

DOCUMENTATION

LAPTOP.PY – THIS IS A SMALL CONSOLE APPLICATION IMITATING LAPTOP SHOP. APPLICATION ALLOWS TO CREATE, EDIT AND DELETE LAPTOP ITEMS. PROGRAM USES CACHE MEMORY TO SAVE THE PROGRES AND DO NOT MAKE SAVES AFTER CLOSING, BUT LOGIC IMITATE OPERATION ON DATABASE. APPLICATION IS EQUIPPED WITH USER INPUT ERRORS HANDLING, WHAT PROTECT PROGRAM FROM UNEXPECTED CRASH.

PART I – CLASS AND METHODS

Class Laptop:

Main class which define object in application. On that objects program performs every operations related to methods in the class. There is two object created by default in code:

```
lap_01 = Laptop('DELL',13.3,['Intel Core i5-8250U','RAM[GB]:8','SSD[GB]:256','Intel UHD Graphics 620','Windows 10'],True,3499)
lap_02 = Laptop('ASUS',14,['Intel Core i3-8145U','RAM[GB]:4','SSD[GB]:256','Intel UHD Graphics 620','Windows 10 Home'],False,2257)
```

Class Variables:

items – List. Contains every object created by Laptop class. Base on this list program can refer to the class objects and preform crucial logic operations. On the beginning contains two object which are created in code by default. User can modify it by his action.

promotion – List. Contains elements – in the assumption the same as class object names - which are in a promotion. Operating with lower letters (*name.lower()*). By default contains one element. User can modify it by his action.

```
items = []
promotion = ['dell']
```

Class Methods:

Name(def)	Arguments and Aggregated Function	Description and Functionality
export_to_csv() (class method)	Arg.: cls , path (str) – path to file	Create csv file with current object (class instances) from items list.
item_list() (class method)	Arg.: cls	Print current object (class instances) from items list in the console interface. Feature for the user.

Show_Laptop_List() (class method)	Arg.: cls , lap_list (list) – mainly items var. Is provided Func.: Set_Price() , Get_Info()	Print all updated information (arguments) of every instance from items list to the console interface. Contains feature related to update price when object is in promotion. Result depends on user action (printing current status).
__init__() (instance method)	Arg.: self , name (str) – model name, screen (str) – size of screen, parameters (list) -technical parameters, IsOnsale (bool) – is Laptop in promotion or not. Additional approval which allows change price arg. (hidden) price – (hidden, int – because of default values on start object. When user provide it, float type is acceptable).	Initialize new class instance – new Laptop object and append it to the items list by default. (<i>items.append(self)</i> – feature)
Get_Info() (instance method)	Arg.: self	Format and print all arguments which includes single instance of Laptop class. Is aggregated in func.: Show_Laptop_List .
add_parameter() (instance method)	Arg.: self , parameter (str) – tech. parameter user want to add to self.parameter arg.(list) in the instance.	Add new technical parameter of Laptop to the parameters list argument. Voluntary.
remove_parameter() (instance method)	Arg.: self , parameter – parameter to remove from self.parameter arg.(list) in the instance.	Adding new technical parameter of Laptop to the parameters list argument. Voluntary. Func. operating on indexes of self.parameter arg.(list)
Add_Item_to_promotion() (instance method)	Arg.: self ,	Adding new element to promotion list. Func. Always add self.name arg. of instance. Operating on lower letter (<i>self.name.lower()</i>)
IsOnSale() (with property) (instance method)	Arg.: self	Return hidden arg. self.__IsOnsale.

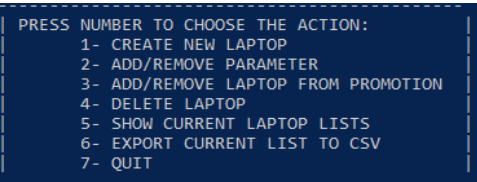
		It is needed in connection with @property func. decorator (functools module).
IsOnSale() (with setter) (instance method)	Arg.: self newStatus (bool) – new status define allowing of instance to being in promotion.	Additional feature which is not available for the user. This func. is changing instance arg. __IsOnSale on True, if user add the instance <i>self.name</i> to the promotion list. (This feature has a potential to became some kind of double check or even approval, given to the app when object is adding to the promotion list. We can imagine business case when end user has access only to the promotion list, but changing price require additional approval of administrator – who has only access to instance hidden arg. __IsOnSale via IsOnSale function.
Set_Price() (instance method)	Arg.: self	Changing instance argument self.price , if instance args.: __IsOnSale = True and self.name.lower() (lower letters) is in promotion list. Changing of price is coded by default - 5% discount of base price.
remove_item() (instance method)	Arg.: self	Deleting instance from items list.
Wrapper_with_Logs() (general func.)	Arg.: activity (str) – define type of tracked activity (Create_New_Obj, Delete_Obj, Add_to_Promotion, Change_Sale_Status) path (str) – path to file Func.: Function_with_wrapper() , Wrapper()	Main activity/log tracker. It is created to track functions such as: Create_Obj(), Add_Item_to_promotion(), IsOnSale(), Remove_item(), and create txt. files with basics logs. Works in connection with decorator “@”
Function_with_wrapper()	Arg.: func – tracked function Func.: Wrapper()	Get the tracked function as argument and return a “real” Wrapper() .

