



InsideSherpa

JPMorgan Chase Software Engineering Virtual Experience

# **Task 1 - software engineering task : code changes**

Module 1 - Interface with a stock price data feed and create tests for it

# Disclaimer

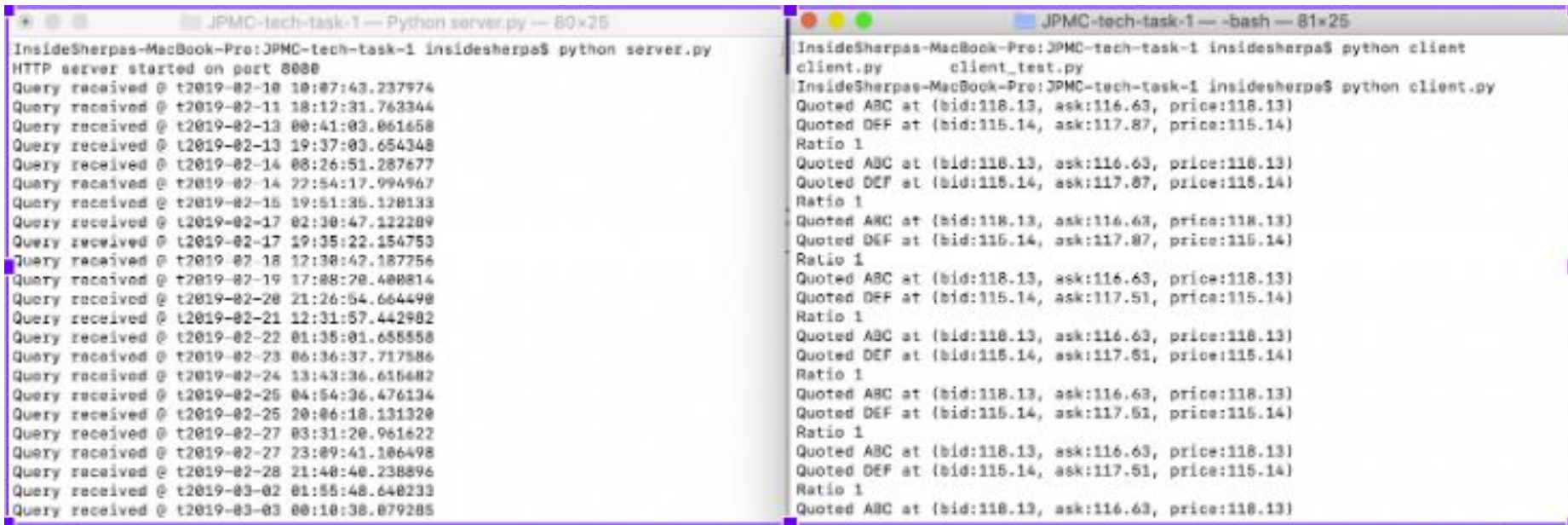
- This guide is only for those who did the setup locally on their machines.
- For those using REPL, see the REPL guide in the Instructions file of your REPL environment

# Prerequisite

- Set up should have been done. This means, your server and client applications should have been running with no problems without introducing any changes to the code yet. You can verify this if you get a similar result to any of the following slides that include a picture of the server and client app running together

