

Assignment 4

Due Friday October , 15th, at Noon

Reading

Review Chapter 8 and read Chapter 9 in **Introduction to Computing using Python: An Application Development Focus, Second Edition** by Ljubomir Perković.

Logistics

In this class programming assignments may be completed in consultation with up to two other classmates. You must identify the classmates with whom you collaborate in a comment at the top of the assignment, and the number of collaborators on any assignment **may not exceed two other people**. You must also submit a comment in your submission for each assignment that describes in detail how each collaborator contributed to the assignment. If you did not collaborate with anyone on the assignment, you must include a comment that says that. You may not under any circumstances discuss the assignments with classmates' other than your identified collaborators. Working so closely with anyone other than your identified collaborators, Mr. Zoko, or the lab assistant, so as to produce identical or near identical code is a violation of the Academic Integrity policy. This policy will be strictly enforced.

Please include the following with your assignment submission:

1. A comment at the top of your Python file identifying any classmates with whom you discussed or in any other way collaborated on the assignment. You may work (directly or indirectly) with **no more than two** other people.
2. Add a comment at the top of your Python file that describes for each person what they contributed to the assignment. This must be at least 2-3 sentences and be **very specific and detailed**.

A submission that does not include a list of collaborators and a comment indicating how you collaborated with classmates will earn a 0. If you worked alone, you must put a comment at the top of your file that indicates that or you will also receive a 0. There will be no exceptions to this rule.

Again, you are subject to all the rules specified in the Academic Integrity policy. Please read it carefully before beginning this assignment.

Assignment

Begin the assignment by downloading the **csc242hw4.py** file from the D2L site. It contains a significant amount of code to get you started with the GUI classes required for this assignment.

For part 1, you must use the provided Car class. It contains all the information about a single car in the garage. Look at the example we discussed related to Animal class and Zoo. Significant points will be taken off if the Car class is not used.

1. 50 Points: Write a class **Garage** that represents a collection of Car instances. You must write the entire class yourself. A template is not included in the .py file, you must create the Container class and a custom Iterator yourself. The class supports six methods:

a __init__() the constructor

b __contains__() which takes a Car as a parameter and returns True if the Car is already in the collection and False otherwise. The function must return a Boolean and not a string. Note: The in operator uses __contains__. **Hint : Compare the Year, Make and Model of the car passed in to the Cars in the grage.**

c. getRandomCar – Returns a random Car object from the Garage

d. getCarsBasedOnMake – Returns a list of Car objects that match the make passed in as a parameter.

e. __len__ operator: returns the count of cars in the garage

e __str__() which returns a string representing the object as seen in the sample output below.

g __iter__() which returns an iterator to traverse all the Cars in the Garage.

Example Usage:

```

....
>>> g=Garage()
>>> g.add(Car('VW','Passat',2021))
>>> g.add(Car('Buick','Enclave',2008))
>>> g.add(Car('Rimac','Nevera',2021))
>>> g.getRandomCar()
Car('Rimac','Nevera','2021')
>>> g.getRandomCar()
Car('Rimac','Nevera','2021')
>>> g.getRandomCar()
Car('Rimac','Nevera','2021')
>>> g.getRandomCar()
Car('Buick','Enclave','2008')
>>> g.getRandomCar()
Car('VW','Passat','2021')

```

```

>>> g.add(Car('VW','Beetle',1966))
>>> g.getCarsBasedOnMake('VW')
[Car('VW','Passat','2021'), Car('VW','Beetle','1966')]
>>> str(g)
'The are 4 cars in the garage'
>>> c=Car('Ford','Pinto',1979)
>>> c in g
False
>>> c=Car('VW','Passat',2021)
>>> c in g
True
>>> c=Car('VW','Passat',2020)
>>> c in g
False

```

```

///
>>> len(g)
4
>>> for c in g:
        print(c)

```

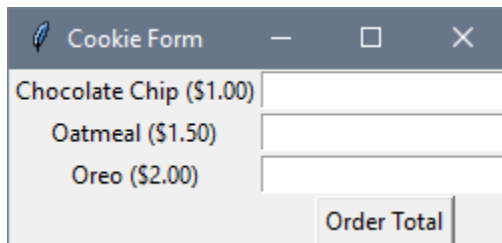
```

I am a VW Passat made in 2021.
I am a Buick Enclave made in 2008.
I am a Rimac Nevera made in 2021.
I am a VW Beetle made in 1966.

