ECE 122: Introduction to Programming for ECE- Summer 2019

Project 2: My Media Store (procedural programming example)

Due Date: see website, class policy and Moodle for submission. This is an individual project- to do it alone (discussions are encouraged but no sharing of code)

Description:

The purpose of software is to manage data. Most applications have three main components: a user interface, or UI, a database, and code that manages the interaction between the UI and the database. Access to the data is provided by an *interface*: a set of functions that performs operations on the data, such as search, add, remove, etc. This project will give you some experience working with this type of code base. Another aspect of this project is working with how data is organized in a *data type*. A data type is (often) a programmer-defined object that models a real-world entity that the program manages. A data type has attributes, or values, and functions. Data types are key to Object-Oriented programming and help keep data organized and provide for a way to keep the complexity of the program more manageable for future development, troubleshooting and debugging.

The goal of this project is to design and implement a basic media library system in which administrators (you) can inquire about the current inventory, and add or remove items. These items are identified either as books or movies. The media library is organized as follows. Each entry in the library can be identified by its four attributes: Reference, media type (Movie or Book), its Title, and its Price. A sample media library comprised of 9 items is given in Table 1 below.

Reference	Media	Title	Price
TU2RL012	Movie	2001: A Space Odyssey	$11.99
GV5N32M9	Book	A Brief History of Time	$10.17
1DB6HK3L	Movie	North by Northwest	$8.99
PO5T7Y89	Movie	The Good, The Bad and The Ugly	$9.99
TR3FL0EW	Book	The Alchemist	$6.99
F2O9PIE9	Book	Thus Spoke Zarathustra	$7.81

R399CED1	Book	Jonathan Livingston Seagull	$6.97
2FG6B3N9	Movie	Gone with the Wind	$4.99
6Y9OPL87	Book	Gone with the Wind	$7.99

Table 1: A sample media library.

In addition, a book media item includes information about by its Author, while a movie media item includes information about its Director and Lead Actor. Examples of these media-specific attributes are provided in Table 2, below.

Reference	Author- Book	Director- Movie	Main Actor- Movie
TU2RL012		Stanley Kubrick	Keir Dullea
GV5N32M9	Stephen Hawking		
1DB6HK3L		Alfred Hitchcock	Cary Grant
PO5T7Y89		Sergio Leone	Clint Eastwood
TR3FL0EW	Paulo Coelho		
F2O9PIE9	Friedrich Nietzsche		
R399CED1	Richard Bach		
2FG6B3N9		Victor Fleming	Vivien Leigh
6Y9OPL87	Margarett Mitchell		

Table 2: Additional information for the sample media library.

At the first execution of the program (the file media_store.py), the output includes a menu containing several options and should look like this:

```
Welcome to BestMedia
====================
Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
```

Once option 0 (for Exit) is selected, the program stops. All of the options 1-7 are described below. The project should be developed incrementally, one option at a time. Write one option, test it, and make sure it works correctly before moving on. Use a debugger as it will help you see how your code works at runtime.

After Task 1 is done, all the other Tasks/options can be completed in any order.

How to start:

Download the project zip file and extract the contents to a folder on your machine. The three .py files that you need for this project are provided. Do not change their names or add any other files. You will add your code to these files to build the functionality specified in this document.

1. MediaItem.py file/module that contains the user-defined type MediaItem. Each Object of this type represents the data for one item in the library.

2. inventory.py file/module that contains all the necessary functions to operate the inventory. This file provides the interface (functions) for working with the library of items. The library is implemented in code as a list.

3. media_store.py file contains the main program. This code carries out the interaction with the user (the UI) and the library (the data). This code uses the inventory.py code to initialize the list when the program starts, and when it needs to perform any operations on the list of MediaItems.

Read this document completely and look over the code provided. Then begin to develop option 0 and 1. Once you have done that, and you have verified that they work correctly, move on to the next option.

Use your preferred IDE to write and save your files- preferably as a project. ***Note: your***

project submission must be a .zip file (not a 7-zip, rar, or tarball) and must be named in this manner:

Your first initial_your last name_ECE122project2.zip For example, my name is Gordon Anderson, so my file would be named:

g_anderson_ECE122project2.zip

I will only accept any files that follow this naming convention. I will only accept files with a .zip extension.

Remember: ***any function definitions should go at the top of the file***. Do not forget to comment your code when necessary. Make sure you obtain the exact same output for the exact same input for the examples below (this includes syntax, blank spaces, and skipping blank lines). Your program will also be graded on coding style as this is a very important aspect of writing code.

Important: Use the style detailed in the ***"Python Coding Style"*** link posted on Moodle. Remember that you are writing code for another human to read and understand. Someone else will need to work with your code at some point. Good coding style makes that much easier, and your code looks professional, which inspires confidence in your work.

