Assignment - 1

Question 1: Online Course Management System

Design a Class Diagram for an **online course enrollment system**. Students can browse available courses, enroll in courses, and make payments. Each course may be taken by multiple students. The system should manage student records, course enrollment, and support various payment methods (such as credit card, debit card, and PayPal). Instructors are responsible for creating and managing courses, as well as tracking student progress.

Instructions:

- Identify key classes (e.g., `Student`, `Course`, `Enrollment`, `Payment`, `Instructor`, etc.)
- Define relevant attributes (e.g., `email`, `courseTitle`, `price`, `enrollmentDate`)
- Define operations/methods (e.g., `enrollCourse()`, `makePayment()`, `addModule()`)
- Clearly represent relationships:
  - **Associations** (1-to-many, many-to-many)
  - **Inheritance** (e.g., different types of `Payment`)
  - **Aggregation or Composition** where applicable

Question 2: Employee Management System

Create a Class Diagram for a company's **employee management system**. Employees can be assigned to multiple projects. Each project is led by one or more managers. Managers can oversee several projects and evaluate employee performance. The HR department handles employee records, hiring, and performance reports.

Instructions:

- Identify main classes (e.g., `Employee`, `Project`, `Manager`, `HR`, `PerformanceReport`)

- Include key attributes (e.g., `employeeId`, `projectName`, `performanceScore`)
- Add appropriate methods (e.g., `assignToProject()`, `evaluateEmployee()`, `generateReport()`)
- Use correct associations:
  - **Many-to-many** between employees and projects
  - **One-to-many** between managers and projects
  - **One-to-one or one-to-many** for performance reports
- Consider whether any classes require generalization (e.g., `Person` as a superclass for `Employee` and `Manager`)