How to Approach Object-Oriented Design Interview (Template)

Object Oriented Design Interview Template

An Object Oriented Design Interview usually lasts for 45-60 minutes. The following template will guide you (with a basic example to get you an idea) on how to manage time duration across various aspects of it -

✅ Requirement Clarifications (3-5 mins)

The OOD interview questions are often abstract and vague. Always ask clarifying questions to understand the problem and find the exact scope of the system that the interviewer has in mind.

👉 Focused use cases to cover (MVP)
👉 Use cases that will not be covered
👉 Who will use the system (Actors)
👉 How the system will be used

e.g. Let's **design an online shopping site**. A few requirements would be -

👉Users should be able to search products based on the name.
👉Users should be able to view/buy products.
👉Users should be able to add/remove product items in their shopping cart.
👉User can place an order.
👉Users should get notifications about orders.
👉User should be able to pay through different modes.

✅ Use cases to cover (3-6 mins)

You are expected to find possible actors of the system and write down different use cases the system will support.

👉Actors could be -
1. Customer
2. Admin
3. System

Some of the use cases could be -

👉 Search products based on the name.
👉 add/remove/modify products in the shopping cart.
👉 Check out to buy items in the shopping cart.
👉 Make payment to place an order
👉 Send a notification to the user about the order.

✅ Identify the Core classes (3-6 mins)

After gathering requirements and drafting a few use cases, our understanding of what we are designing becomes clear. Now we should consider what could be the main classes of the system. You are expected to sketch a class diagram or write down class names.

The way to identify classes or entities is -

👉Nouns in the requirements are possible candidates for Classes.

Some of the core classes could be -

👉 Product
👉 Item
👉 User
👉 ShoppingCart
👉 Order
👉 Payment
👉 Notification

✅ Identify the fields/properties of each class (5-10 mins)

Once we know the core classes/objects of the system, it is expected to draw a class diagram along with class fields. Take each class identified in the above steps and add a few important properties which drive the use cases of the system.
eg.

👉 Product
    - name
    - description
    - price ...

👉 User
    - name
    - email
    - phone …

✅ Identify the relationship between the classes (5-10 mins)

Once we know the core classes/objects of the system, it is expected to draw what is the relationship between the classes. The relationship mainly focuses on is-a and has-a between classes. The different types of relationships to draw or write are -

👉 Are there any classes that are very generic and more concrete classes that can be sketched?
👉 Are there any one-to-one, one-to-many, and many-to-many relationships between the classes. eg.

    👉 Customer, Guest, and Admin inherit from User
    👉 Customer has One Shopping Cart
    👉 Shopping Cart has Many Items
    etc ...

✅ Identify the possible actions of the classes (5-10 mins)

Once we are clear with the requirements, use cases and possible design of the system, etc, it is time to identify the different actions classes will perform based on their relationship.

The way to identify class actions is -

👉 Verbs in the requirements/use cases are possible candidates for actions these classes perform. Those can be taken as methods of the classes.
eg.

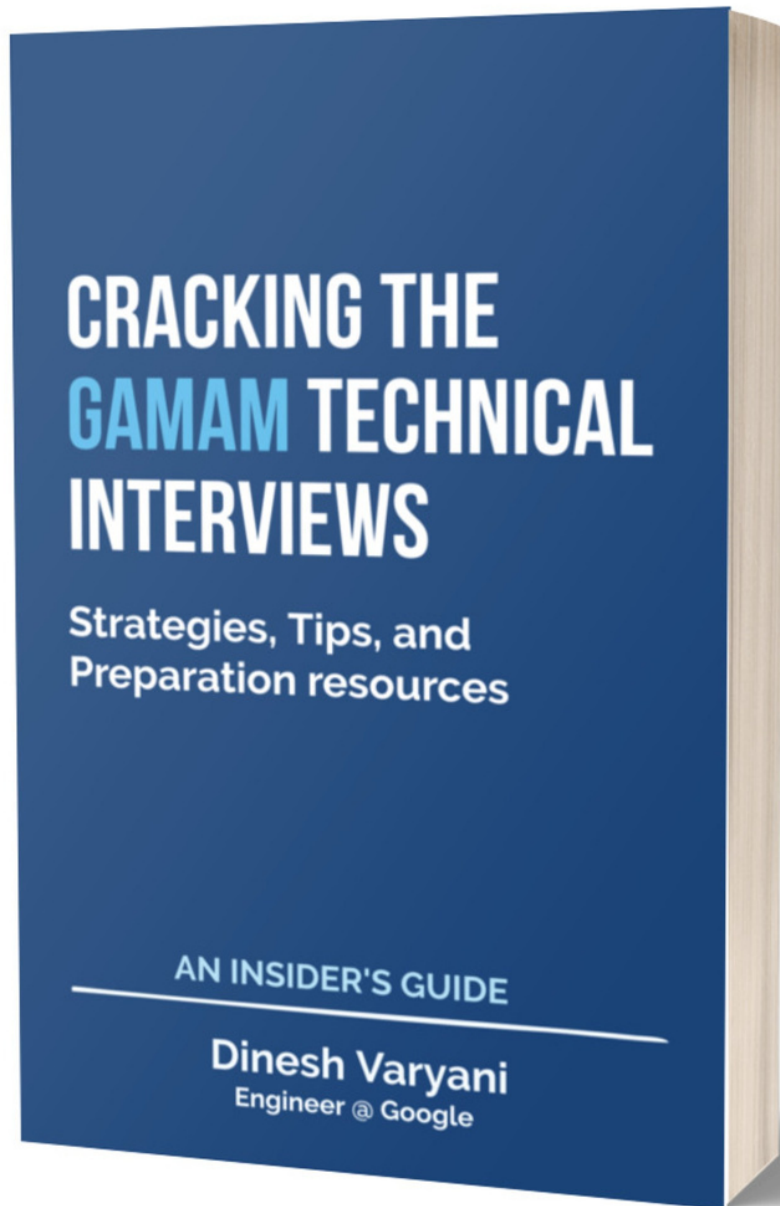👉 Customer can add the item to the shopping cart - **addItemToCart(Item item)**
👉 Customer can place an order - **placeOrder(Order order)** etc…

✅ Code (5-8 mins) (Optional)

The interviewer will ask you to write code for a specific use case by taking the above classes. The class diagram will give you an idea about the class's name, fields, and methods. You are expected to write code for the methods which fulfill the use case interviewer wants or any algorithm/data structure which handles certain use cases.

✅ Resolve bottlenecks and follow-up questions (3-4 mins)

Cracking the GAMAM Technical Interviews

CRACKING THE
GAMAM TECHNICAL
INTERVIEWS

Strategies, Tips, and
Preparation resources

AN INSIDER'S GUIDE

Dinesh Varyani
Engineer @ Google

Buy Now

THANK YOU !!!
For more such content follow

@Dinesh Varyani