CS-182 Lab #5

Student Learning Outcomes

- More experience with the NetBeans IDE
- Creating Java projects in NetBeans
- Java console applications
- Comments
- Variables
- Constants
- Console input
- Console output
- Java loops

Overview

This lab gives some hands-on experience coding loops in Java. You will be using the NetBeans IDE to create Java console applications that apply the various concepts introduced in the chapter.

If you're still unclear on how to create a project in NetBeans or how to submit your lab work on Canvas, please refer to the detailed instructions in Lab 1.

Friendly reminder: Be sure you are saving your work for this lab in your Lab 5 folder. If you are using the lab computers, be sure to save to your USB storage media.

For all exercises, be sure you have the statement *import java.util.Scanner;*

Be sure to read each problem carefully and make sure you understand it. You might have to read it over multiple times! While not formally required for this lab, you may find it helpful to write pseudocode or make a flowchart to work out your logic. Keep in mind there are almost always multiple possible ways to solve these problems!

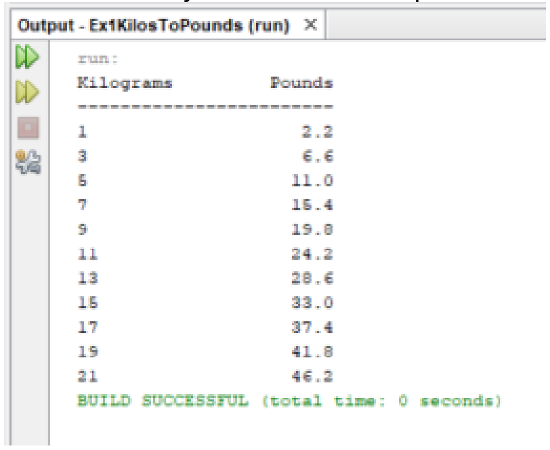
Exercise 1: Kilograms to Pounds

This is exercise 5.3 on page 195 of the text. In this exercise, you will create a simple table of kilogram to pound conversions. 1 kilogram = 2.2 pounds.

1. Create a new project in NetBeans in your Lab 5 folder called Ex1KilosToPounds.

2. Code a solution that creates a table of kilo to pound conversions. To keep things simple, do odd kilos from 1 through 21 kilos. You can code this using either a for loop or a while loop.

3. Be sure to test your work. Your output should look something like this:



```
run:
Kilograms         Pounds
------------------------
1                    2.2
3                    6.6
5                   11.0
7                   15.4
9                   19.8
11                  24.2
13                  28.6
15                  33.0
17                  37.4
19                  41.8
21                  46.2
BUILD SUCCESSFUL (total time: 0 seconds)
```

Exercise 2: Loan Comparison

This is exercise 5.21 on page 197 of the textbook. Some key formulas for this exercise:

monthlyInterestRate = annualInterestRate/ 100/ 12

double monthlyPayment = loanAmount * monthlyInterestRate /
    (1 - (Math.pow(1 / (1 + monthlyInterestRate), numOfYears * 12)))

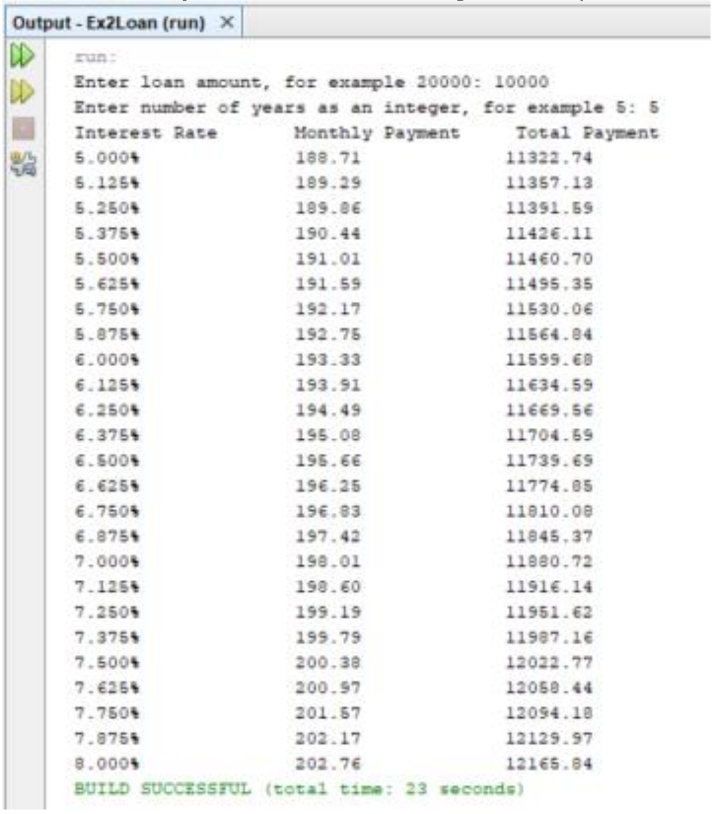
double totalPaymentAmt = monthlyPayment * numOfYears * 12

