Project 3: Using Arrays, For Loops, and Conditionals

Please be sure to include substantial comments in your code. Any class which does not include non-trivial comments will lose significant points (at least 20) during grading. Please be clear about the function of your new classes and each of their methods. Grammar and spelling are also important.

Arrays can be used for keeping track of changes in data as time progresses. For this project, you will use an array to keep track of a company's stock valuation over a ten day period. Over this period, you will also need to note trends in the company's performance and whether the company goes bankrupt (hopefully not!). To accomplish these tasks you will need to use multiple methods, loops, and conditional statements.

Rather than monitoring a real company's stock valuation over a ten day period, you will record the activity of a fictional company which changes every time you run your program. Make a class called CompanyStock that performs the following activities.

- Initializes the stock valuation of your company to a random double between 100.0 and 300.0, inclusive of both. Makes an array that monitors the changes in the stock valuation. The array should be large enough to keep track of the stock's history: you will be monitoring the stock for 10 days, and the stock price can change up to 20 times each day. The initial number of shares of the company is a random value between 50 and 54. Note that the number of shares should remain constant.
- For each day, generates 20 different random double precision values between 0 and 0.5. These random numbers will represent changes in the stock price for that day. For each random number, use a random Boolean value to determine whether the stock price rises or falls by the given amount.
- Every time the price changes, records a new stock valuation in the array of stock valuations. Also calculates and stores the new price of each share.
- Compares the most recently recorded change in the stock's valuation to 20 elements before this change, or to the first element in the array if the 20^{th} element in the array has not yet been filled. Takes the difference of these two values. Determines if the stock is rising, falling, or steady. Considers the stock to be steady if the calculated difference is less than 10; if it is more than 10, the company stock is either rising or falling.
- Reports the stock valuation gain/loss relative to 20 elements prior to the last recorded change in the stock, or if appropriate, the beginning of the array, and the current valuation of the stock.

The activities listed above should be implemented as a series of methods. **Please include lots of comments to indicate the function of your methods.** Notice that you do not have to keep a list of the number of shares (a single variable is sufficient), nor do you have to keep a list of the people who bought or sold stocks.

After the CompanyStock class is finished, write a driver (for example a main method) for the project. Monitor the company's stock valuation for 10 days. Print out the results after each day, even if the company is bankrupt. For example, your output for each day should look like the following:

Day 1: The company stock is steady; the price changed by $3.00.; Total value $279.00
Day 2: The company stock is rising; the price changed by $19.00.; Total value $298.00
Day 3: The company stock is rising; the price changed by $50.00.; Total value $348.00
Day 4: The company stock is rising; the price changed by $85.00.; Total value $433.00
Day 5: The company stock is falling; the price changed by $205.00.; Total value $228.00
Day 6: The company stock is rising; the price changed by $75.00.; Total value $303.00
Day 7: The company stock is rising; the price changed by $25.00.; Total value $328.00
Day 8: The company stock is falling; the price changed by $59.00.; Total value $269.00
Day 9: The company stock is falling; the price changed by $19.00.; Total value $250.00
Day 10: The company stock is falling; the price changed by $50.00.; Total value $200.00

Please format the dollar amounts to 2 decimal places.

BANKRUPTCY: If the company goes bankrupt, cease all trading in the company. That is, if the stock valuation reaches zero, do not allow any further activity in the company stock: every subsequent record of the stock's valuation should be zero. The trend can continue to be determined, but now it should also be able to report that the company is bankrupt. The gain can still be recorded, and should obviously be zero on any day (after the first day) that the company goes bankrupt. The output for a bankrupt company should look like the following:

Day 1: The company stock is rising; the price changed by $45.00.; Total value $151.00
Day 2: The company stock is falling; the price changed by $103.00.; Total value $48.00
Day 3: The company stock is rising; the price changed by $57.00.; Total value $105.00
Day 4: The company stock is bankrupt; the price changed by $105.00.; Total value $0.00
Day 5: The company stock is bankrupt; the price changed by $0.00.; Total value $0.00
Day 6: The company stock is bankrupt; the price changed by $0.00.; Total value $0.00
Day 7: The company stock is bankrupt; the price changed by $0.00.; Total value $0.00
Day 8: The company stock is bankrupt; the price changed by $0.00.; Total value $0.00
Day 9: The company stock is bankrupt; the price changed by $0.00.; Total value $0.00
Day 10: The company stock is bankrupt; the price changed by $0.00.; Total value $0.00

You have some flexibility in completing this project. As long as you meet the requirements of both CompanyStock and its driver class, any coding style (e.g. methods, loops, etc) is acceptable.

Table 1 gives a sample list of methods which you might want to write to satisfy the requirements of the project :

Method Name
CompanyStock
monitorDay
determineTrend
reportResults
Prospers
slumps

Table 1 : Sample names for methods