WORKSHOP KIT ON PYTHON

SOFTPRO LEARNING CENTER
(A UNIT OF SOFTPRO INDIA COMPUTER TECHNOLOGIES (P) LTD)

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
- List
- Tuples
- Dictionary
- Function
- Modules

Day -2 (Phase -3)

- Web Architecture of Python
- CGI Programming
- Transferring data using get and post methods
- Brush-up your database concept
- Database handling using Python

Day – 2(Phase -4)

- Web Application Development (Project)
- Development of Enquiry module.
- SMS API integration
- Development of Registration and Login modules
- Development of Discussion forum module
- Overview of Django framework
- Query Session

About The Company

Softpro India Computer Technologies (P) Limited is one of the growing IT companies in consulting and software development, founded in 2004 by technocrats from IIT Kanpur and IET Lucknow, and a **renowned member of UPDESCO.** A dynamic new generation software solution as well as networking product Development Company.Softpro India's advisory board is a combination of renowned people from academics and corporate on honorary basis like **Prof. Onkar Singh Yadav (VC- MMMUT), Prof. D S Yadav (Director- GEC, Banda), Prof G N Pandey (Ex Director- IET, Lucknow), Prof B B Singh (Ex. Principal, MMMEC), Er. ShashiKatiyar (Intel Corporation), Er. Deepak Sharma (Oracle Corporation), Er. Vikram Singh (Microsoft, US).**

The company has remarkable establishment in the field of software development with more than 200 clients (Domestic & International), with major clients like Medicounsel (US), Gas Authority of India Limited, Indian Oil Corporation Limited, Sahara India, Commander Works Engineer (CWE) & Garrison Engineers (Ministry of Defense).

"Softpro Learning Center", the training division of the company was established in 2008 with the clear vision to reduce the Technology Gap prevailing between IT students and IT professionals. We have come a long way since the start; the company is an ISO 9001: 2000 certified company. Softpro Learning Center has established itself as one of the most promising center for learning across UP and nearby states.

SLC services include imparting Summer Training to B.Tech students, Industrial Training to MCA/BCA students and Apprenticeship Program to pass out students of B.Tech/BCA/MCA. In the year 2015, we had a crowd of approx. 2000 students who did their summer training from Softpro making us the largest learning centers of North India. Moreover 60% of the total trainees of ours are placed in big brands like TCS and Wipro.

Our Mission:

Since its inception, Development and Educational Wings of **Softpro India Computer Technologies Pvt. Ltd.** with its goal oriented mission is perpetually achieving success.

Python Programming

What is Python: - Python is an interpreted, 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:-

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

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!
```

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
```

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
```

Write a program in python to find greatest number in three numbers.

```
a=input("Enter first number : ")
b=input("Enter second number : ")
```

```
c=input("Enter third number : ")

if a>b and a>c:

        print "Greatest Number = ",a

elif b>a and b>c:

        print "Greatest Number = ",b

else:

        print "Greatest Number = ",c
```

Develop a program in Python to calculate bill after taking units from user. Take the following logic to calculate the electricity bill;

For 1 to 150 units rate is Rs. 2.40, for next 150 (150 to 300) units rate is Rs. 3.00 and Rs. 3.20 for units above 300.

```
unit=input("Enter the number of units consumed : ")

if unit<=150:

        bill=unit*2.40

elif unit>150 and unit<=300:

        bill=(150*2.40)+(unit-150)*3

else:

        bill=(150*2.40)+(150*3)+(unit-300)*3.20

print "Your bill = ",bill
```

Iteration Statements in python:-

while loop:- The while is a keyword in python which works as loop control. The syntax of while loop is given below:-

```
initialization of loop counter

while condition:

        Statement 1

        Statement 2

        Updation of loop counter
```

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

```
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.

