RMIT University

EEET2248 – Electrical Engineering Analysis

Lecturer: Katrina Nevilee

Student Name: Nathan Williams (s3707244)

**Lectorial 1 Milestone**

This milestone required a sequential script to be created that converted a selection of units from imperial to metric and/or vice versa. The focus being on converting a selected set of data representing degrees Fahrenheit temperatures to degrees Celsius.

The tools used included basic variable definition and arithmetic. Emphasis was placed on maintaining clear style and conciseness during this project.

The results obtained included 5 test results for Fahrenheit to Celsius and 1 test result each for Celsius to Fahrenheit, centimetres to inches, grams to ounces, km/h to mph, litres to gallons and hectares to acres. Since it is a sequential script the program can perform conversions only for the data that is hardcoded into it upon running, which currently includes the data mentioned above. To add more would require writing further lines as there are currently no user defined functions that can be called. The performance of the program is further discussed below in testing.

**Output and Testing**

A close up of a logo

Description generated with high confidence

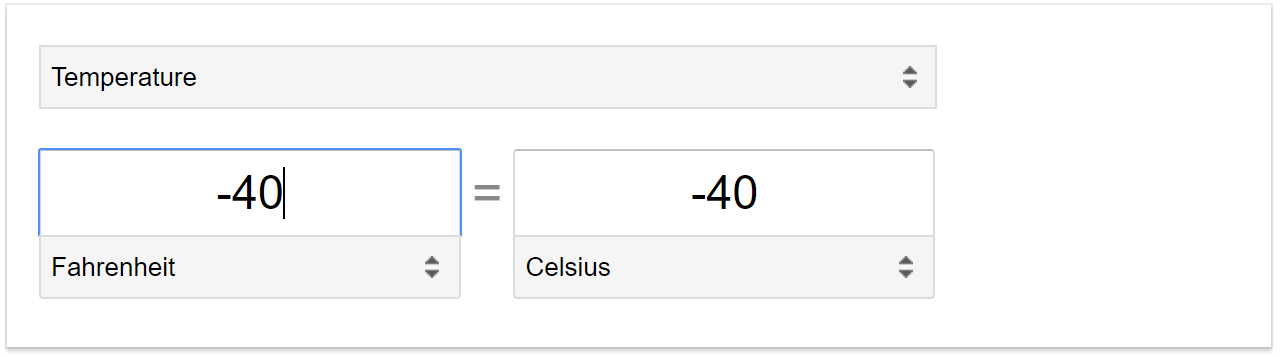
A screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with high confidenceThe above screenshots show the output from the program. The first 5 variables represent the 5 test values after they had been converted to degrees Celsius. Below are screenshots of googles converter when given the same values.

A screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidence

The google converter gave the exact same output for the test data supplied which provides evidence that our program is working correctly. Usually much more testing would be required however in this scenario it is quite a simple function that uses old and tested formulas so for the purpose of this task the output matching Google’s will satisfy as a test of the accuracy of the program. Below are more screenshots of Google’s converter, this time testing the output of the other unit conversions.

A screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidence

Similarly, Google’s converter results agree with results of our other converters. One test is not enough to verify the accuracy but for the purpose of the assignment it will suffice as these conversion formulas are very well tested already and it is simply the application of them in our MATLAB script which would require more testing.

Next the script converts all the converted units back to metric. As you can see in the script file all the variables return to their initially defined values which provides evidence that the unit conversions are accurate in reverse as well. The output is shown below:

A close up of a logo

Description generated with high confidence

From this output we conclude that the program satisfies the requirements of the task by providing successful conversions for the test data and also providing several successful conversions of other units from metric to imperial and vice versa.

**Note: not all output is displayed in the screenshots, for full output please run the .m file in matlab**

In terms of improving the script it would be highly advantageous to rewrite the program with a separate function defined for each unit conversion. This way it could be called repeatedly to test large amounts of data efficiently rather than separately rewriting the lines of code each time data is input to be converted.

**Note:** I wrote an example of this function implementation in the comments section at the bottom of the script. These functions could be called repetitively with large amounts of data and using if/switch statements to check the reverse conversions return the variables to their initial value. This is just one example of how functions can be used to more efficiently implement and test in this scenario