Programming Assignment

Lesson 3

CISC 071

By

Jeremy Reuwer

Date: 2/13/2019

Purpose

* To learn about variables

Rubric

* Correctness: 2 Points. Program should work as specified
* Input/Output: 2 Points. Show the inputs and outputs to the program. If Applicable multiple examples needed
* Coding style/Comments: 1 Points.

Project Assignment

* Use the setup function to set the serial monitor and print “Serial monitor setup done”
* Define Variables in the setup function
  + A variable to store floating point value called “Average”
  + Variables to store integer values called “Score1”, “Score2”, & “Score3”
* Assign value of 2, 7 and 9 to Score1, Score2 and Score3 respectively
* Print the values of these three score variables to the serial monitor.
* Calculate the average of the scores, assign it to “Average”
* Print the value of the variable “Average” to the serial monitor
* Leave the loop function empty, since we only want to do the above calculations once

Learning Notes:

## Variables

## A variable is used in programming to store a value that may change during the life of the program (or sketch). Memory is set aside for storing the variable and the variable is given a name which allows it to be accessed in the sketch.

## One example of a variable is if you were to write a sketch that keeps the total of a teams score in a sports match. The value that the variable is holding (i.e. the teams score) would be displayed on a screen while the sports match is being played. As the score increases, the value that the variable holds will be increased and the display on the screen would be updated showing the new value that the score variable now holds.

## In this example, the score value may change several times during the match and the time that the sketch is running. It is therefore variable as opposed to a fixed or constant value.

## Using a Variable

## The following sketch called variables demonstrates the use of a variable.

## void setup() {

## int count;

## Serial.begin(9600);

## count = 0;

## Serial.println(count);

## count = 1;

## Serial.println(count);

## count = 2;

## Serial.println(count);

## }

## void loop() {

## }

## Variable Definition

## The following statement from the above sketch is a variable definition as it defines the variable type and name:

## int count;

## Variable Type

## In the above example, the variable type is int. This means that the variable can hold an integer value. An integer is a whole number – i.e. not a fraction. For example, the following numbers are integers 2, 5, 0, 100, 1024, -32, etc.

## Variable Name

## The above variable has been given the name count. This variable can now be referenced or used in the sketch by using the name "count".

## By giving the variable a type and name, space is made available in memory for this variable.

## Types of Variables

## The integer (int) variable type is only one type of variable. An example of a different variable type is a float or floating point variable. A floating point variable is used to store a number that contains a fraction, e.g. 1.45, 99.99, 362.5634, -200.21, etc. A floating point variable is defined in the same way as an integer variable, except that the float keyword is used instead of int as shown in the example below.

## float average;

## This floating point variable can now be assigned a floating point value e.g.:

## average = 1.2;

## To print or send a floating point value to the Arduino serial monitor window, the println() function can be used. A second parameter can be passed to the println() function to specify the number of decimal places that must be printed as shown in the floats sketch below.

## void setup() {

## float average;

## Serial.begin(9600);

## 

## average = 12.3299;

## Serial.println(average);

## Serial.println(average, 4);

## }

## void loop() {

## }

## The above sketch assigns a value of 12.3299 to the average floating point variable. When the value of the variable is sent to the serial monitor window, we can see that println() automatically rounds the number off to two decimal places.

## The second time that println() is used to send the value of the variable to the serial monitor window, the number of decimal places is specified as 4. This is done by passing a second parameter value of 4 to the println() function.

## Naming Variables

## Variables can be given any name that you like, so long as it sticks to the rules set out below. It is best to give variables meaningful names that help you and others to understand the sketch better, e.g. score\_total would be a good variable name for a variable that is storing the running total of a score in a sports match.

## Variable Naming Rules

## Variables can consist of any letters (a to z and A to Z)

## Variables can contain the numbers 0 to 9, but may not start with a number, e.g. 3var is not allowed, but var3 is allowed

## Variables may not have the same names as Arduino language keywords, e.g. you can not have a variable named int

## Variables must have unique names i.e. you can not have two variables with the same name

## Variable names are case sensitive, so Count and count are two different variables

## Variables may not contain any special characters, except the underscore (\_), e.g. top\_score

## Initializing Variables

## In the above example sketches, the variables that were used were first defined, and then assigned a value.

## A variable can also be assigned an initial value when it is defined as shown below:

## int total = 0;

## In this example, the integer value named total is defined and assigned a value of zero (0) in one statement.

**For further details refer to the Arduino programming reference guide**

<https://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf>

Program

void setup() {

Serial.begin(9600);

Serial.println("Serial monitor setup done");

int score1 = 2;

int score2 = 7;

int score3 = 9;

float averageScore;

Serial.println(score1);

Serial.println(score2);

Serial.println(score3);

averageScore = ((score1 + score2 +score3) / 3);

Serial.println(averageScore);

}

void loop() {

// put your main code here, to run repeatedly:

}

Inputs/Outputs

Serial monitor setup done

2

7

9

6.00