RMIT University
School of Engineering
EEET2248 – Electrical Engineering Analysis

# Lectorial Progress Report

Lecturer: Dr. Katrina Neville
Student Name: Aiden Contini Student Number: s3780445

Submission Due Date: 5th April 2019

## Abstract:

This milestone had the requirements of making the MATLAB unit converter more user friendly, via introducing user input, as well displaying instructions to the user on how to run the program. In this section, I was able to introduce user input which enables a user to select the conversion they wish to undertake, and then enter a number in which they want to convert, all at run time. In doing so, I included a menu-like opening to the program, to which transitions into various case statements. In this way, a user is able to select a number from a list, associated with each conversion, and in turn only run this section of the program. There has also been the inclusion of user input in which they are able to enter their selection of conversion, as well as their selection of what value they would like to convert. Finally, the inclusion of "fprintf" statements and "disp" statements were introduced to present instructions to a user to enable them to use the program correctly. As of this point, my program can successfully obtain user input and undertake specific conversions of user selected values on an easy to read text interface.

## Software solution and testing :

As the same formula were used in this milestone as were used in milestone one, it can be assumed that the formula used were correct and the accuracy of conversion were correct. In this step however, testing was undertaken to ensure the syntax for each conversion was correct, and the program was displaying the correct "to" and "from" units being converted. This was to ensure that when the program displayed "Enter Celsius value to convert to Fahrenheit", the celsius to Fahrenheit conversion was the one which was undertaken. Similarly, testing was done to ensure the final statement displayed by the program was the correct way around. Below are a series of screenshots of the program running, and performing the conversions to which the text is in the correct places and the program is doing what the user expects it to do.

| Program outputted value | Google checked value |
|---|---|
| Enter ºCelsius value to convert to ºFahrenheit: 0<br>0.000000 ºC = 32.000000 ºF | 32° Fahrenheit |
| Enter ºFahrenheit value to convert to ºCelsius: 100<br>100.000000 ºF = 37.777778 ºC | 37.7778° Celsius |
| Enter centimeter value to convert to inches: 33<br>33.000000 cm = 12.992126" | 12.9921 inches |
| Enter inch value to convert to centimeters: 25<br>25.000000" = 63.500000 cm | 63.5 inches |
| Enter meter value to convert to feet: 235<br>235.000000 m = 770.997375 ft | 770.997 ft |
| Enter feet value to convert to meters: 654<br>654.000000 ft = 199.339200 m | 199.339m |

| Program outputted value | Google checked value |
|---|---|
| Enter kilometers value to convert to miles: 3.65<br>3.650000 km = 2.268005 mi | 2.268005 miles |
| Enter miles value to convert to kilometers: 7.79<br>7.790000 mi = 12.536790 km | 12.53679km |
| Enter grams value to convert to ounces: 156<br>156.000000 g = 5.502743 oz | 5.50274 ounces |
| Enter ounce value to convert to grams: 15<br>15.000000 oz = 425.242500 g | 425.2425 grams |
| Enter kilograms value to convert to pounds: 3.4<br>3.400000 kg = 7.495591 lb | 7.49572 pounds |
| Enter pound value to convert to kilograms: 9.35<br>9.350000 lb = 4.241160 kg | 4.241089 kg |
| Enter km/h value to convert to mph: 0.657<br>0.657000 km/h = 0.408241 mph | 0.40824087 mph |
| Enter mph value to convert to km/p: 1567.92<br>1567.920000 mph = 2523.322644 km/h | 2523.322644 km/h |
| Enter litres value to convert to gallons: 1.56<br>1.560000 L = 0.412105 gal | 0.4121084 gallons |
| Enter gallons value to convert to litres: 57.4<br>57.400000 gal = 217.284256 L | 217.2826 litres |
| Enter hectares value to convert to acres: 16.4<br>16.400000 ha = 40.526040 ac | 40.52528 acres |
| Enter acres value to convert to hectares: 19.54<br>19.540000 ac = 7.907410 ha | 7.9075574 ha |

Through these screenshots, it is clearly shown that the correct units symbols are displayed in the correct spots, and the specified conversions are undertaken by the program correctly at runtime. This said however, the program does not yet handle erroneous input very well. This is primary due to the lack of error testing at this point. Up until this point, when a number that is not in the list of conversions is selected (example: 20 is selected), an “otherwise” statement is used to ensure a statement of error is displayed to the user, enabling them to adjust their mistake by entering a valid number. Unfortunately however, if a non-numeric value is inputed by the user (for example a string), the program does crash without warning, due to the lack of error testing.

Overall, the usability of the program to this stage is very high, as the display functions are able to effectively explain the instructions to a user, and hence allow the user to obtain the conversion they want to undertake.