1 Objectives

In this lab you will focus on the following objectives:

1. Explore more complex array operations in Java

2 Tasks

1. Create a new Java project and call it “Lab 5”.

2. You will write a program that lets a user keep track of their homemade salsa sales!

   (a) Use three 5-element arrays to track the following:
   
      i. The salsa names: mild, medium, sweet, hot, and zesty
      ii. The number of each type of salsa sold
      iii. The price of each type of salsa

   (b) Use initialization lists to store the names and prices into the appropriate arrays at the start of the program.

   (c) Then prompt the user to enter the number of each salsa sold this month.

   (d) Finally, display a summary of the total sales report:

      i. Show the gross amount of money made (before tax).
      ii. Calculate how much the user owes in sales tax (6% of the amount made).
      iii. Then display the net profit (after subtracting the tax).

3. Here is an example output of the program (yours does not have to match exactly):

   Welcome to your salsa profit calculator!

   Enter how many Mild salsas were sold (at $2.25 each): 12
   Enter how many Medium salsas were sold (at $3.00 each): 4
   Enter how many Sweet salsas were sold (at $3.50 each): 0
   Enter how many Hot salsas were sold (at $4.00 each): 9
   Enter how many Zesty salsas were sold (at $4.50 each): 7

   Processing...

   Here is your monthly sales report:
Mild:  12 x 2.25 = 27.0
Medium: 4 x 3.00 = 12.0
Mild: 0 x 3.50 = 0.0
Hot:  9 x 4.00 = 36.0
Zesty: 7 x 4.50 = 31.5
------------------------------
Gross:  106.50
Tax:  (6.39)
------------------------------
Net:  100.11

You made $100.11 this month!

4. Some tips:

(a) You can (optionally) specify formats for decimals by using the `System.out.format` function. For example, to print the variable `gross` with two decimals, you can do: `System.out.format("%.2f", gross);` which will use the format string "%.2f" to indicate you are printing a floating point number (the f) with two decimals (indicated by .2). Any place the format string has the percent sign (%), you will need to pass an extra argument (separated by commas) to the function. We'll be learning a lot more about function soon!

(b) You can help align your output into columns by again using the formatted output. But this time, the format string will have `%Ns` somewhere inside to indicate that the thing you're outputting should take up exactly $N$ spaces where $N$ is an integer. In this example, we set columns of length 20:

```
System.out.format("%-20s %5d\n", "test", 1);
System.out.format("%-20s %5d\n", "test2", 20);
System.out.format("%-20s %5d\n", "test3", 5000);
```

// will output

test 1
test2 20
test3 5000

(c) You can find a lot more information on Java formatted strings here: [https://docs.oracle.com/javase/7/docs/api/java/util/Formatter.html](https://docs.oracle.com/javase/7/docs/api/java/util/Formatter.html)

3 Submission

Be sure to test different situations, show how the program handles erroneous input and different edge cases. Upload your project files to MyClasses as a single .java file.