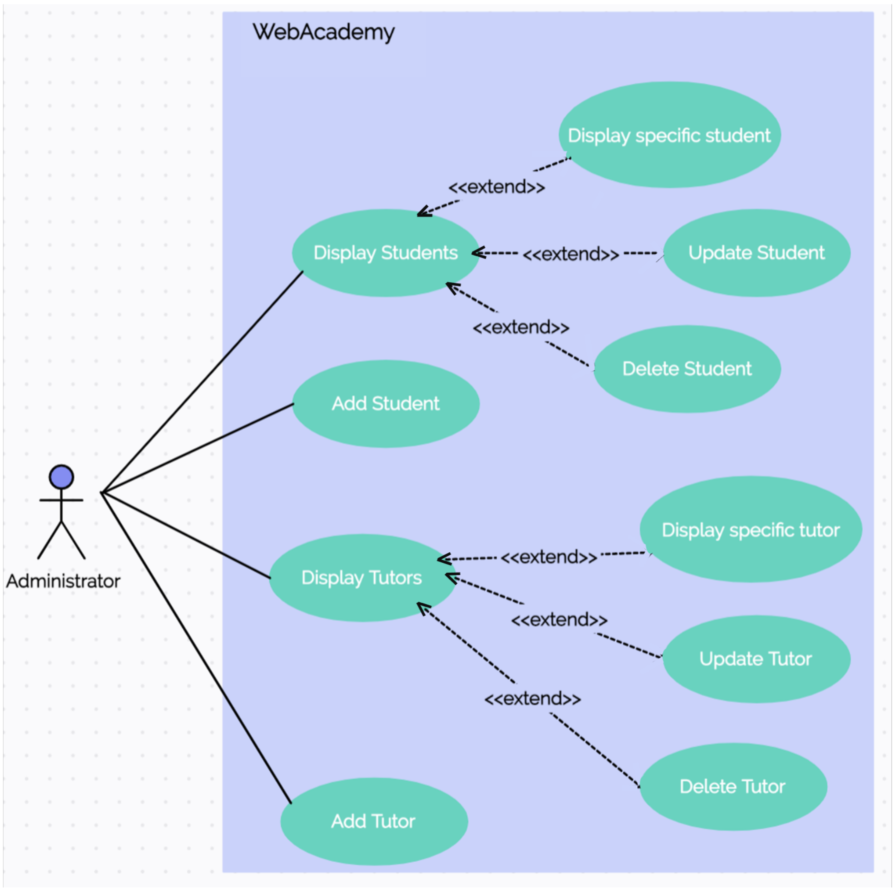
In this coursework, use your knowledge of object-oriented (OO)
programming in order to construct functional components of a web application. You will be creating a prototype to show how specifications of a given set of requirements of a business, can be implemented using OO PHP and MySQL. Furthermore, you are asked to prepare a report where you will explain and discuss how your code meets the specifications of the given scenario and demonstrate your understanding of the OO programming concepts that you have implemented.

Business description: “WebAcademy”

“WebAcademy” is a small web-based business that offers online courses to adults who wish to extend their information technology (IT) skills. More specifically, “WebAcademy” offers the following courses:

- Web Design Principles
- HTML
- Introduction to Databases
- Cyber Security

All users can browse the site and search for a specific course. Users can register as “students” and / or as “tutors”. In order to register, all users (students and tutors) must provide their names and contact details (e.g., email address) and chose a username. Furthermore, students must provide further details such as the title of the course(s) they want to attend. Tutors must also provide the following information: the title of the course they teach and their academic qualification(s). Finally, there is one more actor: Administrator. The following figure is the use case diagram, limited to the scope of this assessment, that shows what an administrator can do.



Based on the above description, you are asked to complete the following tasks, organized in part A and part B.

Part A asks you to implement a console system to be used by the administrator in order to manage tutors and students.

Part B asks you to explain and discuss further, the specific sections of your implementation for part A.

Part A: Implementation 55 marks

<u>Note:</u> You must upload all the files containing your code for this section to University's server.

Using OO PHP and MySql, implement a console system to be used by the administrator in order to manage the details of tutors and students. More specifically, in your implementation, you must address the following:

A1. Preparation and organization of the site [15 marks]

a. Create your database with all the necessary tables and populate them with the following data *[6 marks]:*

Type of user	Name	Email	User name	Course to attend
student	Henry Miller	henry@henry.com	henry	Web Design Principles
student	John Lock	john@john.com	john	Introduction to Databases
student	Jane Austin	jane@jane.com	jane	HTML, Cyber Security
student	Zadie Smith	zadie@eleanor.com	zadie	Cyber Security

Type of user	Name	Email	User name	Course to teach	Qualification
tutor	Philip Brown	philip@academy.com	phil	Web Design Principles	Diploma in Web Design
tutor	Catherine White	catherine@academy.com	catherine	HTML	BSc in Computer Science
tutor	David Blue	david@academy.com	david	Introduction to Databases	Diploma in Databases
tutor	Abigail Black	abigail@academy.com	abigail	Cyber Security, HTML	BSc in Computer Science

b. Create the basic site layout of your site (you could adopt the same template that we have used in the lectures/tutorials). You do **not** get extra marks for design elements (e.g., pictures, background etc.). When you call the site, a welcome message should appear on the screen and the following management options should appear on the navigation bar: 'Display Students"; "Display Tutors"; "Add New Student"; "Add New Tutor". *[4 marks]*

c. Organize your project using the appropriate structure and folders. Make sure you keep the project tidy. *[3 marks]*

d. Connect your site to your database. *[2 marks]*

A2. Create the necessary classes **[15 marks]**

Create all the necessary classes for implementing the console management system, according to the above description. For full marks, you are expected to optimize your code and create a "database class". If no "database class" is created, a penalty of 7 marks will apply.

