**Introduction to the C Compiler**

**Lab 2**

**Section X**

**SUBMITTED BY:**

**NATHAN SHULL**

**SUBMISSION DATE:**

**SEPTEMBER 9, 2016**

**Lab Problem**

The purpose of this lab is to become familiar with the c compiler. Also, to create a program that contains input and output commands that will calculate the area and volume of a rectangle and prism, given the dimension inputs from the user. The objective was to learn how to write simple lines of code in c and to compile it within Cygwin.

**Analysis**

The problem states that I must compute the area, and later volume, of a rectangle/prism given the width, height, and depth provided by the user. Therefore, the problem input consists of any two/three decimal numbers and the output is one decimal number. The formula used for this was (width\*height) for area, or (width\*height\*depth) for volume, as these are basic equations learned in geometry.

**Design**

Our problem was to create a simple program in c designed to perform simple math operations. I divided the problem into a few smaller steps.

1. Get the dimensions of the length/width/height
2. Assign each dimension to a variable
3. Compute area/volume
4. Display area/volume back to user

Using the outline above, I designed my program in steps. I first began by selecting the variable data types. The numeric data was assigned to decimal data types. I modified the source code to utilize the scanf ( ) command to process each input variable, especially the z dimension that was later added in order to find the volume of a prism.

**Testing**

In order to test the validity of my code, I inserted simple number, like 2 and 3, to find the area. Of course the area of a 2 x 3 rectangle is 6. However, I didn’t initially get that result. I found that when copying the skeleton from the pdf, there are blank characters that you can’t see and cause errors in the code. I had to go back and re write the skeleton code. Another problem I ran into while testing was when attempting to calculate volume. Again, I used simple numbers like 2, 3, and 4. This should yield a volume of 24. However, when compiling the code, it resulted in 6, and never gave me the option to input depth. I found that this was because I had my entire line of scanf ( ) in double quotes, rather than just the %d.

**Comments**

In doing this lab, I learned the importance of checking and being aware of every little thing that I write. In only just a few lines of code, I was able to screw up and create errors and unwanted results. I learned a lot about the c compiler, and how to compile the code and Notepad + + and then run it in Cygwin.

