

Abstract Class Challenge

In this challenge, you need to build an application that can be a store front for any imaginable item for sale.

Instead of the Main class we usually create, create a Store class with a main method.

The **Store** class should:

- manage a **list of products for sale**, including displaying the product details.
- manage an order, which can just be a **list of OrderItem** objects.
- have methods to **add an item to the order**, and **print the ordered items**, so it looks like a sales receipt.

Abstract Class Challenge

Also create a **ProductForSale** class that should have at least three fields: a **type**, **price**, and a **description**. This class should also have methods to:

- `getSalesPrice`, a **concrete method**, which takes a **quantity**, and **returns the quantity times the price**.
- `printPricedItem`, a **concrete method**, which takes a **quantity**, and should **print an itemized line item** for an order, with **quantity and line-item price**.
- `showDetails`, an **abstract method**, which represents what might be **displayed** on a product page. **Product type, description, price**, and so on.

Also create an **OrderItem** type, that has at a minimum 2 fields, **quantity** and a **Product for Sale**.

You should also create **two or three classes that extend the ProductForSale class**, that will be products in your store.

The Design

Let's look at my approach.

Here is the class diagram of what you'll be building.

This covers all the requirements I talked about.

Notice I'm specifying that OrderItem will be a record. This is just to keep the code simple.

I'm not really specifying what our store products are, you can really put anything there.