Grading:

This project will be graded out of 100 points:

1. Your program should implement all basic functionality/Tasks and run correctly (90 points).

2. Overall programming style: program should have proper identification, and comments. (10 points).

Options 0 and 1- [30pts]

Option 0: The program ends when option 0 is selected. This is the output, input and corresponding behavior of the program:

```
Welcome to BestMedia
====================
Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
```

```
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>0
Goodbye!
```

Option 1: This is the output, input and corresponding behavior of the program when option 1 is selected (notice that the message “`Welcome to BestMedia`” is only displayed when the program is started and not afterwards):

```
Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>1
Reference/Media/Title/Price
---------------------------
TU2RL012 Movie 2001: A Space Odyssey $11.99
GV5N32M9 Book A Brief History of Time $10.17
1DB6HK3L Movie North by Northwest $8.99
PO5T7Y89 Movie The Good, The Bad and The Ugly $9.99
TR3FL0EW Book The Alchemist $6.99
F2O9PIE9 Book Thus Spoke Zarathustra $7.81
R399CED1 Book Jonathan Livingston Seagull $6.97
2FG6B3N9 Movie Gone with the Wind $4.99
6Y9OPL87 Book Gone with the Wind $7.99

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
```

```
0-Exit
Enter Command:
```

An inventory of the media store is displayed when option 1 is selected. We can see the references, media type, title and price of each item. At the end the menu selection is printed again and the program is waiting for you to make another choice. What you need to implement for option 1:

1. In the media_store.py file:
    a) Add a call to a function initialize that will return a list of items (data objects of type MediaItem...explained further below).
    b) Write a while loop that keeps printing the menu selection and asking the user to enter a command choice; this while loop will exit if the entry is 0 and print a 'Goodbye!' message; the function to print the menu, display menu, is already provided to you in the file inventory.py.
    c) Add the option 1 that contains a call to a function `display`. Note that the header of the file already contains the instruction import inventory. This will allow you to call the functions in the inventory file using the dot operator.

2. In the MediaItem.py file:

    a) Implement the class that defines the type data object MediaItem that includes the constructor (i.e. function __init__). You will consider the following attributes: media, title, price, price, ref, director, lead actor, author. All the attributes can be initialized to `None` by default.

3. In the inventory.py file:

    a) Complete the `initialize` function that creates and returns a list of MediaItem data objects. Ideally we would like to read the entire inventory from a file (so we could easily consider 1000s of items if needed) but we will do that later in the semester. Here, you will need to fill up by hand (hard coded) all the attributes of the objects for our selected 9 items presented in Tables 1 and 2 (a bit long but you can cc-paste title, etc. from this pdf file).

    b) Write the function `display` that displays the items as presented in the output example (you can use\n for line breaks and \t to separate each field of the items). We note that the header of the file contains the instruction from MediaItem import MediaItem which will allow you to use the MediaItem data type.

Option-2- [15pts]

This is the output, input and corresponding behavior of the program when option 2 is selected:

```
Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>1

Inventory is worth $75.89
Most expensive item at $11.99
There are 5 Book(s), and 4 Movie(s)

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
```

The program displays some info about the inventory: the total value of all the media items, the price of the most expensive item, and the total number of books and movies found in the whole inventory. What you need to implement:

1. In the media_store.py file: Write the option 2 that contains a call to the inventory function `info`.

2. In the inventory.py file: the function `info` that displays the items as presented in the output example. You will probably need a for loop that scans through all members of the list of media items. To find the maximum price you could implement the linear search algorithm.

Options-3 and 4- [10pts]

This is the output, input and corresponding behavior of the program when options 3 or option 4 are selected.

Option 3 prints data about only Book items, and option 4 prints data about only Movie items. First, here is an example when option 3 is selected

```
Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>3
Reference/Media/Title/Price
---------------------------
GV5N32M9 Book A Brief History of Time $10.17
TR3FL0EW Book The Alchemist $6.99
F2O9PIE9 Book Thus Spoke Zarathustra $7.81
R399CED1 Book Jonathan Livingston Seagull $6.97
6Y9OPL87 Book Gone with the Wind $7.99
```

This is the output, input and corresponding behavior of the program when option 4 is selected:

```
Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>4
Reference/Media/Title/Price
---------------------------
TU2RL012 Movie 2001: A Space Odyssey $11.99
1DB6HK3L Movie North by Northwest $8.99
PO5T7Y89 Movie The Good, The Bad and The Ugly $9.99
```

```
2FG6B3N9 Movie Gone with the Wind $4.99

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
```

To implement Options 3 and 4:

This is very similar to option 1 but either the media Book items or Movie items are displayed.

In the media_store.py file: Write the options 3 and 4 that contain calls to the inventory function `display`.

In the inventory.py file: you could modify the function `display` by including an additional argument that specified the type of the media: “Book” or “Movie”.

Option-5- [10pts]

This option searches for an item by its reference number and displays its information. The following is the output, input and corresponding behavior of the program when option 5 is selected (here we do it three times, once for a Movie, once for a Book, and once when the reference does not exist):

```
Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>5
Enter item reference:
```

```
>>>1DB6HK3L
Title: North by Northwest (Ref: 1DB6HK3L, Price: $8.99);
Movie Director: Alfred Hitchcock; Lead Actor: Cary Grant

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>5
Enter item reference:
>>>R399CED1
Title: Jonathan Livingston Seagull (Ref: R399CED1, Price:
$6.97);
Author: Richard Bach

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>5
Enter item reference:
>>>ECE122ECE
No such item found!

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
```

```
7-Add Item
0-Exit
Enter Command:
```

The program prompts the user to enter a particular item reference and it will display all information about this item (or returns “No such item found!” if the item is not in the inventory). A Book item will also include information about the author while a Movie item will include information about the director and lead actor. What you need to implement:

1. In the media_store.py file: Write the code for option 5 that prompts the user for the reference, and includes a call to a function `search_item` that returns the MediaItem object if found (could return None if not found). Another call would be to the function `display item` that displays the information for the selected item as presented in the output example.

