CIS 246 – C++ Programming

Program:	**Week 12**
Points:	**20**
Chapter(s):	**12**
File(s) to Submit:	**Company.h, Company.cpp, Technology.h, Technology.cpp, Manufacturer.h, Manufacturer.cpp and Portfolio.cpp (within one zip)**

Summary

Write a C++ application that demonstrates polymorphism using information about companies.

Note – This assignment builds on the program from Week 11. Finish that one first!

Description

You will build a portfolio containing the stock prices of 20 companies. The portfolio will be built by flipping a coin and adding a Technology company or a Manufacturing company to the portfolio based on the outcome of the coin flip.

Your program will then update the stocks in the portfolio using polymorphism and create an updated report.

Your program will use three classes to demonstrate inheritance: Company, Technology, Manufacturer and an implementation file as the driver program: Portfolio.cpp. The details of these classes are outlined below.

Learning Objectives

- Using polymorphism to manipulate objects of derived classes.
- Function overriding

You must write at least the following three classes and driver program!!

Requirements for the Company Class

Data Members – no new data members. Keep the existing data members from program 11.

Member Functions – Keep all existing member functions. Add an **update()** function as described below.

Update function – Implement a **virtual function** called update(), that has no return value and no parameters. It has no functionality in this class. Note that you still must write a function definition in the implementation file for this class (in addition to a prototype in the header file). The definition will use empty curly braces.

Requirements for the Technology Class

The Technology class is derived from the Company class.

Keep all existing member functions. Add an update() function as described below.

Update function – Override the virtual function update() in the base class. This implementation subtracts 1.01 from the current value of the object's stock price.

Requirements for the Manufacturer Class

The Manufacturer class is derived from the Company class.

Keep all existing member functions. Add an update() function as described below.

Update function – Override the virtual function update() in the base class. This implementation adds 0.75 to the current value of the object's stock price.

Requirements for the Portfolio Program

A **main** function is required. You may include other functions if you prefer. Main builds the portfolio of companies and generates a summary report to the screen.

Create an array of pointers to the Company class (size 20). You can use the built-in array style, the <array> class, or a <vector>.

You must use two separate loops, as follows, to access the array to create and update the company information:

1. **Counter-controlled Loop 1:**
    a) Create the appropriate object (existing program 11 functionality) and assign it to a pointer in the array.
    b) Print each company's info (existing program 11 functionality).
2. **Counter-controlled Loop 2:**
    a) Access each element of the array and update the stock price by calling its update function.
    b) Print each company's updated info (existing program 11 functionality).

Sample Run

Remember you are generating these Companies randomly, so it is unlikely that your output will duplicate this output exactly. Your output should all line up in neat columns.

Original Portfolio

Manufacturer	$1.00
Technology	$1.11
Manufacturer	$1.22
Manufacturer	$1.33
Technology	$1.44
Technology	$1.55
Technology	$1.66
Manufacturer	$1.77
Manufacturer	$1.88
Manufacturer	$1.99
Manufacturer	$2.10
Manufacturer	$2.21
Technology	$2.32
Manufacturer	$2.43
Technology	$2.54
Technology	$2.65
Manufacturer	$2.76
Manufacturer	$2.87
Manufacturer	$2.98
Manufacturer	$3.09

Updated Portfolio

Manufacturer	$1.75
Technology	$0.10
Manufacturer	$1.97
Manufacturer	$2.08
Technology	$0.43
Technology	$0.54
Technology	$0.65
Manufacturer	$2.52
Manufacturer	$2.63
Manufacturer	$2.74
Manufacturer	$2.85
Manufacturer	$2.96
Technology	$1.31
Manufacturer	$3.18
Technology	$1.53
Technology	$1.64
Manufacturer	$3.51
Manufacturer	$3.62
Manufacturer	$3.73
Manufacturer	$3.84

Requirements for Full Credit on This Project

COMPLETE AND ACCURATE – Your program must compile, execute, and give accurate output.

FOLLOW ALL REQUIREMENTS ACCORDING TO THE INSTRUCTIONS – Follow the instructions as written for completing this project, even if you [think you] know a "better" way to do something.

COMMENTS – Include comments in your code. There must be a comment at the top of each source code or header file that includes your name, the assignment number, and a description of the code in that file. There must be comments at each important step in your algorithm that describes that step.

BEST PRACTICES – Follow best practices in C++ programming, including, but not limited to, appropriate use of private/public, appropriate use of classes and/or header files, sets & gets, white space, alignment, meaningful variable names, naming conventions, using statements, etc. Points will be deducted for sloppy code that is hard to read, even if it works, so pay attention to these details.

SUBMIT ALL FILES BEFORE THE DUE DATE – See the due date for this assignment on the course calendar in Canvas and review the submission requirements below.

ALL SUBMISSIONS MUST BE YOUR OWN WORK – Review the syllabus regarding plagiarism and the Joliet Junior College Academic Honor Code. Review the Get Help page in every module if you need help with this assignment.

Submission Requirements

All source code file submissions must be compressed into ONE Compressed Folder, or zip file. The dropbox for this assignment will only allow compressed files (.zip extension) to be uploaded.

Instructions for creating a Compressed Folder (zip file) are available here:
https://support.microsoft.com/en-us/help/14200/windows-compress-uncompress-zip-files

Do not submit your code in any other file format but .cpp and .h (within a zip). I will not accept links to online storage. You must submit the actual file. Do not submit executable files. Do not submit project files from an IDE.