

## Savings Dividends - Making Decisions, Simple I/O and Calculations

The next step in learning to program in C++ is to be able to handle making decisions. In this project, you will write a modification of the third project that will read simple stock data from an input file, apply some simple rules to determine the correct values of the output data, and print a summary to a file.

### The input file:

The input file for this program is named "StockFile.txt". A sample input file is given below. This is the same as the input file in project 3 except for 3 things:

1. The addition of the Months Held
2. Input line may be longer than 255 characters.
3. A tab instead of a vertical bar separates the labels and actual data.

Current Dividend data for:	
Company	CSC
Purchase Price	10.00
Selling Price	30.00
# Of Shares	100
Brokerage Fee	25.00
Dividend	0.10
Tax Rate	0.20
Months Held	4

### What must be calculated:

Same as project 3 but you have additional things to consider about the tax.

1. In this exercise we may generate data where the stock fell in value. If the stock went down in value and no money was earned from the sale then do not charge any tax. This is before subtracting for the fee.
2. If the number of months held is less or equal to 3 months, multiply the tax rate by 2.
3. If the number of months held is more than 3 months but less than or equal to 6 months, the tax rate is multiplied by 1.5.
4. If the number of months held is more than 6 months the tax rate stays the same.

Therefore the value for the tax will be different than project 3 and you will need to use conditional logic in your program.

### The output file:

The output file is named "Stockdata.txt". Same as project 3 but notice the value for the tax is different.

Programmer: David Tucker							
CS 1044 Project 4 Spring 2001							
Company Name	Number of shares	Purchase Price	Selling Price	Fee	Dividend	Tax	Gain/Loss
-----							
CSC	100	\$10.00	\$30.00	\$25.00	\$10.00	\$600.00	\$1960.00

If you have read the description of how the Curator scores your program in the *Student Guide*, you know that is important that you use the same spelling and capitalization for all the labels shown above. The horizontal spacing does not effect scoring unless you combine things that should be separate or separate things that should be combined.

## Documentation and other requirements:

You must meet the following requirements (in addition to designing and implementing a program that merely produces correct output):

- write a header comment with your identification information, the required pledge statement (below), and a brief description of what the program does.
- write a comment explaining the purpose of every variable and named constant you use.
- write comments describing what most of the statements in your program do.
- use descriptive identifiers for variables and for constants.
- use named constants instead of “magic numbers” whenever it is appropriate.

## Submitting your program:

You will submit this assignment to the Curator System (read the *Student Guide*), and it will be graded automatically. Instructions for submitting, and a description of how the grading is done, are contained in the *Student Guide*.

You will be allowed up to five submissions for this assignment. Use them wisely. Test your program thoroughly before submitting it. Make sure that your program produces correct results for every sample input file posted on the course website. If you do not get a perfect score, analyze the problem carefully and test your fix with the input file returned as part of the Curator e-mail message, before submitting again. The highest, earliest score you achieve will be counted.

The *Student Guide* can be found at: <http://ei.cs.vt.edu/~eags/Curator.html>

The submission client can be found at: <http://spasm.cs.vt.edu:8080/curator/>

## Evaluation:

Your submitted program will be assigned a score based upon the runtime testing performed by the Curator System. We will be evaluating your submission of this program for documentation style.

## Pledge:

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the header comment for your program:

```
// On my honor:  
//  
// - I have not discussed the C++ language code in my program with  
// anyone other than my instructor or the teaching assistants  
// assigned to this course.  
//  
// - I have not used C++ language code obtained from another student,  
// or any other unauthorized source, either modified or unmodified.  
//  
// - If any C++ language code or documentation used in my program  
// was obtained from another source, such as a text book or course  
// notes, that has been clearly noted with a proper citation in  
// the comments of my program.  
//  
// - I have not designed this program in such a way as to defeat or  
// interfere with the normal operation of the Curator System.  
//  
// <Student Name>
```

**Failure to include this pledge in a submission is a violation of the Honor Code.**