

PnL Calculator

1 Background Knowledge

Calculating profit and loss (PnL) is an important part of post-trade analysis.

The profit (or loss) that is made when you buy some quantity of shares q at price p_B and sell the same quantity of shares for price p_S is $q(p_S - p_B)$.

Things get a little more complicated when we consider the fact that you may not necessarily buy and sell shares in quantities that exactly offset one another. For example, consider the following situation:

1. At time 1, bought 15 shares at price \$11.00
2. At time 2, bought 15 shares at price \$12.50
3. At time 3, sold 20 shares at price \$13.00
4. At time 4, sold 10 shares at price \$12.75

In this scenario, suppose we want to calculate how much PnL was generated at time 3 with the sale of 20 shares. The sale price (p_S in the above formula) is \$13.00, but it is not obvious which value we should use for p_B , because the quantity sold does not exactly match either the quantity bought at time 1 or the quantity bought at time 2.

There are two ways to deal with this: FIFO accounting and LIFO accounting.

With FIFO accounting, we clear sold (bought) positions against bought (sold) positions in the order in which the positions were initially opened. So, in the above example, at time 3, we would clear the 20 shares sold with the 15

shares bought at time 1, and 5 of the 15 shares bought at time 2 (because time 1 occurs *before* time 2). Then at time 4, we would clear the 10 shares sold with the remaining 10 shares bought at time 2. The PnL calculations are as follows:

3 At time 3: $15 \cdot (\$13.00 - \$11.00) + 5 \cdot (\$13.00 - \$12.50) = \$32.50$ of PnL generated

4 At time 4: $10 \cdot (\$12.75 - \$12.50) = \$2.50$ of PnL generated

By contrast, with LIFO accounting, we clear sold (bought) positions against bought (sold) positions in the *reverse* of the order in which the positions were initially opened. So, in the above example, at time 3, we would clear the 20 shares sold with the 15 shares bought at time 2, and 5 of the 15 shares bought at time 1. Note that this is the *reverse* of the order in which the shares were originally bought. Then at time 4, we would clear the 10 shares sold with the remaining 10 shares bought at time 1. The PnL calculations are as follows:

3 At time 3: $15 \cdot (\$13.00 - \$12.50) + 5 \cdot (\$13.00 - \$11.00) = 17.50$ of PnL generated

4 At time 4: $10 \cdot (\$12.75 - 11.00) = \17.50 of PnL generated

2 Deliverables

2.1 C++ program

Implement a program `pnl` calculator in C++ that takes two command line arguments:

The path to an input file of trades executed throughout the day. See below for the input format.

A string, either `fifo` or `lifo`, indicating which accounting scheme to use.

The program should then run through the input file and, at every time when a trade clears against a position on the opposite side (i.e., a “buy” clears with a “sell” or vice versa), output to standard output the amount of PnL generated at that time according to the chosen accounting scheme. See below for details on the output format.

2.2 Complete Git Repository

Submitting a single `cpp` source file will fail this assessment.

Please create a complete git repository for this project to be reviewed and used by other developers on a standard Linux platform, as if you’re

working in a C++ development team. Other operating systems are not allowed.

The git repository must be self-contained with a build script or instructions based on cmake. If it has external dependence, it must contain a script or instructions that others can easily follow to set up the dependent libraries or tools. Standard tools available on Linux system, e.g. c++ compiler, do not need to be included in the git repository. However if the code requires a certain version of the compiler or other tools, the requirements should be clearly stated and checked in the script.

Please build your own unit test cases if time allows. You can assume google test is available. But feel free to automate the installation of google test if that's not too time-consuming.

Please work with a git repository from the beginning, and commit your changes whenever appropriate. The git log should show logical steps to complete this project.

If you have a github account, you might upload your repository to github and share the link with us. Otherwise, please zip the entire repository and send us the zipped file. But please make sure every file in the repository is necessary for publishing and production-ready.

3 Input and Output Formats

3.1 Input Format

The input will be a CSV file where each row contains the following fields:

1. Timestamp
2. Symbol (i.e., stock ticker)

3. Whether the stock was bought or sold (indicated by the character B or S respectively)
4. Price at which the stock was bought or sold
5. Quantity of stock bought or sold

An example input describing the scenario we went through above would be as follows:

```

1  TIMESTAMP,SYMBOL,BUY_OR_SELL,PRICE,QUANTITY 101,TFS,B,11.00,15
2  102,TFS,B,12.50,15
3  103,TFS,S,13.00,20
4  104,TFS,S,12.75,10
5

```

You are guaranteed that the lines of the input are sorted in non-decreasing order by timestamp.

You are guaranteed that, for each symbol, the total number of shares sold is equal to the total number of shares bought.

3.2 Output Format

The output, which must be printed to standard output, must also be in CSV format. The header, which must be the first line of output, is:

```

1  TIMESTAMP,SYMBOL,PnL

```

Thereafter, for every time at which PnL is generated (i.e., every time at which a buy (sell) trade for a particular symbol cancels with one or more previous sell (buy) trades of the same symbol), you must output one row to standard output containing:

1. The timestamp at which the clearing trade occurred,
2. The symbol of the stock that was traded, and
3. The amount of PnL generated by the trade according to the accounting scheme specified as a command-line argument. This number must be printed to a precision of 2 digits after the decimal point.

If the program is run with the input file shown above and the fifo accounting scheme, the output should be:

1	TIMESTAMP,SYMBOL,PNL 103,TFS,32.50
2	104,TFS,2.50
3	