Montgomery College
CMSC 203
Assignment 1 Design

1) Write the pseudo code for Assignment 1 based on the Assignment 1 Description given to you. Refer to the **Pseudocode Guideline** on how to write Pseudocode.

2) Complete the following test table. At this point you only need to complete the **Input** and **Expected Output** columns. Later when the implementation is complete, you will complete the **Actual Input** and **Actual Output** columns and compare them to see if the tests passed or not.

Use the given tests table and data as an example. Record your data for input and output in the following table. **Make sure your tests cover all the possible scenarios.**

Test Case #	Input	Actual Input	Expected Output	Actual Output	Did the test pass?
1	Temp: 30 Wind speed: 20		17.361783756466327	17.361783756466327	Yes
2	Temp: 35 Speed: 5		30.59961078	30.599610777069774	Yes
3	Temp: -46 Speed: 4		-62	-62.025237367382815	Yes

Pseudocode Guideline

Pseudocode is code written for human understanding not a compiler. You can think of pseudocode as "English code," code that can be understood by anyone (not just a computer scientist). Pseudocode is not language specific, which means that given a block of pseudocode, you could convert it to Java, Python, C++, or whatever language you so desire.

Pseudocode will be important to your future in Computer Science. Typically pseudocode is used to write a high-level outline of an algorithm.

As you may already know, an algorithm is a series of steps that a program takes to complete a specific task. The algorithms can get very complicated without a detailed plan, so writing pseudocode before actually coding will be very beneficial.

How to Write Pseudocode

There are no concrete rules that dictate how to write pseudocode, however, there are commonly accepted standards. A reader should be able to follow the pseudocode and hand-simulate (run through the code using paper and pencil) what is going to happen at each step. After writing pseudocode, you should be able to easily convert your pseudocode into any programming language you like.

We use indentation to delineate blocks of code, so it is clear which lines are inside of which method (function), loop, etc. Indentation is crucial to writing pseudocode. Java may not care if you don't indent inside your **if** statements, but a human reader would be completely lost without indentation cues.

Remember: Human comprehension is the whole point of pseudocode. So, what does pseudocode look like?

Finding the Fibonacci numbers till n:

Pseudocode	Real Code in Java

Declare an integer variable called n
Declare an integer variable sum.
Declare an integer variable f1
Declare an integer variable f2
If n is less than 2
sum =n
else
set sum to 0
set f1 and f2 to 1
repeat n times
sum = f1 + f2
f2 = f1
f1 = sum
end loop
print sum
	`int n,k, f1, f2, sum;`
	`if ( n < 2 )`
	`sum =n;`
	`else`
	`{`
	`sum=0;`
	`f1 = f2 = 1;`
	
	`for(k=2; k<n; k++)`
	`{`
	`sum = f1 + f2;`
	`f2 = f1;`
	`f1 = sum;`
	`}`
	`}`
	`System.out.println("Fibonacci of number " + n + " is "+ sum);`

Remember that pseudocode is not language specific so we are not looking for "almost Java" code, but instead, we are looking for a strong understanding of the algorithm at hand.

Declare final double CONST_ONE and give it the value of 35.74

Declare final double CONST_TWO and give it the value of 0.6215

Declare final double CONST_THREE and give it the value of 35.75

Declare final double CONST_FOUR and give it the value of 0.16

Declare final double CONST_FIVE and give it the value of 0.4275

Declare double temperature, double windSpeed, and double windChill

Declare the Scanner reference named "input", which will be used to prompt and read input

Print the title of the program

Prompt for the temperature in degrees F as a double between –45 and 40 and store it in temperature

Prompt for the wind speed in MPH as a double between 5 and 60 and store it in windSpeed

Calculate the wind chill as follows: CONST_ONE + (CONST_TWO * temperature) - (CONST_THREE * (windSpeed^CONST_FOUR)) + (CONST_FIVE * (temperature * (windSpeed^CONST_FOUR))). Store this value in windChill.

Print windChill

Print the programmer's name