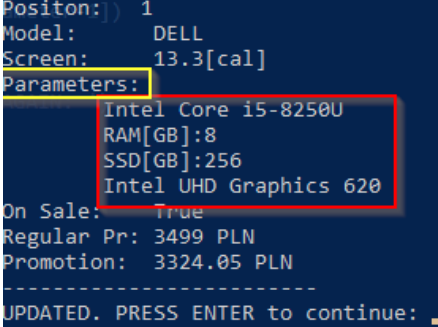
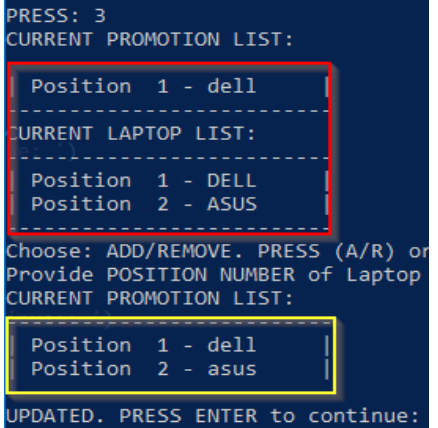
Wrapper()	Arg.: <i>*args</i> – take all argument of tracked function (function which is an argument of <code>Function_with_wrapper(func)</code>)	Creating different txt.files with logs (depends on arguments given). Return tracked function.
------------------	---	---

PART II – CONSOLE INTERFACE

Interface allows user to choose an action. Is responsible to output for the user instruction, what to do to complete chosen task. Most often there are the commands such as: press Enter/(Y/N), select action: 1/2/3, pick item position: 1,2,3...

Functions:

Name(def)	Arguments and Aggregated Function	Description and Functionality
Interface()	Arg.: - Func.: User_action()	Layout of interface. Application initializing function. 
Create_Obj(),	Arg.: - Func.: -	Giving the user possibility to input date (arguments needed to initializing the new instance): <i>Laptop(name, screen, parameters, IsOnSale, price)</i>
User_action()	Arg.: - Func.: CREATE_NEW_LAPTOP(), ADD_OR_REM_PARAMETER(), ADD_OR_REM_FROM_PROM(), DELETE_LAPTOP(), SHOW_CURRENT_LAPTOP_LISTS(), QUIT(), return_to_menu(), export_to_csv()	Initializing of user activity. Function use variable: action to let user choose a task. Aggregate all available tasks.
return_to_menu()	Arg.: - Func.: os.system('cls'), Interface()	Cleaning the screen and back to Interface view.

position()	Arg.: base_list (list) – operating list, kind – description of list	Function pick an element from operating list, which is needed to current action ordered by the user. Basically application uses this function only for items (class) list, but if operations on the other lists will be needed (example: promotion list) feature is already adapted.
ADD_OR_REM_PARAMETER()	Arg.: - Func.: return_to_menu() position()	User can add or remove parameters from instance arg. self.parameters . Function shows updated instance in console: 
ADD_OR_REM_FROM_PROM()	Arg.: - Func.: return_to_menu() position()	Function shows user current status of promotion and item lists (class). Related to chosen action add or remove instance from promotion and properly change instance hidden arg. __IsOnSale . This action has direct influents on price (related to instance Set_Price() func.) Function shows updated lists. 
DELETE_LAPTOP()	Arg.: - Func.: return_to_menu()	Deleting instance from current object (items) list. If object name is in promotion list, is deleting by default as well.
SHOW_CURRENT_LAPTOP_LISTS()	Arg.: - Func.:	Showing current status of promotion and items (instance) list.

	Show_Laptop_List(), return_to_menu()	
QUIT()	sys module	exit