Small Clinic Management System

This is my project using OOP. The clinic has patients and doctors, and they make appointments. There are normal patients and chronic patients ,I use inheritance because chronic is still a patient but with extra things.

For OOA, I think about objects and basic info. Patient has name, id, age, a history list, and appointments list. Chronic patient adds condition type and last checkup date. Doctor has name, id, specialty, and appointments. Appointment has date, time, reason, status, and knows patient/doctor by pointer. Clinic has a name and arrays of everything.

The main classes:

- Patient: name, id, age, medicalHistory, appointments
- ChronicPatient (is a Patient): conditionType, lastCheckUpDate (+ Patient things)
- Doctor: name, id, specialty, appointments
- Appointment: date, time, reason, status, patient*, doctor*
- Clinic: name, patients, doctors, appointments

For methods, I keep only what I need. Patient can add history/appointment, scheduleAppointment, and display. Chronic patient overrides scheduleAppointment and can update last checkup. Doctor adds appointments and displays. Appointment can setStatus and display. Clinic can add things and display all. ChronicPatient is public Patient to keep same interface.

The main functions for each class:

- Patient: addMedicalHistory(), addAppointment(), scheduleAppointment(), displayInfo(), getters
- ChronicPatient: updateLastCheckup(), scheduleAppointment() [override], displayInfo()
- Doctor: addAppointment(), displayInfo(), getters
- Appointment: setStatus(), displayInfo()
- Clinic: addPatient(), addDoctor(), addAppointment(), displayInfo()

In main, I create normal and chronic patient, then a doctor. I make two appointments link to patient or doctor, print info, call scheduleAppointment on both, update chronic patient last checkup, then change appointment status to completed or canceled and show again.

I use fixed-size arrays, link is basic and could duplicate in bigger apps. Destructors only print, no dynamic memory to free.

Testing: I run and check console. It prints patient/doctor/appointment blocks, schedule tests, and status changes. When lists empty it says "No ... yet". When full, add is ignored. It compiles and runs in Visual Studio.

LLM use: I asked ChatGPT about pointer * and reference & (when to link Appointment to Patient/Doctor and when to pass const ref). It also helped debug prints and understand the assignment. And I also use it for coding support.

The system shows OOP basics. It can register simple data, make and update appointments, and display info.