CSC 101 Programming Assignment #3 10/17/17

Due date – Thursday November 9th by 11:59PM

**NOTE:** This assignment asks you to create static methods outside the main to do the work involved. Ignoring this and simply writing code in the main that does the tasks needed with get you **NO** credit on this assignment.

**Program 1:** Using the formula from problem #21 on page 145, do the following: Write a program that will output a list of Fibonacci numbers. Start by asking the user how many values they would like outputted to the screen. Create a static function that will return an integer value of a ***single*** Fibonacci number. This static method should take an integer that represents which value in the sequence we are computing and returns an integer of that value. The formula will calculate this but you will have to cast it to be an integer. Use a loop in the main to call that method repeatedly and output the results one at a time until the numbers of values meets the list the user asked for. Note: method does not output and only returns one value!

**Program 2: PickWord.java**

Write a program that will ask the user to input a phrase (multiple word sentence), and an integer value. A static function will be created that takes the sentence and integer as input parameters and returns a single word of that sentence. If the integer was the number three, then the word returned would be the third word in the sentence. **There should be some protections not allowing the integer to be zero or negative and if it is higher than the number of words in the sentence, then just return the last word.** The main will ask the user for the input and show the output and will also give the user the choice to enter in new information. This should repeat until the user says that they are done. The static function will not do any inputting or outputting, it simply finds the word and returns it to the main as a string. Do **not** use StringTokenizer to break up string and do **not** use the split method from the String class. You may assume each space in the input is the transition between words.

**Program 3: Numbers.java**

Write two static functions that each take two integer parameters, one that calculates and returns the least common multiple of those two values and the other that calculates and returns the greatest common divisor of the two. The main method will ask the user for the two inputs and then output the results from each method call. The user also should be given the option of inputting a new set of numbers. The least common multiple is the smallest value that both numbers divide into. Such as the LCM(12,15) = 60. Algorithm for calculating the LCM can be done easiest by calculating the greatest common divisor first as there is a simple relationship between the two. Calling the GCD method from inside the LCM method is required. To calculate the GCD; the Euclidean Algorithm can do this and is defined on many places on the web. Do not use the recursive method from the book. Do not copy directly from outside source and do not look for an already made method in the JAVA API.

Each of these programs must be done in a separate file. Name them **Fibonacci.java, PickWord.java,** and **Numbers.java**. Name the class in each file these names as well. If you do not name these files correctly you will lose points.

Hand in electronically – (NOT E-mail!!!)

In S-drive CSC 101 folder:

1. Create folder called **projectthree\_firstname\_lastname**
2. Place three files named above in folder. (NOTHING ELSE)

170 // Finding the location of the word choice  
171 if((numberOfSpaces + 1) == numericalValue) {  
172 firstLetterLocation = sentence.indexOf(index);  
173 System.out.println("First Location: " + firstLetterLocation);  
174   
175 }   
176   
177 // Final segment of word choice  
178 if (numberOfSpaces == numericalValue) {  
179 finalLetterLocation = sentence.indexOf(index);  
180 System.out.println("Final Location: " + finalLetterLocation);  
181 }