Lab #2 – LO5 Math Library and Strings

Before beginning any of the programs, create a new project and call it Lab2. You will create all of your programs inside this project. Create a single package to contain **all** of your programs, call it Lab2. Be sure to follow ALL instructions regarding creating various objects for each question.

1. Use the pow() method to apply an exponent and then store/output the result as an integer.

- Create a new Java class within the above package - call it Program1
- Add your public static void main method
- Inside the main method, create three new int variables, x, exponent and result.
- Assign the values of x and exponent to be 2 and 8 respectively.
- Assign result to be the value of 2 to the power of 8, using the pow() method. The format is Math.pow(x,exponent). Note that you are assigning to an int and pow returns a floating point value so you will need to explicitly cast your answer to assign it.
- Inside the main method, add a System.out.println call to output the variable result

2. Output the inputted value as different results using different Math methods.

- Create a new Java class within the above package - call it Program2
- Add your public static void main method
- Add three variables, firstNum, exp, resultExp, resultLog, resultRoot, Declare them as int, int, double, double, double respectively
- Declare your scanner object. Call it myInput
- Add a line to import the Scanner class.
- use myInput to assign the value of firstNum and exp using nextInt()
- calculate the values of the three result variables as follows:

    resultExp -> use pow() with firstNum and exp
    resultLog -> use log10() with firstNum
    resultRoot -> use sqrt() with firstNum

Add System.out.println to output :

    ___ Raised to the ____ power is ____
    The log of ____ is ____
    The square root of ____ is ____

    where the blanks will be replaced with your variables

3. Output the inputted value as different results using different Math methods.

- Create a new Java class within the above package - call it Program3

- Add your public static void main method
- Add four variables, val1, val2, val3, val4. Declare them as doubles
- Declare your scanner object. Call it myDoubleInput
- Add a line to import the Scanner class.
- use myDoubleInput to assign the value of val1, val2, val3, val4 using nextDouble()

Add System.out.println to output :

___ rounded up is ___ -->> ceil()
___ Rounded down is ___ -->> floor()
The min value of 2 and 8 is ___ -->> min()
The min value of 2.0 and 8.0 is ___ -->> min()
The abslote value of -55 is ___ -->> abs()

where the blanks will be replaced with your variables and the method calls

4. Read in two lines of text. Determine whether the second line is found within the first line, displaying the index within the first line at which the second line is found. If the second line is not found within the first line, display a value of -1.

- Create a new Java class within the above package - call it Program4
- Add your public static void main method
- Add three variables, line1, line2, index. Declare them as String, String, int
- Declare your scanner object. Call it myStringInput
- Add a line to import the Scanner class.
- Read in a line of text into variable line1
- Read in another line of text into variable line2
- use the indexOf() method called on line1 using line2 as a parameter to determine the index of the second string within the first. It will return -1 if it finds nothing.
- You should store the result of the last step in the index variable
- Output the variable index

5. Read in a line of text and display the line with the first word moved to the end of the line.

- Create a new Java class within the above package - call it Program5
- Add your public static void main method
- Add four variables, text, index, firstWord and remainder. Declare them as String, int, String, String
- Declare your scanner object. Call it myLineInput
- Add a line to import the Scanner class.
- Read in a line of text into variable text
- Remove any leading or trailing whitespace by calling the trim method on text and assigning the result back to itself
- Find the first space in the line of text and assign the index of it to variable index. That will indicate the end of the first word. Remember we use the indexOf() method for this.

- Use the substring method to extract the first word and assign it to firstWord. You will extract from the start of the string to the above index.
- Use substring again to assign to the remainder variable the rest of the line. You will extract starting at index and go to the end of the string
- Remove the leading space(s) from the rest of the line using the trim method on remainder and assign the result back to remainder
remainder = remainder.trim();
- Output the line with the first word moved to the end of the line by simply outputting remainder concatenated with a space and then firstWord.
System.out.println(remainder + " " + firstWord);

6. Process a string to extract the second word

- Create a new Java class within the above package - call it Program6
- Add your public static void main method
- Add four variables, line and word2. Declare them as Strings
- Add a variable wordSeparator as an int
- Declare your scanner object. Call it stringInput
- Add a line to import the Scanner class.
- Read in a line of text into variable line adding the trim() method to your usual next___() method call to ensure there is no leading space
- Find the first position of a space character (don't worry about tabs right now) using indexOf as we did above and assign the result to wordSeparator
- Set line to line minus the first word by extracting everything after the first space using the substring method from wordSeparator+1 to the end of the line. Dont forget the trim the result you assign back to line
- Repeat the step above to find the next space which will be the end of the second word
- Use substring again to extract from the start of line to wordSeparator and assign the result to word2
- output a message saying ""<<word2>> is the second word in the line"