

Java_Assignment-05

Abstraction - Online Food Delivery System

Scenario: Design an **Online Food Delivery System** where different food ordering platforms extend a base abstract class.

- Abstract class: FoodDeliveryApp
- Properties: appName, deliveryCharge
- Abstract method: placeOrder()
- Method: showDetails() to display app info

Subclasses:

- Zomato → Implements placeOrder() with "Ordering food via Zomato!"
- Swiggy → Implements placeOrder() With "Ordering food via Swiggy!"

Task:

- Take user input for choosing the food delivery app.
- Display app details and place an order accordingly.

Multiple Inheritance using Interfaces - Smart Home System

Scenario: Create a **Smart Home System** where devices implement multiple interfaces.

Interfaces:

- WiFiEnabled: Method connectToWiFi()
- VoiceControl: Method activateVoiceCommand()
- Class: SmartSpeaker (implements both interfaces)
- Implements connectToWiFi() → "Smart Speaker connected to WiFi."
- Implements activateVoiceCommand() → "Listening for voice command..."

Task:

- Take user input for connecting a Smart Speaker.
- Display WiFi and Voice Control status.

Scanner & User Input - Fitness Tracker App

Scenario: Create a **Fitness Tracker** that takes user input and calculates daily calories burned.

- ◆ Class: FitnessTracker
 - Take user input using Scanner for:
 - o stepsWalked
 - workoutMinutes
 - o caloriesBurnedPerMinute
- Method calculateTotalCalories() → Calculates total calories burned

Task:

- Ask the user for input values.
- Calculate and display total calories burned.

Example Output:

Enter steps walked: 5000 Enter workout minutes: 30

Enter calories burned per minute: 5

Total Calories Burned: 150

Multiple Inheritance using Interfaces - Online Payment Gateway

Scenario: Build an **Online Payment Gateway** where a payment processor implements multiple interfaces.

Interfaces:

- UPI: Method processUPITransaction()
- CardPayment : Method processCardPayment()
- Class: PaymentProcessor (implements both interfaces)

- Implements processUPITransaction() → "UPI Transaction Successful!"
- Implements processCardPayment() → "Card Payment Approved!"

> Task:

- Ask user input for choosing a payment mode.
- Display transaction confirmation accordingly.

Abstraction & Scanner Input - Online Exam System

Scenario: Implement an **Online Exam System** where students can attempt MCQ tests.

- ♦ Abstract class: OnlineExam
- Abstract method startExam()
- Method showInstructions()
- ◆ Subclass: JavaExam (implements startExam())
- Displays 3 MCQs, takes user answers, and calculates the score.

Task:

- Take user input for answers.
- Calculate and display final score.

Example Output:

Q1: What is JVM?

- 1. Java Virtual Machine
- 2. Java Variable Method
- 3. Java Verified Module

Enter your answer (1/2/3): 1

• • •

Final Score: 2/3

Submission Guidelines:

- Implement each program separately and test it.
- Use **Scanner** for user input.

• Have fun coding! DM for doubts! 😎