The Food Class

In 2019, diet-related diseases made up 3 of 7 leading causes of death in the US in 2019 [1]. This is largely due to having a food culture which is particularly dense with fats and carbs (e.g. sugars) and deficient in protein – but what constitutes "dense" or "deficient"? We see nutrition labels all the time, but we are never taught the significance of these values.

The purpose of this lab is to build an understanding of object-classes as a way to *organize and synthesize data* through the lens of nutritional health and provide a baseline understanding of how to gauge the relative health of any one food item. To do so, we will build a Food class to generate useful nutritional data and help us assess the health of any product in terms of its *caloric composition*.

A precursory note: it is not a good idea to label food or diets as inherently *good* or *bad*; rather we should identify if a food is better or worse in terms of helping us get closer to meeting our nutritional needs. French fries aren't bad, but since they don't help us achieve our nutritional goals, eating them every day would be a bad choice!

The calories of a single food item are provided at the top of a nutrition label can be *roughly* calculated by from its macronutrients as summarized to the right:

1g of:	Cals per gram
Proteins	4 cals
Carbohydrates	4 cals
Fats	9 cals

The nutrition label to the right is for a half pound of chicken breast. Fill in the table below given the nutrition values to the right:

Macros:	Grams	Calories	%'s (by Calories)
Proteins:			
Carbs:			
Fats:			
Total:	▨		

Nutrition Facts

For a Serving Size of 1 Serving

Calories ______	Calories from Fat ______
Total Fat 17g	-
Saturated fat 3g	-
Carbohydrates 1g	-
Net carbs 1g	-
Sugar 0g	-
Fiber 0g	0%
Protein 76g	

Do the same for a pack of Trolli gummy worms (note the total servings in the nutrition label):

Nutrition facts

6 servings per container

Serving Size	29
Amount per serving	
Calories	____
Total Fat 0g	
Saturated Fat 0g	
Trans Fat 0g	
Total Carbohydrate 25g	
Dietary Fiber 0g	
Sugars 14g	
Includes Added Sugars 14g	
Protein 1g	

Macros:	Grams	Calories	%'s (by Calories)
Proteins:			
Carbs:			
Fats:			
Total:	▨		

1 - Xu JQ, Murphy SL, Kochanek KD, and Arias E. Deaths: Final data for 2019. National Vital Statistics Reports, vol 70 no 08. Hyattsville, MD: National Center for Health Statistics. 2021. DOI: https://dx.doi.org/10.15620/cdc:106058

Create the Food class that has the following:

Fields: name, proteins, carbs, fats, calories, servings

Constructors:

1. Create a default constructor that takes in the food's name, carbs, proteins, fats, calories, and servings.
2. Create an overloaded constructor that does the same as above but calculates the number of calories per serving.
3. Create an overloaded constructor that is the same as the default constructor, but takes in a number of calories sets the number of servings to 1. (Note: Do the servings per container have to be whole numbers?)
4. Create an overloaded constructor that does the same as above but sets the number of servings to 1.

Methods:

toString() – overwrites the toString() method so that printing out an object of the food class will return the following string:

- "[FoodName] nutrient values:
  - Calories: [calories] cals
  - Servings: [servings]
  - Proteins [proteins]g
  - Carbs [carbs]g
  - Fats [fats]g

calcMethods – create methods that calculate and stores each of the following totals for *all* macros. You may need to create new fields for each macros total:

- total grams of each macro
- the total number of calories from each macro
- the percents of each macro as a portion of the total number of calories (rounded to the nearest 100th).
  - Write a new instance variable for percentString that is set equal to a string of the following format: Ratio: P:25% C:34% F:41%

Is404020() – In this lab, a food item is considered "lean" if maintains a ratio of:

- 40% Proteins, 40% Carbs, and 20% fats (as portions of calories)
- Given this rigid criteria however, we will allow our food to still be considered lean if each field is within 10% of the specified ratio:
  - e.g. 35-35-30 will still be considered "lean"
- Update the toString method so that it displays if the item is Lean and what its ratio is
- This is because the idea of "40-40-20" is to maintain this as a daily average rather than something we necessarily need to consider on a *per meal* basis. This allows us to have more of a "ballpark" intuition as opposed to a rigid precision that's hard to follow.
- Write the is404020 method so that it sets a field *isLean* equal to true or false upon being called.

calcAll() – a method that simply calls each of the calcMethods and is404020 described above.

- When this is complete, call this method at the end of each constructor.

Void getters – For each calc method, write a comparable void method that prints out and formats the values generated using the following format:

- "[FoodName] percents:
    - Proteins: [pctProts]%
    - Carbs: [pctCarbs]%
    - Fats: [pctFats]%

Getters and setters for ***every*** field (do this last):

- To generate getters and setters for every field, simply press *alt+insert* and select the proper menu option.

Data Collection:

Fill out the table below with 10 items that are in your pantry and 10 items from your fridge on the left side of the tables below. If the ingredient label is not listed you may need to do some research to fill in the values (e.g. you have 6 apples, so you look up the nutritional values for that type of apple and list 6 for servings).

When you are done, put the data into a notepad file with each row on a new line. Then loop through the file to produce the data on the right side of the table and fill in the results accordingly.
Note: you may need to use underscores in place of spaces for this to work.

From the pantry:									
Name	Servs:	Cals	P (g)	C (g)	F (g)	%P	%C	%F	Lean?

From the fridge:									
Name	Servs:	Cals	P (g)	C (g)	F (g)	%P	%C	%F	Lean?

Submit the lab when you are finished! You should turn in the physical copy of the lab with your data collected in the table. Additionally, you should turn in a ***zip*** folder with your Food class, main method, and notepad data file.

Do not lose these files. The goal is to later use this as a base for learning much of what is left to be covered in the remainder of the year (arrays, ArrayLists, super classes, and GUIs).