Engineer - Software - 2nd Round Interview Coding Test - Fruit Sales Calculator

Overview

You are building a small system for a fruit shop that needs to calculate the price of fruit orders. The shop sells different types of fruit (apples, bananas, cherries, etc.) and occasionally applies different pricing strategies (e.g., by weight, by item count, or with seasonal discounts).

Objective

Build a system that:

- Can represent fruit with **varying pricing models**
- Can calculate the total price of an order based on the fruit and **quantity or weight**
- Can easily support new **pricing strategies** or new fruit types
- Demonstrates use of at least one **design pattern**
- Is well-tested with **unit tests** covering core functionality

Functional Requirements

Each fruit has:

- A name (e.g., “Apple”)
- A base price (e.g., $2.00 per kg or $0.50 per item)
- A **pricing method** (per kg, per item, or with discount logic)

The system must:

- Accept an order consisting of different fruits and quantities/weights
- **Calculate and print** the total cost of the order

Example fruit types:

- Apple: $2.00 per kg
- Banana: $0.30 per item
- Cherry: $5.00 per kg with a 10% discount for more than 2kg

Technical Requirements

- Use C# (.NET 8 or higher)
- Use **at least** one **design pattern** (e.g., Decorator, Factory, Strategy, Command,

etc — your choice)

- Write **unit tests**, using XUnit and arrange, act, assert notation, that validate each method
- **At least two different** pricing method types
- The **total cost calculation** for an order
- Include a README that explains:
  - Your **design decisions**
  - What **design pattern(s)** you used and why
  - How you would **extend** it to **support** new fruit, discounts, or pricing

models

Submission Instructions

- Submit your code in a **GitHub repo** (preferred) or a zip file, do not submit any compiled

output

- You may use any tools at your disposal (any IDE, any libraries, any AI or other

tools etc) to help write your code, but be prepared to discuss

- You must be able to explain your solution and choices in a follow-up interview

Follow-Up Interview

We'll ask you to:

- Walk through your **code and design**
- Explain the **design patterns** you implemented
- Show how you'd add a **new feature** (e.g., bulk discount, loyalty pricing)
- Discuss how you ensured your tests give confidence in the code's behaviour