Jessica Qi
Hari Kuttivelil
CSE 20
3 December 2021

README File

GITHUB

https://github.com/jessicaqi123/DrinksClass.git

Class Documentation

Description of the Class

The class "Drinks" details a boba shop's orders based on the arguments of name of the drink, size, sweetness (in pumps of syrup), and any other add-ons or modifications that may be made to the drink. The class will format the constructor arguments, modify them, and return them when prompted. An added functionality includes finding the sugar level (in percentage) based on the pumps of sugar and size of the drink. Similar to a real world boba shop, users are able to change and view their drink orders.

Description of Each of the Class and Data Variables

shop = "Sammy's Shack"

A class variable that is shared by all object instances of the "Drinks" class, establishing that everything is under the place called "Sammy's Shack".

self.name

A data variable that stores the constructor argument for the name of the drink.

self.__size

A data variable that stores the constructor argument and/or modifications of the size of the drink.

self.__sweetness

A data variable that stores the constructor argument and/or modifications of the sweetness (in pumps of syrup) of the drink.

self.__other

A data variable that stores the constructor argument and/or modifications of the add-on/modifications of the drink.

self.__num

A data variable that stores the number that is used to modify the sweetness.

self.__newsweetness

A data variable that stores the new sweetness after adding or removing the sweetness.

self.__level

A data variable that stores the ounces of syrup given the amount of pumps of syrup.

self.__percent

A data variable that stores the percentage of syrup in the medium or large sized drink when the sweetness has a value of 5.

self.__mediumpercent

A data variable that stores the percentage of syrup in the medium sized drink when the sweetness is 0-4.

self.__largeprecent

A data variable that stores the percentage of syrup in the large sized drink when the sweetness is 0-4.

Description of Each of the Methods

__init__

A method that serves as a constructor to initialize the object's state. It takes in the drink name, size, sweetness (in pumps of syrup), and any other add-on/modification that may be made to the drink, formats them, and sets them to the data variables.

add_sweetness

This method adds the inputted number to the sweetness (in pumps of syrup). If the new sweetness number exceeds 5, it will return "5 (max level)" as the sweetness.

remove_sweetness

This method removes the inputted number to the sweetness (in pumps of syrup). If the new sweetness is either more than 5 or less than 0, it will return "5 (max level)" or "0", respectively.

change_size

This method changes the size of the drink to the inputted size.

change_other

This method changes the add-on/modification of the drink to the inputted add-on/modification.

get_size

This method returns the current size of the drink.

get_sweetness

This method returns the current sweet of the drink and provides insight to how many ounces is in one pump of syrup.

get_other

This method returns the current add-on/modification of the drink and provides suggestions for other add-ons or modifications.

sugar_level

This method computes the percentage of syrup in a drink given the drink size is medium or large. If the drink size is neither medium or large, it will return a message saying the percentage cannot be calculated given the inputted size.

__str__

A magic method that reads the outputs of the given arguments and its possible modifications.

Demo Program Demonstration

Description of the Demo Program

The demo program utilizes all the methods listed above, such as adding/removing sweetness, obtaining the size, sweetness, and other add-on/modifications, calculating the percentage of syrup in a medium and large sized drink, and retrieving certain arguments and providing suggestions. The program also includes how it handles string errors when 5 is set to "5 (max level)" instead of just the value 5, negative numbers for the sweetness, and sizes that are not either medium or large.

Instructions on How to Run the Demo Program

In order to run the demo program, you must download the code and have a preferred terminal open. Users may modify the program by hashing out some of the code to see how the program runs better. For example, hashing out everything related to "coffee" to see how "tea" works, therefore, avoiding any crowding in the terminal, making results easier to read.