SEEK International coding exercise

We are interested in seeing your coding and problem solving style, so we would love if you could complete this open code test.

For the purpose of this exercise, SEEK is in the process of rewriting his job ads checkout system. You have been assigned to build this new system.

We want to offer different level of ads to recruiters:

- **Classic A**d, offers the most basic level of advertisement
- **Standout Ad**, allows advertisers to use a company logo and use a longer presentation text
- **Premium Ad**, same benefits as Standout Ad, but also puts the ad at the top of the results, allowing higher visibility

Each of the ads is billed as follows:

ID	**Name**	**Price**
classic	Classic Ad	$269.99
standout	Standout Ad	$322.99
premium	Premium Ad	$394.99

We established a number of special pricing rules for a small number of privileged customers:

- Unilever
  - Gets a for **3 for 2 deal on Classic Ads**
- Apple
  - Gets a discount on **Standout Ads where the price drops to $299.99 per ad**
- Nike
  - Gets a discount on **Premium Ads when 4 or more** are purchased. The price drops to **$379.99 per ad**

- Ford
    - Gets a **5 for 4 deal on Classic Ads**
    - Gets a discount on **Standout Ads where the price drops to $309.99 per ad**
    - Gets a discount on **Premium Ads when 3 or more** are purchased. The price drops to **$389.99 per ad**

These deals are regularly renegotiated, so we want the pricing rules to be as flexible as possible as they can change in the future with little notice.

The interface to our checkout looks like this (shown in Ruby-ish pseudocode):

```
Checkout co = Checkout.new(pricingRules)
co.add(item1)
co.add(item2)
co.total()
```

Your task is to implement a checkout system that fulfils the requirements described above.

Example scenarios

Customer: default
ID added: `classic, standout, premium`
Total expected: $987.97

Customer: Unilever
SKUs Scanned: `classic, classic, classic, premium`
Total expected: $934.97

Customer: Apple
SKUs Scanned: `standout, standout, standout, premium`
Total expected: $1294.96

Customer: Nike
SKUs Scanned: `premium, premium, premium, premium`
Total expected: $1519.96

Notes on implementation

- We do not want you to lose a weekend trying to solve this problem. Only spend enough time required to produce an appropriate, clean, testable and maintainable solution to the stated problem.
- Back-end implementations won’t be assessed at all on the quality of the UI, if any is delivered. Front-end implementations should cover matters such as usability. In both cases, we are more interested in your approach rather than how shiny it looks.
- Deliver any code, test code and test data required so that this can be fully reviewed for accuracy and completeness of the solution that you prepare.