LAB TASK # 04

1. Create a **Patient** class with properties for **PatientId**, **Name**, **Age**, **Gender**, and **MedicalCondition**.
2. Create a **Doctor** class with properties for **DoctorId**, **Name**, **Specialization**, and **Availability**.
3. Create an **Appointment** class with properties for **AppointmentId**, **Patient**, **Doctor**, **DateTime**, and **Status**.
4. Implement a system to manage patient appointments. The system should have methods to schedule an appointment, cancel an appointment, reschedule an appointment, view all appointments for a given date, and view all appointments for a given patient or doctor.
5. Ensure that the system handles cases where appointments are already scheduled, canceled, or rescheduled.
6. Include input validation to handle invalid inputs gracefully, such as entering non-existent patient or doctor IDs, or scheduling appointments on non-working days.
7. Write a **Main** method that provides a console-based menu for interacting with the appointment system. The menu should include options to schedule an appointment, cancel an appointment, reschedule an appointment, view appointments for a specific date, view appointments for a specific patient or doctor, and exit the program.
8. Ensure that the program handles input validation, such as validating user input for appointment information and handling invalid inputs gracefully.
9. Test the program by scheduling, canceling, and rescheduling appointments, and verify that the appointment management operations work as expected.