Roy Chung

Prof. Zachary Whitten

CMSC 255 Section 1

October 4, 2019

Programming Project 4 – Temperature Conversions

TempConvert.java is a program that asks the user for a temperature value and temperature scale. It then asks the user for the target temperature scale for the conversion. The three temperature scales handled by the program are Celsius, Fahrenheit, and Kelvin. Once the conversion is completed, the program asks if the user would like to perform another temperature conversion. If the user says no, the program ends, otherwise, it will run again. Below is the algorithm for the program:

1. Declare boolean *loopOn* and assign to it a ‘true’ value. This will be used to determine whether the loop continues or terminates.
2. Create the methods *processData, checkValidity, convertFahtoCel, convertFahtoKel, convertCeltoFah, convertCeltoKel, convertKeltoCel, convert KeltoFah*.
3. With the exception of the first two methods listed above, the rest all take in and return double values. *convertFahtoCel* is used by the program when a Fahrenheit value is entered and returns the equivalent Celsius value. *convertFahtoKel* does the same but returns a Kelvin value instead. All possible conversions between Celsius, Fahrenheit, and Kelvin are covered by the six methods.
   1. Fahrenheit to Celsius: C = (F – 32) / (9 / 5)
   2. Celsius to Fahrenheit: F = (C \* 9/5) + 32
   3. Celsius to Kelvin: K = C + 273
   4. Kelvin to Celsius: C = K – 273
   5. Fahrenheit to Kelvin: K = (F – 32) / (9 / 5) + 273
   6. Kelvin to Fahrenheit: F = (K – 273) \* (9 / 5) + 32
4. Create scanner object in the main method.
5. Call *processData* from the main method. This method prompts the user for a temperature value and scale.
   1. Store the temperature value as a double variable *temp*
   2. Store the temperature scale as a string variable *scale*
   3. These values will be passed along to the *checkValidity* method.
6. *checkValidity* will return a Boolean value of *true* if the entered values are all valid.

IF *scale* is Fahrenheit and *temp* is between -459.40 and 1000 degrees, return true.

ELSE IF *scale* is Celsius and *temp* is between -273 and 1000 degrees, return true.

ELSE IF *scale* is Kelvin and *temp* is between 0 and 1000, return true

ELSE return false.

1. If *checkValidity* returns a true value, *processData* will call either *convertFahtoCel, convertFahtoKel, convertCeltoFah, convertCeltoKel, convertKeltoCel, or convertKeltoCel*, depending on what the user inputs for current and target temperature scales. If current and target temperature scales are the same, then this step is skipped and *processData* jumps straight to outputting the result. If *checkValidity* returns a false value, then the following error message is displayed:  
   *temp* degrees *scale* is not a valid temperature.
2. Ask if the user would like to perform another conversion. If the user does not, and enters “quit,” *loopOn* becomes false and the loop ends, otherwise the loop body runs again.