2022W CS 1920
Computer Science 2
Programming Assignment 1
Mark Contribution: 5%

Assignment Objectives
- Design your own class in the Python programming language.
- Implement accessor and mutator methods that allow access to your object data
- Create a dunder method for your class.
- Employ your creativity to design a class that is unique to you.

Scenario
In this assignment we are pretending you are running your own online storefront! It will be the store of your dreams and will sell only the type of product that you want to sell! However, but first you need to store information about those products in a class called Product.

Example
If we were creating a class for a Comic Book Shop selling Comic Books, we could define the Product attributes as being:

- Unique ID: Integer
- Comic Title: String
- Issue Number: Integer
- Price: Float in the interval of 2.99 to 8.99
- Description: String

So an example object instance of a comic issue for the series Saga by Brian K. Vaughn and Fiona Staples would be:
- Unique ID: 1
- Comic Title: Saga
- Issue Number: 56
- Price: 2.99
- Description: Everyone mourns in their own way. The most emotional epic in comics continues.

Note: you are not supposed to create the exact same product class as above. You must create your own class with your own attributes, mutators and valid ranges according to the task below.

2022W CS 1920
Computer Science 2
Programming Assignment 1
Mark Contribution: 5%

Your Sample Data:

You must create a file called **asn1_00000_products.csv** where 00000 is replaced with your student number. This file contains products that fit your product attributes. For example, the following CSV file would work for a set of comics:

Comic_Title,Issue_Number,Price,Description
Saga, 1, 2.99, New parents Marko and Alana risk everything to raise their child amidst a never-ending galactic war.
Saga, 2, 2.99, After deserting their galactic armies former soldiers Marko and Alana must now protect their newborn girl.
Saga, 3, 2.99, Stranded on a mystical alien world new parents Marko and Alana encounter their greatest fear.
Saga, 4, 2.99, See why everyone is talking about this hit new ongoing adventure.
Saga, 5, 3.99, The hunt of Prince Robot IV for Hazel and her parents takes a deadly turn.

Note – you must make sure that none of your strings have a comma within them or the CSV format will fail when you read it in!

Your Product Class:

You are going to write a class Product that contains all of the information about the type of product you want to sell. It must contain:

- A unique identifier stored as an integer.
- A brief description (less than 200 characters long) of a particular instance of a product.
- Exactly 3 additional instance variables: one string, one float and one integer of your own design that are associated with the product.
- The class constructor will receive the description and the values for the above attributes and use them to create an instance of the class. It will also generate the unique identifier starting at 1 from a shared static variable and increment by one every time a product is created. Each time a product is created you will need to call a static method: ***get_next_id()*** which returns the next identifier.
- You must use name mangling to make all instance variables inaccessible from outside the class (including the name and description).
- There must be accessor and mutators for all private instance variables.
- The float instance variable mutator should check whether the provided value is within a given interval (of your design) and only updates the attribute if that's the case. It returns true if the update is successful, false otherwise. In the comic example we restrict the price to be between 3.99 and 8.99.
- You must implement the **__str__** method for printing the product's info.
- You must also provide appropriate ***docstring*** documentation for classes and methods.
- You must provide a driver script as specified below.

Driver Script and Sample Run:

Your code should be contained in 1 file called asn1_00000.py where 00000 is replaced by your student number. Your code should have a main program defined and all driver code should be in that main program function.

Your driver should read in the 5 products from a pre-prepared CSV file. It should then print all 5 objects sequentially as shown in the printout using the __str__ method.

Then, you should have each of the following operations:

For the first product in your list:

- Retrieve the string attribute via its accessor method and print it with a label.
- Print the words "Updating string attribute value …"
- Update the string attribute using the mutator method to a value of your choice.
- Retrieve the string attribute via its accessor method and print it with a label showing the change.