1. Create a new NetBeans project called Ex2Loan.

2. Code a solution that allows the user to enter the loan amount and number of years. Calculate and display the monthly payment and total payments for interest rates starting at 5% through 8% in 1/8 (0.125) increments. Challenge option: You may prompt the user for starting and ending interest rate range. You may use either a for loop or a while loop.

3. Use appropriate variable names, constants and comments.

4. Be sure to test your work. You should get the output below:



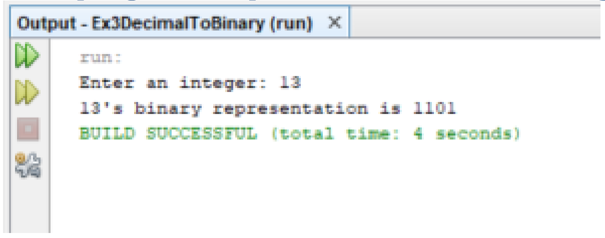
```
run:
Enter loan amount, for example 20000: 10000
Enter number of years as an integer, for example 5: 5
Interest Rate        Monthly Payment     Total Payment
5.000%               188.71              11322.74
5.125%               189.29              11357.13
5.250%               189.86              11391.59
5.375%               190.44              11426.11
5.500%               191.01              11460.70
5.625%               191.59              11495.35
5.750%               192.17              11530.06
5.875%               192.75              11564.84
6.000%               193.33              11599.68
6.125%               193.91              11634.59
6.250%               194.49              11669.56
6.375%               195.08              11704.59
6.500%               195.66              11739.69
6.625%               196.25              11774.85
6.750%               196.83              11810.08
6.875%               197.42              11845.37
7.000%               198.01              11880.72
7.125%               198.60              11916.14
7.250%               199.19              11951.62
7.375%               199.79              11987.16
7.500%               200.38              12022.77
7.625%               200.97              12058.44
7.750%               201.57              12094.18
7.875%               202.17              12129.97
8.000%               202.76              12165.84
BUILD SUCCESSFUL (total time: 23 seconds)
```

Exercise 3: Decimal to Binary Conversion

This is exercise 5.37 on page 200 of the textbook. There's not much code for this exercise but it is a little tricky! Some hints for this exercise: Use string concatenation and use the modulus to get the binary value and integer division to successively divide the number down. You ***may not*** use the Java Integer.toBinaryString() method. Zero points on the exercise if you do!

1. Create a new NetBeans project called Ex3DecimalToBinary.
2. Code a solution that allows the user to enter an integer value. Compute and display its binary value.
3. Use appropriate variable names, constants and comments.
4. Be sure to test your work.

Your program output should look something like this:



```
run:
Enter an integer: 13
13's binary representation is 1101
BUILD SUCCESSFUL (total time: 4 seconds)
```

Exercise 4: Occurrence of Max Numbers

This is exercise 5.41 on page 201 of the textbook.

