Link Apprenticeship Coding Exercise

Why are we asking for a coding submission?

This is not a complex computer science puzzle. Rather, what we're looking for is how you can translate a business problem into code. Remember, there is no perfect solution to this problem!

This coding submission serves two purposes:

1. We'll use it to get a sense of your approach to writing software.
2. In the interview process, we'll be pair programming with you! We'll use your coding submission so you're familiar with the problem and code we're working with.

Instructions

- Please complete the coding exercise described below. This problem gives us a chance to see how you approach testing and problem decomposition. Note—be sure to preserve your commit history (i.e. don't squash your solution into a single commit.)

  Push your solution to a publicly accessible repository (e.g. GitHub). Once you're ready, send along the link to your repo.

- Please also include a reflection on submission (you can create a text file and include it somewhere in the repo.) The intent here is to give you an opportunity to comment on your submission. As such, there's no formal requirements. However, you might consider commenting on what other design options you considered, and what principles guided certain decisions. Also, if there are aspects of your implementation that you had questions about, feel free to discuss that here.

Coding Exercise

Answer the following question using Ruby, JavaScript or Java without using any third party gems, plugins, or jar files with the exception of test frameworks (i.e. MiniTest, RSpec, Mocha/Chai, JUnit, etc).

- Treat this problem as a library. No UI or file/console input is expected or required.

- Add a Readme file that at least includes:
  - runtime versions (e.g. Ruby 2.3)
  - how to install any dependencies
  - how to run your tests from the command line

- Many commits. Don't squash into a single commit.

- Please submit via publicly accessible repo so that we can review your commits.

Note: if you choose to submit a Java solution either include the config files for Eclipse or IntelliJ and/or include a build script written using Ant, Maven, or Gradle so we can easily build your solution and run your tests.

Pricing problem
===============

NuPack is responsible for taking existing products and repackaging them for sale at electronic stores like Best Buy. Companies will phone up NuPack, explain the process and NuPack needs to quickly give them an estimate of how much it will cost. Different markups to the job:

* Without exception, there is a flat markup on all jobs of 5%
* For each person that needs to work on the job, there is a markup of 1.2%

Markups are also added depending on the types of materials involved:

* If pharmaceuticals are involved, there is an immediate 7.5% markup
* For food, there is a 13% markup
* Electronics require a 2% markup
* Everything else, there is no markup

Another system calculates the base price depending on how many products need to be repackaged. As such, the markup calculator should accept the initial base price along with the different categories of markups and calculate a final cost for a project.

The flat markup is calculated first and then all other markups are calculated on top of the base price plus flat markup.

Example 1:

Input: $1,299.99, 3 people, food
Output: $1,591.58

Example 2:

Input: $5,432.00, 1 person, drugs
Output: $6,199.81

Example 3:

Input: $12,456.95, 4 people, books
Output: $13,707.63