

The programs for this lab should be stored in your Eclipse repository. Create a new Java project in your workspace called lab5. Remember to use a location on your P: drive or a USB drive (not the C: drive). When your programs are running correctly, turn in a printout of the Java code.

### **Problem 1**

Write a program called Fact that prompts the user for an integer number ( $n$ ) and computes the factorial of that number ( $n!$ ). This problem is an example of using a **while** loop to repeat something a set number of times. What is the factorial of a number? Wikipedia is usually a good place to look when you have questions like this (see <http://en.wikipedia.org/wiki/Factorial>).

#### ***Example 1:***

```
Enter a non-negative integer: 5
5! = 120
```

#### ***Example 2:***

```
Enter a non-negative integer: 0
0! = 1
```

#### ***Example 3:***

```
Enter a non-negative integer: 1
1! = 1
```

### **Problem 2**

Write a program called Survey that tallies the results for a beverage survey. The program should prompt the user to enter the beverage number chosen by a survey participant until the user enters the value -1. Once that value has been entered, display the survey results (i.e., the number of times each beverage was chosen). You can assume users enter a valid choice when prompted. You will need to create several variables to keep track of the number of times each beverage is chosen. Your output should look similar to the example below. Note: feel free to use beverages of your own choosing.

#### ***Example 1:***

```
1.Coffee  2.Milk  3.Red Bull  4.Water
Please enter the person's choice: 1
1.Coffee  2.Milk  3.Red Bull  4.Water
Please enter the person's choice: 3
1.Coffee  2.Milk  3.Red Bull  4.Water
Please enter the person's choice: 1
1.Coffee  2.Milk  3.Red Bull  4.Water
Please enter the person's choice: 4
1.Coffee  2.Milk  3.Red Bull  4.Water
Please enter the person's choice: -1
```

```

Survey Results
=====
Coffee      2
Milk        0
Red Bull    1
Water       1
    
```

**Problem 3**

Write a new program named RomanNumeralsII that prompts the user to enter an integer between 1 and 10 (inclusive). The program should then display the Roman numeral equivalent for that number (see [http://en.wikipedia.org/wiki/Roman\\_numerals](http://en.wikipedia.org/wiki/Roman_numerals)). The difference between this program and the RomanNumeral program in lab 4 is that this program will use a **separate method** to ask for and validate that the entered number is in the **correct range** and force the user to enter a number in the specified range. It should keep prompting the user to enter integers until they enter one in the correct range. Since the user has to enter a number in the correct range to continue, the if-else in the main program no longer needs to handle numbers that are not in the correct range.

***Example 1:***

```

Enter an integer from 1 to 10 (inclusive): 4
The Roman numeral for 4 is IV.
    
```

***Example 2:***

```

Enter an integer from 1 to 10 (inclusive): -3
-3 is not a valid choice.
Enter an integer from 1 to 10 (inclusive): 2
The Roman numeral for 2 is II.
    
```

***Example 3:***

```

Enter an integer from 1 to 10 (inclusive): 347
347 is not a valid choice.
Enter an integer from 1 to 10 (inclusive): -9
-9 is not a valid choice.
Enter an integer from 1 to 10 (inclusive): 0
0 is not a valid choice.
Enter an integer from 1 to 10 (inclusive): 9
The Roman numeral for 9 is IX.
    
```