

CS 2213, Spring 2009
Assignment 6: Capital Gains / Loss Calculator (v3)

~~Due: 5pm, Friday, April 3, 2009~~

Due: 5pm, Monday, April 6, 2009

Goal: The purpose of this assignment is to get practice implementing Queues and Stacks with Linked Lists and Dynamic Arrays.

Objectives: be able to use a linked-list to implement a queue containing information about stocks owned, their purchase prices and tax basis, ordered by their date of purchase; be able to implement a dynamic array list of linked-lists queues.

Capital Gains Tracker

In this assignment, you will write a program that can track the capital gains and losses based on information about the purchases and sales of lots of shares of stocks. This is information similar to what would be needed to complete Schedule D of Form 1040 for the IRS using FIFO accounting.

Purchase and Ownership Information For purposes of accounting, multiple shares of stock are usually aggregated into lots. A lot of shares is a set of identical shares purchased at the same time for the same price. Since each of these shares are identical, it is more efficient to have a structure that records information about a lot of shares instead of individual records for each share. For each lot it is necessary, to keep track of the identity of the security (i.e., the stock symbol for the company), the purchase date, the purchase price (for the entire lot), and the number of shares purchased for that price.

For example, this is the information contained in the English sentence, "I purchased 10,000 shares of HPQ for \$29,150 on 3/19/09."

In addition, since it is possible to sell only some of the shares in a lot, it will be necessary to update the data to keep track of the number of shares remaining unsold in the lot and the remaining 'basis' of the lot. (The 'basis' is an IRS term for the amount of value you have in an asset. In our case, this is the purchase price, adjusted for the amount of 'basis' used up when shares are sold.)

For example, this is the information contained in the English sentence, "I have 7,560 shares of HPQ with a basis of \$22,037." Presumably, this is after having sold 2,440 shares of HPQ with a basis of \$7,113 ($\frac{2440}{10000}29150 = 7112.6$, rounded to the nearest dollar, this is \$7113). That is, the basis of a sale is pro-rated by the proportion of shares (out of those currently owned) involved in the transaction, and the basis of the remaining shares is adjusted by subtracting the amounts involved in the sale.

Sales Information The basis is independent the sales price. In fact the profit is calculated as the difference between the sales price and the basis. In the above example, if those 2,440 shares had, for example, been sold for \$5,000, there would have been a \$2,113 capital loss on the sale. If they had been sold for \$8,000, in which case there would have been a \$887 capital gain.

Schedule D of the IRS's Form 1040 requires one to list information about sales of securities and calculate the capital gain or lost on the items sold. For each sale, it is necessary to identify what lots (or portions of lots) were sold, their purchase date and basis. This is used with the sales price to calculate the capital gain (loss) on the sale. If the number is positive it is a capital gain; if it is negative, it is a capital loss. The capital gains and losses for all the transactions in a given tax year are added together to come up with a net capital gain or loss. If, for a given year, there is a net capital gain for all transactions, that gain will be taxed. If there is a net loss, it can, in some circumstances, be deducted from ones taxable income.

In this assignment, you will be reading in purchase and sales information from a file, and the printing out, information for each sale similar to what would need to be recorded on a Schedule D.¹

Input/Output Specification

Input File The program should input stock purchases and sales information from a file specified as the first command-line argument. The file will contain a sequence of lines, each of which describes and purchase or sale of some number of stocks at a particular price. These lines will be formatted as in the following example:

```
01/21/2005 buy 5000 KO $11123
03/19/2005 buy 10000 LUV $29150
03/11/2007 sell 2440 LUV $8000
09/01/2007 sell 2000 KO $2000
04/01/2008 sell 5000 LUV $15221
07/01/2008 buy 440 LUV $1000
01/01/2009 sell 3000 LUV $7233
```

This example means that 5,000 shares of KO were purchased on 1/21/07 for \$11,123. Then, 10,000 shares of LUV common stock were purchased for \$29,150 on 3/19/05. (LUV and KO are stock symbols for two different companies.) Then, 2,440 shares of LUV were sold for \$8,000 on 3/11/07. Then 2000 shares of KO were sold for \$2,000 on 9/1/2007. Then 5,000 shares of LUV were sold for \$15,221 on 4/1/08. Then 440 shares of LUV were bought for \$1,000 on 7/1/2008. Then 3000 shares of LUV sold for \$7,233 on 1/1/2009.