# Prerequisite

Mac OS (left side server, right side client)



```

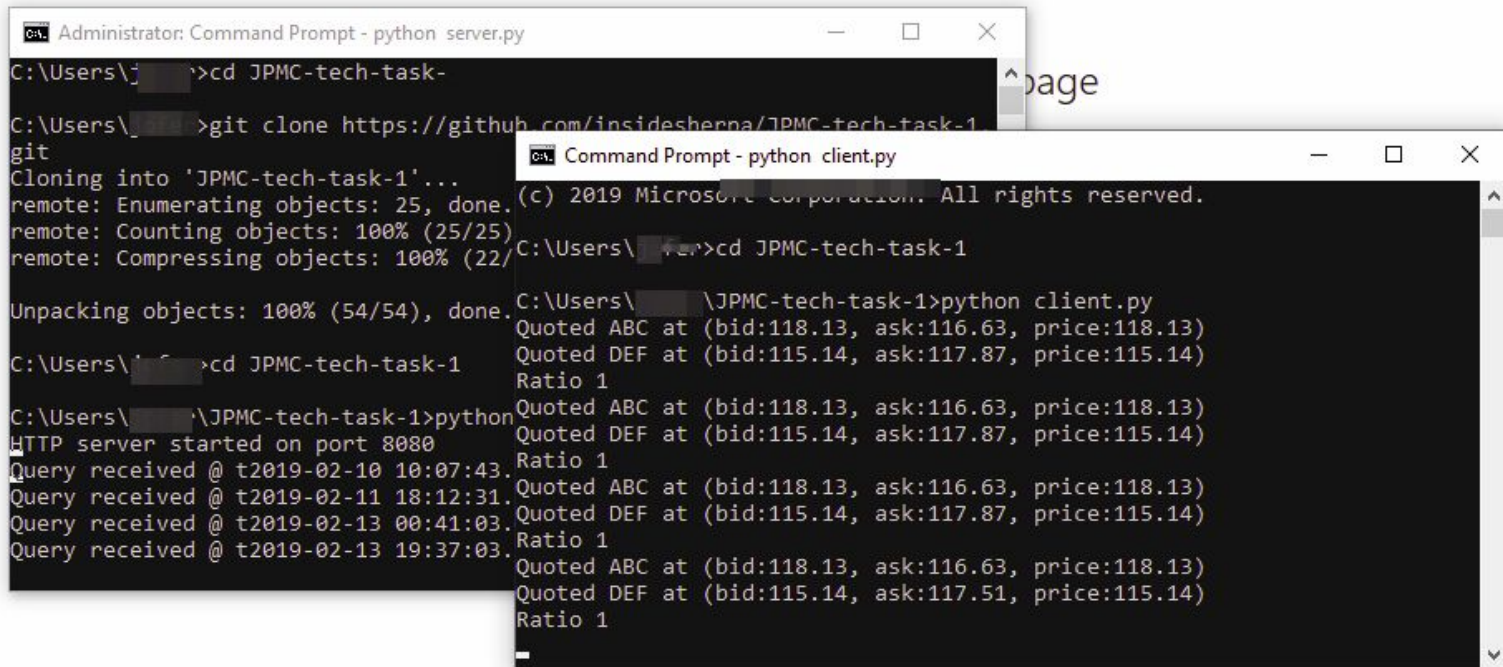
JPMC-tech-task-1 — Python server.py — 80x25
InsideSherpas-MacBook-Pro:JPMC-tech-task-1 insidesherpa$ python server.py
HTTP server started on port 8080
Query received @ t2019-02-10 10:07:43.237974
Query received @ t2019-02-11 18:12:31.763344
Query received @ t2019-02-13 00:41:03.061658
Query received @ t2019-02-13 19:37:03.654348
Query received @ t2019-02-14 08:26:51.287677
Query received @ t2019-02-14 22:54:17.994967
Query received @ t2019-02-15 19:51:35.120133
Query received @ t2019-02-17 02:30:47.122209
Query received @ t2019-02-17 19:35:22.154753
Query received @ t2019-02-18 12:30:42.187756
Query received @ t2019-02-19 17:08:20.400814
Query received @ t2019-02-20 21:26:54.664498
Query received @ t2019-02-21 12:31:57.442982
Query received @ t2019-02-22 01:38:01.658558
Query received @ t2019-02-23 06:36:37.717586
Query received @ t2019-02-24 13:43:36.616482
Query received @ t2019-02-25 04:54:36.476134
Query received @ t2019-02-26 20:06:18.131320
Query received @ t2019-02-27 03:31:20.961622
Query received @ t2019-02-27 23:09:41.106498
Query received @ t2019-02-28 21:40:40.238896
Query received @ t2019-03-02 01:55:48.640233
Query received @ t2019-03-03 00:10:38.079285

JPMC-tech-task-1 — -bash — 81x25
InsideSherpas-MacBook-Pro:JPMC-tech-task-1 insidesherpa$ python client
client.py
InsideSherpas-MacBook-Pro:JPMC-tech-task-1 insidesherpa$ python client.py
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.87, price:115.14)
Ratio 1
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.87, price:115.14)
Ratio 1
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.51, price:115.14)
Ratio 1
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.61, price:115.14)
Ratio 1
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.61, price:115.14)
Ratio 1
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.51, price:115.14)
Ratio 1
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)

