Introduction

The test should take no longer than 2 hours to complete.

Please complete the test on your own machine and in the same way that you would if you were doing it for work. You will need the same machine with the code running for the next stage of the interview so please keep your solution. If for any reason working on your own machine isn't an option for you please let me know and we will be happy to make alternative arrangements.

Please complete this test in one of the following Languages:

- Java
- Kotlin
- dotnet (C#)
- NodeJS with Typescript

Task

We are working with a client who wants to launch an e-commerce site to promote their brand, below are some of the products that they will be listing:

ID | Name | Price

0001 | Water Bottle | £24.95

0002 | Hoodie | £65.00

0003 | Sticker Set | £3.99

The marketing team would like to run the following promotions;

- If you spend over £75 then you get a 10% discount
- If you buy two or more water bottles then the price drops to £22.99 each

Multiple promotions can be applied to the same checkout

The checkout system needs to be able to scan the items in any order then apply the promotion rules. These rules should be allowed to change over time. The pseudocode below outlines the interface for the checkout

```
checkout = new checkout(promotionalRules)
checkout.scan(items) //List of item objects created from the products above
```

```
println(checkout.total()) //e.g. £68.99
```

Implement a checkout that meets these requirements.

Do not worry about the storage or retrieval of any items in a database, for the purpose of this test they only need to be processed.

Test Data

Items: 0001,0001,0002,0003

Total Price: £103.47

Items: 0001,0001,0001

Total Price: £68.97

Items: 0002,0002,0003

Total Price: £120.59

Submitting the test

Once you have completed the test please reply to this email with a zip file or tarball containing your solution.