Correct input will be in chronological order. You may assume that the stock ticker code for a company will be no more than 5 characters. In addition, you may assume that the number of shares in a lot will be less than 1 million and the price for a lot will be less than 100 million. But your program should not crash even if the input is not correct due to its ordering, format, selling more shares than were purches, etc.

1. Note, that for this assignment, you do not need to deal with complications not mentioned in this document, such as the distinction between short and long term capital gains.

Output For each year, starting with the earliest year (i.e., chronological order), the program should list information about the sales of that year (in chronological order by sales date) under the heading for the year (which should be the string "YEAR", a space, and then the four digit year).

Under each year's heading, sales information should be listed in chronological order by date of the sale. For each sale, the program should print out information about the sale. It should print out one line per lot involved in the sale. (Note: if a sale involves shares from multiple lots, the sales price will need to be allocated to each of the lots according to the number of shares in that lot. If the sale doesn't completely consume a lot, the basis of that lot will need to be allocated according to the number of shares sold from that lot relative to the total number of shares in that lot.) Each line should contain:

1. a space, the number of shares, followed by a tab,
2. the stock ticker symbol, followed by a tab,
3. the purchase date (for the lot sold), followed by a tab,
4. the sales date, followed by a tab,
5. the allocated sales price, followed by a tab,
6. the total basis of the shares sold, followed by a tab, and
7. the capital gain (or loss as a negative number) on the sale.

After the sales listings for a given year, the program should also print out the total net capital gain (or loss as a negative number) for that year. This line should also be indented with a single space, followed by the text "NET GAIN: " followed by the amount of the net gain or loss.

Dollar amounts should be prefixed by a dollar sign.

For The example input above, the program should produce the output:

```
YEAR_2007
_2440 -> LUV -> 03/19/2005 -> 03/11/2007 -> $8000 -> $7113 -> $887
_2000 -> KO -> 01/21/2005 -> 09/01/2007 -> $2000 -> $4449 -> $-2449
_NET_GAIN: _$ -1562

YEAR_2008
_5000 -> LUV -> 03/19/2005 -> 04/01/2008 -> $15221 -> $14575 -> $646
_NET_GAIN: _$646

YEAR_2009
_2560 -> LUV -> 03/19/2005 -> 01/01/2009 -> $6172 -> $7462 -> $-1290
_440 -> LUV -> 07/01/2008 -> 01/01/2009 -> $1061 -> $1000 -> $61
_NET_GAIN: _$ -1229
```

Since the IRS, allows items to be rounded to the nearest dollar, you we will only track and print out integer numbers of dollars. (But you may need to use doubles and the round() for calculations to get the rounding correct.)

Implementation Constraints

Storing Purchases and Current ownership. You should have a structure that contains information about lots of shares of stocks currently owned.

This structure should contain fields for:

1. date of purchase (a structure)
2. number of shares purchased as part of this lot (an int)
3. stock ticker for the company (a string)
4. total purchase price of all of the shares in the lot (an int)
5. number of shares still owned from this lot (an int)
6. current basis²

For each company, there should be a queue (implemented using a linked-list) holding information about lots of shares of stocks that have not yet been sold. A pointer to each of these queues should be an element in the master dynamic array list holding all of the queues for all of the companies. (So each element of the dynamic array list will be structure that has a field for the stock symbol name and another field for the queue.) As purchases are processed, they should be placed at the end of the appropriate queue.

Processing Sales. When a sale is processed, it will be necessary to check if it introduces a new year, and if so, print out the summary of the previous year and the heading for the new year. (Except for the first sale, which should cause the program to just print out a heading for that year.)

As sales are processed, ownership information about purchases of the corresponding number of shares should be read off of the appropriate queue. (If a sale cannot be satisfied from the first lot, the sale will need to be split across two lots. If a partial lot is sold, the number of shares owned and the basis for the lot will need to be updated, and this modified lot will need to remain at the head of the queue. Otherwise the head lot can be removed from the queue and free'd.) The information from the purchase lot structure and from the sales entry can be used to calculate the profit/loss for the sale, which can then be printed out.

Files Split your program across a main C file named `capgains.c`, a header file named `capgains.h` and an auxiliary C file named `lotqueue.c`. Provide a Makefile that produces an executable named: `capgains`. It is recommended that you place separately executable test cases into C files beginning with `test-`.

Extra Credit: Wash Sales.

