Creating an object model for a Ready-Made Garments (RMG) system in Java involves defining the classes, their attributes, and the relationships between them. Below is a detailed specification for this object model, which includes the necessary classes, attributes, methods, and their relationships.

Object Model Specification for Ready Made Garments are -

**Classes**

**1. Garment**

**Attributes:**

String id - Unique identifier for the garment.

String name - Name of the garment (e.g., T-shirt, Pants, etc.).

String description - Description of the garment.

String size - Size of the garment (S, M, L, XL, etc.).

String color - Color of the garment.

double price - Price of the garment.

int stockQuantity - Number of items available in stock.

**Methods:**

void updateStock(int quantity) - Updates the stock quantity.

double calculateDiscountPrice(double discountPercentage) - Calculates the discounted price based on the provided percentage.

**2. Fabric**

**Attributes:**

String id - Unique identifier for the fabric.

String type - Type of the fabric (e.g., Cotton, Polyester).

String color - Color of the fabric.

double pricePerMeter - Price per meter of the fabric.

**Methods:**

double calculateCost(double meters) - Calculates the cost based on meters.

**3. Supplier**

**Attributes:**

String id - Unique identifier for the supplier.

String name - Name of the supplier.

String contactInfo - Contact information of the supplier.

List<Fabric> suppliedFabrics - List of fabrics supplied by the supplier.

**Methods:**

void addFabric(Fabric fabric) - Adds a fabric to the supplier's list.

List<Fabric> getSuppliedFabrics() - Returns the list of supplied fabrics.

**4. Order**

**Attributes:**

String orderId - Unique identifier for the order.

Date orderDate - Date when the order was placed.

List<Garment> garments - List of garments included in the order.

double totalAmount - Total amount for the order.

**Methods:**

void addGarment(Garment garment) - Adds a garment to the order.

double calculateTotalAmount() - Calculates the total amount of the order.

void printOrderDetails() - Prints order details.

**5. Customer**

**Attributes:**

String customerId - Unique identifier for the customer.

String name - Name of the customer.

String email - Email address of the customer.

String phone - Phone number of the customer.

**Methods:**

void placeOrder(Order order) - Places an order.

List<Order> viewOrders() - View orders made by the customer.

**6. Inventory**

**Attributes:**

List<Garment> garments - List of garments in inventory.

**Methods:**

void addGarment(Garment garment) - Adds a new garment to inventory.

void removeGarment(String id) - Removes a garment from inventory by ID.

Garment findGarment(String id) - Finds a garment by ID.

**Relationships Among Classes**

Garment uses Fabric: Each garment is made of a fabric, so an attribute in the Garment class could refer to the Fabric class.

Order contains multiple Garments: The Order class will hold a collection of garments that are being ordered.

Supplier provides Fabric: A supplier provides various fabrics, so there’s a one-to-many relationship (one supplier can supply many fabrics).

Customer can place multiple Orders: A customer can place multiple orders, establishing a one-to-many relationship between Customer and Order.