1. Object-Oriented Analysis (OOA)

Step 1: Identify Objects (Nouns):

- **Patient**:
- **ChronicPatient**
- **Doctor**
- **Appointment**
- **Clinic**

Step 2: Identify Attributes (Descriptive Nouns):

- **Patient**: name, id, age, medicalHistory, appointments
- **ChronicPatient**: conditionType, lastCheckupDate
- **Doctor**: name, id, specialty, appointments
- **Appointment**: date, time, reason, status
- **Clinic**: list of patients, doctors, and appointments

Step 3: Identify Methods (Verbs):

- **Patient**: scheduleAppointment(), cancelAppointment(), addMedicalHistory(), displayInfo()
- **ChronicPatient**: (override) scheduleAppointment(), displayInfo()
- **Doctor**: addAppointment(), cancelAppointment(), displayInfo()
- **Appointment**: updateStatus(), displayAppointment()
- **Clinic**: addPatient(), addDoctor(), addAppointment(), scheduleAppointment(), cancelAppointment()

Step 4: Inheritance Relationships:

- **ChronicPatient** inherits from **Patient**.
  This allows code reuse while specializing for chronic conditions
- **2. Class Design & Inheritance Explanation**

  The design has 5 main class . Include:

- **Patient** is base class, save patient infomation and medical history.

- **ChronicPatient** inherits from Patient, attributes conditionType (bệnh mãn tính) và lastCheckupDate.
- Override method **scheduleAppointment()** và **displayInfo()** to handle specific requirements for chronic patients
- **Doctor** manage doctor information and list of appoinment
- **Appointment** represents a meeting between a patient and doctor, including details such as time, reason, and current status (Scheduled, Completed, Canceled)
- **Clinic** the central manager that holds patients, doctors, and appointments, coordinating scheduling and cancellations

The use of **inheritance** avoids redundant code and enables the system to extend functionality to different types of patients without modifying the base Patient class.

3. Code Walkthrough

- **Appointment Class**:
  Provides updateStatus() to change the status (e.g., Scheduled → Canceled) and displayAppointment() to print details.
- **Patient Class**:
  Contains patient data, medical history, and methods to schedule or cancel appointments.
  Uses virtual functions so subclasses (like ChronicPatient) can override behavior.
- **ChronicPatient Class**:
  Extends Patient with chronic disease information.
  Its scheduleAppointment() method not only schedules the appointment but also reminds about periodic check-ups.
- **Doctor Class**:
  Stores doctor information and manages appointments with patients.
- **Clinic Class:**
  Coordinates between patients and doctors.
  Provides scheduleAppointment() to link a patient, a doctor, and an appointment, and cancelAppointment() to remove the appointment and update its status.
- **Main Function:**
  - **Creates patients, doctors, and appointments.**
  - **Demonstrates adding medical history.**
  - **Shows scheduling appointments for both regular and chronic patients.**
  - **Cancels one appointment and updates its status to Canceled.**

4. Test Results

The patient Duy has successfully scheduled an appointment

Schedule Appointment

Patient Information

Name: Duy

Id: 123

age: 19

medicalHistory: Visited on 2025-09-01: General Checkup - Normal

Doctor Information

Name: A

Id: 1

Specialty: General

Appointment Infomation

Date: 2025-09-10

Time: 10:00

Reason: General Checkup

Status: Scheduled

Appointment status updated to: Scheduled

The patient Vu has successfully scheduled an appointment

Chronic Patient Vu condition: Hypertension, check-up every 3 months

Schedule Appointment

Chronic Patient Information

Patient Information

Name: Vu

Id: 345

age: 30

medicalHistory: Visited on 2025-09-01: Blood sugar high - Adjusted medication

Condition: Hypertension

Last check update: 09-09-2025

Doctor Information

Name: B

Id: 2

Specialty: Cardiology

Appointment Infomation

Date: 2025-09-12

Time: 14:00

Reason: Diabetes Checkup

Status: Scheduled

Appointment status updated to: Scheduled

Canceling Appointment

Appointment status updated to: Canceled

This demonstrates that:

- Patients can successfully schedule appointments.
- Appointment details are printed correctly.
- Chronic patients display extended information.
- Cancellations update the status to **Canceled** and remove the appointment from records.

5. LLM Usage

During development, I used **ChatGPT** as an assistant for brainstorming ideas:

- For example, I asked: *"Suggest methods for an Appointment class in a clinic system"*.
- The model suggested methods like updateStatus() and displayAppointment(), which I adapted to fit my design.
- I also consulted the model about when to use **virtual** and **override** keywords in C++ inheritance, which helped refine the Patient–ChronicPatient relationship.