**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.

a. What is the index and how is it used?

The index is the point in a series of numbers or strings that a loop is at, it is commonly used in loops to keep them going until it has looped a certain amount of specified times.

b. When does the for loop end?

When the condition is false the loop would end, in this case, when the number has reached the specified number (3).

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

The “for” loop repeats for a certain amount of time which is often determined by the programmer. They are different from “do” loops because “do” loops will loop a minimum of one time whereas a “for” goes on until the program tells it to stop. The “while” loop will loop for an infinite number of times until the statement is false.

3. Research the “<” Comparitor.

a. List all the other comparitors defined for the C language.

Comparison Operators:

x == y (x is equal to y)

x != y (x is not equal to y)

x< y (x is less than y)

x > y (x is greater than y)

x <= y (x is less than or equal to y)

x >= y (x is greater than or equal to y)

b. Modify the “for” loop to use the “<=” comparator

The program does the same operation as the statement with “<”, but it would also check if the value id equal to or less than.

4. Research the “++” incrementor operator.

a. Explain how this is different from the “=+ 1” assignment

The “++” operator repeatedly adds 1 to the to the given value, and =+1 makes the program do ans = ans + 1.

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

**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.

You can make different statements that can only work with some LEDs. For example, there can be different if statements for LEDs for red, yellow and green that can select a code.

3. Modify the program to replace the three blocks of code (green, yellow, red light) with a single “for” loop.

a. Use an “if” statement to create a special case for the yellow light.

b. 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.

a. Colour of the light

b. Countdown index for the yellow light

**Question 2 code**

int redled =3;

int yellowled = 5;

int greenled =7;

void setup(){

pinMode(redled, OUTPUT);

pinMode(yellowled, OUTPUT);

pinMode(greenled, OUTPUT);

Serial.begin(9600);

}

void loop(){

Serial.println("The GREEN LED is on.");

Serial.println(" ");

digitalWrite(greenled, HIGH); // turn on green LED

delay(5000); // wait 5 seconds

digitalWrite(greenled, LOW); // turn off green LED

for(int i=3;i>-1;i--){ // blinks for 3 times

Serial.println("The YELLOW LED is on.");

delay(500);// wait 0.5 second

digitalWrite(yellowled, HIGH); // turn on yellow LED

delay(500); // wait 0.5 second

digitalWrite(yellowled, LOW); // turn off yellow LED

//Serial.println(i," seconds left ");

Serial.print("seconds left: ");

Serial.println(i);

Serial.println(" ");

}

delay(500); // wait 0.5 second

Serial.println("The RED LED is on.");

Serial.println(" ");

digitalWrite(redled, HIGH);// turn on red LED

delay(5000); // wait 5 second

digitalWrite(redled, LOW);// turn off red LED

}

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.

**Question 3 code**

int redled =3;

int yellowled =5;

int greenled =7;

String ledColour = " ";

void setup()

{

Serial.begin(9600);

pinMode(redled, OUTPUT);

pinMode(yellowled, OUTPUT);

pinMode(greenled, OUTPUT);

}

void loop()

{

Serial.print("Enter a colour LED.");

while(Serial.available()==0);

Serial.print("You have selected: ");

ledColour = Serial.readString();

Serial.println(ledColour);

digitalWrite(redled, LOW);

digitalWrite(yellowled, LOW);

digitalWrite(greenled, LOW);

if (ledColour == "green"){

digitalWrite(greenled, HIGH); // turn on green LED

}

if (ledColour == "yellow"){{

digitalWrite(yellowled, HIGH); // turn on yellow LED

}

}

if (ledColour == "red"){ // wait 0.5 second

digitalWrite(redled, HIGH);// turn on red LED

}

}

**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.

**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.

a. What is the index and how is it used?

The index is the point in a series of numbers or strings that a loop is at, it is commonly used in loops to keep them going until it has looped a certain amount of specified times.

b. When does the for loop end?

When the condition is false the loop would end, in this case, when the number has reached the specified number (3).

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

The “for” loop repeats for a certain amount of time which is often determined by the programmer. They are different from “do” loops because “do” loops will loop a minimum of one time whereas a “for” goes on until the program tells it to stop. The “while” loop will loop for an infinite number of times until the statement is false.

3. Research the “<” Comparitor.

a. List all the other comparitors defined for the C language.

Comparison Operators:

x == y (x is equal to y)

x != y (x is not equal to y)

x< y (x is less than y)

x > y (x is greater than y)

x <= y (x is less than or equal to y)

x >= y (x is greater than or equal to y)

b. Modify the “for” loop to use the “<=” comparator

The program does the same operation as the statement with “<”, but it would also check if the value id equal to or less than.

4. Research the “++” incrementor operator.

a. Explain how this is different from the “=+ 1” assignment

The “++” operator repeatedly adds 1 to the to the given value, and =+1 makes the program do ans = ans + 1.

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

**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.

You can make different statements that can only work with some LEDs. For example, there can be different if statements for LEDs for red, yellow and green that can select a code.

3. Modify the program to replace the three blocks of code (green, yellow, red light) with a single “for” loop.

a. Use an “if” statement to create a special case for the yellow light.

b. 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.

a. Colour of the light

b. Countdown index for the yellow light

**Question 2 code**

int redled =3;

int yellowled = 5;

int greenled =7;

void setup(){

pinMode(redled, OUTPUT);

pinMode(yellowled, OUTPUT);

pinMode(greenled, OUTPUT);

Serial.begin(9600);

}

void loop(){

Serial.println("The GREEN LED is on.");

Serial.println(" ");

digitalWrite(greenled, HIGH); // turn on green LED

delay(5000); // wait 5 seconds

digitalWrite(greenled, LOW); // turn off green LED

for(int i=3;i>-1;i--){ // blinks for 3 times

Serial.println("The YELLOW LED is on.");

delay(500);// wait 0.5 second

digitalWrite(yellowled, HIGH); // turn on yellow LED

delay(500); // wait 0.5 second

digitalWrite(yellowled, LOW); // turn off yellow LED

//Serial.println(i," seconds left ");

Serial.print("seconds left: ");

Serial.println(i);

Serial.println(" ");

}

delay(500); // wait 0.5 second

Serial.println("The RED LED is on.");

Serial.println(" ");

digitalWrite(redled, HIGH);// turn on red LED

delay(5000); // wait 5 second

digitalWrite(redled, LOW);// turn off red LED

}

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.

**Question 3 code**

int redled =3;

int yellowled =5;

int greenled =7;

String ledColour = " ";

void setup()

{

Serial.begin(9600);

pinMode(redled, OUTPUT);

pinMode(yellowled, OUTPUT);

pinMode(greenled, OUTPUT);

}

void loop()

{

Serial.print("Enter a colour LED.");

while(Serial.available()==0);

Serial.print("You have selected: ");

ledColour = Serial.readString();

Serial.println(ledColour);

digitalWrite(redled, LOW);

digitalWrite(yellowled, LOW);

digitalWrite(greenled, LOW);

if (ledColour == "green"){

digitalWrite(greenled, HIGH); // turn on green LED

}

if (ledColour == "yellow"){{

digitalWrite(yellowled, HIGH); // turn on yellow LED

}

}

if (ledColour == "red"){ // wait 0.5 second

digitalWrite(redled, HIGH);// turn on red LED