Lab 03: Generic Cheese Shop
CSE-021-01 **Points: 20**

Overview

We will review the following statements in this lab:

- Scanner
- Variables (access types)
- System.out.print/println/printf
- Conditional statements (if, else, else if)
- Arrays
- Loops

Getting Started

After starting Eclipse, create a new project called **Lab21_3**. Import `Error.java` from the assignment page.

Examine at the code given in `Error.java`. It will run as is, so go ahead and run it. You should see the following output (prices per pound for cheese types D – J might be different than shown):

```
We sell 10 kinds of Cheese
Humboldt Fog: $25.0 per pound
Red Hawk: $40.50 per pound
Teleme: $17.25 per pound
Cheese Type D: $9.15 per pound
Cheese Type E: $2.5 per pound
Cheese Type F: $8.74 per pound
Cheese Type G: $9.88 per pound
Cheese Type H: $2.91 per pound
Cheese Type I: $6.66 per pound
Cheese Type J: $0.36 per pound
```

Let us analyze the code to figure out how it is working. To begin with, we see a new kind of declaration for a variable:

```
final int MAXCHEESE = 10;
```

Final (`final`) is an access modifier that prevents the variable's value from changing after its initialization. Such an initialized variable, called a ***constant variable***, is typically used when we want a value that cannot be accidently overwritten in the rest of the program. Thus, the value of a constant variable is set only once [cf. Section 2.9].

Next we declare 3 arrays:

```
String[] names = new String[MAXCHEESE];
double[] prices = new double[MAXCHEESE];
int[] amounts = new int[MAXCHEESE];
```

Next, we set values corresponding to Humboldt Fog cheese (from Lab 02) as the first entry in the `names` and `prices` arrays:

```
names[0] = "Humboldt Fog";
prices[0] = 25.0; ...
```

The code repeats for the other two special cheeses. Note the meaningful variable names that self-describe their purpose [cf. Section 2.3]. The program then prints out how many types of cheese are currently being sold, and a list of names and prices:

```
System.out.println("We sell " + MAXCHEESE + " kinds of Cheese");
System.out.println(names[0] + ": $" + prices[0] + " per pound");
System.out.println(names[1] + ": $" + prices[1] + " per pound");
System.out.println(names[2] + ": $" + prices[2] + " per pound");
```

The code so far handles the first three cheeses. But this time, we want to sell up to `MAXCHEESE` (=`10`) cheeses. We assume the remaining cheeses besides the first three are generic cheeses whose names and prices aren't too important. We first need to come up with a way to set the prices of these generic cheeses. Instead of having predetermined prices, we decide to set them randomly at the beginning. To do this we use a random number generator object [cf. Section 2.18] that will help us (like a Scanner) generate the prices. And to ensure that the Random object generates the same random sequence of integers each time the program is run (so that our randomly chosen prices stay the same), we specify a seed -- the input to the equation that generates the random values -- of 100. This is done as follows:

```
Random ranGen = new Random(100);
```

We can now output information about the rest of the cheeses with the following for-loop.

```
for (int i = 3; i < MAXCHEESE; i++) {
   names[i] = "Cheese Type " + (char)('A' + i);
   prices[i] = ranGen.nextInt(1000)/100.0;
   amounts[i] = 0;
   System.out.println(names[i] + ": $" + prices[i] + " per pound");
}
```

In the code above, the name for a generic cheese is created from a typecast [cf. Section 2.12] of an integer to a character with the `(char)(…)` command. The typecast takes the expression `('A' + i)`, which first adds `i` to the ***value*** of character `A` (65) [cf. Section 2.14] due to an implicit type conversion of `char` to `int`. So, `A + 3` (notice, `i`'s initial value is `3`) results in 68, which is cast to a `char` to get the character `D`. Hence, you see the `Cheese Type D` in the console output. We then set the price by first calling `ranGen.nextInt(1000)` that generates an integer between 0 and 999, then dividing the random number by 100.0 to get a floating-point number between 0 and 9.99 that is set as the price. The next statement initializes the order amounts to 0. And finally, the print statement outputs the name and price of each of the generic cheeses.