```

# Prerequisite

**Windows OS** (left side server, right side client)



The image shows two overlapping Windows Command Prompt windows. The background window is titled 'Administrator: Command Prompt - python server.py' and shows the process of cloning a repository and starting a server. The foreground window is titled 'Command Prompt - python client.py' and shows the process of running a client program that interacts with the server.

```

Administrator: Command Prompt - python server.py
C:\Users\j...>cd JPMC-tech-task-
C:\Users\j...>git clone https://github.com/insidesherpa/JPMC-tech-task-1
git
Cloning into 'JPMC-tech-task-1'...
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (25/25)
remote: Compressing objects: 100% (22/22)
Unpacking objects: 100% (54/54), done.
C:\Users\j...>cd JPMC-tech-task-1
C:\Users\j...>python server.py
HTTP server started on port 8080
Query received @ t2019-02-10 10:07:43.
Query received @ t2019-02-11 18:12:31.
Query received @ t2019-02-13 00:41:03.
Query received @ t2019-02-13 19:37:03.

Command Prompt - python client.py
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\j...>cd JPMC-tech-task-1
C:\Users\j...>\JPMC-tech-task-1>python client.py
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.87, price:115.14)
Ratio 1
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.87, price:115.14)
Ratio 1
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.87, price:115.14)
Ratio 1
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.51, price:115.14)
Ratio 1

```

# Prerequisite

Linux OS (left side server, right side client)

```
→ JPMC-tech-task-1 git:(master) python server.py 467/492
HTTP server started on port 8080
Query received @ t2019-02-10 10:07:43.237974
Query received @ t2019-02-11 18:12:31.763344
Query received @ t2019-02-13 00:41:03.061658
Query received @ t2019-02-13 19:37:03.654348
Query received @ t2019-02-14 08:26:51.287677
Query received @ t2019-02-14 22:54:17.994967
Query received @ t2019-02-15 19:51:35.120133
Query received @ t2019-02-17 02:30:47.122289
Query received @ t2019-02-17 19:35:22.154753
Query received @ t2019-02-18 12:30:42.187256

→ JPMC-tech-task-1 git:(master) python client.py 1465/1469
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.87, price:115.14)
Ratio 1
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.87, price:115.14)
Ratio 1
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.87, price:115.14)
Ratio 1
Quoted ABC at (bid:118.13, ask:116.63, price:118.13)
Quoted DEF at (bid:115.14, ask:117.51, price:115.14)
Ratio 1
```



# Objectives

- If you closely inspect the output of the client applications in the previous slides, there are two incorrect things...
  - (1) Ratio is always 1
  - (2) The price of each stock is always the same as its bid\_price.
- These are obviously wrong so your job is to fix those things...
- Don't worry we'll walk you through how to get these things done

# How to make changes to code

- You'll be making changes to the code in the some of the files within the repository you cloned or downloaded to achieve the objectives the task.
- To do this, you can use any text editor your machine has and just open the files in the repository that must be changed (the guide will instruct you in the following slides which files these will be)
- Our recommendation of editors you can use would be [VSCode](#) or [SublimeText](#) as these are the most commonly used code editors out there.



