CIS 2541: C++ Language Programming
Program Assignment

Name: _____

Program Assignment	#4 – Classes
Due Date *(beginning of class)*	05/09/2016
Points	**Stock Sector Portfolio** C++ Program header _____/ 5 pts. alignment _____/ 5 pts. file input _____/ 5 pts. class definition _____/ 20 pts. functions _____/ 20 pts. accuracy _____/ 15 pts. file output _____/ 5 pts. output formatting _____/ 15 pts. demo _____/ 10 pts. **Total**** _____**/ 100 pts.** ***70 point maximum for code that cannot compile or execute.*

On your own, create a modular C++ program to calculate the gain or loss status of a Stock portfolio. The data maintained for a stock object consists of the following (*validation not required for this project, assume input data in valid ranges*):

Trading Symbol	C++ string object containing the abbreviation for the trading symbol (does not contain spaces)
Company Name	C++ string object containing the full name of the company (may contain spaces)
Sector	enumerated type indicating sector to which this stock belongs (enumeration value in parentheses): • Technology (10) • Health (20) • Financial (30) • Consumer Goods (40) • Utilities (50)
Number of Shares	integral number (greater than 0 and less than or equal to 10,000 shares)
Purchase Price	floating point number representing the purchase price of single stock (greater than 0 and less than or equal to 100.00)
Current Price	floating point number representing the current price of single stock (greater than or equal to 0 and less than or equal to 100.00)

Define an ADT **Stock class** data type to contain all the **private** data for a single stock (*only* the data listed above). Include **public constructors**, **set**, and **get** methods for the private data in your definition. Only the **set** and **get** methods should be declared **inline**. Also include the following **public methods** in your definition (no parameters are passed to any of the following methods):

- **`CurrValue`** Returns the current value of the stock (Number of Shares * Current Price).
- **`CurrStatus`** Returns enumerated type of **GAIN**, **LOSS**, or **BREAKEVEN** depending upon whether stock has GAIN (current price is greater than purchase price plus 5 cents), LOSS (current price is less than purchase price minus 5 cents), or BREAKEVEN (current price is within plus or minus 5 cents of purchase price).
- **`GainAmt`** If **Status** method returns **GAIN**, calculate and return amount of gain, otherwise return 0.0. The gain of a stock is calculated as number of shares times the difference between the current and purchase price.
- **`TaxGainAmt`** If **GainAmt** method returns non-negative number, calculate and return calculated tax on gain amount using long term capital gain tax rate, otherwise return 0.0. Use a **Stock** class **public static named constant** for the long term capital gain tax rate which is **15%.**

Create an array of these stock class objects to store the data for all stocks within a portfolio. The maximum size of this portfolio array is **25** and a **named constant** should be used in its definition.

Prompt the user for an input data file name. This data file contains the portfolio's stock data where each line has one stock's data (there can be any number of lines in the file—only read until the end of file is reached or the maximum number of array elements are read, **whichever comes first**):

Field	Data Type	Digits After Decimal	Delimiter
Trading Symbol	Character		space
Company Name	Character		#
Sector	Numerical	0	space
Number of Shares	Numerical	0	space
Purchase Price	Numerical	1-2	space
Current Price	Numerical	1-2	newline

Store each line of data into an element of the stock objects array. After all data has been read from the file, **sort the array of stock objects according to sector**. Create an output report file that lists the detail and summary statistics for each sector and stock in the following format:

```
Prepared Portfolio Analysis Summarized by Sector
Long Term Capital Gain Tax Rate = 0.15

Symbol    Company                                      Status
======    =======                                      ======
CA        CA Inc                                    BREAKEVEN
SYMC      Symantec Corp                                  LOSS
TWTR      Twitter Inc                                    GAIN

               Summary for sector:     TECHNOLOGY
                      Gain amount:          29.40
                      Tax on gain:           4.41

Symbol    Company                                      Status
======    =======                                      ======
JNJ       Johnson & Johnson                              GAIN
LLY       Eli Lilly and Company                          GAIN
RPTP      Raptor Pharmaceutical Corp                BREAKEVEN

               Summary for sector:         HEALTH
                      Gain amount:          23.81
                      Tax on gain:           3.57

Symbol    Company                                      Status
======    =======                                      ======
BAC       Bank of America Corp                           LOSS
WFC       Wells Fargo & Co                          BREAKEVEN
V         Visa                                           LOSS

               Summary for sector:      FINANCIAL
                      Gain amount:           0.00
                      Tax on gain:           0.00
```

Your program should present a modular approach -- no global variables; use parameters and return values for transferring data between modules, call class public methods for getting/setting/calculating data for a stock object. Your program should create and properly use (pass parameters, check return values, etc.) **at least** the following functions (prototypes and descriptions are given):

- **`void main();`**
  This controlling function creates the needed variable parameters, calls the function to read the stock data from file, the function to sort the read stock data, and then the function to output the stock portfolio report to file.

- **`int ReadStkData(Stock stks[], int maxSize);`**
  This function prompts the user for a file name, opens the file, reads data from the file (until end of file or maxSize has been reached, whichever comes first) into the passed array of stock class objects, closes the file, and returns the number of elements read from the file.
- **`void SortStks(Stock stks[], int size);`**
  This function sorts the array of passed Stock objects according to increasing sector value.
- **`void CalculatePortfolio(Stock stks[], int size);`**
  This function prompts the user for a file name, opens the file, writes out the stock portfolio report to file, closes the file, and displays an output message to the user indicating the report has been created.

a) Add a **block comment** at the top of the file to identify your name, file, date, class, assignment, and short description of the program.

b) Use proper alignment and spacing in your source code. Use textbook examples as a guide.

c) Compile the source code until no syntax errors are found. Run the program with a given data file and manually verify the expected and actual results. Debug as appropriate to ensure accuracy.

d) Attach a copy of the source code file(s) and generated output file to this sheet for full possible points as indicated above Upload a copy of ***only*** your source code file in Blackboard under the appropriate program assignment. Multiple source code files must be combined into a single .zip file before upload.

e) Be prepared to give a demo of your code and describe its functionality at the ***beginning*** of class on the listed due date. Note that the demo data file will be *different*, but with the *same layout*, as the one provided with the program assignment.