

Software Engineer Homework

Introduction

At Carta, we help companies manage and understand **ownership**. Corporations represent ownership on a ledger called a *capitalization table*, or “cap table” for short. This table lists each of the company’s shareholders (investors, employees, and advisors), the number of shares they own, the price they paid to purchase them, and their ownership percentage. Keeping a cap table accurate and up to date is a challenging problem, and [most cap tables are broken](#) as a result.

In this exercise, we’d like you to show off your software engineering skills by modeling a simplified version of a cap table that accounts only for investors. Below is an example of such a capitalization table for the fictitious company *Unicorn, Inc.*:

Investor	Shares	Cost	Ownership %
Ben Horowitz	1,000	\$7,490	10%
Mary Meeker	5,000	\$37,450	50%
Arlan Hamilton	4,000	\$29,960	40%
Total	10,000	\$74,900	100%

The Problem

Given a list of comma-separated values representing investments, we’d like you to produce a summary cap table in JSON.

Input

```
#INVESTMENT DATE, SHARES PURCHASED, CASH PAID, INVESTOR
2016-04-03,1000,10000.00,Sandy Lerner
2017-11-14,1000,12000.00,Don Valentine
2018-01-20,2000,40000.00,Don Valentine
2018-03-20,2000,40000.00,Ann Miura-Ko
2019-01-02,2000,50000.00,Sandy Lerner
2019-01-02,1500,13500.00,Fred Wilson
```

Output

```
{
  "date": "02/01/2019", // mm/dd/yyyy format date the captable was requested
                        // (if no filter is implemented, then today)
  "cash_raised": 165500.00, // decimal representing total cash paid by
                           // investors up to date
  "total_number_of_shares": 9500, // integer representing the total number
                                  // of shares bought by investors
  "ownership": [ // list of investors and their aggregated investments
    {
      investor: "Don Valentine", // investor name
      shares: 3000, // total amount of shares owned by that given investor
      cash_paid: 52000.00, // total investment made by this investor
      ownership: 31.58 // decimal representing total investor ownership
                       // (31.58 means 31.58%)
    },
    ...
  ],
}
```

Requirements

The program you write should take in a path to a CSV file as an argument, and support an optional date parameter, in YYYY-MM-DD format. If a date argument is provided, the summary cap table should be computed using investments made on or before the specified date. If no date is given, calculate the cap table as of today. The program you write should take command line arguments.

You may use any OOP programming language, libraries, or frameworks that you like, provided it is easy for us to build and run your code.

Grading

Your submission will be evaluated on three axes:

1. Is its output **correct**?
2. Is the code **clear** and **readable**?

3. Is the code **well structured** and **production quality**?

Submittal Instructions

Submit your code as a zip file (no GitHub links, please!) named `FIRSTNAME-LASTNAME-CARTA.zip`, ie. `HENRY-WARD-CARTA.zip`

Your submission should also include a plaintext file `README.txt`, explaining how to build and run your code, and a few sentences describing any key design decisions and documenting any assumptions you made.

So we can ensure that your assignment remains anonymous during grading, please **do not** include your name *anywhere* in your code or in your `README`. The only place your name should appear in your submission is in the name of the zip file you submit.