



## Programming Assignment 2

### Assignment Objectives

By completing this assignment you are demonstrating your ability to:

- Designing your own related set of classes in the Python programming language.
- Create new classes using class composition.
- Implement accessor and mutator methods that allow access to your class data
- Create dunder methods for your class.
- Read data from disk.

### Scenario

In the previous assignment, you worked on a Product class to store information about the products that you want to sell in your store. In this assignment, we are working on something similar. You are given a Product class that will store information about the products. You are also given a Store class that will store information about the stores. Your objective is to design a StoreDB class that is going to create a database of stores and products.

### Given

As part of this assignment, you have been given the following classes:

#### Product class

The Product class generates a unique product ID for every new product. Its constructor receives the following product information: product name (string), products sold (int), products available (int), product cost (float), and product price (float). The `__str__` method prints the product information.

## Store class

The Store class generates a unique store ID for every new store. Its constructor receives the following store information: store name (string), store city (string), store area (string). Each store maintains a dictionary of products where unique product ids act as keys and instances of product class act as values. There are also two methods to add and remove products from/to the dictionary.

## inventory.csv

This file contains the database of stores and products. There are 8 columns with labels: InventoryId, StoreName, City, Area, ProductName, QuantitySold, QuantityAvailable, Cost, and RetailPrice. The first three represent a store object and the last 5 represent a product object.

## Objective - StoreDB class

You need to write a StoreDB class that maintains the database of all stores and products. As part of this assignment, you have been given a StoreDB class with driver code. You need to write your code in the given StoreDB class. It must contain:

1. The class constructor will receive the filename and use it to create an instance of the class. It will also initialize an empty store dictionary *store\_dict* and an empty store object list *store\_obj\_list*.
2. The *read\_file* method will read the file named filename. It will store the file contents in *store\_dict*. For each row of the csv file, cells in the first column (*InventoryId*) will be added as keys. The cells from the remaining columns will be added as values in the dictionary in the form of a list. In the end, this method will print '*xx records added to the dictionary.*', where *xx* is the length of the dictionary. Example: The first record of the *inventory.csv* file is "1, National Stores, Ouro Branco, 9 Springview Point, Chocolate Bar – Smarties, 1, 11, 4873.07, 1500.39". The *read\_file* will save this record as *store\_dict['1'] = ['National Stores', 'Ouro Branco', '9 Springview Point', 'Chocolate Bar – Smarties', '1', '11', '4873.07', '1500.39']*.
  - *Hint:* You will need to use a UTF-8 file encoding to read in the file. You will need to investigate how to do that correctly.
3. The *store\_exists* method will receive a store object and check if the *store\_obj\_list* contains another store object with identical attribute values (store name, store city, and store area). It will return the index of the identical store object if found, otherwise, it will return -1.
4. The *add\_store\_products* method will read the store dictionary *store\_dict* and add the unique stores to the store object list *store\_obj\_list*. Please note that for each record (value as a list) in the dictionary, the first three attribute values (store name, store city, and store area - all strings) represent a store object. The remaining 5 attribute values represent a product object (product name - string, products sold - int, products available - int, product cost - float, and product price - float). This method will use the *store\_exists* method to check if the store object with identical values already exists in the *store\_obj\_list* or not. If the current record represents a unique store not currently in the *store\_obj\_list*, then a new store object will be added to the *store\_obj\_list*. You can assume that the product objects will always be unique, so they will always be added to their respective stores. To summarize, for each record in the dictionary:
  - (a) You add a new store object to the *store\_obj\_list* if it does not exist in the list already.

- (b) Then you add the product object to its respective store object using the `add_product` method of the `Store` class.
- (c) In the end, print '*xx records added to unique yy stores.*', where *xx* is the length of the `store_dict` and *yy* is the length of the `store_obj_list`.