Write a program in python to find sum of digits of given number.

```
n=input("Enter a number to find sum of digits : ")

sum=0

while n>0:

        r=n%10

        sum=sum+r

        n=n/10

print "Sum of digits : ",sum
```

Output:

```
Enter a number to find sum of digits : 123

Sum of digits : 6
```

Write a program in python to find reverse of digits of given number.

```
n=input("Enter a number : ")

rev=0

while n>0:

        r=n%10

        rev=rev*10+r

        n=n/10
```

```
print “Reverse of digits = ”,rev
```

Write a program in python to check the given number is prime or not.

```
n=input(“Enter a number : ”)
c=0
i=1
while i<=n:
    ifn%i==0:
        c=c+1
    i=i+1
if c==2:
    print “The number”,n,”is prime”
else
    print “The number is not prime”
```

for loop:- The for is a keyword which work as loop control. Basically, the for loop in python work like foreach loop in java or C#. It is used for traversing of a sequence. Some examples using for loop are given below:-

Write a program in python to print the characters of given string.

```
str=”PYTHON”
for c in str:
    print c
```

Output:

```
P
Y
T
H
O
N
```

List in Python

The list is a most versatile datatype available in Python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type.

For example:-

```
list1=[“Rohit”,”Mohit”,”Shobhit”,”Mudit”]

list2=[10,20,30,40,50]

list3=[“a”,”e”,”i”,”o”,”u”]
```

Traversing the elements of list:-

Write a program in python to make a list and traverse the elements of list.

```
names=[“Rohit”,”Satyam”,”Brijesh”]

for element in names:

        print element
```

Output:

```
Rohit

Satyam

Brijesh
```

Python includes the following list functions:-

Sr.No.	**Function with description**
1.	cmp(list1, list2) Compares elements of both lists.
2.	len(list) Gives the total length of the list.
3.	max(list) Returns item from the list with max value.
4.	min(list) Returns item from the list with min value
5.	list(seq) Converts a tuple into list.
6.	list.count(obj) Returns count of how many times obj occurs in list
7.	list.extend(seq) Appends the contents of seq to list
8.	list.insert(index, obj) Inserts object obj into list at offset index
9.	list.remove(obj) Removes object obj from list
10.	list.reverse() Reverses objects of list in place
11.	list.sort([func]) Sorts objects of list, use compare func if given

Write a program in python to create a list of ten numbers by taking input from user, find maximum and minimum number of list.

```
list=[]
i=0
while i<10:
        item=input("Enter the number : ")
        list.insert(i,item)
        i=i+1
print "Maximum value of list: ",max(list)
print "Minimum value of list: ",min(list)
```

Write a program in python to create a list of students by taking input from user. Now display the names of students in ascending and descending order.

```
names=[]
n=input("Hoe many names you want to store in list? ")
print "Enter",n,"names"
i=0
while i<n:
        element=raw_input();
        names.insert(i,element)
        i=i+1
names.sort()
print "Names in ascending order"
for element in names:
        print element
names.reverse()
print "Names in descending order"
for element in names:
        print element
```

Tuple in Python

A tuple is a sequence of immutable Python objects. Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets.

For example:

tup1 = ('physics', 'chemistry', 1997, 2000)

tup2 = (1, 2, 3, 4, 5)

tup3 = ("a", "b", "c", "d")

The empty tuple is written as two parentheses containing nothing:

tup1 = ()

Basic Tuples Operations:-

Tuples respond to the + and * operators much like strings; they mean concatenation and repetition here too, except that the result is a new tuple, not a string.

Python Expression	**Results**	**Description**
len((1, 2, 3))	3	Length
(1, 2, 3) + (4, 5, 6)	(1, 2, 3, 4, 5, 6)	Concatenation
('Hi!',) * 4	('Hi!', 'Hi!', 'Hi!', 'Hi!')	Repetition
3 in (1, 2, 3)	True	Membership
for x in (1, 2, 3): print x	1 2 3	Iteration

Built-in type tuple functions:-Python includes the following tuple functions:

Sr. No.	Function with Description
1.	cmp(tuple1, tuple2) Compares elements of both tuples
2.	len(tuple) Gives the total length of the tuple
3.	max(tuple) Returns item from the tuple with max value.
4.	min(tuple) Returns item from the tuple with min value
5.	tuple(seq) Converts a list into tuple

Dictionary in Python

Each key is separated from its value by a colon (:), the items are separated by commas, and the whole thing is enclosed in curly braces. An empty dictionary without any items is written with just two curly braces, like this: {}.

