COMP 1020
Lab 6

MATERIAL COVERED

- File reading with BufferedReader and FileReader and text parsing with Scanner
- Subclasses and object hierarchies

Notes:

- You should definitely do at least the Bronze and Silver exercises. You can do these
  - before the lab, on your own, using any Java programming environment you like, or
  - during the lab, with assistance from the lab TA, or
  - after the lab, on your own.
- For credit, you must demonstrate at least one working exercise (Bronze), in the lab.
- Make sure your TA has recorded your mark before leaving.
- The three questions are sequential – each builds on the previous one.
- Always try to complete as many exercises as you can on every lab.

You will be adapting your **`VendingMachine`** class to also provide nutritional information.

Download a **`csv`** file from Health Canada and adapt your Lab project to import various nutritional information.

This is a typical format used for data and is easy to understand. You will have to skip the first few lines of the file, then you will reach the headings (highlighted in yellow). It also appears that there are some subtitles in there which may mess things up. You don't need to detect these other lines programmatically (although it would be a good challenge) however just write your code to read in one specific file (for now).

The Canadian Nutrient File: Nutrient Value of Some Common Foods
Health Canada.
https://open.canada.ca/data/en/dataset/a289fd54-060c-4a96-9fcf-b1c6e706426f

From the Baked Goods for example:

Nutrient Value of Some Common Foods - 2008 Edition,,,,,,,,,,,,,,,,,,,,
BAKED GOODS,,,,,,,,,,,,,,,,,,,,
Food name,Measure,Weight ,Energy,Energy,Protein,Carbohydrate,Total Sugar,Total Dietary Fibre,Total Fat,Saturated Fat,Cholesterol,Calcium,Iron,Sodium,Potassium,Magnesium,Phosphorus,Thiamin,Riboflavin,Niacin,Folate
"BISCUITS, CROISSANTS, AND MUFFINS",,,,,,,,,,,,,,,,,,,,
"Biscuit, plain or buttermilk, fast food",1,51,186,776,3,25,2,0.7,8,1.3,1,25,1.7,537,114,9,219,0.2,0.15,2.4,58
"Biscuit, plain or buttermilk, from mix, baked",1,30,97,404,2,14,3,0.4,4,1,2,54,0.6,273,55,7,136,0.1,0.11,1.4,31

Tip: Remember you can use Scanners useDelimiter method to set a new delimiter to separate the columns, or you could use String methods (which we didn't cover in class)

Bronze:

Read in the file and store an array of **`Food`** objects that represents the lines in the file. You should extract the **`Food name`**, **`Weight`**, **`Protein`**, **`Carbohydrate`** and **`Total Sugar`** values. Do some research on the website to figure out the units of measurements.

Silver:

You should then make a subclass of **`Food`** to hold the particular type of food you chose (such as Beverages, or Baked Goods). Look at some other files to compare what is the same and what is different between them. Next, update your code to import the Vitamins as well (a few of them at least) such as Calcium, Iron, Sodium or Potassium (or others). Each different **`Food`** type has different vitamins and measurements. Which ones are common to all Foods (and should be in the **`Food`** class) and which ones are specific to the type you are reading in (and should be in the subclass you created)

Gold

You obviously have a knack for this so finish the job. Import two more food files into two separate subclasses of **`Food`** (that you create). Update your **`VendingMachine`** methods to work with these new subclasses. What do you need to change? What can stay the same? This will require some foresight so plan your strategy carefully before you start coding.