**Project Scope**

1. **Instructor functionalities**:
   * Create courses
   * Upload course content (text/videos)
   * Create quizzes for courses
   * Issue certificates
2. **Student functionalities**:
   * Register and log in
   * Enroll in courses
   * Take quizzes and receive scores
   * Earn certificates after passing

**OOP Concepts to Focus On**

1. **Encapsulation** – Keep data private and use getter/setter methods.
2. **Inheritance** – Derive Instructor and Student from a common User class.
3. **Polymorphism** – Use function overriding for role-specific behaviors.
4. **Abstraction** – Use abstract base classes for common behaviors.
5. **File Handling** – Save user progress, quiz scores, and certificates.
6. **STL (Vectors, Maps)** – Store lists of courses, students, and instructors.

**Class Structure**

**1. User (Base Class)**

**4. Course Class**

**5. Quiz Class**

**6. Question Class**

**7. Certificate Class**

**Project Phases**

1. **Week 1**: Plan and finalize class structure, define attributes and methods.
2. **Week 2**: Implement core functionalities (user management, course creation, quizzes).
3. **Week 3**: Add file handling, test input/output, and refine UI (console-based).
4. **Week 4**: Debug, optimize, and prepare final documentation.

**Team Division**

1. **Member 1** – User authentication & profile management.
2. **Member 2** – Course and quiz implementation.
3. **Member 3** – Student activities (enrollment, quizzes, certificates).
4. **Member 4** – File handling, testing, and UI improvements.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   userfile types Student and Teachers  
   Course   
    {CourseCreation and Quiztaking}