Keys are unique within a dictionary while values may not be. The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples.

For example:-

dict = {'Name': 'Brijesh', 'Age': 34, 'Designation': 'Sr.Developer'};

Python includes following dictionary methods

Sr.No.	Methods with description
1.	dict.clear() Removes all elements of dictionary dict
2.	dict.copy() Returns a shallow copy of dictionary dict
3.	dict.fromkeys() Create a new dictionary with keys from seq and values set to value.
4.	dict.get(key, default=None) For key key, returns value or default if key not in dictionary
5.	dict.has_key(key) Returns true if key in dictionary dict, false otherwise
6.	dict.items() Returns a list of dict's (key, value) tuple pairs
7.	dict.keys() Returns list of dictionary dict's keys
8.	dict.setdefault(key, default=None) Similar to get(), but will set dict[key]=default if key is not already in dic
9.	dict.update(dict2) Adds dictionary dict2's key-values pairs to dict
10.	dict.values() Returns list of dictionary dict's values

Function in Python

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

As you already know, Python gives you many built-in functions such as print() and but you can also create your own functions. These functions are called user-defined functions

Defining a Function:-You can define functions to provide the required functionality. Here are simple rules to define a function in Python.

- Function blocks begin with the keyword def followed by the function name and parentheses ( ) .
- Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
- The first statement of a function can be an optional statement - the documentation string of the function or docstring.
- The code block within every function starts with a colon (:) and is indented.
- The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

Syntax of defining function:-

```
deffunctionname( parameters ):
  "function_docstring"
function_suite
return [expression]
```

Write a program to find greatest number in two numbers using function.

```
def greatest(a,b):
        if(a>b):
                return a
        else:
                return b
x=input("Enter first number : ")
y=input("Enter first number : ")
g=greatest(x,y)
print "Greatest Number = ",g
```

Write a program in python to find factorial of given number using 'Recursion'.

```
def fact(n):
        if n==0 or n==1:
                return 1
        else:
                return n*fact(n-1)
x=input("Enter a number to find factorial: ")
f=fact(x)
print "Factorial =",f
```

Modules in Python

A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.

Simply, a module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code.

Now make a module named MyCalc.py.

```
def add(a,b):
    return (a+b)
def sub(a,b):
    return (a-b)
defmult(a,b):
    return (a*b)
def div(a,b):
    return (a/b)
```

Now test the module MyCalc.py

Make the file Test.py

```
importMyCalc
x=input("Enter first number : ")
y=input("Enter second number : ")
print "Summation =",MyCalc.add(x,y)
print "Subtraction =",MyCalc.sub(x,y)
print "Multiplication =",MyCalc.mult(x,y)
print "Division =",MyCalc.div(x,y)
```

Make a module TempConv.py with two methods ctof() and ftoc() to convert temperature from centigrade to forenhite and forenhite to centigrade.

```
defctof(c):
    f=(9*c)/5+32
    return f
defftoc(f):
    c=(f-32)*5/9
    return c
```

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.

First CGI Program:

```
#!/usr/bin/python


print "Content-type:text/html\r\n\r\n"

print “<html>”

print “<head>”

print “<title>Hello Word - First CGI Program</title>”

print “</head>”

print “<body>”

print “<h2>Hello Word! This is my first CGI program</h2>”

print “</body>”

print “</html>”
```

GET and POST Methods

You 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

Here is a simple URL, which passes two values to hello_get.py program using GET method.

/cgi-bin/hello_get.py?first_name=Brijesh&last_name=Mishra

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

```
# Import modules for CGI handling

importcgi, cgitb


# Create instance of FieldStorage

form = cgi.FieldStorage()


# Get data from fields

first_name = form.getvalue('first_name')

last_name  =form.getvalue('last_name')


print "Content-type:text/html\r\n\r\n"

print "<html>"

print "<head>"

print "<title>Hello - Second CGI Program</title>"

print "</head>"

print "<body>"

print "<h2>Hello %s %s</h2>" % (first_name, last_name)

print "</body>"

print "</html>"
```

Output:

Hello Brijesh Mishra

Simple FORM Example: GET Method

This 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 the information in exactly the same way as GET methods, but instead of sending it as a text string after a ?in the URL it sends it as a separate message. This message comes into the CGI script in the form of the standard input.

