# Chapter 8 Practice - Structures – Drink Machine Simulator

Write a program that simulates a pop machine. The program should use a **structure** that stores the following data: **Drink Name**, **Drink Cost**, & **Number of Drinks in 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 |
| Cola | .75 | 20 |
| Root Beer | .75 | 20 |
| Lemon-Lime | .75 | 20 |
| Grape Soda | .80 | 20 |
| Cream Soda | .80 | 20 |

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, he or she will next enter the amount of money that is to be inserted into the drink machine.
* The program should display the amount of change that would be returned, and subtract one from the number 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.

Input Validation: When the user enters an amount of money, do not accept negative values or values greater than $1.00

## Sample Output

1) Cola 0.75

2) Root Beer 0.75

3) Lemon-Lime 0.75

4) Grape Soda 0.80

5) Cream Soda 0.80

6) Leave the drink machine

Choose one: 5

Enter an amount of money: 1.25

Enter at least 0.80 and not more than 1 dollar.

.85

Thump, thump, thump, splat!

Enjoy your beverage!

Change calculated: 0.05

Your change, 0.05 has just dropped into the Change Dispenser.

There are 19 drinks of that type left.

1) Cola 0.75

2) Root Beer 0.75

3) Lemon-Lime 0.75

4) Grape Soda 0.80

5) Cream Soda 0.80

6) Leave the drink machine

Choose one: 6

Total earnings: $0.80

# Chapter 8 Practice – classes – trivia game

Create a simple trivia game for two players. The program (**triviaGame.cpp**) will work like this:

* Starting with player 1, each player gets a turn at answering five trivia questions. (There are a total of 10 questions.) When a question is displayed, four possible answers are also displayed. Only one of the answers is correct, and if the player selects the correct answer, he or she earns a point.
* After answers have been selected for all of the questions, the program displays the number of points earned by each player and declares the player with the highest number of points the winner.

In this program, you will design a Question class (**Question.h**) to hold the data for a trivia question. The Question class should have member variables for the following data:

* A trivia question
* possible answer #1
* possible answer #2
* possible answer #3
* possible answer #4
* the number of the correct answer (1, 2, 3, or 4)

The Question class should have appropriate constructor(s), accessor, and mutator functions.

The program should create an array of 10 Question objects, one for each trivia question. Your program should read in all the questions and possible answers and correct answer selection from the **trivia.txt** text file, which you can download from ilearn.

## partial Sample Output

Question for PLAYER 1:

----------------------

(3) The constructor function return type is \_\_\_\_\_.

ANSWERS:

1. int

2. float

3. char

4. nothing. Constructors do not return data.

Enter the correct answer: 4

Correct!

Question for PLAYER 2:

----------------------

(4) A \_\_\_\_\_ is a member function that is automatically called when a class object is \_\_\_\_\_\_.

ANSWERS:

1. destructor, created

2. constructor, created

3. static function, deallocated

4. utility function, declared

Enter the correct answer: 3

Sorry, that's incorrect. The correct answer is 2. constructor, created