**CMSC203**

**Assignment #1**

**Wind Chill Temperature Index**

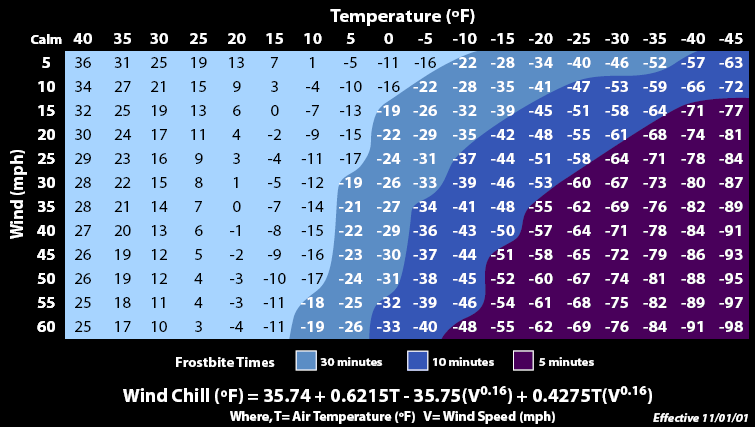
**Wind Chill** is the term used to describe the rate of heat loss on the human body resulting from the combined effect of low temperature and wind. As winds increase, heat is carried away from the body at a faster rate, driving down both the skin temperature and eventually the internal body temperature.

The first wind chill formulas and tables were developed by [Paul Allman Siple](https://en.wikipedia.org/wiki/Paul_Allman_Siple) and [Charles F. Passel](https://en.wikipedia.org/wiki/Charles_F._Passel) working in the Antarctic before the [Second World War](https://en.wikipedia.org/wiki/Second_World_War) They were based on the cooling rate of a small plastic bottle as its contents turned to ice while suspended in the wind on the expedition hut roof, at the same level as the [anemometer](https://en.wikipedia.org/wiki/Anemometer) (pictured above). The so-called Windchill Index provided a pretty good indication of the severity of the weather.

**In late 2001, the National Weather Service implemented a new Wind Chill Temperature (WCT) Index. The reason for the change was to improve upon the old index used by the NWS and the Meteorological Services of Canada (MSC) which was based on the 1945 Siple and Passel Index.**

**Assignment Description**

Develop a Java application that prompts for and reads two double numeric amounts that represent the Fahrenheit temperature and the wind speed. The temperature must be between -45 and 40 inclusively. The wind speed must be between 5 and 60 inclusively. Use these two amounts in the formula below to calculate the wind chill temperature in degrees Fahrenheit.



Here is how you calculate the **New Wind Chill Index:**

Wind Chill (oF) = 35.74 + 0.6215T - 35.75(V0.16) + 0.4275T(V0.16), where

V is the Wind Speed in MPH, and

T is the temperature in degrees F.

**Concepts tested by this assignment**

* Creating a class
* Variables
* Strings
* Input/output facilities
* Conditional statements
* Logical operators
* Eclipse IDE
* GitHub

**Classes**

Your program must use the single class WindChill with only one static void main method

**Assignment Details**

* Use the Scanner class for input and use System.out.print or System.out.println for output. Please make sure that the Scanner is closed when your project terminates.
* You must use "named constants" for any literal values that will not change during program execution.
* You must use meaningful variable names and good indentation.
* Ask the user for the Fahrenheit temperature and the wind speed.
* Remind the user that temperature values should be greater than or equal to -45, but less than or equal to 40, and the wind speeds should be greater than or equal to 5, but less than or equal to 60, but don't validate these values in your code.
* Display a header for your application.
* Display the wind chill temperature that results.
* Display your name as programmer at the end.
* In your Learning Experience: highlight your lessons learned and learning experience from working on this project. What have you learned? What did you struggle with? What will you do differently on your next project? Include what parts of the project you were successful at, and what parts (if any) you were not successful at.
* GitHub: In your repository (see Lab 1), upload the files initially provided in Blackboard for the project. When you are finished with the design and programming, upload your design file and java file. You will want to upload these files as contents of a directory so that future uploads can be kept separate. Take a screen shot of the GitHub repository.
* Just in case you know what an array is -- you may not use arrays (or any Java collections of any kind) while implementing this project.
* Run your application with the command line (javac, java) and with an IDE Integrated Development Environment (Eclipse)

