**Programming Python on the Raspberry Pi**

**Lesson One: Input & Output**

**Objectives:**

There are 3 objectives to this lesson:

1. Use the print function to display text to the screen.
2. Use the input function to ask the user for information.
3. Use the “+” operator to concatenate strings and variables

**Program Description:**

This program will print to the display the message “Hello World”. Next the user will be asked their first name and then asked their last name. Finally the program will display the message “Hello <firstname> <lastname>. I am pleased to meet you.” where the <firstname> and <lastname> are the values entered by the user.

**Example 1 Source:**

#!/usr/bin/env python3

print('Hello World')

first=input('Please enter your first name: ')

last = input('Please enter your last name: ')

print('Hello ' + first + ' ' + last + '. I am pleased to meet you.')

**Lesson Two: A simple game**

**Objectives:**

There are 5 objectives to this lesson:

1. Importing function from the standard library
2. Converting string input to numeric using the int function
3. Using the random number generator
4. Using the While loop
5. Using the if – elif – else statement

**Program Description:**

Using the random number generator, the program will select a number between 1 and 100. The user will be prompted for their guess. If the number entered by the user is too low, the program will tell the user that the number is too low and ask the user to guess again. If the number is too high, report that the guess was too high and guess again. If the number entered equals the selected number, report success, select another number, and restart the game. If the user enters “0” then the game ends.

**Example 2 Source:**

#!/usr/bin/env python3

from random import randint

var = 1

while var == 1:

 myNum = randint(1,100)

 Guess = -1

 print('I am thinking of as number between 1 and 100. Can you guess it?')

 print('Enter 0 to quit the game.')

 while Guess != myNum:

 Guess = int(input('What is your guess? '))

 if Guess == 0:

print('Goodbye!')

 quit(0)

 elif Guess == myNum:

 print("That's correct!!")

 elif Guess > myNum:

 print('Your number is too high.')

 elif Guess < myNum:

 print('Your number is too low.')

 else:

 exit(0)

**Lesson Three: GPIO programming**

**Objectives:**

There are 3 objectives to the lesson

1. Creating a circuit connecting an LED to the Raspberry Pi
2. Using the Raspberry Pi GPIO library, controlling a pin
3. Using the for loop to flash the LED 10 times

**Program Description:**

One of the unique things about the Raspberry Pi is the availability of GPIO pins. In this lesson we will be using the GPIO pins to wire an LED light and make it flash.

**Wiring Diagram:**

Connect Pin 11 (GPIO 17) on the Raspberry Pi to the positive leg of an LED light. Connect Pin 9 (GND) on the Raspberry Pi to one leg of a resister. Connect the other leg of the resister to the negative leg of the LED light. Your circuit should look something like the picture below.

**Example 3 Source:**

#!/usr/bin/sudo python3

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

GPIO.setup(17, GPIO.OUT)

# blink GPIO17 10 times

for i in range(0,10):

 GPIO.output(17,GPIO.HIGH)

 time.sleep(.25)

 GPIO.output(17,GPIO.LOW)

 time.sleep(.25)

GPIO.cleanup()

**Lesson Four: Sensors**

**Objectives:**

There are 3 objectives to the lesson

1. Using the circuit from lesson 3, add a motion sensor to trigger the LED.
2. Create a Python function
3. Setting an interrupt to sense an event

**Program Description:**

Now that we can turn an LED on and off, let's add a motion sensor to our project to turn the LED on when motion is detected and turn off when the motion has stopped.

**Wiring Diagram:**

Using the circuit from lesson 3, add a passive infrared (PIR) motion sensor to your breadboard. Connect the ground to the existing ground, connect Pin 2 (5v) to the VCC, and connect Pin 13 (GPIO 27) to the PIR sensor out. Your circuit should look something like the picture below.



**Example 4 Source:**

#!/usr/bin/sudo python3

import RPi.GPIO as GPIO

import time

# Setup variables & GPIO pins

GPIO.setmode(GPIO.BCM)

PIR\_Pin = 27

LED\_Pin = 17

GPIO.setup(PIR\_Pin, GPIO.IN)

GPIO.setup(LED\_Pin, GPIO.OUT)

# Setup the callback function

def PIRDetect(PIR\_Pin):

 if GPIO.input(PIR\_Pin):

 print('Motion')

 GPIO.output(LED\_Pin,GPIO.HIGH)

 else:

 print('Stopped')

 GPIO.output(LED\_Pin,GPIO.LOW)

# Main program loop

print('PIR Module Test (CTRL+C to exit)')

GPIO.output(LED\_Pin,GPIO.LOW)

try:

 GPIO.add\_event\_detect(PIR\_Pin, GPIO.BOTH, callback=PIRDetect)

 while 1:

 time.sleep(10)

except KeyboardInterrupt:

 print('Finished')

 GPIO.cleanup()