A couple of hints for this method:

- *Hint:* You may have multiple records with the same store name, say Walmart. So, you also need to check the store city and area to ensure that you are adding a unique store.
  - *Hint:* To add a product object to the store object, you need to find the index of the store object that it belongs to in the `store_obj_list`.)
5. The `print_store_info` method will receive a store name and print the information of all stores that match the input store name in the required format as shown in the sample run screenshots. The method will start with printing '*Searching for a store named xx*', where *xx* is the input store name. There may be multiple stores with the same name. This method will iterate over the `store_obj_list` to search for an input store name and print the store information as '*Store\_ID:a;Store\_Name:b;Store\_City:c;Store\_Area:d*', where *a* is store id, *b* is store name, *c* is store city, and *d* is store area. In the end, the method should print '*xx records found for a store named yy*', where *xx* is the number of records found and *yy* is the input store name.
  6. The `print_product_info` method will receive the store id and print the information of all products that the respective store carries. The method will start with printing '*Printing product list for store ID xx*', where *xx* is the input store id. This method will iterate over the `store_obj_list` to search for an input store id and print the product information as '*Product ID:a; Name:b; Sold:c; Available:d*', where *a* is the product id, *b* is the product name, *c* is the products sold, and *d* is the products available. Please note that one store may have multiple products. If no store matches the input store id, then print '*Store not found!*'. If the store is found but there is no product for that store, then print '*No Product found!*'.  
*Hint:* You can use `get_product()` method using the store objects.
  7. The `remove_store_data` method will receive the store id and remove the store if it exists in the `store_obj_list`. If the store is found, then it will remove the store object and print '*Successfully removed store with store id: xx*', where *xx* is the input store id. Otherwise, it will print '*Store not found!*'.
  8. The `add_new_store` will receive store name, store city, and store area. It will create a store object and add it to the `store_obj_list` if it does not exist already. If a new store object is added to the list, then it will print '*Successfully added new xx store!!*', where *xx* is the input store name.
  9. The `add_new_product` will receive store id, product name, products sold, products available, product cost, and product price. If a store with input store id does not exist, then the method will print '*Store not found! Product can't be added*'. Otherwise, the method will create a product object and add it to the store object using the `add_product` method.