A3. Display tutors and students on the screen **[4 marks]**

When the administrator clicks the option "Display Students" in the site, all student names and email addresses should be displayed on the screen. Similarly, when the administrator clicks the option "Display Tutors" in the navigation bar, all tutor names and email addresses should be displayed on the screen.

A4. Display specific details for a student and a tutor **[4 marks]**

When the administrator clicks the name of a student, a new page should be generated, where all the details about this specific student (name, email, username, course(s) to attend) are presented. Similarly, when the administrator clicks the name of a tutor, a new page should be generated, where all the details about this specific tutor are presented (name, email, username, course to teach, academic qualifications).

A5. Add, delete, update **[12 marks]**

Furthermore, the administrator must be able to perform the following actions:

- **add** a new student; add a new tutor [2 marks each]
- **delete** an existing student; delete an existing tutor [1 mark each]
- **update** the details of a student (e.g., modify the email or list of courses the student attends); update the details of an existing tutor (e.g., modify the qualifications of a tutor) [3 marks each]

A6. Comments **[5 marks]**

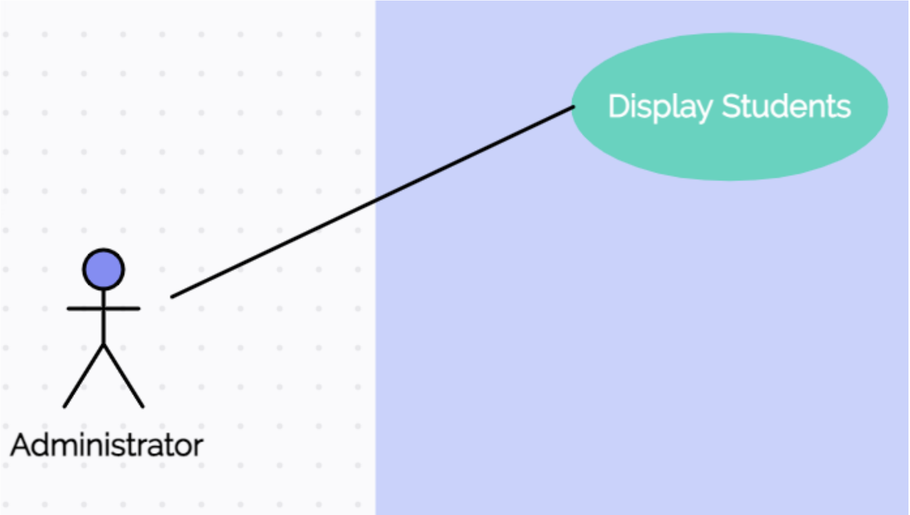
Inside your code, you must provide meaningful comments to explain it. They must be your own comments. If you just copy the comments from the tutorial files, then you will get no marks for the comments.

Part B: Explanation – Discussion 45 marks

Answer the following questions. The recommended word count for each answer is 300 to 350 words. Maximum word count is 350.

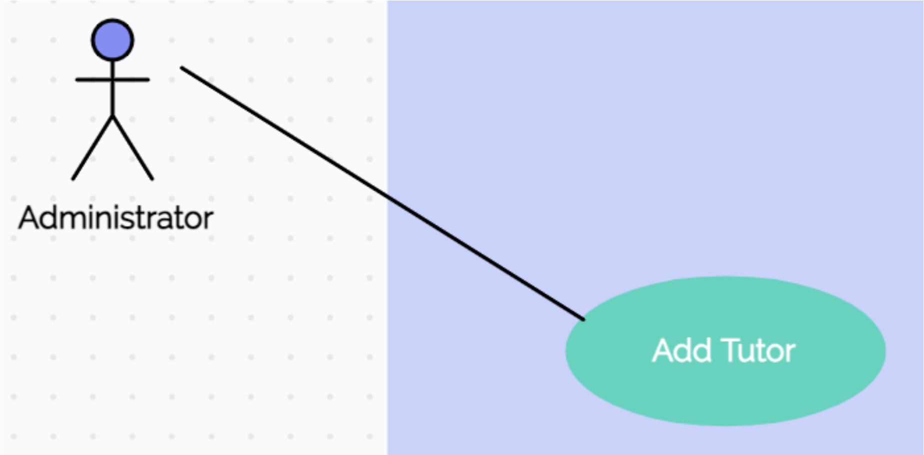
B1. According to the "inheritance principle" instead of implementing two classes as "Student" and "Tutor", you could implement one parent-class (super-class) e.g., "User" and define "Student" and "Tutor" as sub-classes. Discuss the "inheritance principle" and explain what its advantages are, using this example (5 marks); create the code for the parent class and the sub classes and present and explain it (no need to upload it to the server). (10 marks)

B2. Consider the following case of the above use case diagram.



Present and explain the function(s) you created in order to implement the above case. Discuss your implementation in detail and explain what each line of the code does. In your presentation and discussion, you must justify the name of the function(s) you created as well as the name(s) of the variable(s) you used.

B3. Consider the following case of the above use case diagram.



Present and explain the function(s) you created in order to implement the above case. Discuss your implementation in detail and explain what each line of the code does. In your presentation and discussion, you must justify the name of the function(s) you created as well as the name(s) of the variable(s) you used.