Columbia College
Vancouver, Canada

Introduction to Computer Science and Programming 1
CSCI120

Chapter12: Inheritance and Polymorphism

Assignment 12-2

Note: This document has been designed and developed as part of an initiative for creating an OER (Open Education Resource) package for the course CSCI 120 at Columbia College.

Please contact Alireza.davoodi@gmail.com for any comment, modification, and questions.

Terms of use: Please feel free to customize this document as needed

Last Modified: July 2022

If it is a group assignment, please add the information here

# of Students in the Group:		
Student 1	*First name, last name*	*Student-ID*
Student 2	*First name, last name*	*Student-ID*
Student 3	*First name, last name*	*Student-ID*
Student 4	*First name, last name*	*Student-ID*

Requirements

- You need to do the Assignment 12-1 first before you work on this.
- Please use meaningful name for your classes, properties and methods
- Try to reuse your solutions as much as possible.
- You can define the classes in one file or separate files.
- Define one main function (one for each file)
- Define some instances (if needed from the classes you have defined) and try them in the main function.
- Do not use methods, functions, statements that we have not covered in the previous lectures.

Problem1

a) Create a Python class called Product.
b) Add the following instance properties to the Product class:

Properties	**Type**
ProductID	Integer
productName	String
productPrice	Float
productMadeInCountry	String

c) Define and implement a constructor (initializer) for this class.
d) Create a test class and call it TestProduct.
e) In this class define a static method and call it test.
f) Inside the test method create one object/instance called sampleProduct from the Product class with the following values:

Properties	**Type**
ProductID	110
productName	"Diet Pepsi"
productPrice	2
productMadeInCountry	USA

Problem2

a) Define another Python class for each of the following entities (Drink, Food, Cloth) using the Product class you have defined in Problem 1. (Use inheritance!)

Note: Feel free to change the name of the instance variables if needed. Similar to the Shape example we did in the class in which we had to change the names of the instance variables and called them dim1 and dim2.

Class: Drink	**Type**
drinkID	Integer
drinkName	String
drinkPrice	Float
drinkMadeInCountry	String
isDrinkDiet	Boolean
drinkSize	Integer

Class: Food	**Type**
foodID	Integer
foodName	String
foodPrice	Float
foodMadeInCountry	String
foodCalorie	Integer
foodSize	Integer
foodIngredients	Array of String

Class: Cloth	**Type**
ClothID	Integer
ClothName	String
ClothPrice	Float
ClothMadeInCountry	String
ClothMaterials	Array of Material (defined below)

Also create another Python class and call it Material using the following properties.

Class: Material	**Type**
MaterialCode	Integer
MaterialName	String

b) Define and implement one constructor (initializer) for each of the above classes (Drink, Food, Cloth, Material).
c) Define the properties of the classes based on the above tables.
d) In the TestProduct class defined already, create one object from each of these classes with some arbitrary values.

Problem3

a) Define another Python class called ShoppingCart. The shoppingCart class works like a container for your shopping. Imagine you have bought the following items:
   - You have purchased 4 drinks (3 Pepsi, 1 Ginger Zero)
   - You have purchased 4 Food items (2 Chicken, 2 Pasta)
   - You have purchased 1 cloth item.
   - The following tables showed the above purchases.
   - Hint: Each purchase is one object/instance. For instance the user has purchased 3 Pepsi as shown the table below. That means you need to create 3 objects/instances from the class Drink and similarly for the others. So do so in the test method of the TestProduct class

Drinks	**Amount: 3**
drinkID	412
drinkName	Pepsi
drinkPrice	2$ per can
drinkMadeInCountry	USA
isDrinkDiet	NO
drinkSize	150 per can

Drinks	**Amount: 1**
drinkID	183
drinkName	Ginger Zero
drinkPrice	3$ per can
drinkMadeInCountry	Canada
isDrinkDiet	Yes
drinkSize	200 per can

Food	**Amount: 2**
foodID	100
foodName	Chicken
foodPrice	8$/lb
foodMadeInCountry	Canada

foodCalorie	350
foodSize	4lb
foodIngredients	“chicken”, “oil”, “chees”

Food	**Amount: 2**
foodID	101
foodName	Pasta
foodPrice	18$ / lb
foodMadeInCountry	Canada
foodCalorie	250
foodSize	3lb
foodIngredients	“Pasta”, “meat”, “spinach”

Cloth	**Amount: 1**
ClothID	701
ClothName	T-shirt
ClothPrice	15$
ClothMadeInCountry	China
ClothMaterials (list of materials)	(“cotton”, 10), (“Nylon”, 11)

b) The ShoppingCart class has an instance variable called basket. Basket is a list of Products (Drink, Food, Cloth)
c) The ShoppingCart also has an instance variable called totalPrice which is the total price of the items in the basket.
d) For each of the above classes (Drink, Food, Cloth) define an instance method called calculatePrice. The calculatePrice for each item is implemented differently.
    a. The calculatePrice for 1 food item is foodSize*foodPricde.
    b. The calculatePrice for 1 drink item is the value of drinkPrice
    c. The calculatePrice for 1 cloth item is the value of clothPrice
e) The Cloth, Food and Drink classes should be child of the class Product. Define a method in the Product class and call it calculatePrice.
    a. ~~#This method does not do anything just return 0.~~
    b. Override, this method in the child classes (Food, Drink, Cloth) using the formula mentioned in the d) option above. (In any class needed)
f) Also, define and add the following instance methods for the ShoppingCard class:
    a. A method using which you can add your purchases items (products) (listed above) to your shopping cart.

   b. A method using which you can calculate the total amount you need to pay for your entire purchase.
   c. A method using which you can print just the name of the items you have purchases.

g) Now in your TestProduct class, create an instance (object) from class ShoppingCart with some arbitrary values.
h) Call all the instance methods (getTotalPrice and addProductItem methods) of the ShoppingCart with their appropriate input arguments.

Good Luck ☺