

Programming Assignment #7

Simulate a Vending Machine

CS 1428.003 and 004, Fall 2015

Instructor: Jill Seaman

Due: in class Monday, 11/30/2015 (upload electronic copy by 10:00am)

Problem:

Write a program that simulates a soft drink machine. The program should use a structure that stores the following data:

Drink Name
Drink Cost
Number of units currently in the machine

The program should create an array of five structures. The elements should be initialized with the following data:

Drink Name	Cost	Number in Machine
Coke (can)	0.75	10
Coke (bottle)	1.25	10
Mountain Dew (can)	0.75	10
Water	1.25	10
Full Throttle	2.00	10

Each time the program runs, it should enter a loop that performs the following steps: A list of drinks is displayed on the screen. The user should be allowed to either quit the program or pick a drink. If the user selects a drink (that has not sold out), the program should display the cost of the drink and subtract one from the number of units of that drink left in the machine. If the user selects a drink that has sold out, a message should be displayed. The loop then repeats. When the user chooses to quit the program it should display the total amount of money the machine earned.

Sample output is on the class website.

Additional Requirements:

- Your program **must compile** and run, otherwise you will receive a score of 0.
- Your program must pass **Test Case 0** (below) or you will receive a score of 50 or less with no credit for the other grading categories (correctness/constraints/style):

Input: 1 2 3 4 5 6. (1 of each type of drink, then quit.)

Expected output: Amount earned: \$6.00. It should also show the menu before each selection. Your code must calculate the earnings by iterating over the array AFTER the user quits (not during the selection process) to pass Test Case 0.

- Use named constants for the number of drinks and the initial quantity.
- The drinks array must be processed using loops. (Do not use literals for subscripts: drinks[0], drinks[1], etc., use a variable instead: drinks[j]).
- Use functions as we will discuss in class (or as you see fit) (or use no functions for partial credit). Since the array is always full (not a partially filled array) you do not need to pass the size as a parameter to the functions (but you can if you want to). For full credit, you need at least three functions that have an array parameter.
- I recommend using integers for the drink choices (like a menu). However:
- Do not use a switch to process the drink choices. If the choice is any one of the drinks (1-5) you will process them using the same code.
- Validate the drink choice input from the user (1-6).
- Calculate the earnings by iterating over the array AFTER the user has chosen to quit the program (do not compute the earnings inside the main loop). [This will give you more practice with structures and arrays.]

Style:

See the Style Guidelines document on the course website. Especially pay attention to the comments required for functions. **Include the full comments required for your functions this time!** The grader will deduct points if your program violates the style guidelines.

Logistics:

Name your file **assign7_XXXXX.cpp** where XXXXX is your TX State NetID (your txstate.edu email id). The file name should look something like this: assign7_js236.cpp

There are two steps to the turn-in process:

1. Submit an electronic copy using the Assignments tool on the TRACS website for this class (tracs.txstate.edu). Submit the .cpp file, (NOT a .cbp file!).
2. Submit a printout of the source file at the beginning of class on the day the assignment is due. Please print your name on the front page, and staple if there is more than one page.

See the assignment turn-in policy on the course website (cs.txstate.edu/~js236/cs1428) for more details.