Industry Problem Statement

Hospital Patient Billing & Appointment Management System (Python)

Business Background

A multi-specialty hospital wants to build a Python-based internal system to manage:

- Patient registration
- Doctor appointments
- Consultation charges
- Treatment billing and payment tracking

The system should reflect **real-world hospital workflows** used in HMIS (Hospital Management Information Systems).

Task 1: Capture Patient Registration Details (Input Validation)

Objective

Register a patient in the hospital system.

Requirements

Write a program to accept:

- Patient ID
- Patient Name
- Age
- Gender

Business Rules

- Age must be greater than 0

- Gender must be one of: Male / Female / Other
- Patient name must not be empty

Expected Outcome

Validated patient record.

Task 2: Capture Doctor Details

Objective

Maintain doctor master data.

Requirements

Accept:

- Doctor ID
- Doctor Name
- Specialization
- Consultation Fee

Business Rules

- Consultation fee must be greater than 0

Expected Outcome

Validated doctor record.

Task 3: Appointment Booking

Objective

Schedule a patient appointment.

Requirements

- Select doctor
- Accept appointment date

Business Rules

- Appointment date cannot be in the past

Expected Outcome

Confirmed appointment entry.

Task 4: Appointment Status Tracking

Objective

Track patient visit.

Status Options

- Scheduled
- Completed
- Cancelled

Expected Outcome

Updated appointment status.

Task 5: Consultation Charge Application

Objective

Apply consultation fee after visit.

Requirements

- Add doctor’s consultation fee to patient bill

Expected Outcome

Consultation charges recorded.

Task 6: Diagnostic Test Billing

Objective

Add diagnostic services.

Requirements

- Accept multiple test names and charges

Business Rules

- Each test charge must be > 0

Expected Outcome

Accumulated diagnostic bill.

Task 7: Treatment Cost Calculation

Objective

Calculate total treatment cost.

Formula

```
Total Treatment Cost = Consultation Fee + Sum of Test Charges
```

Expected Outcome

Accurate treatment bill amount.

Task 8: Insurance Coverage Processing

Objective

Apply insurance benefits.

Requirements

- Accept insurance coverage percentage

Business Rules

- Coverage must be between 0% and 100%

Formula

```
Insurance Deduction = Total Cost × Coverage %
```

Expected Outcome

Reduced payable amount.

Task 9: Net Payable Amount Calculation

Objective

Determine final bill.

Formula

```
Net Payable = Total Cost - Insurance Deduction
```

Expected Outcome

Correct patient bill.

Task 10: Procedural Billing Summary

Objective

Generate billing snapshot.

Summary Should Include

- Patient Name
- Doctor Name
- Consultation Fee

- Test Charges
- Insurance Deduction
- Net Payable

Task 11: Patient Class Design (OOP)

Objective

Model patient as an object.

Create class `Patient` with:

Attributes

- patient_id
- name
- age
- gender
- appointments
- bills

Task 12: Appointment Management Method

Objective

Encapsulate appointment logic.

Method

- book_appointment()

Task 13: Billing Methods

Objective

Encapsulate billing logic.

Methods

- add_consultation_fee()
- add_test_charge()
- calculate_total_bill()

Task 14: Insurance Processing Method

Objective

Automate insurance deduction.

Method

- apply_insurance()

Task 15: Final Patient Invoice Generation

Objective

Generate a professional hospital invoice.

Output Format (Example)

```
Patient ID        : P305
Patient Name      : Suresh Mehta
Doctor            : Dr. Ananya Rao (Cardiology)
Appointment Date  : 12-Mar-2025

Consultation Fee  : ₹800
Diagnostic Tests  : ₹2,200
Insurance Covered : ₹900
--------------------------------
Net Payable       : ₹2,100
Payment Status    : Pending
```