

WORKSHOP REPORT ON PYTHON

**DEPARTMENTS OF COMPUTER SCIENCE &
ENGINEERING / INFORMATION TECHNOLOGY
(CSE/IT - 2ND YEAR)**

**BABU BANARASI DAS
NORTHERN INDIA INSTITUTE OF
TECHNOLOGY**

PROGRAMMING THROUGH PYTHON

Objective of the Course:

The course is designed to provide Basic knowledge of Python. Python programming is intended for software engineers, system analysts, program managers and user support personnel who wish to learn the Python programming language.

Learning Outcomes:

Problem solving and programming capability

Duration of the Course: 2 Days

Books Recommended for Reading and Reference:

1. Python-(Mark Lutz)
2. Python Training guide (BPB Publications)

Agenda

Day -1 (Phase -1)

- Getting familiar with Python
- Getting expertise on Python programming
- Development of Console Applications using Python

Day -1 (Phase -2)

- Physical Computing (Raspberry Pi) using Python
- Modules

Day -2 (Phase -3)

- Web Architecture of Python
- CGI Programming
- Transferring data using get and post methods
- Database handling using Python

Day – 2(Phase -4)

- Web Application Development (Project)
- Development of html page
- Development of Registration and Login modules

Python Programming

What is Python?

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Python was developed by Guido Van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Features of Python:-

Less line of codes than other languages (fast execution)

Platform independent

Open Source

Object Oriented

- It supports functional and structured programming methods
- It can be used as a scripting language or can be compiled to byte-code for building large Applications.
- It provides very high-level dynamic data types and supports dynamic type checking
- It supports automatic garbage collection
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java

How to start the Python?

First of all install the Python 2.7.12 on your system. After installation it will create a directory in C: drive named Python27. In Python27 there is a file python.exe. Copy the path of python.exe from address bar.

To set the path of python in “path” system variable:-

- Copy the path of python.exe from address bar.
- Right click on Computer then click on Properties.
- Click on Advanced System Settings then click on Environment Variables.
- Choose path variable and click on Edit button.
- Deselect the path and stuff a semicolon at last and paste the copied path.
- Now click on OK button.

1. First Python Program:-

Open the command prompt, type python and press enter key.

The python prompt will open. Now type the following code:-

print “Hello Python!”press enter key it will display following **output** :-

Hello Python!

2. Make simple calculator using python:-

On command prompt make a directory PythonProgs using md command. Then use cd command to open PythonProgs directory. Now type notepad SimpleCalc.py. The notepad editor will open and type the following code:-

```
a=input(“Enter first number : ”)
```

```
b=input("Enter first number : ")
print "Summation = ",(a+b)
print "Subtraction = "(a-b)
print "Multiplication = "(a*b)
print "Division = ",(a/b)
```

Now save the file SimpleCalc.py and close the file.

Now at command prompt type python SimpleCalc.py it display the following output:-

```
Enter first number: 10
Enter second number: 5
Summation = 15
Subtraction = 5
Multiplication = 50
Division = 2
```

Decision Making Statements:-

Use of if statement:-if is a keyword which is used for decision making. The syntax of if statement is given below:-

```
if condition:
Statement 1
Statement 2
```

Notes:-In Python there is no need to used curly braces { }. In python we use indentation to make code block.

Use of if – else statement:-if – else is the variation of if statement. We attach condition with if statement if given condition is true then if block code will executed and if the given condition is false then else block code will executed. The syntax of if – else statement is given below:-

```
if condition:
Statement 1
Statement 2
else:
Statement 3
Statement 4
```

3. Write a program in python to check the given number is even or odd.

```
n=input("Enter a number : ")
if n%2==0:
print "The number ",n,"is even"
else:
print "The number",n,"is odd"
```