Now CHANGE the line `final int MAXCHEESE = 10;` to `final int MAXCHEESE = 0;`. You should see the following error:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0
	at Lab21_3_Error.main(Error.java:14)
```

The error is in line 14 of the file. It is of the error type "out of bounds" when indexing the array. You can click on the line 14 link and Eclipse will highlight the error for you. If not, the following line causes the error:

```
names[0] = "Humboldt Fog";
```

The reason is that when `MAXCHEESE` is 0, the arrays are of size 0, and access in line 14 to the first entry of the array at index 0 fails. So, think about the CONDITION that has to be true BEFORE we should access the array, and do the following exercise.

Part 1: Fix `Error.java`

Fix the code so the program will run correctly for `MAXCHEESE` values of 0 to 20 (inclusive). Note that the value of `MAXCHEESE` is set by changing the value in the code itself. If you are not sure of how it should work then look at the **Sample Runs** of the next part. This part handles the beginning where it lists all the cheese types available and their prices. Note: it is a very simple fix that needs to be added to all the statements that have an array access.

Part 2: Create `GenCheeseShop.java`

The machine should dispense ANY types of cheese, in **half-pound packages**. Your program will STILL do the following:

A. List all the cheese types available and the prices
B. Asks the user how many pounds of each type of cheese to purchase in 0.5 lb increments.
C. Calculate Original Sub Total (price*amount of each cheese added together)
D. Calculate discounts based on how many pounds for Humboldt Fog and Red Hawk cheeses the user entered, as well as the total purchase amount
   - **See Discount Calculation below for details**
E. Ask the user if they would like to see a list of what they purchased
   - If yes, a list comes up showing how much of each type of cheese they bought and the cost of each cheese. Display only the cheese they actually bought
   - If no items are purchased, then display a message stating the same (see Sample Run 4)
   - If the user answers no, then no itemized information is displayed
F. Display Original Sub Total, Specials and New Sub Total, Additional Discount and Final Total as shown in the Sample Runs below (see Discount Calculation below to see how these are calculated).

Note, you must use printf to format your output [cf. Section 9.3 in zyBooks]. For instance, you will use the following print statement to print costs of individual items in your itemized list:

```
System.out.printf("%.1f lb of %s @ $%.2f = $%.2f\n", amounts[i], names[i],
                                                prices[i], prices[i]*amounts[i]);
```

Discount Calculation

For this lab, there are two types of discounts. As before, the first discount is based on the amounts of Humboldt Fog and Red Hawk bought, with the following specials offered:

- Humboldt Fog: Buy 1 package get 1 package free
- Red Hawk: Buy 2 packages get 1 package free

See the previous lab assignment for details on how to calculate these discounts. Assuming the user buys 4.5 lb of Humbolt Fog and 4.5 lb of Red Hawk and no Teleme, the itemized list and discount calculation for this order will appear as follows:

```
Display the itemized list? (1 for yes) 1
4.5 lb of Humboldt Fog @ $25.00 = $112.50
4.5 lb of Red Hawk @ $40.50 = $182.25

Original Sub Total:              $294.75
Specials...
Humboldt Fog (Buy 1 Get 1 Free): -$50.00
Red Hawk (Buy 2 Get 1 Free):     -$60.75
New Sub Total:                   $184.00
```

Here the values are as follows:

- Original Sub Total is the price*amount of each cheese added together
- $50 discount for Humboldt Fog corresponds to the 2 lb (or 4 pkgs) in the order that were free
- $60.75 discount for Red Hawk corresponds to the 1.5 lb (or 3 pkgs) in the order that were free
- New Sub Total = Original Sub Total – (Humboldt Fog discount +Red Hawk discount)

After obtaining the New Sub Total, the Additional Discount is calculated as follows:

- A 10% discount if New Sub Total is over $150
- An additional 15% discount if New Sub Total is over $250

And finally, the Final Total is obtained by subtracting the Additional Discount from the New Sub Total.

Sample Runs (user input shown in green, with each run separated by a dashed line):

```
-----------------------------------------------------------------------------SAMPLE RUN 1
 Enter the number of Cheeses for shop setup: 0

 We sell 0 kinds of Cheese (in 0.5 lb packages)


Original Sub Total:                $0.00
Specials...
None                              -$0.0
New Sub Total:                     $0.00
Additional 0% Discount:           -$0.0
Final Total:                       $0.00
-----------------------------------------------------------------------------SAMPLE RUN 2
Enter the number of Cheeses for shop setup: 1

We sell 1 kinds of Cheese (in 0.5 lb packages)
Humboldt Fog: $25.0 per pound

Enter the amount of Humboldt Fog in lb: 1

Display the itemized list? (1 for yes) 1
1.0 lb of Humboldt Fog @ $25.00 = $25.00

