



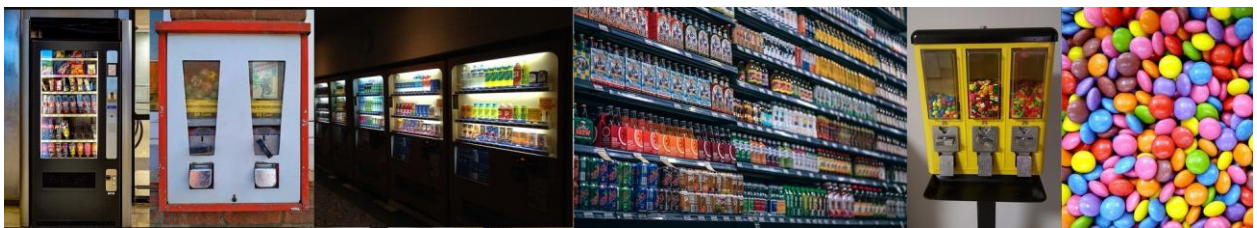
Python OOP – Object Oriented Programming for Beginners

Mini Project

Inheritance - Methods

Project Description:

Now you work for a software development company that has been hired to develop the software of the **vending machines** in your local hospitals and schools.



- Your job is to **customize the behavior of the existing generic VendingMachine class using method inheritance and overriding.**
- You need to create two subclasses: HospitalVendingMachine and SchoolVendingMachine that inherit from VendingMachine (this class is already defined in the VendingMachine.py file that you need to download for this project).

Both

- These subclasses must have a custom greeting message for the user, so you must override the `sales_menu` method in these two subclasses. You must make them print the corresponding message (shown below) **before** calling the `sales_menu` method of the `VendingMachine` superclass.

The custom messages that must be printed before calling salesMenu() in the subclasses are:

For HospitalVendingMachine:

```
print("\n===== Welcome to our Hospital Vending Machine  
===== \nWe hope you are feeling better today!\n")
```

For SchoolVendingMachine:

```
print("\n===== Welcome to our School Vending Machine  
===== \nWe hope you have a great day full of learning!")
```

Note: \n indicates that a new line will be printed.

- Add their corresponding `__init__()` to make them inherit the attributes of VendingMachine.
- Add a class attribute `snack_prices` in these two subclasses to replace the value defined in the superclass. You can customize the prices (values) to your liking but please maintain the structure of the original dictionary.
- Override the method `find_snack_price` in both subclasses to make them use the class attribute that corresponds to the subclass. (Tip: copy/paste the method from VendingMachine and make the necessary modifications).

HospitalVendingMachine

- In HospitalVendingMachine, add a method `print_days_until_maintenance` that prints a string with the number of days remaining until the machine needs maintenance (Tip: it's an instance attribute of the superclass).

SchoolVendingMachine

- In SchoolVendingMachine, add a class attribute `student_debt` (a dictionary with students' names as the keys and their corresponding debt with the vending machine as values).
- Then, add a method `print_student_debt` that takes the name of the student as an argument and prints the value of his/her debt.

Important: the `VendingMachine` class has many methods defined, but you already know how to work with every element in the file! You have worked very hard during the course and you are now capable of understanding this program. **Congratulations! You should be very proud of yourself.**

I suggest taking some time to read through the code and familiarize yourself with this class. This is intended to help you practice working with existing code and reading code written by other developers. I included descriptive comments for each one of the methods, but please do not hesitate to ask if you have any questions.

Note: You will be able to check your solution with a sample answer as soon as you submit your mini project.