CASH CODING EXERCISE V3.1

Guidelines

We want you to spend around 2 hours on this exercise, please don't spend much longer than that. Once you're done please zip it up and email it to me at nabila@squareup.com. During the onsite interview we will pair with you to extend this exercise. We review these anonymously to reduce bias, please don't put your name in the code or host it in a public place.

Exercise

Code a bank with the following functionality:

- Deposit, withdraw and hold money for multiple customers
- Return a customer's balance or the bank's total balance
- Prevent customers from withdrawing more money than they have

Example test scenario

When Alice deposits $30 and withdraws $20
Then Alice's balance will be $10 and the banks balance will be $10
Alice will be prevented from withdrawing another $20 to prevent her balance going negative

We expect you to add more tests.

What we're looking for

Code in the language and tools you are most familiar and comfortable with. We've had successful submissions in a variety of languages both object oriented or functional.

We will be evaluating the following:

- The modeling of the bank domain and how it's translated into code
- The demonstrated knowledge of the language and tools used
- We're looking for code that is readable and maintainable by being simple, concise, idiomatic and tested
- The README is important, we need instructions and context. It's a time efficient way to communicate your assumptions, the design decisions and trade offs you're making and what you would do with more time.

To stop people spending extra time, we will mark down entries that add functionality not needed to robustly implement the requirements. We also mark down entries that add a user interface, command line tool, implement persistence or multi threaded concurrency. Classes and tests are all you need to do well.