Below is same hello_get.py script which handles GET as well as POST method.

```
#!/usr/bin/python

# Import modules for CGI handling

importcgi, cgitb

# Create instance of FieldStorage

form = cgi.FieldStorage()


# Get data from fields

first_name = form.getvalue('first_name')

last_name  =form.getvalue('last_name')

print "Content-type:text/html\r\n\r\n"

print "<html>"

print "<head>"

print "<title>Hello - Second CGI Program</title>"

print "</head>"

print "<body>"

print "<h2>Hello %s %s</h2>" % (first_name, last_name)

print "</body>"

print "</html>"
```

Let us take again same example as above which 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="post">

First Name: <input type="text" name="first_name"><br />

Last Name: <input type="text" name="last_name" />

<input type="submit" value="Submit" />

</form>
```

Database Access using Python

The Python standard for database interfaces is the Python DB-API. Most Python database interfaces adhere to this standard.

You can choose the right database for your application. Python Database API supports a wide range of database servers such as:

- GadFly
- mSQL
- MySQL
- PostgreSQL
- Microsoft SQL Server 2008
- Informix
- Interbase
- Oracle
- Sybase

Here is the list of available Python database interfaces: Python Database Interfaces and APIs .You must download a separate DB API module for each database you need to access. For example, if you need to access an Oracle database as well as a MySQL database, you must download both the Oracle and the MySQL database modules.

The DB API provides a minimal standard for working with databases using Python structures and syntax wherever possible. This API includes the following:

- Importing the API module.
- Acquiring a connection with the database.
- Issuing SQL statements and stored procedures.
- Closing the connection

We would learn all the concepts using MySQL, so let us talk about MySQLdb module.

What is MySQLdb?

MySQLdb is an interface for connecting to a MySQL database server from Python. It implements the Python Database API v2.0 and is built on top of the MySQL C API.

```
importMySQLdb
```

Database Connection:

Before connecting to a MySQL database, make sure of the followings:

- You have created a database TESTDB.
- You have created a table EMPLOYEE in TESTDB.
- This table has fields FIRST_NAME, LAST_NAME, AGE, SEX and INCOME.
- User ID "testuser" and password "test123" are set to access TESTDB.
- Python module MySQLdb is installed properly on your machine.

Example:

Following is the example of connecting with MySQL database "TESTDB"

```
#!/usr/bin/python

importMySQLdb

 # Open database connection

db = MySQLdb.connect("127.0.0.1","testuser","test123","TESTDB" )

 # prepare a cursor object using cursor() method

cursor = db.cursor()

 # execute SQL query using execute() method.

cursor.execute("SELECT VERSION()")

 # Fetch a single row using fetchone() method.

data = cursor.fetchone()

print "Database version : %s " % data

 # disconnect from server

db.close()
```

If a connection is established with the datasource, then a Connection Object is returned and saved into db for further use, otherwise db is set to None. Next, db object is used to create a cursor object, which in turn is used to execute SQL queries. Finally, before coming out, it ensures that database connection is closed and resources are released.

Creating Database Table

Once a database connection is established, we are ready to create tables or records into the database tables using execute method of the created cursor.

Let us create Database table EMPLOYEE:

```
#!/usr/bin/python

importMySQLdb

 # Open database connection

db = MySQLdb.connect("127.0.0.1","testuser","test123","TESTDB" )

 # prepare a cursor object using cursor() method

cursor = db.cursor()

 # Drop table if it already exist using execute() method.

cursor.execute("DROP TABLE IF EXISTS EMPLOYEE")

 # Create table as per requirement

sql = """CREATE TABLE EMPLOYEE (

         FIRST_NAME  CHAR(20) NOT NULL,

         LAST_NAME  CHAR(20),

         AGE INT,

         SEX CHAR(1),

         INCOME FLOAT )"""

cursor.execute(sql)

