Lab 2 README

ECE 13

Sean Manger

Prof: Steve McGuire

F24

Author: <Sean Manger> (smanger@ucsc.edu)

Collaborators: I did not collaborate with anyone for this Lab. All code is my own original work. I did however discuss the concepts in the lab with / assist Donny Tang.

Purpose of Lab & Program

The program implemented here is a Simple Calculator that performs standard arithmetic operations plus rounding of a number, finding the tangent of a number, average of two numbers, absolute value of a number, and temperature conversions from Farenheit To Celcius & Celcius to Farenheit. The user selects a desired operation and then enters their desired operand(s) to perform the operation with.

Operations are either binary (accepting two inputs) or unary (one input depending on the function. The binary operations are addition, subtraction, multiplication, division, and average. The unary operators are absolute value, tan, round, and the temperature conversions. There is also an option to quit the calculator program using the ‘q’ key. This Calculator performs error checking for invalid operational and numeric input and prompts the user to try again.

Functions: Sample Outputs

Multiply():

```
Welcome to Sean's calculator program! Compiled at Oct 13 2024 21:46:06
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): *
Enter Your 1st Operand: -3
Enter Your 2nd Operand: 3
Result of Multiply (-3.000000 * 3.000000) = -9.000000
```

Divide():

```
Welcome to Sean's calculator program! Compiled at Oct 13 2024 21:46:06
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): /
Enter Your 1st Operand: 4
Enter Your 2nd Operand: 2
Result of Divide (4.000000 / 2.000000) = 2.000000
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): /
Enter Your 1st Operand: 1
Enter Your 2nd Operand: 0
Error: You Cannot Divide By Zero!
```

Add():

```
Welcome to Sean's calculator program! Compiled at Oct 13 2024 21:46:06
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): +
Enter Your 1st Operand: 5
Enter Your 2nd Operand: -6
Result of Add (5.000000 + -6.000000) = -1.000000
```

Subtract():

```
Welcome to Sean's calculator program! Compiled at Oct 13 2024 21:46:06
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): -
Enter Your 1st Operand: 2
Enter Your 2nd Operand: -7
Result of Subtract (2.000000 - -7.000000) = 9.000000
```

Average()

```
Welcome to Sean's calculator program! Compiled at Oct 13 2024 21:46:06
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): m
Enter Your 1st Operand: 5
Enter Your 2nd Operand: 2
Average of (5.000000 & 2.000000) = 3.500000
```

AbsoluteValue()

```
Welcome to Sean's calculator program! Compiled at Oct 13 2024 21:46:06
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): a
Enter Your Operand: -45
|-45.000000| is = 45.000000
```

FarenheitToCelsius()

```
Welcome to Sean's calculator program! Compiled at Oct 13 2024 22:41:52
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): c
Enter Your Operand: 20
(20.000000 deg F --> C): -6.666667
```

CelsiusToFarenheit()

```
Welcome to Sean's calculator program! Compiled at Oct 13 2024 22:45:40
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): f
Enter Your Operand: 20
(20.000000 deg C --> F): 68.000000
```

Tangent()

```
Welcome to Sean's calculator program! Compiled at Oct 13 2024 22:48:48
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): t
Enter Your Operand: 30
tan (30.000000) = 0.577350
```

Round()

```
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): r
Enter Your Operand: -5.6
Round (-5.600000) = -6.000000
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): r
Enter Your Operand: 2.3
Round (2.300000) = 2.000000
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r):
```

Quitting Function

```
User, Please Enter a Mathematical Operation you Desire to Perform (*, /, +, -, m, a, c, f, t, q, r): q
Calculator has exited; restart to perform more calculations.
```

Possible Improvements To Program

While I did my utmost to resolve any & all bugs in my program I am confident there are likely some as yet undiscovered bugs remaining within my program. The fact is virtually any program can be broken in unexpected ways with enough effort. One thing I was not able to solve was the fact that when I am prompted to enter an operation if I type say “tangerine” the program will execute the Tangent() function since scanf() reads in the ‘t’ character. I did not discover this bug until the evening of submission. In the interest of time I decided not to attempt to fix it and instead work on this readme & further function testing. I believe the problem can easily be fixed by perhaps working with strings and doing some checking with if statements. However, it would be too costly this late in the game to attempt to fix this in time for submission since it is not an insignificant rewrite. Overall though, my calculator functions as intended and I am happy overall with the result.

Q&A

1. What happens if the line scanf(“ ”); executes? Why?

A: I ran this line of code myself to see what would occur and an infinite loop ensued. This makes sense because we are running scanf(“%format_specifier”, &variable) without a

format specifier or an address to store the read information. This is of course undefined behavior. As a result scanf() does not know how to handle such a situation and chaos in the form of an infinite loop ensues since scanf() is looking for information it won't receive, in a format it doesn't know, in order to store it in an unknown location (address). As a result scanf() will keep executing until it is interrupted or the program eventually crashes.

2. In this lab, we forbid you from using printf() or scanf() inside of certain functions. Explain why a rule like this is useful?

A: **As a general rule of good programming practice we should keep input & output separate from computation. scanf() & printf() are useful and necessary functions but using them irresponsibly is a recipe for disaster. You wouldn't wanna use either of these functions within another function since input & output from the called function can be easily overridden and altered which may lead to all sorts of difficult to track down bugs. As such it is best to partition input from computation to minimize the possibility for errors. If scanf() and printf() are separate from computation it is much easier to resolve any issues which may arise with the code.**

3. How long did this lab take you? Was it easier or harder than you expected? What challenges did you encounter?

A: I managed to code most of this lab yesterday afternoon and used today for any mop up, error checking, and commenting I wanted to do. I would say that the most challenging part of this lab for me was simply getting the user input to function the way I intended within our semi-infinite calculator loop. scanf() is certainly a departure from the C++ style cin >> method of input. However, I think I prefer scanf() since it is a bit simpler to work with since you must specify type identifiers and variable names within the function. As such I think scanf() is easier to

understand from a human readability standpoint. Other than that I thought the mathematical functions for the calculator were fairly easy to understand and implement in code. Overall, this was a fun lab that was neither easy or hard but rather simply as expected. In general, I like to spend a day or so thinking about a program before working on it. It helps to have a semi-blueprint in your mind of what you're going to do and how to do it.