Main class:

This code operates by simulating various actions in a store. It creates instances of **SalableProduct**, **InventoryManager**, and **ShoppingCart**. Then, it adds products to the inventory and displays the inventory contents. Next, it adds products to the shopping cart, displays its contents, and removes a product from it. Afterward, it displays the updated contents of the shopping cart and confirms its emptiness by emptying the shopping cart.

SalableProduct Class:

This code defines a class that represents products for sale. It encapsulates product attributes such as **productId**, **name**, and **quantity**. The class provides a constructor to initialize these attributes and getter and setter methods to access and modify them. It serves as a blueprint for creating objects representing products available in the store.

InventoryManager Class:

This class manages the store's inventory, which consists of **SalableProduct** objects. It maintains a list of products and provides methods to add and remove products from the inventory, as well as retrieve the inventory contents. However, the code lacks implementation details for adding and removing products, which would need to be implemented elsewhere in the program.

StoreFront Class:

This class contains methods to integrate the **InventoryManager** and **ShoppingCart** components into the store's system. However, the method bodies are empty, serving as placeholders for future implementation. In a real application, these methods would likely contain logic to connect and coordinate actions related to inventory management and shopping cart functionalities.

ShoppingCart Class:

This class represents a shopping cart and manages its contents, which are **SalableProduct** objects. It provides methods to add and remove products from the cart, retrieve the cart contents, and empty the cart. Similar to the **InventoryManager** class, the implementation details for adding and removing products are not provided in the code.

Weapon Class:

This class defines attributes and methods for a type of weapon product in the store. It encapsulates data such as **productId**, **name**, and **quantity**. Additionally, it implements the **Comparable** interface to enable comparison between **Weapon** objects based on their names. However, the code lacks the implementation of the **compareTo** method, which is required by the **Comparable** interface and would need to be added for proper functionality