## Driver Script and Sample Run

The `storeDB.py` file has been provided as part of this assignment. This file includes the necessary driver code. You should write your solution in the given `storeDB.py` file. A sample run of the completed code is shown below:

```

*****
1000 records added to unique 997 stores.
*****
Searching for a store named "Walmart"...
Store_ID:29;Store_Name:Walmart;Store_City:Si Racha;Store_Area:2 Laurel Drive
Store_ID:69;Store_Name:Walmart;Store_City:Montpellier;Store_Area:6 Ridgeway Hill
Store_ID:171;Store_Name:Walmart;Store_City:Th? Tr?n C?m Th?y;Store_Area:2514 Summit Court
Store_ID:181;Store_Name:Walmart;Store_City:Klau;Store_Area:82167 Russell Junction
Store_ID:192;Store_Name:Walmart;Store_City:Pordapor Barat;Store_Area:655 Union Center
Store_ID:196;Store_Name:Walmart;Store_City:Parai;Store_Area:57004 Southoff Avenue
Store_ID:277;Store_Name:Walmart;Store_City:Kiel;Store_Area:1830 Forest Dale Parkway
Store_ID:286;Store_Name:Walmart;Store_City:Baijiang;Store_Area:881 Eagan Way
Store_ID:363;Store_Name:Walmart;Store_City:Chornobay;Store_Area:51591 Morning Trail
Store_ID:372;Store_Name:Walmart;Store_City:Matriz de Camaragibe;Store_Area:2 Texas Lane
Store_ID:377;Store_Name:Walmart;Store_City:Ágios Vasíleios;Store_Area:47909 John Wall Avenue
Store_ID:393;Store_Name:Walmart;Store_City:Dinan;Store_Area:25 Corscot Road
Store_ID:407;Store_Name:Walmart;Store_City:Gielniów;Store_Area:1062 Brickson Park Pass
Store_ID:428;Store_Name:Walmart;Store_City:Älvsjö;Store_Area:7 Mayfield Crossing
Store_ID:441;Store_Name:Walmart;Store_City:Meiqi;Store_Area:329 Blackbird Crossing
Store_ID:457;Store_Name:Walmart;Store_City:Purac;Store_Area:7140 Springview Pass
Store_ID:466;Store_Name:Walmart;Store_City:Bich ??ng;Store_Area:0855 Eagan Lane
Store_ID:511;Store_Name:Walmart;Store_City:Purwojoyo;Store_Area:985 West Road
Store_ID:516;Store_Name:Walmart;Store_City:Gorey;Store_Area:4 Anniversary Plaza
Store_ID:561;Store_Name:Walmart;Store_City:Columbus;Store_Area:81058 Goodland Place
Store_ID:595;Store_Name:Walmart;Store_City:Lis;Store_Area:0 Cambridge Court
Store_ID:631;Store_Name:Walmart;Store_City:Sevan;Store_Area:92 Del Sol Alley
Store_ID:652;Store_Name:Walmart;Store_City:P?shm?l;Store_Area:3719 Moose Alley
Store_ID:658;Store_Name:Walmart;Store_City:Adelaide;Store_Area:369 Fairfield Pass
Store_ID:712;Store_Name:Walmart;Store_City:Flore?ti;Store_Area:71 Wayridge Point
Store_ID:727;Store_Name:Walmart;Store_City:Sapataria;Store_Area:881 Lakewood Gardens Point
Store_ID:809;Store_Name:Walmart;Store_City:Bombon;Store_Area:0 Fairview Point
Store_ID:936;Store_Name:Walmart;Store_City:Chambéry;Store_Area:867 Fallview Junction
Store_ID:969;Store_Name:Walmart;Store_City:Volgo-Kaspiyskiy;Store_Area:660 Esker Alley
Store_ID:976;Store_Name:Walmart;Store_City:Zhuanqiang;Store_Area:93318 Cascade Park
Store_ID:979;Store_Name:Walmart;Store_City:Bulqizë;Store_Area:29193 Lunder Lane
Store_ID:993;Store_Name:Walmart;Store_City:Meixi;Store_Area:59238 Shopko Terrace
Store_ID:996;Store_Name:Walmart;Store_City:Anserma;Store_Area:7 Roxbury Place
33 records found for a store named "Walmart"
*****

```

Figure 1: Sample run - part 1

```

*****
Printing product list for store ID 29....
Product ID:29; Name:Soup - Base Broth Beef; Sold:6; Available:5
Product ID:161; Name:Cheese - Brick With Pepper; Sold:12; Available:3
Product ID:411; Name:Cake Circle, Foil, Scallop; Sold:1; Available:12
Product ID:915; Name:Fondant - Icing; Sold:1; Available:6
*****
Successfully removed store with store id: 996
*****
Successfully added new Walmart store!!
*****
Successfully added the new product!!
*****