**Examples – Tests**

Example 1:

Wind Chill Calculator

Enter the temperature in Fahrenheit (must be >= -45 and <= 40): 30

Enter the wind speed (must be >= 5 and <= 60): 20

Wind chill temperature: 17.361783756466327 degrees Fahrenheit

Programmer: (insert your name here)

Example 2:

Wind Chill Calculator

Enter the temperature in Fahrenheit (must be >= -45 and <= 40): -15.5

Enter the wind speed (must be >= 5 and <= 60): 35.3

Wind chill temperature: -48.842359110042835 degrees Fahrenheit

Programmer: (insert your name here)

Example 3:

Wind Chill Calculator

Enter the temperature in Fahrenheit (must be >= -45 and <= 40): -9.3

Enter the wind speed (must be >= 5 and <= 60): 22.8

Wind chill temperature: -35.55509110244696 degrees Fahrenheit

Programmer: (insert your name here)

Example4:

Wind Chill Calculator

Enter the temperature in Fahrenheit (must be >= -45 and <= 40): 0

Enter the wind speed (must be >= 5 and <= 60): 22.5

Wind chill temperature: -23.09357636894822 degrees Fahrenheit

Programmer: insert your name here

**Good Faith Attempt - Test**

To satisfy the “Good Faith Attempt” (see Blackboard) your code must compile and print the prompts. Your GFA need not read the prompts, compute the windchill, or print the results.

Note that the GFA is not graded, so you must submit your totally-working code by the submission deadline to have a non-zero grade.

**Test Plan**

Test your program with at least two more test cases. Use the given 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 chill:  20 |  | 17.361783756466327 |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

**Deliverables**

Design: Submit your initial design (pseudo-code and test cases) in Blackboard.

Implementation: Run the application twice and turn in screen shots of both runs in Blackboard.

* Run it once using command-line Java (javac to compile, and java to run). Your screen shot should show both compilation and execution. If your computer cannot find the javac or java commands, see the lecture slides explaining how to set the path for Java.
* Run it again using an Integrated Development Environment (IDE). I would recommend Eclipse. Provide a screen shot of the console output. See the lecture slides explaining how to download and install Eclipse.
* Turn in your final design (pseudo-code and test cases)
* Turn in the working java file WindChill.java (not the .class file).
* Learning Experience: highlight your lessons learned and learning experience from working on this project. What have you learned? What did you struggle with? What will you do differently on your next project? Include what parts of the project you were successful at, and what parts (if any) you were not successful at.
* GitHub: In your repository (see Lab 1), upload the files initially provided in Blackboard for the project. When you are finished with the design and programming, upload your design file and java file. You will want to upload these files as contents of a directory so that future uploads can be kept separate. Take a screen shot of the GitHub repository.

**Submission Detail**

* Upload two .zip files for each assignment. The first .zip file will contain **all** the files required for the assignment (final design with pseudocode and table of test cases, screen shot of runs and the GitHub repository, and the .java file, and will be named: LastNameFirstName\_Assignment1.zip.
* The second .zip file will only contain the .java files and will be named: LastNameFirstName\_Assignment1\_Moss.zip. **This .zip will not have any folders in it – only .java files.**

Here’s an example for Assignment 1:

KartchnerJeannette\_Assignment1.zip [a compressed file containing the following]

PseudoCode\_and\_TestTable.docx

Learning\_Experience.docx

CmdLine\_output.jpg

Eclipse\_console\_output.jpg

GitHub\_repo.jpg

WindChill.java

KartchnerJeannette\_Assignment1\_Moss.zip [a compressed file containing only the following]

WindChill.java