Dedan Kimathi University of technology

Department of Electrical and Electronic Engineering

EEE2406: Instrumentation

Lab 1

Title: Introducing Python

October 2, 2015

1 Introducing Python

1.1 Objectives

The aim of this laboratory practice is to

- Introduce the student to programming in Python
- Equip the students with important skills in programming with Python to be used later in proceeding labs

1.2 Requirements

1. Laptop
2. Raspberry Pi with Python installed
3. PuTTY installed in laptop

2 Introduction

Python is a very high level language. The Raspberry PI was actually developed to run Python as its programming platform, hence the term PI in the name Raspberry PI. We are going to explore a bit of programming syntax used in Python and write some codes and run in the Raspberry PI. This will give us some background knowledge in Python programming. As is the tradition of learning every new programming language we are going to start exploring Python by writing the Hello, World! program in it. Here is the code for this program.

To start with, once you have obtained your Raspberry PI and managed to access it remotely on your laptop using PuTTY, which you are going to be guided through, run python in the terminal window by typing: python and pressing enter. In the python file type the following code.

```
#!/usr/bin/python

print "Welcome to the world of Python"
```

When you press enter your code will run and display the statement on the terminal window.

2.1 Entering Variables in Python

When you want to input some data into the Python for use later then here is how you declare and feed the variable. Assuming you want to input your name:

```
#!/usr/bin/python
myName = raw_input("Enter your name: ")
```

You can display this as:

```
#!/usr/bin/python
```

myName = raw_input("Enter your name: ")
print "Welcome to Python programming, ", myName

2.2 Common Arithmetic in Python

When adding, subtracting, multiplying and dividing variables in Python, the variables are captured as has been explained above. The following normal operators are used for respective operation:

$'+'$ for Addition
$'-'$ for subtraction
$'*'$ for multiplication
$'/'$ for division

Remember to use the appropriate data type for each variable such as int, float, double, etc. The following code for summing, subtracting, multiplying and dividing a set of two numbers should give more insight on how to go about this.

#!/usr/bin/python
firstNumber = float(raw_input("Enter the first number: "))
secondNumber =float(raw_input("Enter the second number: "))

print firstNumber, "added to ", secondNumber, "gives", firstNumber + secondNumber
print firstNumber, "subtract ", secondNumber, "gives", firstNumber secondNumber
print firstNumber, "multiplies by ", secondNumber, "gives", firstNumber * secondNumber
print firstNumber, "divided by ", secondNumber, "gives", firstNumber / secondNumber

2.3 Vectors and Matrices in Python

A matrix can be written in Python using square brackets and commas. For instance, the following is a 3 by 2 matrix.

X = [[1, 2], [4, 5], [5, 6]]

To select the first raw in matrix X you type: X[0]

To select the element in the first raw first column you type: X[0][0]

Addition and multiplication of matrices can be achieved by using nested loops. Let's explore this after discussing loops and flow control.

2.4 Flow control

Just like in most programming languages, while loop, for loop and if else if statements are used in Python for flow control.

1. IF statement
   Below is a simple code for showing how if condition works.

myNumber = 6

if myNumber == 6:

print "I just wanted to confirm that nobody has changed my number."

2. For loop
An example of a for loop to print out integers from 1 to 10 is show below.

myCount = range(1, 11)

for count in myCount:

print(count)

3. while loop
A while loop can be used as well. A simple program employing a while loop for displaying even numbers from 1 and 50 is shown below.

print These are the even number between 1 and 50:

$n = 1$

while $n \leq 50$:

if $n\%2 == 0$

print n

$n = n + 1$

print "There they are!"

Using loops, we can write elements into matrices and vectors. Below is a simple code to show how this can be done.

$i = 0$

myNumbers = [ ]

while $i < 10$:

print "At the top i is %d" % i

numbers.append(i)

$i = i + 1$

print "The new set of numbers: ", numbers

print "At the bottom i is %d" % i

print "The numbers: "

for num in numbers:

print num

Now we can explore the use of nested loops to add matrices. Consider the example below.

$X = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]$

$Y = [[10, 11, 12], [13, 14, 15], [16, 17, 18]]$
$result = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]$

we can now iterate through rows

for i in range(len(X)):

Then we iterate through columns

    for j in range(len(X[0])):
        $result[i][j] = X[i][j] + Y[i][j]$
for r in result:
    Print r

2.5 Functions in Python

Functions can be written in Python and called at any point of the program. Whenever a file starts with a function Python treats it as a function. Therefore, when you intend to write a script containing multiple functions then it is necessary that the first line be anything other than one of the functions. For instance, you can start by writing a meaningless command with no effect on the entire program like:

```
1;
```

The following code shows how you can define multiple functions and call then in the main function or in any other function.

```
#!/usr/bin/python
PI = 3.14159
def myCircumference(radius):
    return 2*PI*radius
def myArea(radius):
    return PI*radius*radius
def main():
    radius = float(raw_input("Enter the radius of your circle: "))
    print"The circumference for radius", radius, "is: ", myCircumference(radius)
    print"The area with radius ", radius, "is: ", myArea(radius)
main()
```

The last command main() runs the main function of the program and displays the area and circumference upon entering the radius by the user.

2.6 Plotting in Python

Python uses a library called matplotlib for plotting functions. For now, your raspberry PI does not have this library installed. So we are going to explore a more cumbersome

means of plotting just for illustration. The library which we are going to use is called turtle but it is not very suitable for plots which we are interested in.

The code below shows how you can plot a sine wave in Python using the turtle library.

```
#!/usr/bin/python
import math
import turtle
window = turtle.Screen()
myPlot = turtle.Turtle()
for angle in range(360):
        y = math.sin(math.radians(angle))
        myPlot.goto(angle, y*100)
window.exitonclick()
```

3 Exercises

1. Write a code which can allow a user to add, subtract, multiply and divide sets of two numbers as many times as he likes provided he enters 1 (or any character you choose), at the end of subsequent operations. (Hint: use while loop.)

2. Write a code for multiplying a 3 by 3 matrix by a 3 by 4 matrix. Choose the elements of your matrices as you please and use nested loops to achieve this.

3. Write a python program which prompts user to enter the necessary parameters and then it evaluates the surface area and volume of a frustum. Use multiple functions and display your answers with the main() function.

4. Write a code that uses the turtle library to plot the sine and cosine of angles within a given range of your choice. (Hint: use append to store your values of sine and cosine in an array. Also use an amplitude scaling factor to make visible curves. Lastly use setpos for plotting)

5. Write a python code for evaluating power in a sine wave for one cycle of the wave. Sample the wave at a frequency of 100 Hz. Use for loop for this exercise and remember to convert the wave into a discrete time system.

$$x = sin(2 * pi * t)$$

$$P_x = \sum_{k=0}^{100} |sin\,(2 * pi * k * Ts\,)\,|^2$$