Use of if-else ladder:-In Python language the switch statement is not worked. In Python we use if-else ladder in spite of using switch. The syntax of if-else ladder is given below:-

```
if condition1:
Statement 1
elif condition 2:
Statement 2
elif condition 3:
Statement 3
```

else:
Statement 4

4. Write a program in python for reverse counter, display the numbers from 10 to 1 using delay of 1second.

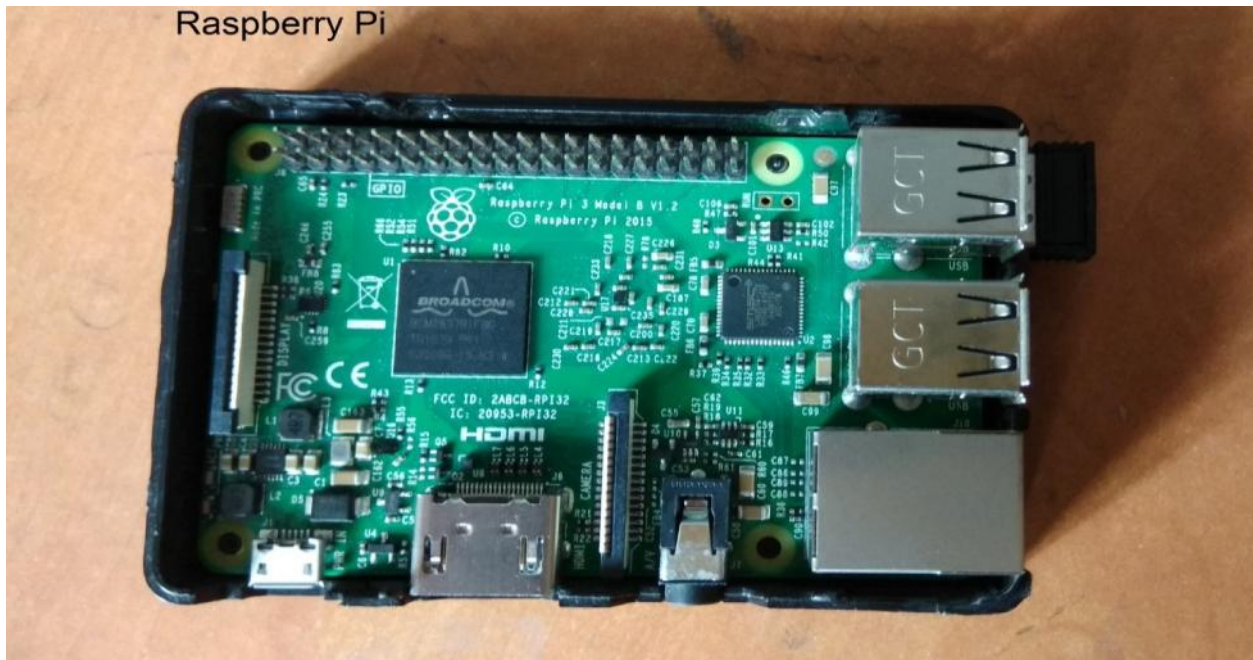
```
import time
i=10
while i>0:
    print i
    time.sleep(1)
    i=i-1
```

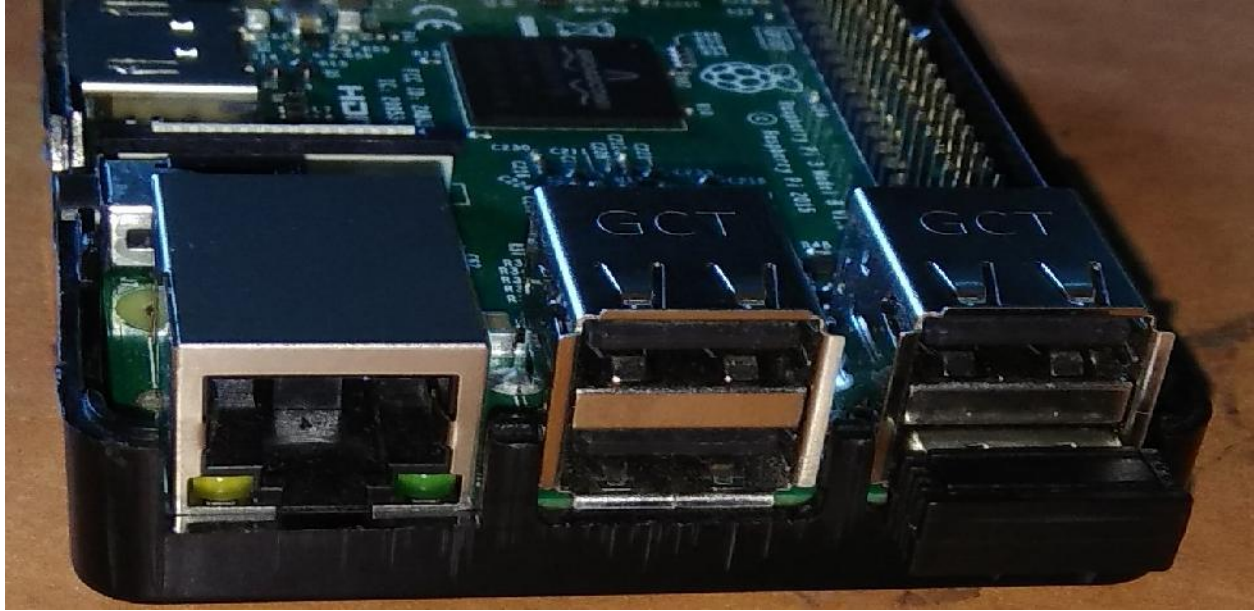
The above code will display the numbers from 10 to 1 in interval of 1 second.