Original Sub Total:                $25.00
Specials...
Humboldt Fog (Buy 1 Get 1 Free): -$12.50
New Sub Total:                     $12.50
Additional 0% Discount:           -$0.0
Final Total:                       $12.50
-----------------------------------------------------------------------------SAMPLE RUN 3
Enter the number of Cheeses for shop setup: 2

We sell 2 kinds of Cheese (in 0.5 lb packages)
Humboldt Fog: $25.0 per pound
Red Hawk: $40.5 per pound

Enter the amount of Humboldt Fog in lb: 4.5
Enter the amount of Red Hawk in lb: 4.5

Display the itemized list? (1 for yes) 1
4.5 lb of Humboldt Fog @ $25.00 = $112.50
4.5 lb of Red Hawk @ $40.50 = $182.25

Original Sub Total:                $294.75
Specials...
Humboldt Fog (Buy 1 Get 1 Free): -$50.00
Red Hawk (Buy 2 Get 1 Free):     -$60.75
New Sub Total:                     $184.00
Additional 10% Discount:          -$18.40
Final Total:                       $165.60
-----------------------------------------------------------------------------SAMPLE RUN 4
Enter the number of Cheeses for shop setup: 3

We sell 3 kinds of Cheese (in 0.5 lb packages)
Humboldt Fog: $25.0 per pound
Red Hawk: $40.5 per pound
Teleme: $17.25 per pound

```

```
Enter the amount of Humboldt Fog in lb: 0
Enter the amount of Red Hawk in lb: 0
Enter the amount of Teleme in lb: 0

Display the itemized list? (1 for yes) 1
No items were purchased.

Original Sub Total:               $0.00
Specials...
None                             -$0.0
New Sub Total:                    $0.00
Additional 0% Discount:          -$0.0
Final Total:                      $0.00
-------------------------------------------------------------------------------SAMPLE RUN 5
Enter the number of Cheeses for shop setup: 4

We sell 4 kinds of Cheese (in 0.5 lb packages)
Humboldt Fog: $25.0 per pound
Red Hawk: $40.5 per pound
Teleme: $17.25 per pound
Cheese Type D: $9.15 per pound

Enter the amount of Humboldt Fog in lb: 2.3
Invalid input. Enter a value that's multiple of 0.5: 2.5
Enter the amount of Red Hawk in lb: 3.5
Enter the amount of Teleme in lb: 0
Enter the amount of Cheese Type D in lb: 1.7
Invalid input. Enter a value that's multiple of 0.5: -1.5
Invalid input. Enter a value >= 0: 1.5

Display the itemized list? (1 for yes) 1
2.5 lb of Humboldt Fog @ $25.00 = $62.50
3.5 lb of Red Hawk @ $40.50 = $141.75
1.5 lb of Cheese Type D @ $9.15 = $13.73

Original Sub Total:               $217.98
Specials...
Humboldt Fog (Buy 1 Get 1 Free): -$25.00
Red Hawk (Buy 2 Get 1 Free):     -$40.50
New Sub Total:                    $152.48
Additional 10% Discount:         -$15.25
Final Total:                      $137.23
--------------------------------------------- ---------------------------------SAMPLE RUN 6
Enter the number of Cheeses for shop setup: 10

We sell 10 kinds of Cheese (in 0.5 lb packages)
Humboldt Fog: $25.0 per pound
Red Hawk: $40.5 per pound
Teleme: $17.25 per pound
Cheese Type D: $9.15 per pound
Cheese Type E: $2.5 per pound
Cheese Type F: $8.74 per pound
Cheese Type G: $9.88 per pound
Cheese Type H: $2.91 per pound
Cheese Type I: $6.66 per pound
Cheese Type J: $0.36 per pound

Enter the amount of Humboldt Fog in lb: 1
Enter the amount of Red Hawk in lb: 2
Enter the amount of Teleme in lb: 3
Enter the amount of Cheese Type D in lb: 0
Enter the amount of Cheese Type E in lb: 0
Enter the amount of Cheese Type F in lb: 0.3
Invalid input. Enter a value that's multiple of 0.5: 4
Enter the amount of Cheese Type G in lb: 5
Enter the amount of Cheese Type H in lb: 0
Enter the amount of Cheese Type I in lb: 0
Enter the amount of Cheese Type J in lb: 6

