

39. QUESTION:

Design and implement a console-based Telemedicine app to schedule online consults, record diagnoses, issue e-prescriptions, and track follow-ups using OOP in Java.

Requirements:

1. Create at least 4 classes:

- o User – base with id, name, email, role (Doctor/Patient).
- o Doctor – extends User, specialization, slots, licenseNo.
- o Consultation – consultId, doctor, patient, dateTime, notes, status.
- o Prescription – rxId, consultation, medicines, dosage, validity.

2. Each class must include:

- o ≥ 4 instance/static variables.
- o A constructor to initialize values.
- o ≥ 5 methods (getters/setters, bookSlot(), startConsult(), addDiagnosis(), issuePrescription()).

3. Demonstrate OOPS Concepts:

- o Inheritance → Doctor/Patient extends User.
- o Method Overloading → bookSlot() by exact time or by preferred day+slot.
- o Method Overriding → custom displayProfile() for doctor vs patient.
- o Polymorphism → manage User list and invoke role-specific behavior.
- o Encapsulation → guard consultation state transitions and prescription edits.

4. Write a Main class (TeleMedAppMain) to test:

- o Register doctors/patients, book and conduct consults.
- o Issue e-prescriptions, schedule follow-ups.
- o Print doctor schedules and patient prescription historie

user.java

```
public class User {
    protected String id;
    protected String name;
    protected String email;
    protected String role;
    public User(String id, String name, String email, String role) {
        this.id = id;
        this.name = name;
        this.email = email;
        this.role = role;
    }

    public String getId() { return id; }
    public void setId(String id) { this.id = id; }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public String getEmail() { return email; }
    public void setEmail(String email) { this.email = email; }

    public String getRole() { return role; }
    public void setRole(String role) { this.role = role; }

    public void displayProfile() {
        System.out.println("User: " + name + " (" + role + ")");
    }

    public void sendNotification(String msg) {
        System.out.println("Notification to " + name + ": " + msg);
    }

    public void updateEmail(String newEmail) {
        this.email = newEmail;
        System.out.println("Email updated for " + name);
    }

    public void login() {
        System.out.println(name + " logged into TeleMed.");
    }

    public void logout() {
        System.out.println(name + " logged out of TeleMed.");
    }
}
```

Doctor.java

```
import java.util.ArrayList;
import java.util.List;

public class Doctor extends User {
    private String specialization;
    private String licenseNo;
    private List<String> availableSlots;
    private List<Consultation> consultations;

    public Doctor(String id, String name, String email, String specialization, String licenseNo) {
        super(id, name, email, "Doctor");
        this.specialization = specialization;
        this.licenseNo = licenseNo;
        this.availableSlots = new ArrayList<>();
        this.consultations = new ArrayList<>();
    }

    public String getSpecialization() { return specialization; }
    public void setSpecialization(String specialization) { this.specialization = specialization; }

    public String getLicenseNo() { return licenseNo; }
    public void setLicenseNo(String licenseNo) { this.licenseNo = licenseNo; }

    public List<String> getAvailableSlots() { return availableSlots; }

    public void addSlot(String slot) {
        availableSlots.add(slot);
    }

    public boolean bookSlot(String exactSlot) {
        if (availableSlots.contains(exactSlot)) {
            availableSlots.remove(exactSlot);
            return true;
        }
        return false;
    }

    public boolean bookSlot(String day, String time) {
        String slot = day + " " + time;
        return bookSlot(slot);
    }

    @Override
    public void displayProfile() {
        System.out.println("Dr. " + name + " [" + specialization + "] - License: " + licenseNo);
    }

    public void addConsultation(Consultation c) {
        consultations.add(c);
    }

    public void viewSchedule() {
        System.out.println("Consultations for Dr. " + name + ":");
        for (Consultation c : consultations) {
            System.out.println(" - " + c.getConsultId() + " with " + c.getPatient().getName() + "
at " + c.getDateTime());
        }
    }
}
```

```

    }
    public void cancelSlot(String slot) {
        if (!availableSlots.contains(slot)) {
            availableSlots.add(slot);
            System.out.println("Slot " + slot + " reopened.");
        }
    }
}

```

Patient.java

```

import java.util.ArrayList;
import java.util.List;

public class Patient extends User {
    private int age;
    private String gender;
    private List<Consultation> consultHistory;

    public Patient(String id, String name, String email, int age, String gender) {
        super(id, name, email, "Patient");
        this.age = age;
        this.gender = gender;
        this.consultHistory = new ArrayList<>();
    }

    public int getAge() { return age; }
    public void setAge(int age) { this.age = age; }

    public String getGender() { return gender; }
    public void setGender(String gender) { this.gender = gender; }

    @Override
    public void displayProfile() {
        System.out.println("Patient: " + name + " (" + age + ", " + gender + ")");
    }

    public void addConsultation(Consultation c) {
        consultHistory.add(c);
    }

    public void viewPrescriptionHistory() {
        System.out.println("Prescription history for " + name + ":");
        for (Consultation c : consultHistory) {
            if (c.getPrescription() != null) {
                c.getPrescription().displayPrescription();
            }
        }
    }

    public void requestConsultation(Doctor doctor, String slot) {
        if (doctor.bookSlot(slot)) {
            System.out.println(name + " booked slot: " + slot + " with Dr. " + doctor.getName());
        } else {
            System.out.println("Slot unavailable for Dr. " + doctor.getName());
        }
    }
}

