

The University of Alabama in Huntsville  
Electrical and Computer Engineering  
Project 4 (20 points)

**Submit Your Solution Using ANGEL by Noon, Friday February 1, 2013**

(A late submission dropbox will be available on 2/1/13 from Noon to 2pm)

**<Project 4 Directions>**

***On Project 4, you may only use concepts presented in Chapters 1 - 4 of your textbook!!***

Using your favorite text editor, type your solution and save it as a file named **Project\_04.cpp** within your **CPE112\_SPR13/Project\_04** directory. If there are syntax errors, correct them and compile again. Once your program successfully compiles, run it and verify that the output for your modified program matches the output from the solution executable – **Project\_04\_solution**.

It is recommended that you run the sample solution **Project\_04\_solution** by typing the following at a command prompt in a terminal window:

**/home/work/cpe112/Executables/Project\_04/Project\_04\_solution**

**Sample inputs #1:** Apple 100 600 500

**Sample inputs #2:** Pepsi 200 50 100

These values are contained in P4\_in1.txt and P4\_in2.txt which can be used for input redirection. P4\_in1.txt and P4\_in2.txt are included in the input zip file named P4\_in.zip which can be downloaded from ANGEL

**Running the sample solution will provide you examples on what information has to be entered and in what order it is to be entered.**

You can run the compare solution script for this project by typing the following at a command prompt in a terminal window that is in the directory containing your source code Project\_04.cpp.

**/home/work/cpe112data/Project\_04/CompareSolution.bash Project\_04.cpp**

Once you are satisfied with your solution, submit **Project\_04.cpp** via ANGEL.

**NOTE: make sure that you do not change the order in which the information is entered. An automatic script is used to process all lab submissions, and if the order of the input information is modified, the script will not work properly with your program.**

**Remember:**

- Submissions by e-mail are unacceptable and will receive **no credit (0)**.
- Submissions, which do not compile are unacceptable and will receive no credit (0).
- **Make sure that your program will compile on the Suns before submitting it.**

## <Allowed C++ Techniques for Solving Project 4>

You cannot use any C++ techniques that are covered in Chapters 5 and higher. Material from Chapters 1 through 4 and any extra code in this handout is allowed, only. Even though functions are discussed in chapters 2 and 3, you cannot write them until they are covered in Chapter 8. Loops are not necessary to solve this project.

## <Project 4 Description>

In this program mathematical calculations involving the purchase and sale of stock are to be performed. The user will be prompted for the name of the stock, the number of shares of the stock and the buy price of the stock. Six months after buying the stock, the stock is sold for a sell price that is entered by the user. From this information several calculations are made and information regarding the purchase and sale of the stock is output.

In both cases of selling and buying the stock, the broker is paid a commission. For buying the stock, the commission is 2% and for selling the stock the commission is 1.5%. Furthermore the profit (or loss) of the stock is subject to a 25% tax rate. In the case of a loss on the sale of stock the calculated amount will be negative and will be a tax credit of that amount.

The input order of information, the calculations to be performed and the order of output information are listed below (this information is also observed by running the sample solution).

### **Input order of information**

- Name of the stock – read into a string variable
- Number of shares of stock – read into an integer variable
- Purchase price of the stock – read into a floating point variable
- Sell price of the stock – read into a floating point variable

### **Calculations to be performed (all variables holding these values should be floats):**

- The broker commission for purchasing the stock (2% of the cost of the stock)
- The total cost of the purchase: cost of the stock **plus** the commission
- The broker commission for selling the stock (1.5% of the sell price)
- The total return from the sale of the stock: return from the sale of the stock **minus** the commission paid
- The net profit (or loss) from the stock purchase (total return **minus** the total cost of the purchase). If this value is negative then it is a loss.
- Tax on the profit (or tax credit on the loss) which is 25% of the net profit or loss

### **Output order of information:**

- The number of shares and name of the stock
- The commission for purchase of the stock
- The total cost of the purchase
- The commission for the selling of the stock
- The total return from the sale
- The net profit (or loss)
- The tax on the net profit (or tax credit on the net loss)

The format and information required is illustrated in the sample solution. **All floating point values are to be written with two digits of precision.**

### <Project 4 Short Detail of program requirements>

- (1) **Use variables of the data types specified in the description.**
- (2) Obtain the necessary stock information as specified above.
  - Prompt for a value, read the value and **echo print\*** the value read
  - Run the sample solution to see the order of input and the prompt phrases
- (3) Perform the calculations for the values specified above.
- (4) Output the information to the terminal as illustrated in the sample solution.
- (5) **All information written to the terminal by your program is to match the output created by the sample solution.**
- (6) **Precision of floating point numbers is to be two decimal places.**

#### **\*Echo Print Example:**

```
cout << "Enter in an integer: "; // prompt for a value
cin >> number;                  // read in the value
cout << number << endl;        // echo print the value entered
```

The use of echo printing has an effect on what is seen on the terminal. When running the program using the comparison script or redirected input, the values used will be seen one time. When running the executable from the command line, the number will appear twice – once from the keyboard entry and once from the echo print.

### < Hints for This Program >

- The stock calculations header has two rows of 25 asterisks – these are written using the C++ statement: `cout << string(25, '*') << endl;`
- Header for the output section has a row of 35 – (hyphens) under it.
- Field width for the output phrases is 20 characters and **left justified**
- For the output, the periods after the phrases and before dollar amounts are obtained by using `setw` and `setfill` as shown below (`setfill` is invoked one time only, `setw` is invoked for each phrase):

```
cout << setfill('.') << "Total Return" << "$" << ...
cout << setw(20) << "Total Return" << "$" << ...
```

### < C++ Output Manipulators Useful for This Program >

- Use header files `#include <iostream>`, `#include <iomanip>` and `#include <string>`
- **endl** – ends current line of output
- **fixed** – forces the decimal notation output
- **showpoint** – forces printing of the decimal point even for whole floating point numbers
- **setprecision(#)** (Where # represents some integer value) – sets the number of digits that are output to #. If `fixed` is also set, then for floating-point values `setprecision` sets the number of digits printed to the right of the decimal point to #.
- **setw(#)** – Sets the fieldwidth to # for the next item output
- **setfill(char)** – Sets the fill character for unused spaces in `setw` to the character specified. The character remains as the fill character until changed to another character.