COM3003 OBJECT ORIENTED PROGRAMMING PROJECT

In this project, you are expected to create a console application for online courses. In this application, the student should be able to view the registered courses and their details. The student can add selected course(s) to their cart. Since an actual payment system is not in place, the payment section will be simulated. After entering payment information, the student will see a message indicating the payment was successful. The student should also be able to view the courses they have purchased.

To structure this, first, create a class called **OnlineCourse**. This class should include pure virtual methods such as displayInfo(), addStudent(), and checkCapacity(), along with data members such as course name, instructor, total capacity, number of enrolled students, course rates, total duration, price, level, discount rate, and prerequisites.

Afterward, use this class to create **SoftwareCourses**, **LanguageCourses**, and **ProfessionalDevelopmentCourses** classes. The SoftwareCourses class should have additional data members for programming language and development environment used in the course. The LanguageCourses class should include data members for the foreign language being taught and information on whether the instructor is a native speaker. The ProfessionalDevelopmentCourses class should have data members indicating if the course provides certification and the topics covered.

You are also required to create a fourth course type similar to these classes. Assessment will be based on the creativity and uniqueness of the class and concept.

You need to write a discount calculation function. However, this function does not belong to any specific class and should be able to operate on courses of any type. It should take the course as a parameter. It should return the newly calculated discounted price.

You are also expected to create **Student** and **Cart** classes. The Student class should store information such as student name, surname, email address, and courses they are enrolled in. However, these data members should not be directly accessible. In the Cart class, an array should hold the selected courses. The cart should allow adding, removing, and confirming courses. Once confirmed, courses cannot be removed from the cart. A student can enroll in a maximum of three courses, and cannot enroll in a course they have already registered for. After courses are added to the cart, the new price for each course will be calculated based on its own discount rate. When the student clicks the 'View Cart' option, both the prices before and after the discount will be displayed, and the total price will be calculated based on the discounted prices. Once the cart is confirmed, the courses should be added to the student's enrolled courses.

Additionally, students should be able to rate the courses they have taken between 1 and 5. The new ratings should be added to the average rating of the courses.

You are required to design an appropriate menu based on these operations. The program should not terminate until the user wants to close it.

Important Note: Do not share your solution with others! Do not collaborate with each other! You have to do your assignment/project by yourself. Otherwise, your submission will not be evaluated and both you and the person that you get the solution/a part of the solution from will get zero points from the assignment/project. Each submitted assignment/project will be checked for plagiarism using a comparison tool to determine similarity and duplicates. If ChatGPT or any AI tool is detected in the Assignment/Exercise solutions, the assignment/exercise will not be graded.

The items listed below must be included in your program. You are expected to have applied these concepts wherever necessary. Explain why you define each class, method and variable in your program and how it works, by adding comments. All of them must be used. Points will be deducted for methods used unnecessarily or incorrectly.

- Classes
- Constructors and Destructors
- Accessor and Mutator Methods
- Method overloading
- Virtual Methods
- Inheritance
- Template function
- Operator overloading

PROJECT SUBMISSION:

- You must have explained your project ***with comments in detail***.
- You should also run your program and upload screenshots. These screenshots should include examples of all types of courses and operations.
- The class diagram must be submitted along with the project files.
- You are expected to submit entire solution folder. Use the provided template for project naming.
    - YourStudentID_OOP_Project
        - Example: 1600201294_OOP_Project

PRESENTATION GUIDELINES:

- Presentation dates will be announced later.
- Class diagrams for the entire program should be drawn.
- You should demonstrate how the entire program works, explain the structure of your own program.
- Your program must be functional and executable for the presentation.
- The project will be considered incomplete unless clear and precise answers are provided to the questions during the presentation. You must be able to explain every part of the code you have written.

Important Note: Do not share your solution with others! Do not collaborate with each other! You have to do your assignment/project by yourself. Otherwise, your submission will not be evaluated and both you and the person that you get the solution/a part of the solution from will get zero points from the assignment/project. Each submitted assignment/project will be checked for plagiarism using a comparison tool to determine similarity and duplicates. If ChatGPT or any AI tool is detected in the Assignment/Exercise solutions, the assignment/exercise will not be graded.