Pre-Lab 5 Exercise

CS 106: Introduction to Data Structures
Week 5

Note: this should be completed before starting Lab 3. You should begin this exercise *before* lab on Friday Feb 21 – it is okay to complete it during lab (you should be checked off by the end of lab).

Goals:

- Practice working with command line inputs

Review: The `main` method is special but still has many similar behaviors as others you write. Anybody can pass arguments (a.k.a. parameters) into the method using command-line arguments. In Eclipse, this can be done by `Run` → `Run Configurations…` → `arguments` tab. Enter the arguments you would like to pass into the `main` method, separated by spaces. The arguments will then exist in a String array called `args` in the order entered.

The lab:
Imagine a produce stand that carries vegetables and fruits. Create a new package in your Lab 3 (right click on src/main/java) called "prelab". Then create a new class: "FarmInventory.java." In this file, space-separated command line arguments will be passed in with respective tag preceding each item. Assume that the arguments will be a tag, followed by the item representative of the tag. Create a `main` method for the following code.

Implement the following in "FarmInventory.java"

a.) In a `for` loop, add the vegetable or the fruit passed in as command-line arguments into two ArrayLists, one for fruit and one for vegatables.

i.) The ArrayLists of vegetables and fruits can be initialized with:

```
ArrayList<String> fruits = new ArrayList<String>();
ArrayList<String> veggies = new ArrayList<String>();
```

ii.) Each item will be passed in with `-v` or `-f` tag based on whether it is a veggie or fruit. Remember that arguments passed in will be in the String array, `args`. Use .equals to compare Strings.

iii.) For example, if `-v lettuce -f apple` are passed in, `args` will look like the following: `args = {"-v", "lettuce", "-f", "apple"}`.

iv.) Use a for loop to go through all the command line arguments:

```
for (int i=0; i < args.length; i++) {
```

b.) Run the program with these arguments:

```
-f tomato –v carrot –v onion -v spinach -f
pineapple –v asparagus
```

The output look something like the following:

```
The set of veggies are: [carrot, onion, spinach, asparagus]

The set of fruits are: [tomato, pineapple]
```

c.) Add “-a” tag which is followed by a number, to record the fruit stand’s assets. The value followed by the tag should be added to a `totalAssets` variable.

d.) To the arguments described in part (b), add these arguments:

```
-a 12.50 –a 42
```

The output should be something like the following:

```
The set of veggies are: [carrot, onion, spinach, asparagus]

The set of fruits are: [tomato, pineapple]

The total asset is 54.50
```