Display the itemized list? (1 for yes) 0

Original Sub Total:               $244.27
Specials...
Humboldt Fog (Buy 1 Get 1 Free): -$12.50
Red Hawk (Buy 2 Get 1 Free):     -$20.25
New Sub Total:                    $211.52
Additional 10% Discount:         -$21.15
Final Total:                      $190.37
-------------------------------------------------------------------------------SAMPLE RUN 7
Enter the number of Cheeses for shop setup: 8

We sell 8 kinds of Cheese (in 0.5 lb packages)
Humboldt Fog: $25.0 per pound
Red Hawk: $40.5 per pound
Teleme: $17.25 per pound
Cheese Type D: $9.15 per pound
Cheese Type E: $2.5 per pound
Cheese Type F: $8.74 per pound
Cheese Type G: $9.88 per pound
Cheese Type H: $2.91 per pound

Enter the amount of Humboldt Fog in lb: 4
Enter the amount of Red Hawk in lb: 5
Enter the amount of Teleme in lb: 0
Enter the amount of Cheese Type D in lb: 0
Enter the amount of Cheese Type E in lb: 0
Enter the amount of Cheese Type F in lb: 3.7
Invalid input. Enter a value that's multiple of 0.5: 3.2
Invalid input. Enter a value that's multiple of 0.5: 3.5
Enter the amount of Cheese Type G in lb: 5.5
Enter the amount of Cheese Type H in lb: 0

Display the itemized list? (1 for yes) 1
4.0 lb of Humboldt Fog @ $25.00 = $100.00
5.0 lb of Red Hawk @ $40.50 = $202.50
3.5 lb of Cheese Type F @ $8.74 = $30.59
5.5 lb of Cheese Type G @ $9.88 = $54.34

Original Sub Total:               $387.43
Specials...
Humboldt Fog (Buy 1 Get 1 Free): -$50.00
Red Hawk (Buy 2 Get 1 Free):     -$60.75
New Sub Total:                    $276.68
Additional 25% Discount:         -$69.17
Final Total:                      $207.51
```

Part 3: (Assessment) Logic Check for `Error.java` and `GenCheeseShop.java`

Create a Word document or text file named `Part3` that contains answers to the following:

1) Answer the following questions about `Error.java`
   a. What happens if you add the following line to the program after the declaration of `MAXCHEESE`, where it is set to `10` (in line 7):

```
MAXCHEESE = 20;
```

b. What are the data types of the arrays `names`, `prices` and `amounts`?

c. How many entries get created for each of the arrays, `names`, `prices` and `amounts`?

d. State whether the following statements are valid or invalid:

```
names[3] = 1;
prices[3] = 1;
amounts[3] = 1;
```

e. Give the output of the following statements:

```
System.out.println("Cheese " + 'A' + 10);
System.out.println("Cheese " + (char)'A' + 10);
System.out.println("Cheese " + (int)'A' + 10);
System.out.println("Cheese " + (char)('A' + 10));
System.out.println("Cheese " + (int)('A' + 10));
```

f. Why is the initial value of `i = 3`?

2) What parts did you copy from Lab 02 (A-F)?
3) What parts did you copy from fixed `Error.java` (A-F)?
4) How many loops did you use in `GenCheeseShop.java`?
5) What types of loops did you use in `GenCheeseShop.java`?

Specification Compliance

The following are some additional instructions to make sure your project complies with specifications:

1. Your program must produce an output that **exactly resembles the Sample Output shown below, including identical wording of prompts, spacing, input locations, etc**.
2. Your program must ensure the following:
   a. Your program should not display items that are not bought (amount is 0) when listing them out [cf. Sample Run 7, Lines 25-28]. If no items are purchased, then a special message is displayed [cf. Sample Run 4, Line 13].
   b. If no discounts are applied, a special message is displayed [cf. Sample Run 4, Line 17].
3. Your program must check for invalid inputs when entering the amounts [cf. Sample Run 5, Lines 9-15].
4. Before you submit, in Eclipse, type CTRL-A (to select everything) followed by a CTRL-I (to fix indentation) on each of your java programs. In MacOS the corresponding keystrokes are Cmd-A followed by Cmd-I.

What to hand in

When you are done with this lab assignment, submit all your work through CatCourses.

Before you submit, make sure you have done the following:

- Attached the file named `Part3` containing answers to Assessment questions (1 – 5).
- Attached the created `GenCheeseShop.java` file and corrected `Error.java` file.
- Filled in your collaborator's name (if any) in the "Comments…" text-box at the submission page.

Also, remember to demonstrate your code to the TA or instructor before the end of the grace period.