 # disconnect from server

db.close()
```

INSERT Operation

It is required when you want to create your records into a database table.

Example

The following example executes SQL INSERT statement to create a record into EMPLOYEE table:

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

```
importMySQLdb

 # Open database connection

db = MySQLdb.connect("127.0.0.1","testuser","test123","TESTDB" )

 # prepare a cursor object using cursor() method

cursor = db.cursor()

 # Prepare SQL query to INSERT a record into the database.

sql = """INSERT INTO EMPLOYEE(FIRST_NAME,

         LAST_NAME, AGE, SEX, INCOME)

         VALUES ('Mac', 'Mohan', 20, 'M', 2000)"""

try:

   # Execute the SQL command

cursor.execute(sql)

   # Commit your changes in the database

db.commit()

except:

   # Rollback in case there is any error

db.rollback()

 # disconnect from server

db.close()
```

READ Operation

READ Operation on any database means to fetch some useful information from the database.

Once our database connection is established, you are ready to make a query into this database. You can use either fetchone() method to fetch single record or fetchall() method to fetch multiple values from a database table.

- fetchone(): It fetches the next row of a query result set. A result set is an object that is returned when a cursor object is used to query a table.
- fetchall(): It fetches all the rows in a result set. If some rows have already been extracted from the result set, then it retrieves the remaining rows from the result set.
- rowcount: This is a read-only attribute and returns the number of rows that were affected by an execute() method.

The following procedure querries all the records from EMPLOYEE table having salary more than 1000:

```
#!/usr/bin/python

importMySQLdb

 # Open database connection

db = MySQLdb.connect("localhost","testuser","test123","TESTDB" )

 # prepare a cursor object using cursor() method

cursor = db.cursor()

 # Prepare SQL query to INSERT a record into the database.

sql = "SELECT * FROM EMPLOYEE \

      WHERE INCOME > '%d'" % (1000)

try:

   # Execute the SQL command

cursor.execute(sql)

# Fetch all the rows in a list of lists.

results = cursor.fetchall()

for row in results:

fname = row[0]

lname = row[1]

age = row[2]

sex = row[3]

income = row[4]

     # Now print fetched result

print "fname=%s,lname=%s,age=%d,sex=%s,income=%d" % \

            (fname, lname, age, sex, income )

except:

print "Error: unable to fecth data"
```

```
# disconnect from server

db.close()
```

Python Questionnaire

Q.1 Develop a program using Python to calculate volume and surface area of cuboid.

Q.2 Develop a program using Python to find roots of quadratic equation $ax^2+bx+c=0$.

Hint : Use formula x_1=(-b+math.sqrt(b*b-4*a*c))/(2*a) ,

x_2=(-b-math.sqrt(b*b-4*a*c))/(2*a)

Q.3 Write a Python program to convert given number of days to a measure of time given in years, weeks and days. For example 375 days is equal to 1 year 1 week and 3 days (ignore leap year).

Q.4 Write a Python program to convert the given binary number into decimal.

Q.5 Develop a program in Python to calculate bill after taking units from user. Take the following logic to calculate the electricity bill;

For 1 to 150 units rate is Rs. 2.40, for next 150 (150 to 300) units rate is Rs. 3.00 and Rs. 3.20 for units above 300.

Q.6 Develop a program in Python to reverse the digits of given number.

Q.7 Develop a program in Python to print the series of prime numbers from 1 to 100.

Q.8 Develop a program in Python to generate Fibonacci sequence up to n terms where the value of n is entered by user.

Q.9 Develop a program in Python to convert currency from Rupees to Dollar and Dollar to Rupees based on user choice. (Use if-else ladder).

Q.10 Develop a list in Python store five names in list and sort the names in ascending and descending order.

Q.11 Develop a list in Python store ten numbers in this list. Now take a number as input and search the given number in list.

Q.12 Develop a program using Python to take full name as input and display the short name. E.g. Ajay Pratap Singh as A.P.Singh

Q.13 Develop a program in Python to check the given string is Palindrome or not.

Q.14 Develop a program in Python to take a string as input. Find a substring and replace the given string with another string. Use replace function to do this.

Q.15 Develop a program in Python to find factorial of given number using ‘Recursion’.

Q.16 Develop a module TempConv having two functions CtoF() which convert temperature from centigradeto forenhite and FtoC() which converttemperature fromforenhite to centigrade. Test the given module.

Q.17 Design a module to represent a bank account. Include the following members and methods:-

Data Members

i.)Name of the depositor

ii.)Account Number

iii.)Type of Account

iv.)Balance Amount in Account

Methods

i.)To assign initial values

ii.)To deposit an amount

iii.)To Withdraw an amount after checking balance.

iv.) To display the name and balance.

Test this module.

Q.18 Develop a program in Python to ask user enter his name or any English word. Display total value of his name, total value means sum of all alphabets assuming A=1,B=2,……..Z=26. Test it with word “attitude”.

Q.19 Develop a program in Python to take as string as input and display all characters by using for loop.

Q.20 Develop a program using Python to take two tuples tup1 and tup2. Now combine tuples tup1, tup2 into another tuple tup3. Iterate the elements of tup3.

Q.21 Develop a web page using Python which show the following form to user and store the data given by user in a table (Use MySql as backend).

Name	
Mobile	
Email	
Address	
	Save

Name	Mobile	Email	Address

Q.22 Develop a Login page which ask user to enter his Login Id and Password and pressing Login button it should check the validation of Login Id and Password from database and display a Welcome page if user is valid user and display error message on login page itself if user is invalid.

Q.23 Develop a webpage using Python to ask user name and after click on submit button the name should be displayed on another web page. (Use the concept of session).

Q.24 Develop a web page using Python which should display the current date and time in following manner:-

Today's Date :dd/MM/yyyy

Current Time : HH:MM:SS

Q.25 Develop a web application using Python which can generate the given number of recharge voucher code in the following format:-

2394-5341-3412-9145

Q.26 Develop a Web Application which can display the result of a student on entering the RollNo on following manner:-

First Page:

Enter Your Roll Number:

GO

Second Page:

Roll No.:			Name:	
S. No.	Subjects	Marks Obtn	Max Marks	Grade

You have design the database to hold the above information yourself with reasonable column types and size.

Q.27 Develop a webpage using Python to perform simple arithmetic operations addition, subtraction, multiplication and division of two numbers.

Q.28 Develop a function which returns character represented by a ASCII number.

Q.29 Develop a program using Python to take date of birth as input and display the age on current date in years, months and days.

Q.30 Develop a program using Python to take date of birth as input and find the Lucky number. The lucky no. will be sum of all digits of date of birth and it must be a single digit. Based on calculated Lucky no. predict the future.

PHYSICAL COMPUTING WITH PYTHON (RASPBERRY PI)

TASK 1: ARITHMETIC GAME:

Connections: Use the PIN 33, 35, 37 as the GPIO output pins of the raspberry pi for Green Bulb, Red Bulb, Buzzer.

A User will play a game of arithmetic calculations. Here user will enter two numbers (NUM1) & (NUM2) and will enter the result There are two bulbs in the arrangement (RED and GREEN). For each correct answer Green bulb should blink and the score of user should get incremented and printed on the Python shell along with a message "Right Answer your score is __". Similarly for the incorrect answer the Red bulb should blink along with a buzzer indicating a wrong answer. The score should get decremented and a message "Wrong Answer your score is __" should be displayed.

TASK 2: TRESSPASSER CHECK:

Connections: Take a PIR Sensor and attach its output on PIN 37 of Raspberry Pi. Take an LED and attach it on PIN35 of the Raspberry Pi.

Objective: The object is to identify presence of any trespasser near your door. If PIR sensor detect an object the LED should turn ON and OFF (5 times) and you should get a message "Tresspasser Detected" on Python shell.

☺☺☺☺☺

For any queries related to python or any other subject/technology, drop us a mail at brijesh@softproindia.org or visit our website at www.softproindia.org

You can also us at +91 70808120170, +91 7080102011, +91 7080102008

Thank,

Brijesh Mishra

Sr. Manager(IT)

Softpro India