ICS4U Lab 1:

All methods below should be static methods, defined in a **single Java class** called **Assignment1Utility**. All methods should have appropriate comments, and any method body that is not trivial should have appropriate comments.

Note: Class name, method names, input types or output types not matching the assignment will result in a drastic loss of marks as all tests will fail.

1. Write a method called welcome. This method should take in no parameters and return nothing. It should simply print “Welcome to ICS4U!”. This methods returns nothing (The type for this is ‘void’)

2. Write a method called stringCombiner. This method takes in 2 strings and returns them concatenated together. Look up how to use a string builder and use that in your implementation.

ex)

```java
public static void main(String[] args) {
    // Example 1: Both strings are non-empty
    String str1 = "Hello, ";
    String str2 = "world!";
    String combinedString1 = stringCombiner(str1, str2);
    System.out.println("Example 1: " + combinedString1);

    // Example 2: One of the strings is empty
    String str3 = "";
    String str4 = "Empty String Test";
    String combinedString2 = stringCombiner(str3, str4);
    System.out.println("Example 2: " + combinedString2);

    // Example 3: Both strings are empty
    String str5 = "";
    String str6 = "";
    String combinedString3 = stringCombiner(str5, str6);
    System.out.println("Example 3: " + combinedString3);
}
```

```
Example 1: Hello, world!
Example 2: Empty String Test
Example 3:
```

3. Write a method called sumOfMultiples(first, mult) that takes in 2 integers and returns an integer. The first integer is the inclusive stopping point of numbers to check, the second is the target multiple. The returned value is the sum of all multiples of mult from 1 to first. You MUST use a **while loop** for this question. Using a for loop will be treated as a non submission.

```
public static void main(String[] args) {
    // Example 1: Sum of multiples of 3 from 1 to 10
    int first1 = 10;
    int mult1 = 3;
    int result1 = sumOfMultiples(first1, mult1); // 18

    // Example 2: Sum of multiples of 5 from 1 to 20
    int first2 = 20; // Stopping point
    int mult2 = 5;   // Target multiple
    int result2 = sumOfMultiples(first2, mult2); // 50

    // Example 3: Sum of multiples of 7 from 1 to 50
    int first3 = 50; // Stopping point
    int mult3 = 7;   // Target multiple
    int result3 = sumOfMultiples(first3, mult3); // 392
}
```

4. Write a method called nthFibNum. This should take in an int and return an int. The input is a positive integer and the input is the fibonacci number associated with that number. As a reminder the fibonacci numbers are defined as (https://en.wikipedia.org/wiki/Fibonacci_sequence). Return -1 if the input is negative. For this method specifically add a comment estimating how long this took you to complete from start to end.

Ex) nthFibNum(4) -> 2
nthFibNum(10) -> 34
nthFibNum(1) -> 0

5. Write a method called longestStringLen(String[]). This method takes in a list of strings and returns the length of the longest string in that list.

```
public static void main(String[] args) {
    // Example 1: Regular strings
    String[] strings1 = {"apple", "banana", "kiwi", "orange"};
    int longestLength1 = longestStringLen(strings1); // 6

    // Example 2: Empty strings
    String[] strings2 = {"", "", "", ""};
    int longestLength2 = longestStringLen(strings2); // 0

    // Example 3: Mixed strings
    String[] strings3 = {"", "cat", "", "dog", "elephant"};
    int longestLength3 = longestStringLen(strings3); // 8
}
```