Under IRS rules, if you sell a security (e.g., a share of stock) at a loss and also purchase a 'substantially identical' security within 30 days before or after the purchase. (For our purposes, 'substantially identical' means that the sale and purchase have the same stock ticker. The actual definition is more complicated and subject to interpretation.) The loss cannot be claimed for that year, but instead can be added to the basis of the new purchases. The wash sale rule applies on a per-share basis, so the purchase of one share can only wash the loss from the sale of one share. Matching with purchased shares for a sale is done FIFO within the 61-day window extending from 30 days before the sale to 30 days after the sale. In some cases, it may be necessary to split up lots, since the increase basis may only apply to the subset of the shares in a lot.

2. For our purposes, this is the purchase price minus the basis used when any shares were previously sold from this lot. But the IRS has some more complicated rules that apply in certain circumstances. See for example the 'extra credit' on 'Wash Sales.'

For extra credit, implement support for the wash sale rule. In the output continue to write out the full amount of all sales that have net losses as a negative number, but if there is a wash sale, on the line below that, put the text "WASH SALE" and as a positive number, the amount of the loss that should be disallowed due to the wash sale. (This will be the same amount that is added to the basis of other shares.)

Describe the data structures and algorithms you used to solve this problem in the EXTRA CREDIT section of the README.

Submission

Please create a README file divided into three or four sections: NOTES TO GRADER, EXTRA CREDIT (optional), PROGRAM VALIDATION, and CERTIFICATION.

1. Under NOTES TO GRADER, add any notes, you wish the TA/instructor to consider while grading.
Please document any known problems with your program. If gcc produces any warning or error messages (with the switches: `-std=c99 -pedantic -Wall -Wextra -Wstrict-prototypes -Wno-unused-parameter`), or if valgrind reports any problem, please list them, describe what you think they mean, and say whether you think it indicates a problem in your code or if gcc/valgrind is just being 'too picky.'
Please, also note approximately how many hours you spent to complete the assignment. This information will not effect your grade, but will be used for future course planning.
2. If you implemented the extra support for tracking Wash Sales, please add a section entitled EXTRA CREDIT. In this section describe what algorithm and data structures you used to implement the IRS's wash sales rules.
3. Under PROGRAM VALIDATION, please describe what you did to test your program or otherwise check that it is behaving correctly. In particular: What inputs did you try? How did you determine if the outputs were correct? Did you try any invalid inputs and make sure that your program doesn't crash? Did you use valgrind when running these tests? (See also the hints of Testing and Debugging given in the instructions to Assignment 5.)
4. Under CERTIFICATION, please answer the following two questions in your README file:
 - (a) Did you use any code/materials other than those provided by or referenced by the instructor or TA in completing this assignment? If so, please cite the source (including its URL if it is online) and the nature and extent of your use of that material.
If such material is prohibited by the directions for this assignment, points may be deducted, but as long as such use is completely disclosed in the README file (including source, nature, and extent of use), it will not be considered to constitute scholastic dishonesty.
 - (b) Did you work with anyone or get help in completing this assignment from anybody other than the instructor or TA? (This includes mentors and tutors.) If so, please describe the nature and extent of the help received.
If such help or collaboration exceeds what is allowed for this assignment according to these directions, points may be deducted, but as long as the help or collaboration is completely disclosed in the README file, it will not be considered to constitute scholastic dishonesty.

Below your answer to those two questions, include the following statement:

I certify that except as noted above, the work I submit for this assignment has been completed solely by me without any outside help and without looking at any code for solving this problem other than what is contained in this document or included in other material/links provided by the instructor or TA.

Below this statement, please write your name and date.

Use svn to commit your final submission by 5pm on the due date.

1. Make sure that you have all of the required files and they've all been added to svn. The the `assgn6` directory in the subversion repository is expected to include: `assignment6.pdf`, `README`, `Makefile`, `capgains.c`, `capgains.h`, and `lotqueue.c`.
2. You can continue to make changes, and re-commit as often as you want. The last version committed before 5pm on the due date will be graded.

Permissible Collaboration and Resources. This assignment is to be completed *individually*. You are allowed to discuss general strategies for solving assignments with fellow students or other individuals, but it is no longer a 'general strategy' if the discussion gets to the level of detail of what would be done in actual lines of code found in your programs. In addition, discussions of 'general strategy' should not be taking place while either party is editing their source code. (It is, however, perfectly acceptable to discuss C language constructs and concrete examples of them which are not taken directly from anyone's solution to an assignment.) In addition, you may seek more specific help from mentors and lab tutors in trying to understand why their code doesn't work. Furthermore, you may also seek help and guidance of any kind from the TA and instructor.

It is not permitted to view or copy other code solving this problem. The code supplied in lecture, in the extra notes supplied by the instructor, and in the text book can be consulted and adapted without limitation.