Exercise 3: Vending Machine Revamped

Python Programming Bootcamp by Dr Rohitash Chandra
UNSW, 2021

Introduction

The goal of this assignment is to help you practice programming with emphasis to classes and object oriented programming. Your are required to use ***object oriented programming*** techniques in your program.

Workload: 15 Hours. You can work on this assignment on your own or as a team of 2 members. You can also work on your own.

Description

You are to write a program that simulates a vending machine, i.e. the sale of items, coin counting and change giving.

A customer will deposit an initial amount of money and buy item from the vending machine by making a sequence of selections. Your program must check that the customer makes a valid selection. The price of each item will be deducted from the initial deposit. Once the customer has finished his/her selections, your vending machine calculates the denominations of coins that are to be returned to the customer as change. Assume that only $1, 50 cents, 20 cents, 10 cents and 5 cents coins are available for making change.

You may assume that the customer has deposited sufficient cash for the sum of all his selections.

There are 10 of each item at the beginning of each day. If a customer selects an item that is no longer available, then the vending machine will display an error message. Your vending machine starts when it is switched on and only stops when power is shut off.

The following selections are on offer;

1. Water	$0.75
2. Coke	$1.20
3. Diet Coke	$1.20
4. Iced Tea	$1.00
5. Swiss Chocolate	$1.50
6. Candy	$0.95
7. Chips	$1.10
8. Bubble Gum	$0.50
9. Turkish Delight	$1.20
10. End of Transaction	

Programming Requirements

1. Create a VendingMachine class and implement methods ShowMenu(), MakeSelection(), ReturnChange(), and DisplayErrorMessage() with the appropriate parameters. Appropriate Data Encapsulation techniques should be used.

2. Your program should be written in such a way that it can easily be changed to accommodate new items for sale and different coin denominations, i.e. changes to your program should be limited to the declaration parts.

3. If the current balance is not sufficient to cover a customer's selection, then the vending machine will prompt the user to deposit more money before delivering the selected item. The customer has to keep topping up the amount until the balance is sufficient to cover the price of the selected item.

4. As part of routine maintenance – when selection 99 is entered, the vending machine will expect a

secret pin code in order for maintenance to proceed; do not display this selection to the customer – all items in the vending machine are restocked. Also, the number of sales for each item, the total balance given and the net income for that day are displayed.

5. Assume that the vending machine starts every morning with a fixed number of coins of each denomination. If the vending machine has run out of a particular coin, then alternate means for giving change must be found. If no accurate change can be returned to the customer, then his/her current selection will be ignored.

6. Your machine should monitor if you have enough items. If you do not have enough items, your machine should print a message that can be either emailed, else placed to a file.

7. Assume that you have a number of vending machines around the country. Each vending machine has a unique ID with details about it address, and who is responsible to manage. This information can be implemented as another Class that is a friend of Class Vending machine. A Class called ***MachineList*** will contain a list of all the vending machines that can be search via ID or location and their information of transaction can be displayed. You should use appropriate OOP features such as Friendship and Inheritance to design the necessary classes that makes a nation wide Vending Machine System.

Grading

Your program will be graded as follows:

Grading Point	**Marks**
OOP Design, Menu and Make selection	20
Return change	10
Display error message and Cash top-up	10
Vending machine maintenance	10
Alternate change	10
Efficiency	10
Enough Items in Machine	10
Code quality (e.g., variable names, formulation of selection statements and loops, etc) OOP features	1

Resources:

1. Python Tutorial: http://www.tutorialspoint.com/python/
2. Python Examples: http://sandbox.mc.edu/~bennet/python/code/

Extra work (this is not part of this bootcamp – but can be done later)

Implement a Graphics User Interface (GUI) that creatively highlights the features of your vending machine. (20 Points)

Make your Vending Machine Transactions become updated in a Web Server that runs SQL or MySQL. Therefore, each time a vending machine transaction happens, information about the particular vending machine should be updated to the database, with location and ID. (20 Points)

ss