

Programming exercise – supermarket checkout

Written in the Java programming language, implement the code for a supermarket checkout that calculates the total price of a number of items.

In a normal supermarket, items for sale are identified using Stock Keeping Units, or 'SKUs'. In our store, we'll use individual letters of the alphabet (A, B, C, and so on) to represent these SKUs. Our goods are priced individually, however, some items are multi-priced: buy **n** of them, and they'll cost you **y** instead.

For example, item 'A' might cost **£0.50** individually, but this week we have a special offer: buy three 'A's and they'll cost you **£1.30**. In fact, this week's prices are:

Item	Unit Price	Special Price
A	50	3 for 130
B	30	2 for 45
C	20	2 for 38 3 for 50 ¹
D	15	5 if purchased with an 'A' ²
E	5	

- This exercise will be used to gauge how you approach a software engineering problem – the processes you use, the quality of your code and the robustness of your solution.
- **DO** use as many or as few technologies and processes as you normally would when working as a Software Engineer
 - Feel free to write tests, use version control and rely on the third party libraries provided.
- **DON'T** get hung up on the specifics of the implementation
 - The problem is intentionally abstract, giving you the freedom to come up with your own unique solution.
 - This is an opportunity to demonstrate your way of working and your approach to creative problem solving – there are no precise user requirements (aside from the specification above).
- A template project has been set up in IntelliJ IDEA on a feature branch called **feature-exercise-<yourname>** to get you started.
 - The JUnit and Google Guava libraries have been included as Maven dependencies for you to make use of – if you wish.
- This exercise will last approximately **1.5 hours**

¹ The price calculated for any quantity of an SKU with multiple special prices will be the cheapest combination of its special prices. For example: If you buy 5 'C's you would get **2 for 38 + 3 for 50**. If you buy 4 'C's you would get **3 for 50 + 1 for 20** rather than **2 for 38 + 2 for 38**.

² For every 'D' purchased, if there is also an 'A' purchased, it will cost **5** instead of **15**. For example, if you buy **10** 'D's and **6** 'A's, **6** of the 'D's will cost **5 each** whilst the other **4** will cost **15 each**.