In the name of God

# Advanced Programming – Final Project (1403-2)

Professor: Dr. Hosseinpour
Teaching Assistants: Alireza Ranjbar – Saba Famil Feizi
**Deadline: 1404/5/1 (2025/7/23)**

## Hospital Management System

### Project Overview

Build a Hospital Management System (HMS) using Python with an Object-Oriented Programming (OOP) approach. The system will manage:

- Patients (registration, medical records)
- Doctors (specialization, availability)
- Appointments (scheduling, cancellations)
- Prescriptions (medication, dosage)

The system should support CRUD operations (Create, Read, Update, Delete) and provide a menu-driven CLI interface.

### *Features & Requirements*

### 1. Core Classes (OOP Structure)

Implement the following classes with proper **encapsulation, inheritance, and polymorphism**:

**A.** Hospital **(Main Class)**

- Manages all operations (patients, doctors, appointments).
- Uses **composition** to hold lists of Doctor, Patient, and Appointment.

**B.** Person **(Abstract Base Class)**

- Inherited by Doctor and Patient.
- Contains common attributes:
  - name, age, gender, contact

**C.** Doctor **(*Inherits from* Person)**

- Additional attributes:
    - doctor_id, specialization (e.g., Cardiologist, Neurologist)
    - availability (e.g., {"Monday": "9AM-5PM"})
- Methods:
    - view_appointments()

**D.** Patient **(*Inherits from* Person)**

- Additional attributes:
    - patient_id, medical_history (list of past treatments)
- Methods:
    - view_medical_history()

**E.** Appointment

- Attributes:
    - appointment_id, doctor, patient, date, time, status (Pending/Completed/Cancelled)
- Methods:
    - schedule(), cancel(), reschedule()

**F.** Prescription

- Attributes:
    - prescription_id, patient, doctor, medication, dosage, instructions
- Methods:
    - generate_prescription()

## *2. Functionalities (Menu-Driven CLI)*

The system should provide a **text-based menu** with the following options:

**A. Patient Management**

1. Register a new patient
2. View patient details
3. Update patient information
4. Delete patient record

**B. Doctor Management**

1. Add a new doctor
2. View doctor details
3. Update doctor availability
4. Remove doctor from system

**C. Appointment Management**

1. Book an appointment
2. View all appointments
3. Cancel an appointment
4. Check doctor availability

**D. Prescription Management**

1. Issue a new prescription
2. View patient's prescriptions

**E. Data Persistence (Optional Advanced Feature)**

- Save and load data using:
  - **JSON files** (json module)
  - **CSV files**

**F. Additional Features (Bonus)**

- **Exception handling** (e.g., prevent duplicate appointments).
- **Search functionality** (find patients/doctors by name/ID).
- **Logging** (track system activities in a log file).