# Making changes in `client.py` (client3.py for python3)

- All the changes you have make to get the right output will be in the client.py file inside the repository (client3.py if you're in python3 REPL)
- The changes you need to make will be in the following methods of the file
  - getDataPoint
  - getRatio
  - Main
- The changes for each method will be dissected for each method on the next slide

# Making changes in `client.py` (client3.py for python3)

## getDataPoint

**getDataPoint.** In this method, you'll have to make the modifications to compute for the right stock price. This means you have to change how `price` is computed for. The formula is  $(bid\_price + ask\_price) / 2$ .

YOU DO NOT NEED TO CHANGE the return value as that is representational of the entire data point. You should end up with something like:

```
32 def getDataPoint(quote):
33     """ Produce all of the needed values to generate a datapoint """
34     """ ----- Update this function ----- """
35     stock = quote['stock']
36     bid_price = float(quote['top_bid']['price'])
37     ask_price = float(quote['top_ask']['price'])
38     price = (bid_price + ask_price)/2
39     return stock, bid_price, ask_price, price
```

# Making changes in `client.py` (client3.py for python3)

## getRatio

**getRatio.** In the original case, this method just returns 1 all the time. To correct this, you must change the return value to the ratio of stock **price\_a** to stock **price\_b**

```
41 def getRatio(price_a, price_b):
42     """ Get ratio of price_a and price_b """
43     """ ----- Update this function ----- """
44     """ Also create some unit tests for this function in client_test.py """
45     if (price_b == 0):
46         # when price_b is 0 avoid throwing ZeroDivisionError
47         return
48     return price_a/price_b
```

note: that we've also added the condition of the case where in price\_b could be zero, i.e. division by zero, in the rare chance that it might happen...

# Making changes in `client.py` (client3.py for python3)

## main

**main method.** Now that you've fixed the two other methods, it's just a matter of printing the correct values. For every iteration in the main method, you need to store the datapoints you get from getDataPoint method so that you can properly call getRatio and print the right ratio out. *(the below image is for python 2.7.x version)*

```

50 # Main
51 if __name__ == "__main__":
52
53     # Query the price once every N seconds.
54     for _ in xrange(N):
55         quotes = json.loads(urllib2.urlopen(QUERY.format(random.random())).read())
56
57         """ ----- Update to get the ratio ----- """
58         prices = {}
59         for quote in quotes:
60             stock, bid_price, ask_price, price = getDataPoint(quote)
61             prices[stock] = price
62             print "Quoted %s at (bid:%s, ask:%s, price:%s)" % (stock, bid_price, ask_price, price)
63
64         print "Ratio %s" % getRatio(prices['ABC'], prices['DEF'])

```

# Making changes in `client.py` (client3.py for python3) main

main method. *(the image here is now for python 3.x version)*

```

50 # Main
51 if __name__ == "__main__":
52
53     # Query the price once every N seconds.
54     for _ in range(N):
55         quotes = json.loads(urllib.request.urlopen(QUERY.format(random.random())).read())
56
57         """ ----- Update to get the ratio ----- """
58         prices = {}
59         for quote in quotes:
60             stock, bid_price, ask_price, price = getDataPoint(quote)
61             prices[stock] = price
62             print ("Quoted %s at (bid:%s, ask:%s, price:%s)" % (stock, bid_price, ask_price, price))
63
64             print ("Ratio %s" % (getRatio(prices['ABC'], prices['DEF'])))
65

```

python 2.7.x uses xrange.  
in python 3.x we use range

python 2.7.x print does not  
enclose the text it will print

python3.x encloses the text  
it prints in parenthesis

There's also a slight difference in the urllib that's used  
for python3 but you don't have to bother with this

# Making changes in `client.py` (client3.py for python3) **main**

- To review the changes in main (*whether it was in python2 or python3*), what we did was create a **prices** dictionary to store the stock prices. Think of a dictionary as a key-value store wherein you can specify a key and be able to retrieve a value. In our case, the key was the stock name and the value was the price.
- We then used this **prices** dictionary at the end to pass in the right values in the **getRatio** function.

## Next step

- You're done with the main task. You can either create your patch file now by [following this guide](#), or try and do the [bonus task](#) first