CS5001 Project 8 Report

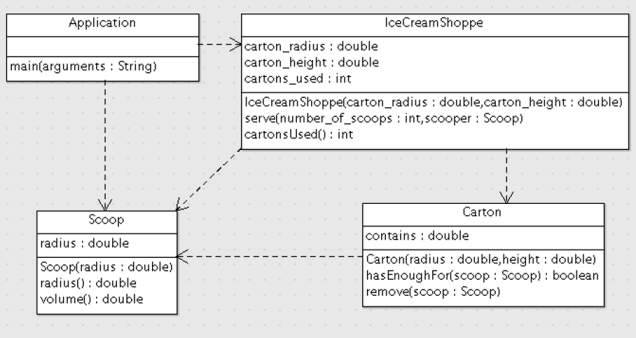
Shang Xiao

April 6, 2022

1 Problem Description

1.1 UML (Unified Modeling Language) diagram is similar to the flowchart diagrams we explored earlier in this course. In project 8, we are given a UML diagram to interpret the information within the diagram and use appropriate classes and methods to complete our code.

The project requires three classes: Scoop, Carton, and IceCreamShoppe, each of them will be explained within their section, and finally, the main function which combines these three classes and then prompts the user to enter the radius of the two Scoops, the dimensions of the Cartons, and any additional ice cream that user wants, lastly print how the amount of ice cream used.



2 Project Implementation Discussion

2.1 class Scoop first creates a new instance of a Scoop and calculates the radius and volume of a Scoop of ice cream using formula $4/3(pi)r^3$

2.2 class Carton first creates a new instance of a Carton and calculates the volume using formula $(pi)r^2h$, which is stored in the constructor function. Next, we create two methods, with first called hasEnoughFor to determine whether or not the Carton contains enough ice cream to make a Scoop, and lastly, we create a method called remove to find the Scoop to be used on the Carton.

2.3 class IceCreamShoppe first creates a new instance of an IceCreamShoppe and counts the number of Cartons used every time we create a Carton. The first method in this class is called serve, which determines the number of Scoops and the specific Scoop to use, and the second method is called cartonUsed to keep tracking the amount of cartons has been used so far.

2.4 The main function accomplishes the task required by prompting the user to enter the radius of each Scoop and the volume of each Carton, as well as if the user wants more ice cream until they are satisfied, and returns the amount of Cartons used in the end.

3 Reflection

3.1 In this module, we progressed even further than simply understanding common functions and their methods; instead, we began designing our own class and objects using self-constructed functions and methods. One of the most significant things I have learned in this module is the flexibility and possibilities of creating my own class. This means, for one unique question, there can be more than one approach, all depends on the program designer - us, to determine the appropriate functions and methods.

At first, I encountered difficulties writing the serve() function in the file IncCreamShoppe. I quickly visited the Python official documentation on class section 9.3.5 to understand the concept of 'instance variables are for data unique to each instance and class variables are for attributes and methods shared by all instances of the class', and realized that I could import the class method which I had previously built in hasEnoughFor in class Carton.

Another huge takeaway that I want to reflect on is this module has taught me the essential part to complete my final project, which is on the Guitar chords diagram. I can now start building the logical blocks of my code and to explore other possible classes and objects to structure the guitar program. This module is also the very first step to moving into Object-Oriented Programming.

4 Acknowledgements

4.1 Website consulted (lecture notes and documentations can be found in these links):

`https://docs.python.org/3/tutorial/classes.html` - Official Documentation for classes and objects

`https://www.geeksforgeeks.org/constructors-in-python/` - Constructors

`https://www.youtube.com/watch?v=RSl87lqOXDE` - This YouTube video made by Corey really solved my questions while answer the last question (question 5) in the module quiz on inheritance.

`https://docs.python.org/3/tutorial/classes.html` - Section 9.5 explains the concept of override on subclass inheritance.

Website used for debugging codes and loops for each line execution:
`https://pythontutor.com/visualize.html#mode=edit`