```

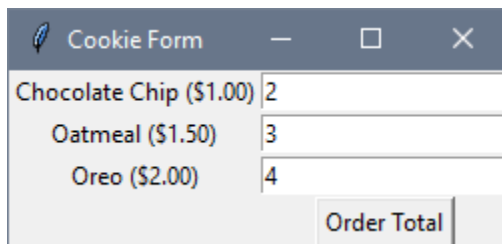
2. 50 Points. Develop a GUI named **CookieOrderForm** that can be used to calculate the total of a cookie order.

This is the default view. **You need to use grid() for layout.**

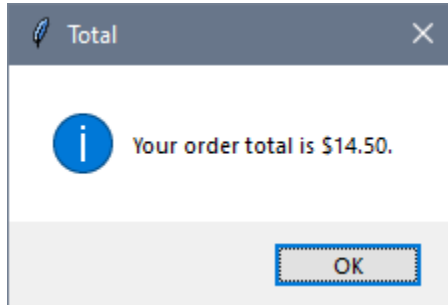


The layout consists of 3 labels, 3 entry text boxes and 1 button.

For example. When calculating the order with the following values:



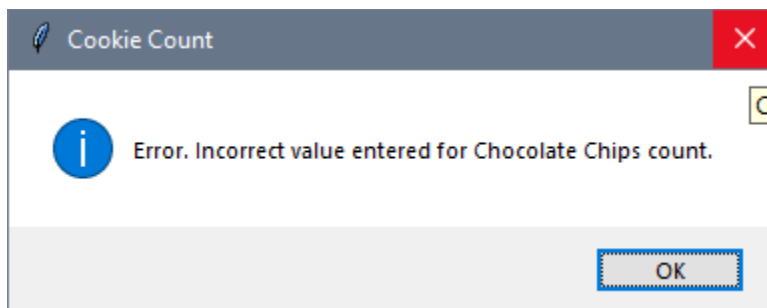
A popup shows the result. Notice formatting! I will be taking **10 points** off for incorrect format!



User enters an incorrect value:

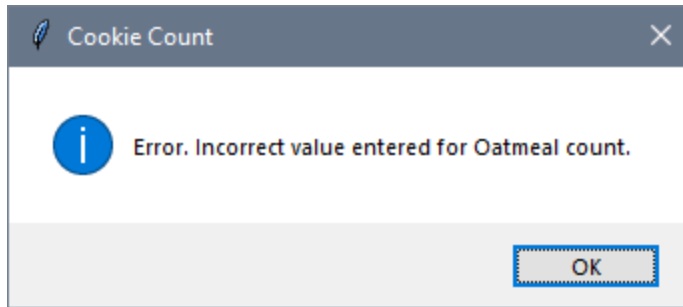
A dialog box titled "Cookie Form" with standard window controls (minimize, maximize, close). It contains three input fields: "Chocolate Chip (\$1.00)" with the value "a", "Oatmeal (\$1.50)" with the value "3", and "Oreo (\$2.00)" with the value "4". At the bottom right, there is an "Order Total" button.

They are warned that an incorrect value was entered for Chocolate Chips.



This applies to Oatmeal and Oreos as well:

A dialog box titled "Cookie Form" with standard window controls (minimize, maximize, close). It contains three input fields: "Chocolate Chip (\$1.00)" with the value "2", "Oatmeal (\$1.50)" with the value "a", and "Oreo (\$2.00)" with the value "4". At the bottom right, there is an "Order Total" button.



Note: The UI should only show one error. Should not annoy the user by constantly giving errors after the first one.

You will run the program using `CookieOrderForm ().mainloop()`.

Do not change methods provided in the template. Instead, you must fill in all the methods.

Submitting the assignment

You must submit the assignment using the assignment 4 dropbox on [the D2L site](#). Submit a Python file (csc242hw4.py) with your implementation in it and comments describing your collaboration status. Submissions after the deadline listed above will be automatically rejected by the system. See the syllabus for the grading policy.

Grading

The assignment is worth 100 points. Any student who does not submit comments in the Python file describing the contributions of each team member or indicating that he/she worked alone will earn a 0 on the assignment.