**Solving Simple Problems in C**

**LAB # 2**

**SECTION # 11**

**Kenneth Schueman**

**SUBMISSION DATE: 9/16/2021**

**9/13/2021**

## Problem 1: Creating Your Own Program

Modify the lab02-skeleton.c file to display your full name as well as the course title and today’s date, each on a new line.

# Analysis

Since the skeleton of the code is already given simply modifying the .c file will be simple.

# Design

Changing the placeholder text for the program to output the data I need as seen in Figure 1 screenshot.

# Testing

Test once to make sure I didn’t delete any important syntax, and then clear for screenshot.

# Comments

Problem was a nice way to get my code sided brain functioning.

# Screen ShotsGraphical user interface, text, application, email Description automatically generated

Figure 1

# Text Description automatically generated

Figure 2

## Problem 2: A Simple Program with Input

int x, y;

printf(“Enter a width: ”);

scanf(“%d”, &x);

printf(“Enter a height: ”);

scanf(“%d”, &y);

printf(“A %d by %d rectangle’s area is %d\n”, x, y, x\*y);

Use this given code above and modify it to account for rectangular cubes.

# Analysis

I recognized that all I need to do was add another int variable for the third dimension and then multiply them all together to get the volume of a three-dimensional rectangle.

# Design

Added another int and respective printf and scanf statements to correctly output users request as seen in figure 3 below.

# Testing

Tested multiple different integer values to make sure that the program was outputting current mathematics.

# Comments

There was no need to grab the formula from outside source because of previous knowledge.

# Screen Shots

Figure 3

# 

Figure 4

## Problem 3: A Mysterious Output

In this problem we are given a code that seems to be outputting multiple incorrect answers and are tasked with fixing each line of code.

# Analysis

There are multiple lines of code that have a variety of errors.

# Design

All errors have been corrected and the program is now outputting the correct int as seen in figures 5 & 6.

# Testing

I recompiled and ran the program after reach attempt to fix the string in order not to get ahead of myself and make sure my corrections worked.

# Comments

These errors are very often oversights from even experienced coders because I have made quite a few of these errors myself.

# Screen Shots

Figure 5

# 

Figure 6

## Problem 4: Simple Arithmetic

Part 1.

* 1. int = 6427 + 1725
  2. int = (6971 \* 3925) – 95
  3. double = 79 + 12 / 5
  4. double = 3640.0 / 107.9
  5. int = (22 / 3) \* 3
  6. int = 22 / (3 \* 3)
  7. double = 22 / (3 \* 3)
  8. double = 22 / 3\* 3
  9. double = (22.0 / 3) \* 3.0
  10. int = 22.0 / (3 \* 3.0)
  11. double = 22.0 / 3.0 \* 3.0

Part 2

a. Calculate the area of a circle with circumference 23.567

* 1. Convert 14 feet to meters.
  2. Convert 76 degrees Fahrenheit to Celsius.

(1 ft. = 0.3048 m Tc = (TF - 32) / 1.8)

# Analysis

This problem is asking for multiple arithmetic functions to be completed in a single program and for all for there outputs to be organized. The key to this problem will be to stay organized in my lines of code.

# Design

I spaced each different part of the problem into its own section with comments as seen in figure 7 and the consequence of this extra effort is a neatly printed program as seen in figure 8

# Testing

After each part of the function, I paused and recompiled and ran the program to make sure the equations and arithmetic was outputting correctly and neatly.

# Comments

This problem was very tedious simple because of the amount of looking back and forth inorder to copy the equations correctly.

# Screen Shots

Figure 7

# 

Figure 8

## Problem 5: Working with I/O – Pythagorean Theorem

Creating a program that calculates the missing side of the triangle using Pythagorean’s theorem (a + b = c) ^2.

# Analysis

At first the problem seemed to be simple but proved to be much more difficult then intended, I know I need to square both of my inputted integers and then run them through a square root function.

# Design

I ended up doing all calculations in a separate function and then assigning it to c so the sqrt function could be kept simple as seen in figure 9.

# Testing

There was a lot of testing because of mysterious outputs being caused by integer overflow.

# Comments

The defining and limits of the sqrt function caused some headache.

# Screen Shots

Graphical user interface, text

Description automatically generated

Figure 9

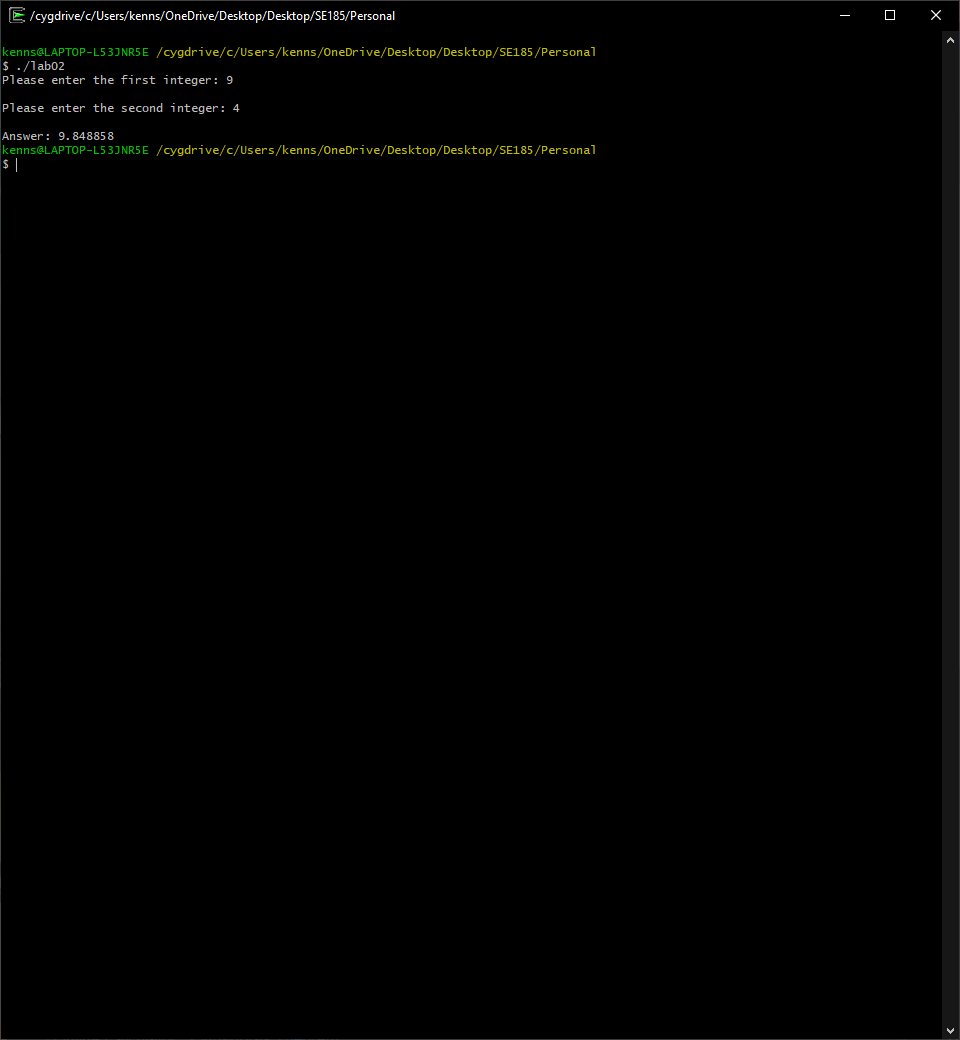


Figure 10