Introduction to Computer Programming

Exercises - Week 5: Classes

Part 1. Classes in Python

Exercise 1 - Defining classes

The goal of this exercise is to write a program that creates a shopping list and then prints out all of the items and the total price.

1. Write a class called `Item` that has a constructor `__init__(self)` that prints "This is an item". From your main program, create an object of class Item called Apple. Run the program.
2. Change the constructor to include 3 additional arguments:
   `__init__(self, Description, Number, UnitPrice)`.
3. Change the print statement to print "Created a new item: `X`", where `X` is the item description.
4. From your main program, create an object of class Item called Apple with parameters "Apple", 1, and 0.5. Call `print(Apple.Description,Apple.Number,Apple.UnitPrice)` to print out the information.
5. Include a function in your class called `PrintItemInfo(self)` that prints all the information about the item. Call `Apple.PrintItemInfo()` from the main program.
6. Override the built-in `__str__()` function so that printing the instance of `item` prints the name of that item. Then, create a list called `ShoppingList`, add the Apple, and 2 other items to the list. Loop through the list and print out each item using the overridden `__str__()` method (hint: to do so, you can wrap each item in `str()`).
7. Write a loop in the main program to go over all items in `ShoppingList`, print out the item information and sum the total price (the price for one item is `Number*UnitPrice`). Print out the total price at the end.

Part 2. Inheritance

Exercise 2 - Deriving a class with inheritance.

1. Add a new class called `SpecialItem` which *inherits* from the `Item` class. The class signature should look like the following:
   `def __init__(self, Description, Number, UnitPrice, SpecialInfo):`
   and should call the `__init__()` function of the Item class passing in the `Description`, `Number`, and `UnitPrice` arguments, but storing the new variable `SpecialInfo` as a member variable of the new class.
2. As we did in the `Item` class, override the built in `__str__()` function to print the item description, but this time have it also print out the special information via `self.SpecialInfo`.
3. Override the `PrintItemInfo()` method of the `Item` class.
4. Add a few special items to your shopping list that require instructions via the `SpecialInfo` argument, such as `Paracetamol:  Take two tablets every 6 hours`.