5. Magic Age

```
Name=raw_input("Enter your name:")
Age=input("Enter your age :")
if Age<=50;
Age=Age+10
else:
Age=Age-10
print "Hello!", Name,"you look",Age,"years old"
```

INTRODUCTION TO RASPBERRY PI





Raspberry pi is a series of small single-board computers developed in United Kingdom by Raspberry foundation to promote teaching of basic computer science in different part of country. It has 40 pins, one HDMI and 4 port or socket.

Pin NUMBER	CONNECTION
2	VCC
3	Ground
33	Input 1(green bulb)
35	Input 2(Red bulb)

6. Program in Raspberry pi :-

```

From RPi import GPIO
From time import sleep
GPIO.setmode(GPIO.BOARD)
GPIO.setup(33,GPIO.out)
GPIO.setup(35,GPIO.out)
GPIO.output(33,GPIO.HIGH)
GPIO.output(35,GPIO.HIGH)
a=input ("Enter first number :")
b=input ("Enter second number :")
c=input ("Enter sum :")
d=a+b
if c==d:
    GPIO.output(33,GPIO.LOW)
    sleep(10)
    GPIO.output(33,GPIO.HIGH)
else:
    GPIO>output(35,GPIO.LOW)
    sleep(10)

```

```
GPIO.output(35,GPIO.HIGH)
```

Run the above code and give input 10, 10, 20 it will glow green bulb

7. Now make TestTempConv.py to test the TempConv.py module.

```
importTempConv
print "Enter 1 for c to f"
print "Enter 2 for f to c"
ch=input()
if(ch==1):
    c=input("Enter temperature in c")
    f=TempConv.ctof(c)
    print "Temperature in f =",f
elif (ch==2):
    f=input("Enter temperature in f")
    c=TempConv.ftoc(f)
    print "Temperature in c =",c
else:
    print "Invalid choice"
```

Web Application development using Python

CGI Programming

The Common Gateway Interface, or CGI, is a set of standards that define how information is exchanged between the web server and a custom script.

What is CGI?

The Common Gateway Interface, or CGI, is a standard for external gateway programs to interface with information servers such as HTTP servers.

The current version is CGI/1.1 and CGI/1.2 is under progress.

Web Browsing

- To understand the concept of CGI, let us see what happens when we click a hyper link to browse a particular web page or URL.
- Your browser contacts the HTTP web server and demands for the URL, i.e., filename.
- Web Server parses the URL and looks for the filename. If it finds that file then sends it back to the browser, otherwise sends an error message indicating that you requested a wrong file.
- Web browser takes response from web server and displays either the received file or error message.
- However, it is possible to set up the HTTP server so that whenever a file in a certain directory is requested that file is not sent back; instead it is executed as a program, and whatever that program outputs is sent back for your browser to display. This function is called the Common Gateway Interface or CGI and the programs are called CGI scripts. These CGI programs can be a Python Script, PERL Script, Shell Script, C or C++ program, etc.

