

## Wind-Chill Index Conversion

You have been commissioned to write a wind-chill index conversion application. The factors that must be considered are the wind speed and outside temperature. The National Weather Service (NWS) provides documentation to explain the calculation on their [website](#).

## Specifications

1. The program should first display a welcome screen that includes your name, course and section number, and the assignment name.

```
*****
* John McMeen * CISP 1010 S# * Project 1 *
*****
```

2. The program should then prompt the user for the required data. When valid numbers have been entered for temperature and wind speed, the program will calculate the wind-chill index and display it to the user as illustrated below.

```
*****
* John McMeen * CISP 1010 S# * Project 1 *
*****

Enter a value for temperature (F): 35.2
Enter a value for wind speed (MPH): 4.3
The wind-chill index is: 31.473087610760995
```

3. The program should check to make sure temperature and wind speed are valid for the NWS calculation. If not, display an error message to the user and exit (call the `System.exit(0)` method to exit immediately).

```
Enter a value for temperature (F): -60.2
Temperature is out of range, must be > -58F and < 70F

Process finished with exit code 0
```

```
Enter a value for temperature (F): 45
Enter a value for wind speed (MPH): 1
Wind speed is out of range, must be > 2 MPH

Process finished with exit code 0
```

4. After reading the NWS documentation, you have concluded the equation to calculate Wind-chill Index from outside temperature and wind speed is as follows.

$$\text{Wind-chill index} = 35.74 + 0.6215T - 35.75 (V ^ 0.16) + 0.4275T (V ^ 0.16)$$

**Where, T = outside temperature measured in degrees Fahrenheit and  
V = wind-speed or velocity as MPH.**

**This formula is only accurate if the temperature is > -58.0 F and < 70.0 F,  
and the wind speed is > 2.0 MPH.**

5. The program should store temperature, wind speed, and wind-chill Index in variables of data type double.
6. All user prompts, messages, and results must be neat, appropriately organized, and easily understood by the user. An automated script will be used to test your program, and data will be supplied in the order shown above. Failure to observe the correct prompting order will result in incorrect operation of your program and a loss of points.
7. You can test your results by using the [Meteorological Conversions and Calculations Wind Chill Calculator](#).
8. Make sure your code has the required documentation, as outlined in the CS Java Documentation Policy under Course Info on D2L.
9. This is an individual assignment. By submitting your work to D2L, you acknowledge you have read the NSCC Computer and Information Science Department's Honor Code and Documentation Policy and are following its policies to the best of your ability.

## Deliverables

Make sure your code has the required documentation, as outlined in the CISP Java Documentation Policy on the course website.

This is an individual assignment. By submitting your work to D2L, you acknowledge you have read the NESCC Computer and Information Science Department's Honor Code and Documentation Policy and are following its policies to the best of your ability.

Once you are satisfied with your code, compress your files with a **.zip** file and upload it the Project 1 D2L drop box. Your **.zip** file should contain the following. Make sure your code is in the following package structure **edu.northeaststate.cs1.projects.project1**.

1. Your source code under the following directory structure: **src\edu\northeaststate\cs1\projects\project1\**. Please **do not** include IDE project files such as .iml, .idea, or the out directory. In your **.zip** file, please prune off any unneeded packages such as examples, labs, or other projects.
2. Provide a citation document in Word **.docx** format with links or a write-up if you utilized any outside resources to complete your assignment.

## Evaluation

Five factors will be considered in grading your project:

1. **Compiles** (10%): does the Java code compile with no errors?
2. **User Interface** (10%): does the program interact with the user as expected?
3. **Design** (40%): does the code meet functionality requirements and design specifications?
4. **Deliverables** (20%): are all deliverables included, named, and organized appropriately, including package structure and diagrams?
5. **Standards** (20%): does the code follow good programming practices, coding standards, and commenting requirements?