```

Figure 2: Sample run - part 2.

```

Searching for a store named "Walmart"....
Store_ID:29;Store_Name:Walmart;Store_City:Si Racha;Store_Area:2 Laurel Drive
Store_ID:69;Store_Name:Walmart;Store_City:Montpellier;Store_Area:6 Ridgeway Hill
Store_ID:171;Store_Name:Walmart;Store_City:Th? Tr?n C?m Th?y;Store_Area:2514 Summit Court
Store_ID:181;Store_Name:Walmart;Store_City:KLau;Store_Area:82167 Russell Junction
Store_ID:192;Store_Name:Walmart;Store_City:Pordapor Barat;Store_Area:655 Union Center
Store_ID:196;Store_Name:Walmart;Store_City:?arai;Store_Area:57004 Sauthoff Avenue
Store_ID:277;Store_Name:Walmart;Store_City:Kiel;Store_Area:1830 Forest Dale Parkway
Store_ID:286;Store_Name:Walmart;Store_City:Baijiang;Store_Area:881 Eagan Way
Store_ID:363;Store_Name:Walmart;Store_City:Chornobay;Store_Area:51591 Morning Trail
Store_ID:372;Store_Name:Walmart;Store_City:Matriz de Camaragibe;Store_Area:2 Texas Lane
Store_ID:377;Store_Name:Walmart;Store_City:Ágios Vasíleios;Store_Area:47909 John Wall Avenue
Store_ID:393;Store_Name:Walmart;Store_City:Dinan;Store_Area:25 Corscot Road
Store_ID:407;Store_Name:Walmart;Store_City:Gielniów;Store_Area:1062 Brickson Park Pass
Store_ID:428;Store_Name:Walmart;Store_City:Älvsjö;Store_Area:7 Mayfield Crossing
Store_ID:441;Store_Name:Walmart;Store_City:Meiqi;Store_Area:329 Blackbird Crossing
Store_ID:457;Store_Name:Walmart;Store_City:Purac;Store_Area:7140 Springview Pass
Store_ID:466;Store_Name:Walmart;Store_City:Bich ??ng;Store_Area:0855 Eagan Lane
Store_ID:511;Store_Name:Walmart;Store_City:Purwojoyo;Store_Area:985 West Road
Store_ID:516;Store_Name:Walmart;Store_City:Gorey;Store_Area:4 Anniversary Plaza
Store_ID:561;Store_Name:Walmart;Store_City:Columbus;Store_Area:81058 Goodland Place
Store_ID:595;Store_Name:Walmart;Store_City:Lis;Store_Area:0 Cambridge Court
Store_ID:631;Store_Name:Walmart;Store_City:Sevan;Store_Area:92 Del Sol Alley
Store_ID:652;Store_Name:Walmart;Store_City:P?shm?l;Store_Area:3719 Moose Alley
Store_ID:658;Store_Name:Walmart;Store_City:Adelaide;Store_Area:369 Fairfield Pass
Store_ID:712;Store_Name:Walmart;Store_City:Flore?ti;Store_Area:71 Wayridge Point
Store_ID:727;Store_Name:Walmart;Store_City:Sapataria;Store_Area:881 Lakewood Gardens Point
Store_ID:809;Store_Name:Walmart;Store_City:Bombon;Store_Area:0 Fairview Point
Store_ID:936;Store_Name:Walmart;Store_City:Chambéry;Store_Area:867 Fallview Junction
Store_ID:969;Store_Name:Walmart;Store_City:Volgo-Kaspiyskiy;Store_Area:660 Esker Alley
Store_ID:976;Store_Name:Walmart;Store_City:Zhuanqiang;Store_Area:93318 Cascade Park
Store_ID:979;Store_Name:Walmart;Store_City:Bulqizë;Store_Area:29193 Lunder Lane
Store_ID:993;Store_Name:Walmart;Store_City:Meixi;Store_Area:59238 Shopko Terrace
Store_ID:1001;Store_Name:Walmart;Store_City:Charlottetown;Store_Area:1 abc st
33 records found for a store named "Walmart"
*****
Printing product list for store ID 1001....
Product ID:1001; Name:Laptop; Sold:300; Available:200
*****
Process finished with exit code 0

```

Figure 3: Sample run - part 3.

## Submitting your assignment

Your submission will consist of one zip file containing solutions to all problems (see below). This file will be submitted via a link provided on our Moodle page just below the assignment description. This file must be uploaded by 5pm (Moodle time) on the due date in order to be accepted. You can submit your solution any number of times prior to the cutoff time (each upload overwrites the previous one). Therefore, you can (and should) practice uploading your solution and verifying that it has arrived correctly.

### Example:

Submission file: asn2\_studentnum.zip

Where studentnum is your student ID number.

Your zip file should contain a single Python file storeDB.py.