For the second product in your list:

- Retrieve the integer attribute via its accessor method and print it with a label.
- Print the words "Updating integer attribute value …"
- Update the integer attribute using the mutator method to a value of your choice.
- Retrieve the integer attribute via its accessor method and print it with a label showing the change.

For the third product in your list:

- Retrieve the float attribute via its accessor method and print it with a label.
- Print the words "Updating float attribute with illegal value…"
- Update the float attribute using the mutator method to a value that is outside the allowed range.
- Print the return value of that mutator call with "Update of float attribute:" and its value.
- Retrieve the float attribute via its accessor method and print it with a label.
- Print the words "Updating float attribute with legal value…"
- Print the return value of that mutator call with "Update of float attribute:" and its value.
- Retrieve the float attribute via its accessor method and print it with a label.

The output should be like Figure 1, but replaced with your product, attributes and data.

At the end of the driver, write out the set of products to a new CSV file called asn1_00000_updated_products.csv where 00000 is replaced with your student number. This file should have the same headings as the original with an additional column at the front for the unique identifier, and then the updated products, as shown in the following example.

Unique_ID, Comic_Title,Issue_Number,Price,Description
1,Saga TPB,1,2.99,New parents Marko and Alana risk everything to raise their child amidst a never-ending galactic war.
2,Saga,2,2.99,After deserting their galactic armies former soldiers Marko and Alana must now protect their newborn girl.
3,Saga,3,2.99,Stranded on a mystical alien world new parents Marko and Alana encounter their greatest fear.
4,Saga,4,2.99,See why everyone is talking about this hit new ongoing adventure.
5,Saga,5,3.99,The hunt of Prince Robot IV for Hazel and her parents takes a deadly turn.

Figure 1: Code output showing the run of the program.


```
ID: 1; Comic Title:Saga; Issue:  1; Price:  2.99; Desc:  New parents Marko and Alana risk everything to raise their child amidst a never-ending galactic war.
ID: 2; Comic Title:Saga; Issue:  2; Price:  2.99; Desc:  After deserting their galactic armies former soldiers Marko and Alana must now protect their newborn girl.
ID: 3; Comic Title:Saga; Issue:  3; Price:  2.99; Desc:  Stranded on a mystical alien world new parents Marko and Alana encounter their greatest fear.
ID: 4; Comic Title:Saga; Issue:  4; Price:  2.99; Desc:  See why everyone is talking about this hit new ongoing adventure.
ID: 5; Comic Title:Saga; Issue:  5; Price:  2.99; Desc:  The hunt of Prince Robot IV for Hazel and her parents takes a deadly turn.
Comic Title of Product 1: Saga
Updating string attribute value
Updated Comic Title of Product 1: Saga TPB 1
Comic Issue of Product 2:  2
Updating integer attribute value ...
Comic Issue of Product 2: 999
Comic Price of Product 3:  2.99
Updating float attribute with illegal value ...
Update of float attribute:False
Comic Price of Product 3:  2.99
Updating float attribute with legal value ...
Update of float attribute:True
Comic Price of Product 3: 3.99

Process finished with exit code 0
```

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. When you submit your file you should download that file and determine if you have uploaded the correct files. We can only mark what is handed in on the day of the assignment

What's in your zip file?

A zip file is a compressed file that can contain any number of folders and files. Name your zip file using this format: asn1_00000.zip where 00000 is replaced with your student number. Your zip file must only contain the .py file and the original .csv file for your products. There should be no other files contained within. If any other files of any kind are provided there will be a deduction applied to your assignment.

Late Penalties

An assignment received between 5:01pm on the due date (usually Friday) and 5:00pm on the following day (usually Saturday) will be accepted as 1 day late and will have a 10% penalty applied. An assignment received between 5:01pm on the first day and 11:59pm on the second day following the assignment due date (usually Sunday) will have a 30% penalty applied. All assignments not received by that date will be considered incomplete and will be given a 0.

2022W CS 1920
Computer Science 2
Programming Assignment 1
Mark Contribution: 5%

.