```

Consulation.java

```
public class Consultation {
    private String consultId;
    private Doctor doctor;
    private Patient patient;
    private String dateTime;
    private String notes;
    private String status;
    private Prescription prescription;

    public Consultation(String consultId, Doctor doctor, Patient patient, String dateTime) {
        this.consultId = consultId;
        this.doctor = doctor;
        this.patient = patient;
        this.dateTime = dateTime;
        this.notes = "";
        this.status = "Scheduled";
    }

    public String getConsultId() { return consultId; }
    public Doctor getDoctor() { return doctor; }
    public Patient getPatient() { return patient; }
    public String getDateTime() { return dateTime; }
    public String getNotes() { return notes; }
    public String getStatus() { return status; }
    public Prescription getPrescription() { return prescription; }

    public void startConsult(String notes) {
        this.status = "In Progress";
        this.notes = notes;
        System.out.println("Consultation started with notes: " + notes);
    }

    public void completeConsult() {
        this.status = "Completed";
        System.out.println("Consultation " + consultId + " completed.");
    }

    public void addDiagnosis(String diagnosis) {
        this.notes += "\nDiagnosis: " + diagnosis;
    }

    public void issuePrescription(Prescription p) {
        if (this.status.equals("Completed")) {
            this.prescription = p;
            System.out.println("Prescription issued for consultation " + consultId);
        } else {
            System.out.println("Cannot issue prescription until consult is completed.");
        }
    }
}
```

Prescription.java

```
public class Prescription {
    private String rxId;
    private Consultation consultation;
    private String medicines;
    private String dosage;
    private String validity;

    public Prescription(String rxId, Consultation consultation, String medicines, String dosage,
String validity) {
        this.rxId = rxId;
        this.consultation = consultation;
        this.medicines = medicines;
        this.dosage = dosage;
        this.validity = validity;
    }

    public String getRxId() { return rxId; }
    public String getMedicines() { return medicines; }
    public void setMedicines(String medicines) { this.medicines = medicines; }

    public String getDosage() { return dosage; }
    public void setDosage(String dosage) { this.dosage = dosage; }

    public String getValidity() { return validity; }
    public void setValidity(String validity) { this.validity = validity; }

    public void displayPrescription() {
        System.out.println("Prescription ID: " + rxId);
        System.out.println("Medicines: " + medicines + ", Dosage: " + dosage + ", Valid till: " +
validity);
    }

    public void extendValidity(String newDate) {
        this.validity = newDate;
        System.out.println("Validity extended till " + newDate);
    }

    public void updateMedicines(String newMedicines) {
        this.medicines = newMedicines;
        System.out.println("Medicines updated: " + newMedicines);
    }
}
```

TeleMedAppMain.java

```
public class TeleMedAppMain {
    public static void main(String[] args) {
        Doctor doc1 = new Doctor("D139", "Kishore", "kishore@med.com", "Cardiology", "LIC1234");
        doc1.addSlot("2025-09-12 10:00");
        doc1.addSlot("2025-09-12 11:00");

        Patient pat1 = new Patient("P301", "Gowtham", "gowtham@mail.com", 20, "Male");

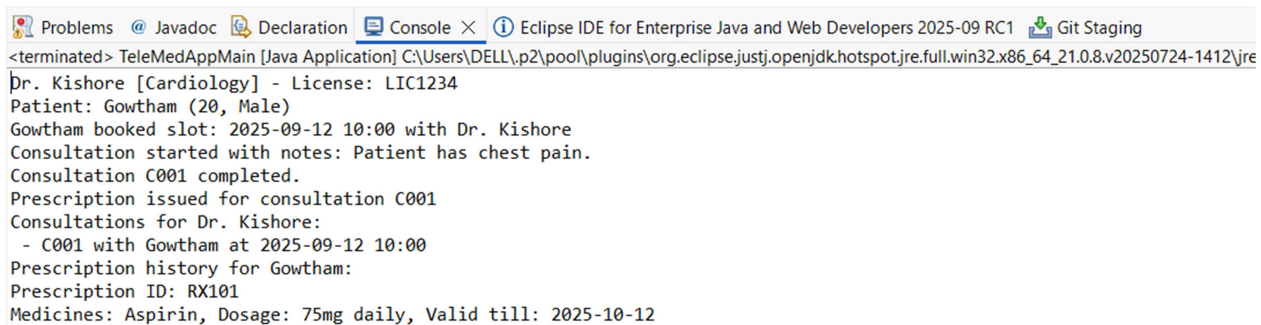
        doc1.displayProfile();
        pat1.displayProfile();
        pat1.requestConsultation(doc1, "2025-09-12 10:00");
        Consultation consult1 = new Consultation("C001", doc1, pat1, "2025-09-12 10:00");
        doc1.addConsultation(consult1);
        pat1.addConsultation(consult1);

        consult1.startConsult("Patient has chest pain.");
        consult1.addDiagnosis("Possible Angina");
        consult1.completeConsult();

        Prescription rx1 = new Prescription("RX101", consult1, "Aspirin", "75mg daily", "2025-10-12");
        consult1.issuePrescription(rx1);

        doc1.viewSchedule();
        pat1.viewPrescriptionHistory();
    }
}
```

OUTPUT:



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output text is as follows:

```
<terminated> TeleMedAppMain [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre
Dr. Kishore [Cardiology] - License: LIC1234
Patient: Gowtham (20, Male)
Gowtham booked slot: 2025-09-12 10:00 with Dr. Kishore
Consultation started with notes: Patient has chest pain.
Consultation C001 completed.
Prescription issued for consultation C001
Consultations for Dr. Kishore:
- C001 with Gowtham at 2025-09-12 10:00
Prescription history for Gowtham:
Prescription ID: RX101
Medicines: Aspirin, Dosage: 75mg daily, Valid till: 2025-10-12
```

Git Repository link:

<https://github.com/717824p139-PerianayagiDivya/TeleMedicineApp.git>