# [redacted] Take-Home Coding Test

## Guidelines

1. Feel free to use any resource you would have access to in your day-to-day
2. Take as long as you like with the problem
3. Write clean, well-designed code
   - We are looking at how well you solve the problem, not just if you can solve it
   - Readability, modularity, scalability, performance, immutability, testing, and overall design are several factors we will highly consider when evaluating your submission.
   - Pay attention to the system behaviors, as well as inputs and outputs. Design your system to handle potential future change.
4. Solve the problem in the language you are strongest in
   - Java is highly preferred
5. When you submit your solution, please provide
   - Source code
   - README.txt
     - Language and version information so we can run it
     - Instructions on how to build and run your code
       - Please provide us with as much data here as possible about the environment used – e.g. version of language, what IDE etc.
       - If you couldn't finish the problem, please explain that
6. Prepare to discuss in an subsequent interview your decisions regarding the choice of your algorithms, implementations, architectures and designs
7. DO NOT SHARE THIS TEST WITH ANYONE

## Beta Service

### Problem Statement

Beta is a measure of the volatility—or systematic risk—of a security or portfolio compared to the market as a whole. It is calculated by comparing the return series for 2 securities against each other.

Your task is to design and create an API that will be able to compute and return a timeseries of beta coefficient values from the underlying prices of 2 securities.

You will be judged not only on correctness of the solution, but also on OO design, time & space complexity of the solution, as well as other standard software engineering principles.

### Inputs

The input into this process will be an array of stock closing prices. Fields for each row include Date, Ticker, ClosePrice. To be responsive to calls to this API, the data must be cached at start up into an easily accessible data structure. The data structure should be optimized to be able to read data. Care should be taken to understand how performance changes as the number of tickers, days, etc. is expanded (i.e. what if we needed to cache all stocks for past 30 years?).

### Outputs

The API will expose 1 method, calcBeta. It will take parameters of: Ticker, TickerBaseline, StartDate, EndDate, BetaDurationDays.

- Ticker = stock ticker for which to calculate beta
- TickerBaseline = baseline stock ticker to compare returns
- StartDate = First date to provide Beta values from
- EndDate = Last date to provide Beta values to
- BetaDurationDays = number of days to include when calculating daily beta.

The return value will an array of double representing the beta coefficient value for each of the days requested (days with a price between StartDate and EndDate). Each return value should be rounded to 4 decimals.

*Note:* To understand the relationship between StartDate, EndDate, and BetaDurationDays, consider the following example:

StartDate = 8/30/2021, EndDate = 9/3/2021, BetaDurationDays = 20. In this case, there are 5 weekdays between start date and end date, and the size of the return array will thus be 5. When calculating beta, however, you must include the prior 20 days of prices in the beta calculation. On the first date (8/30/2021), the prior 20 days would start at 8/3/2021 and go through 8/30/2021. On the next day (8/31/2021), the prior 20 days would start at 8/4/2021 and go through 8/31/2021, and so forth.

*Note:* in the case where the number of BetaDurationDays exceeds the number of days of returns available, as many days are as available should be included.

**Sample API calls:**

calcBeta(MSFT, SPY, 9/23/2020, 9/22/2021, 252)

calcBeta(AAPL, MSFT, 9/23/2020, 9/22/2021, 1260)

calcBeta(NVDA, SPY, 9/22/2021, 9/22/2021, 252)

**Beta coefficient calculation can be defined as:**

$$\text{Beta coefficient}(\beta) = \frac{\text{Covariance}(R_e, R_m)}{\text{Variance}(R_m)}$$

**Variance can be defined as:**

$$\text{variance } \sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}$$

Or, as applied to our case, SUM((Daily Return of Ticker – AVERAGE(Return of Ticker over timeseries)) ^2) / (Number of days – 1)

**Covariance can be defined as:**

$$\text{Covariance} = \frac{\sum (Return_{ABC} - Average_{ABC}) * (Return_{XYZ} - Average_{XYZ})}{(\text{Sample Size}) - 1}$$

Or, as applied to our case, SUM((Daily Return of Ticker – AVERAGE(Return of Ticker over timeseries)) * (Daily return of Baseline Ticker – AVERAGE(Return of Baseline Ticker over timeseries))) / (Number of days – 1)

**Daily return can be defined as:**

Ret = Math.log(1+ ($P_1$ – $P_0$) / $P_0$)

Where $P_1$= Price of ticker on current day, $P_0$ = Price of ticker on prior day. Note this represents the natural log of the daily return which is used in the beta calculation. Math.log with just 1 parameter returns the base e log (natural log) in both Python and Java libraries.