2. In the inventory.py file: implement the `search_item` and `display_item` functions as described above and in the code.

Option-6- [10pts]

This option allows an item to be removed from the library. This is the output, input and corresponding behavior of the program when option 6 is selected. Here we select option 6, followed by option 1, option 3 and option 2 to view the library after the option 6 operation.

```
Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>6
Enter item reference:
>>>GV5N32M9

Menu
====
1-List Inventory
2-Info Inventory
```

```
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>1
Reference/Media/Title/Price
---------------------------
TU2RL012 Movie 2001: A Space Odyssey $11.99
1DB6HK3L Movie North by Northwest $8.99
PO5T7Y89 Movie The Good, The Bad and The Ugly $9.99
TR3FL0EW Book The Alchemist $6.99
F2O9PIE9 Book Thus Spoke Zarathustra $7.81
R399CED1 Book Jonathan Livingston Seagull $6.97
2FG6B3N9 Movie Gone with the Wind $4.99
6Y9OPL87 Book Gone with the Wind $7.99

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>3
Reference/Media/Title/Price
---------------------------
TR3FL0EW Book The Alchemist $6.99
F2O9PIE9 Book Thus Spoke Zarathustra $7.81
R399CED1 Book Jonathan Livingston Seagull $6.97
6Y9OPL87 Book Gone with the Wind $7.99

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
```

```
7-Add Item
0-Exit
Enter Command:
>>>2
Inventory is worth $65.72
Most expensive item at $11.99
There are 4 Book(s), and 4 Movie(s)

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
```

This option asks the user to enter a particular reference. The corresponding item will be removed from the inventory as seen in the example when you select option 1 the book “A Brief History of Time”is not there anymore. It also disappears from the list of books (since it is a book). Option 2 can return the new info about the new inventory that now only contains 8 items. What you need to implement:

1. In the media_store.py file: Write the option 6 code that includes a call to a function `search_item_index` to return the index of the item to remove from the list. We have seen in class how the built-in del function can be used to then remove an item with a given index from a list (and left shift all the other items with higher indexes).

2. In the inventory.py file: Implement the method above.

Option-7- [15pts]

This option allows a user to add either a new Book or Movie to the library. This is the output, input and corresponding behavior of the program when option 7 is selected. There are three examples of option 7 here, followed by option 1 and then 2 to show the results of adding a Book and a Movie).

```
Menu
====
1-List Inventory
2-Info Inventory
```

```
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>7
Book or Movie?
>>>Book
Enter Book Title:
>>>Hamlet
Enter Book Reference:
>>>J45K99EE
Enter Book Price:
>>>4.59
Enter Author Name:
>>>William Shakespeare

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>7
Book or Movie?
>>>Movie
Enter Movie Title:
>>>The Great Escape
Enter Movie Reference:
>>>ER89QW43
Enter Movie Price:
>>>7.99
Enter Director Name:
>>>John Sturges
Enter Lead Actor Name:
>>>Steve McQueen

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>7
Book or Movie?
>>>Novel
Wrong input!

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>1
Reference/Media/Title/Price
---------------------------
TU2RL012 Movie 2001: A Space Odyssey $11.99
1DB6HK3L Movie North by Northwest $8.99
PO5T7Y89 Movie The Good, The Bad and The Ugly $9.99
TR3FL0EW Book The Alchemist $6.99
F2O9PIE9 Book Thus Spoke Zarathustra $7.81
R399CED1 Book Jonathan Livingston Seagull $6.97
2FG6B3N9 Movie Gone with the Wind $4.99
6Y9OPL87 Book Gone with the Wind $7.99
J45K99EE Book Hamlet $4.59
ER89QW43 Movie The Great Escape $7.99

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
```

```
6-Remove Item
7-Add Item
0-Exit
Enter Command:
>>>2
Inventory is worth $78.3
Most expensive item at $11.99
There are 5 Book(s), and 5 Movie(s)

Menu
====
1-List Inventory
2-Info Inventory
3-List of All Books
4-List of All Movies
5-Item Description
6-Remove Item
7-Add Item
0-Exit
Enter Command:
```

For option 7, once you ask the user to enter which media to select (Book or Movie), you will need a new function create item that returns a new data object MediaItem and append it to the list (creating then a new inventory)

1. In the media_store.py file: Write the option 7 code that determines which media type to add, prompts the user for the appropriate data, creates a new MediaItem object and adds it to the library. This code includes a call to the inventory function `create_item`.

2. In the inventory.py file: Implement the `create_item` method above. It prompts the user for the data needed to create the MediaItem.