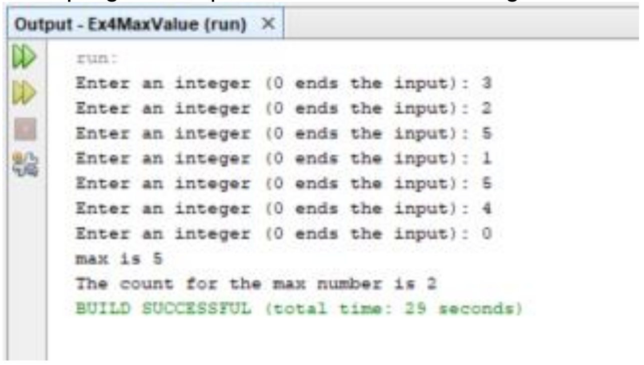
1. Create a new NetBeans project called Ex4MaxValue.

2. Code a solution that allows the user to enter integer values. Keep track of the largest value entered and count the number of occurrences. When the user enters 0 to end the program, display the largest number entered nd the number of occurrences.

3. Use appropriate variable names, constants and comments.

4. Be sure to test your work. Test with both valid and invalid values and make sure your program responds appropriately.

5. Your program output should look something like this:



```
run:
Enter an integer (0 ends the input): 3
Enter an integer (0 ends the input): 2
Enter an integer (0 ends the input): 5
Enter an integer (0 ends the input): 1
Enter an integer (0 ends the input): 5
Enter an integer (0 ends the input): 4
Enter an integer (0 ends the input): 0
max is 5
The count for the max number is 2
BUILD SUCCESSFUL (total time: 29 seconds)
```

Exercise 5: Count Vowels and Consonants

This is exercise 5.49 on page 203 of the textbook.

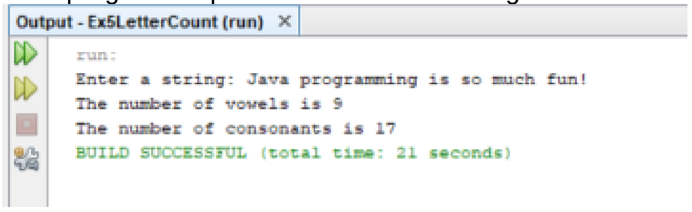
1. Create a new NetBeans project called Ex5LetterCount.

2. Code a solution that allows the user to enter a string, count and display the number of vowels and consonants. Hint: The logic to analyze each character is almost identical to an exercise from Lab 4. Challenge option: Display the string length and count other characters (I.e., digits, spaces, punctuation characters, etc.).

3. Use appropriate variable names, constants and comments.

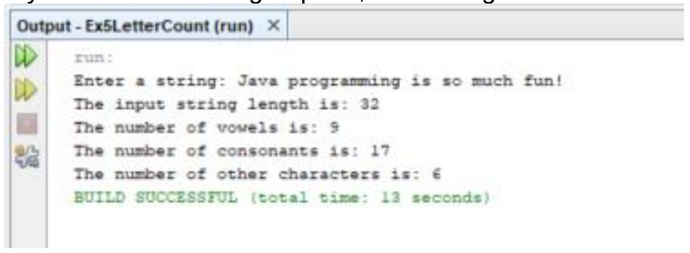
4. Be sure to test your work.

5. Your program output should look something like this:



```
run:
Enter a string: Java programming is so much fun!
The number of vowels is 9
The number of consonants is 17
BUILD SUCCESSFUL (total time: 21 seconds)
```

If you did the challenge option, something like this:



```
run:
Enter a string: Java programming is so much fun!
The input string length is: 32
The number of vowels is: 9
The number of consonants is: 17
The number of other characters is: 6
BUILD SUCCESSFUL (total time: 13 seconds)
```

Grading Criteria:

Deliverable	Points	Breakdown
Ex. 1 Kilograms to Pounds	6	Coded, produces correct output
Ex. 2 Loan Comparison	6	Coded, produces correct output
Ex. 3 Decimal to Binary	6	Coded, produces correct output
Ex. 4 Max Number	6	Coded, produces correct output
Ex. 5 Vowels & Consonants	6	Coded, produces correct output
Lab Total	**30**	