Before you begin, read and understand the exam rules and guidelines

1. Work independently
   * You are not allowed to communicate with anyone during the exam
2. No copy of this exam document shall be shared with anyone or posted online
3. Understanding the requirements is part of the exam
4. Your **program must compile**
   * If the program does not compile, you would automatically get **0% on the exam**.
   * Remove/Delete any code that makes the program uncompilable.
5. Avoid unnecessary codes
   * Any code unrelated to the requirement would be considered unnecessary and useless.
   * **- .5 marks for each line of unnecessary and useless codes**
6. Cite the source(s)
   * If the code was copied from the internet, provide the site (as comment in your program)
     1. Example: if a loop was copied from a website, then on top of the loop you write the source

/\* <http://someWebsite.com/> \*/

// the loop

* + If the code was copied from a book, provide the book’s title, author, and page (as comment in your program)

1. Be aware of the deadline
   * Make sure that you **submit all the requirements before the deadline**. If the exam ends at 14:20, all documents should be on D2L on or before 14:20.
   * **2% deduction for every minute late or fraction of it.** 
     1. If the timestamp says 14:22:23 (hours:minutes:seconds), this means 6% deduction (2 minutes and 23 seconds late = 3 minutes)
2. Rename this document: **CPSC1160\_midterm\_yourStudentid.docx**
   * You need to submit this document at the end of the exam
   * Failure to submit: **5% deduction**
3. An open brackets **[ ]** would be provided for each requirements. Put an “F” inside (e.g. **[F]**) if you fulfill/implement the requirement without any errors. **[“P”]** if partially implemented. You are **responsible to provide sufficient proof that your code works**.

|  |  |
| --- | --- |
| Student name | Jay Seung Yeon Lee |
| Student id | 100357736 |

Given Data: foodAndCalories.csv (do not alter the data inside this .csv)

There are 3 fields: food, serving, and calories

# of rows: 31 (including the headers)

Application name: diet\_yourStudentid.cpp

Executable file: mydietaryplan

Write a C++ application with the following features:

1. Create a function that would read the given data file and store the information in memory using classes
   1. Store the header in an array. The header will always contain 3 strings [1 mark][F ]
   2. Create a class to represent 1 food element.
      1. Class name: Food\_yourStudentid
      2. Files: Food\_yourStudentid.h and Food\_yourStudentid.cpp
      3. Private fields: foodname , serving, and calories
      4. Create the setters and getters.
   3. Create an array of Food\_yourStudentid to store/hold the collection of food read from the given data file. There are 30 of them. [4 marks][ F ]
   4. The read should be done once the program starts
   5. Implement operator-= (int calories) [2 marks][F ]
      1. The value passed would be subtracted from the calories
      2. Example:

Assume food[1] is Apricot and has 17 cal

food[1] -= 20 ;

This would change the value of calories to 0 cal.

If the value passed is greater than the current value, the result would be 0.

Otherwise, just do 17 – value passed

1. Menu [1 mark][ F ]
   1. Use loop to continuously display the loop until the user wants to exit.

Your name

1. Display all – sort by food name (descending order)

2. Display the food with lowest calories

3. meal plan

4. Exit

1. Display all – sort by food name (descending order) [3 marks][ P ]
   1. Use Quick Sort
   2. Create a function that would display in descending order, based on the food name, the contents of your food\_yourStudentId (array). The output to the console should be well formatted and easy to read.
2. Display the food with lowest calories [2 marks][P ]
   1. Do not hardcode the output
   2. In the given data file, the answer would be

Acerola 1 acerola (4.8 g) 1 cal

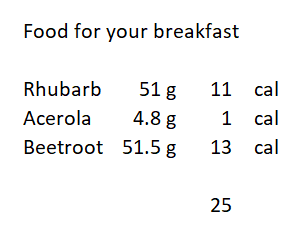
Note: If your program is fed with a different data, the result might not be the same.

1. meal plan
   1. Ask for user input (meal, max calories) [1 mark][F ]
   2. display the suggested food along w/ the serving (g or ml only) and cal [6 marks][ ]
      1. the total calories must match the entered max calories
         1. you can increase or decrease the servings to meet the requirements
      2. it can be any combination of the food in your collection
         1. there should be at least 3 foods mentioned

Sample run

Meal (breakfast, lunch, dinner): breakfast

Max calories: 25



Beetroot 1 beet (82 g) 35 cal

Since we only need 13 cal, the suggested serving is approx. 51.5 g

What to submit

1. All .h and .cpp files
2. A working Makefile
3. .csv file

Put all the mentioned files in 1 zip file, Midterm\_yourStudentID.zip