

RMIT University

School of Engineering

EEET2248 – Electrical Engineering Analysis

Lectorial Milestone 2

Unit Converter

Lecturer: Dr. Katrina Neville

Student Name: Nathan Paul Williams

Student Number: s3707244

Submission Due Date: 6th April 2018

**Lectorial 2 Milestone**

**Abstract**

This milestone required a script that took user input representing a unit system, type and quantity and converted that data into a complimentary unit of equivalent quantity in the opposite unit system (e.g. imperial to metric or vice versa). This output then needed to be displayed in the Command Line Interface (CLI).

New tools used in this milestone include the implementation of ‘if’ and ‘switch functions’ to allow our program to make decisions based on the input. This program also uses the input function to take input from users in the command line interface, this data is then passed into our program at various points to perform the required functions. ‘fprintf’ and ‘disp’ are also used to print instructions as well as print variables and the results upon termination. ‘if’ and ‘switch’ are functions that allow different code to be executed based on a condition. The difference is switch uses different cases to execute code whereas ‘if’ simply will execute the following code if the condition is met. ‘fprintf’ and ‘disp’ are functions that print output, however fprintf is more versatile/complex allowing for printing of different variables, text and other processes within the single fprintf. Disp is a more simple version used to print a single variable or a specific set of text. The design of my program revolves around using disp/fprintf to display the instructions and then if and switch statements to correctly manipulate the relevant variables. The output is then printed using a single fprintf statement that print 2 strings and 2 variables in between some text to neatly present the result.

The program can now convert all specified units from imperial to metric or vice versa with any valid input by the user. The units specified (in metric form) are degrees Celcius, Centimetres, Metres, Kilometres, Grams, Kilograms, Kilometres, Litres and Hectares. It also displays an error message if any input does not match a valid option however it does not loop upon error, rather the program simply terminates.

**A screenshot of a cell phone

Description generated with very high confidenceOutput and Testing**

**A screenshot of a cell phone

Description generated with very high confidence**

Figure 1: Standard Output Example Figure 2: Error Output Example

A screenshot of a cell phone

Description generated with high confidence

Figure 3:Error Output Example 2

Figure 1 shows the output in a standard example converting 100km to imperial. Figure 2 and 3 shows examples of the 2 basic error situations that are dealt with in this script without looping.

|  |  |
| --- | --- |
| ***Google Unit Converter Output*** | |
| **Original Input** | **Converted Output** |
| 55 degrees Farenheit | 12.7778 degrees Celcius |
| 20 centimetres | 7.87402 inches |
| 5 metres | 16.4042 feet |
| 10 km | 6.21371 miles |
| 40 grams | 1.4096 oz |
| 88 kilograms | 194.007 lbs |
| 100 km/h | 62.1371 mph |
| 20 Litres | 5.28344 |
| 50 hectares | 123.553 |
| -12 degrees celcius | 10.4 degrees Farenheit |
| 5 inches | 12.7 centimetres |
| 20 feet | 6.096 metres |
| 3 miles | 4.82803 km |
| 28 oz | 793.787 grams |
| 200 lbs | 90.7185 kg |
| 88 mph | 141.622 |
| 4 gallons | 15.1416 Litres |
| 75 acres | 30.3514 |

Figure 4: Table showing output from Google's Unit Converter

A screenshot of a cell phone

Description generated with very high confidence

Figure 5: Compiled screenshots of program output under identical input

Using figures 4 and 5 we can compare the output of our program to that of Google’s unit converter to test accuracy. The results show that all our results agree to at least 4 significant figures. Most results are identical but some disagree slightly when getting to more than 5 significant figures. This could be inaccuracy in our program but it seems more likely that Google uses slightly different formulas than those provided in lectorial 2. Regardless, for the purpose of this milestone we can say this program appears to be accurate however further testing is required. Currently the program fulfils the requirements of taking user input and by making correct decisions based on the input converts the unit to a specific equivalent in the opposing unit system. This output is displayed in a user friendly format in the form of text in the CLI. The instructions provided to the user are clear and some basic error checking is even included without looping to handle invalid input.