

Student Name: Rana Asad Ur Rahman

Registration No.: SP23-BSE-029

Design Principles and Patterns for the “Place Order” Use Case

1. GRASP Patterns (3 Marks)

Pattern	Intent	Implementation in Code
Controller	Assign system-event processing to a non-UI class that embodies the session or use-case.	OrderController.placeOrder() receives the call, decouples UI from business logic, and forwards to OrderService.
Creator	Assign the creation of an object to the class that uses it frequently or aggregates it.	OrderFactory.createOrder(customerId, items) centralizes Order instantiation (including ID generation).
High Cohesion	Group related behaviors together and keep each class tightly focused on a particular duty.	<ul style="list-style-type: none">– Order/OrderItem: data carriers– OrderFactory: object creation– OrderService: business rules– InventoryService/PaymentService: reaction logic– OrderController: orchestration

2. GoF (Gang of Four) Patterns (2 Marks)

Pattern	Intent	Implementation in Code
Factory Method	Provide an interface for creating objects, leaving the intricacies of instantiation to a factory class.	OrderFactory.createOrder(...) encapsulates all setup (UUID generation, initial state) for Order objects.
Observer	Permit an object (subject) to automatically notify several dependents (observers) of changes in its condition.	OrderService maintains a list of OrderObserver instances (InventoryService, PaymentService) and calls onOrderPlaced(order) on each when an order is created.

