**Outline**

Implement the remaining Arduino beginner lessons. Modify these lesson programs to expand the use of “for” loops, “if” statements, and serial console input and output.

**Objectives**

* Use strings,
* Demonstrate the ability to manipulate string data in a computer program
* Use assignment statements correctly with arithmetic expressions in computer programs
* Use comparison operators (i.e., equal to, not equal to, greater than, less than, greater than or equal to, less than or equal to),
* Write programs that incorporate user input,
* Write programs that incorporate screen output;
* Demonstrate the ability to manipulate and convert data in a computer program (e.g., parse strings; convert data types such as numeric to string, and string to numeric; convert ‘yes’ or ‘no’ to Boolean);
* Use sequence control structures to create programming solutions;
* Use repetition control structures to create programming solutions;

**Materials**

* Arduino Development Environment (IDE)
* Arduino proto board
* Web Documentation: “Smraza UNO Documentation” folder in the “ICS3C0” Repository
  + Getting started guide.pdf
  + Public\_materials🡪Ebook🡪Arduino book.pdf
  + Lessons Folder

**Level 1: LED Trailing Effects**

1. Implement the lesson titled “LED Trailing Effects”.
2. Locate on-line documentation that describes the C language “for” loop.
   1. What is the index and how is it used?

The “for” statement is used to repeat a block of statements which are enclosed in curly braces. Usually, an increment counter is used to increment (increase) or terminate the loop. The “for” statement is useful for any repetitive operation. It is often combined and used with arrays to operate on collections of data/pins.

* 1. When does the for loop end?

The loop ends when the condition becomes false

* 1. How is a “for” loop different from a “while” and a “do” loop?

A “while” loop loops continuously and infinitely until the expression in the parenthesis (brackets) becomes false. Something must change the tested variable, otherwise the while loop will never exit.

1. Research the “<” Comparitor.
   1. List all the other comparitors defined for the C language.

!= (not equal to)

<= (less than or equal to)

== (equal to)

> (greater than)

>= (greater than or equal to)

* 1. Modify the “for” loop to use the “<=” comparitor

1. Research the “++” incrementor operator.
   1. Explain how this is different from the “=+ 1” assignment.

The “++” incrementor operator increments the value of a variable by 1 whereas the “=+ 1” assignment sets a variable ( in this case, the variable is set to 1).

* 1. Modify the “for” loop to use the “=+” assignment

for(int x = 2; x < 100; x = x \* 1.5){

println(x);

}

**Level 2: Traffic Light**

1. Implement the lesson titled “Traffic Light”.
2. Research the C language “if” statement. Think about how to use it to select the different actions for different traffic lights.
3. Modify the program to replace the three blocks of code (green, yellow, red light) with a single “for” loop.
   1. Use an “if” statement to create a special case for the yellow light.
   2. Implement the countdown timer for the yellow light as a “nested” loop.

**Level 3: Traffic Control**

1. Locate on-line documentation for the implementation and use of the Arduino serial monitor.
2. Modify the traffic light to print status information to the serial monitor.
   1. Colour of the light
   2. Countdown index for the yellow light
3. Modify the traffic control program to read a string from the serial console. If the string is “red”, “green’, or “yellow”, the light should immediately change to that colour.

**Level 4: Fading**

1. Implement the lesson titled “Fading”.
2. Modify the main “for” loop to decrement from 255 down to 0.
3. Modify the program to read a number from serial monitor and use that number in the fade down to loop.
4. Modify the program to add code to check that the value read from the serial monitor is a valid number and not some random string.