Lab #4 - LO7 Loops

Before beginning any of the programs, create a new project and call it Lab4. You will create all of your programs inside this project. Create a single package to contain **all** of your programs, call it Lab4. Be sure to follow ALL instructions regarding creating various objects for each question.

1. Create a multiplication table for numbers 1 to 9.

- Create a new Java class within the above package - call it Program1
- Add your public static void main method
- Create two for loops, nested. Set both to count from 1 to 9
- Use a print() in the inner loop to output i*j
- Use println() in the outer loop to create a new line

2. Convert an inputted number from decimal to hex.

- Create a new Java class within the above package - call it Program2
- Add your public static void main method
- Add two variables: decimal and hexVal as int and String
- Declare your scanner object. Call it myDecInput
- Add a line to import the Scanner class.
- use myDecInput to assign the value of decimal using nextInt()
- Set up a while loop using a check on the decimal variable. Your loop should continue as long as decimal does not equal 0
- set up a variable inside the loop called hexValue and initialize it to the modulus of decimal and 16.
- Use the conditional operator to set a char variable called hexDigit based on checking if hexValue is between 0 and 9, inclusive. If it is, set hexDigit to hexValue + '0'. If not, set it to hexValue - 10 + 'A'. The adding and subtracting is generating the correct character value. It's a good idea to use the debugger to watch values here as some operations are regular math and some are concatenation. Note that the debugger will show you something like '4' 52 for a char value. That means char '4' which is integer value 52.
- add hexDigit to hex making sure you add in that order so that hexDigit is always added on the left. Remember, the + is concatenation when working with string variables.
- divide decimal by 16. That completes the loop
- Outside of the loop, output the result, hex

3. Compute the greatest common divisor of two numbers.

- Create a new Java class within the above package - call it Program3
- Add your public static void main method
- Add three variables: n1, n2 and gcd as int variables

- Declare your scanner object. Call it myNumInput
- Add a line to import the Scanner class.
- use myNumInput to assign the value of n1 and n2 using nextInt()
- initialize gcd to 1
- create a counter k of type int and initialize it to 2. This is outside of the loop.
- set up a while loop with a condition that will continue looping as long as k is <= to both n1 and n2
- inside the loop, add a condition to see if n1 and n2 are both divisible by k. if so, set gcd equal to k
- after the condition, increment k
- output the gcd result

4. Determine the number of times a repeated character appears on an input line.

- Create a new Java class within the above package - call it Program4
- Add your public static void main method
- Add three variables: sString and nDupOccurances as String and int variables
- Declare your scanner object. Call it myStringInput
- Add a line to import the Scanner class.
- use myStringInput to assign the value of sString using nextLine()
- initialize nDupOccurances to 0
- set up a for loop to step through the string one character at a time. use the string length as the continuation condition
- set up an if condition to check if the current character(i) is equal to the next character(i+1). If they do, increment nDupOccurances
- output the result

5. Enter a line and a character and count how many times the char occurs in the line.

- Create a new Java class within the above package - call it Program5
- Add your public static void main method
- Add three variables: sLine, sChar and nCount as String, char and int variables
- Declare your scanner object. Call it myStringInput
- Add a line to import the Scanner class.
- use myStringInput to assign the value of sLine and sChar using nextLine()
- initialize nCount to 0
- set up a for loop to step through the string one character at a time. use the string length as the continuation condition
- at each character, check against sChar and increment the counter wherever the current character equals sChar
- output the result