CS2613: Programming Languages Laboratory (FR02A)
Lab #17 – Winter 2024

Language: Python (#4)
# of Tasks:
Topics:

- Class Creation
- Instance Variables
- Methods
- Inheritance

*All tasks are to be completed individually in line with the academic offense guidelines detailed on the syllabus and are **due before the end of the lab period** unless stated otherwise.*

Task #1

Task Style: Planning and Programming (in Group of 3)
Submission Method: Move onto Task #2

Description:

For this lab, you will be working in a group of 3 to create 3 classes and a driver for them. It is recommended that you read the entire lab and spend some time planning for method names, class names, etc. to ensure that all pieces of the code will work together appropriately. Your grade for this Lab will be assigned to all members of your group.

Following the guidelines of object-oriented programming, create an abstract parent class called "TradingCard". Make sure to review resources below to understand how to make an abstract class in Python.

This class should have the following instance variables: ID, Rarity (scale of 1-10), and Year Release. All should have accessor methods. Only Rarity should have a mutator method; however, the mutator method should not allow the rarity value to be set to a value outside of the 1-10 range.

Create a to_string method that returns a string in the following format:

```
#1234567 (2021): Rarity: 1
```

Finally, include an abstract method for cost.

When complete, share your code with your group and move onto Task #2.

Resources:

- https://realpython.com/python3-object-oriented-programming/
- https://www.geeksforgeeks.org/abstract-classes-in-python/
- https://blog.enterprisedna.co/python-how-to-import-a-class/
- https://www.w3schools.com/python/ref_func_round.asp

Task #2

Task Style: Programming (Solo)
Submission Method: Move onto Task #3

Description:
Two group members will create their own sub-class to the TradingCard class while the third creates a driver. Note – costs must be rounded correctly

Partner 1: HockeyCard

Hockey Cards will have a player name and their jersey number. They will also store the number of games the player has won. None require mutators, but all require accessors.

The cost of a Hockey Card is calculated as follows:

Num Games Won * (2023 – Release Year of Card)/10 * (0.15 + Rarity/5)

The to_string method of this class should print in the following style:

```
#1234567 (2021): Rarity: 1   Cost: $XX.XX
      Player Name (#13) – 101 Games Won
```

Partner 2: PlayingCard

Playing Cards will have holographic status (true or false) as well as a condition "Mint", "Good", or "Poor". Holographic status should only have an accessor; however, condition should have both an accessor and a mutator. The mutator here should not allow the condition to be set to anything other than the three previously mentioned condition statuses.

The cost of a Playing Card is calculated by **multiplying** the condition value **by** the holographic status value.

If the card is in Mint Condition:

5.15 * (Rarity / 2)

If the card is in Good Condition:

2.15 * (Rarity / 2)

If the card is in Poor Condition:

0.5 * (Rarity / 2)

If the card is Holographic:

* 5

If the card is not Holographic:

* 1

The to_string method of this class should print in the following style:

```
#1234567 (2021): Rarity: 1   Cost: $XX.XX
      Holographic: True      Condition: Mint
```

Partner 3: Driver

Create a driver program that tests the rest of your groups' classes. This driver should create multiple different versions of each object and store them in a list. The driver should then cycle over the list and print out the total value of the cards in the list. Finally, create a function that will cycle over a list of cards and print out the details of the 3 highest valued cards.

Task #3

Task Style: Executing and Correcting Code (in Group of 3)
Submission Method: Submit to D2L Drop Box

Description:

Send all your code to the group and make sure that it works as a single program. When finished, submit to D2L. Only one group member needs to submit, but make sure all group members' names are listed at the top of your code in the comments.