ENR 161

Ch3 Variables & Arithmetic Homework

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Grade \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Exercise 1:

The following exercise is designed to show you the importance of using the minimum data type size necessary for your needs in order to obtain the fastest calculations possible. Recall from lecture that some data types have larger memory allocations, and can therefore hold larger numbers but as a consequence, typically take longer to process calculations with. It is important to know exactly how much more impact certain data types have than others.

Setup:

Open your Arduino IDE

Open the **DataTypeSpeeds** example from Github. Upload and run the program, be sure to open the Serial Monitor.

Complete the questions below.

1. How many milliseconds did each set of assignments take?

1. byte
2. int
3. float
4. double

2. Is there a significant difference between a byte and an int?

3. Why do the float and double assignments take significantly longer?

## Exercise 2:

The following exercise is a simple example showing how to identify how many bytes of memory a variable type uses in memory.

Setup:

Open your Arduino IDE

Open the **DataTypeSizes** example from Github. Upload and run the program, be sure to open the Serial Monitor.

As it stands, the example shows the size of a **byte** type. Under each of the following comment lines, add the code to show the size of each variable data type. Submit your final sketch to your Github repository to complete this exercise.

## Exercise 3:

The following exercise is designed to give you a better understanding of the scope of variables as it pertains to function calls. Read through the code before executing and reading the output. If you read the output first, it may be more confusing so it is better to read the code and get an idea for what is going on. When you read through the code you should see that there is a global variable assigned with an initial value of 10. There is also member variable created in setup() assigned with the value 10. This member variable is not visible to any other function unless it is passed to that function directly. Later, the custom function is called and the member variable is passed to it. Recall from lecture that variables are passed by value only, this means that the custom function will have access to the value of the member variable but won’t be able to change that value directly.

Setup:

Open your Arduino IDE

Open the **VariableScope** example from Github. Upload and run the program, be sure to open the Serial Monitor.

For this exercise, write/paste the output from running the program in the space below. The important part of this exercise is that you understand and read all of the comments so add a paragraph after your output explaining the flow of the program and how it works.

## 

## 

## 

## Exercise 4:

For the following exercise you will be using your knowledge of Arduino arithmetic to perform the calculation for the volume of a cylinder to the nearest decimal.

Setup:

Open your Arduino IDE

Open the **CylinderVolumeCalc** example from Github. Add a calculation for the volume of a cylinder to the nearest decimal. *Hint: You will need to adjust the variable data types*

Submit your sketch named **CylinderVolumeCalc** to your github repository to complete this exercise.