1. **Shopping cart 1**
   Create a new class called Item with the following properties/attributes:

   name
   price
   stock
   sold
   Create 3 instances of item.

   Now add a constructor method to the class and require the user to specify the name, price, and stock of each instance. In the constructor also specify in the code so that the initial sold is set to be 0 whenever a new instance is created.

   Add the following functions for this class:

   logDetails() - have this method display the item's name, price, number of stock, and the total sold. buy() - have it display "Buying" on the screen and increase the total sold by 1. return() - have it display "Returning" on the screen and decrease the total sold by 1.

   Have the first instance buy three times, return once, and have it logDetails(). Have the second instance buy twice, return twice, and have it logDetails(). Have the third instance return three times and logDetails().

   What would you do to prevent the instance from having negative sold, and buying out-of-stock?

2. **House**
   Create a class called House. In the constructor, allow the user to specify the following attributes: location, price, lot, and type. If the type is Pre-selling, set the discount to 20%. Otherwise, set the discount to be 5%.

   Create five different instances of the class House. In the class have a method called show_all() that returns all the information about the house as a string. In your constructor,

call this show_all() method to display information about the house when a new house is created.

A sample output would be like this:

```
Location: La Union
Price: 1500000
Lot: 100sqm
Type: Pre-selling
Discount: 0.2
Total Price: 1200000

Location: Metro Manila
Price: 1000000
Lot: 150sqm
Type: Ready for Occupancy
Discount: 0.05
Total Price: 950000
```

3. Email builder

Create an “EmailBuilder” class that allows you to build an email message with various components. The class should have methods to set the recipient, subject, body, and attachments. Have these methods chained to achieve a real flow on sending an email.

Add a “send” method that logs the constructed email details (in a real-world scenario, you would implement actual email sending logic).

Below should be how it should look like:

```
Sending email:
Recipient: john@example.com
Subject: Meeting Tomorrow
Body: Hi John, see you there!
Attachments: Array(1)
```