Development of html page

```
<html>
<head>
<title>My First Project</title>
<style>
#outer
{
width:1000px;
height:1200px;
//border:1px solid white;
margin:0px auto;
border-radius:20px;
}
#menu
{
width:1000px;
height:50px;
background-color:white;
border-radius:20px 20px 0px 0px;
}
#slider
{
width:1000px;
height:250px;
background-color:aqua;
margin-top:5px;
}
#parent
{
width:1000px;
height:600px;
margin-top:5px;
}
#lside
{
width:300px;
height:600px;
background-color:aqua;
float:left;
}
#rside
{
width:700px;
height:600px;
background-color:lightgray;
float:right;
}
#footer
{
```

```

width:1000px;
height:270px;
border-radius:20px;
background-color:white;
margin-top:20px;
}
#menu ul
{
list-style-type:none;
}
#menu ul li
{
display:inline;
padding-left:40px;
}
#menu ul li a
{
font-size:30px;
text-decoration:none;
}
</style>
</head>
<body bgcolor="lime">
<div id="outer">
<div id="menu">
<ul>
<li><a href="index.html">Home</a></li>
<li><a href="#">About Us</a></li>
<li><a href="register.py">Register</a></li>
<li><a href="#">Contact Us</a></li>
<li><a href="#">Enquiry</a></li>
</ul>
</div>
<div id="slider">

</div>
<div id="parent">
<div id="lside"></div>
<div id="rside"></div>
</div>
<div id="footer">
</div>
</div>
</body>
</html>

```

Save the above page with name: index.html

Creation of registration page using python and html

```
#!C:\Python27\python.exe

print "Content-Type:text/html \n\n"
print """
<html>
<head>
<title>Registration Page</title>
</head>
<body>
<form action="code.py" method="post">
<h1>Registration Form</h1>
<hr/>
Name:
<input type="text" name="name"/><br/><br/>
Father Name:
<input type="text" name="fname"/><br/><br/>
Contact No:
<input type="text" name="contact"/><br/><br/>
Email:
<input type="text" name="email"/><br/><br/>
Password:
<input type="password" name="password"/><br/><br/>
<input type="submit" name="Register" value="Register"/>
</form>
</body>
</html>
"""
```

Save the above page with register.py

Now install wampserver 2.0

And host your offline server and copy and paste your project folder containing index.html and register.py in C:\wamp\www

Now open chrome and type localhost: 70 and check your project

Creating code to connect page with database

```
#!C:\Python27\python.exe
import MySQLdb
import cgi
print "Content-Type:text/html\n\n"
data=cgi.FieldStorage()
name=data.getvalue('name')
fname=data.getvalue('fname')
contact=data.getvalue('contact')
email=data.getvalue('email')
password=data.getvalue('password')
con=MySQLdb.connect("127.0.0.1","root","","mydb",3306)
```

```
cur=con.cursor()
query="insert into register values('"+name+"','"+fname+"','"+contact+"','"+email+"','"+password+"')"
cur.execute(query)
con.commit()
con.close()
print "<h2>Rgistration Successfull</h2>"
```

Save this file in “C:\wamp\www\newproject” with name “code.py”

The above command use two methods of database that are:-

GET and POST Methods

Must have come across many situations when you need to pass some information from your browser to web server and ultimately to your CGI Program. Most frequently, browser uses two methods two pass this information to web server. These methods are GET Method and POST Method.

Passing Information using GET method:

The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the ?character as follows:

<http://www.test.com/cgi-bin/hello.py?key1=value1&key2=value2>

Simple URL Example: Get Method

Simple URL, which passes two values to hello_get.py program using GET method. /cgi-bin/hello_get.py?first_name=Abhishek&last_name=Pandey

Simple FORM Example: GET Method

Example passes two values using HTML FORM and submit button. We use same CGI script hello_get.py to handle this input.

```
<form action="/cgi-bin/hello_get.py" method="get">
First Name: <input type="text" name="first_name"><br />
Last Name: <input type="text" name="last_name" />
<input type="submit" value="Submit" />
</form>
```

Passing Information Using POST Method

A generally more reliable method of passing information to a CGI program is the POST method. This packages information in exactly the same way as GET methods, but instead of sending it as a text string after in the URL it sends it as a separate message. This message comes into the CGI script in the form of the standard input.

END
