# K.D.K. COLLEGE OF ENGINEERING

Approved by AICTE, New Delhi DTE, Mumbai and Affiliated to R.T.M. Nagpur University, Nagpur

*Department of Computer Science & Engineering*

LAB MANUAL

**Subject**

**Computer Workshop - II**

#### Semester

**IV SEMESTER**

EXPERIMENT NO:- 1

Aim:- Study about python installation and demonstratiom

Theory:- Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Prerequisites You should have a basic understanding of Computer Programming terminologies. A basic understanding of any of the programming languages is a plus.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

• Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

• Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

• Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

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. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include –

• Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

• Easy-to-read − Python code is more clearly defined and visible to the eyes. • Easy-to-maintain − Python's source code is fairly easy-to-maintain.

• A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

• Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

• Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

• Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

• Databases − Python provides interfaces to all major commercial databases. • GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

• Scalable − Python provides a better structure and support for large programs than shell scripting.

First Python Program:

1. Open notepad and type following program Print (“Hello World”)

2. Save above program with name.py

3. Open command prompt and change path to python program location

4. Type “python name.py” (without quotes) to run the program.

Operators are the constructs which can manipulate the value of operands. Consider the expression 4 + 5 = 9. Here, 4 and 5 are called operands and + is called operator.

Python Variables: Declare, Concatenate, Global & Local

What is a Variable in Python?

A Python variable is a reserved memory location to store values. In other words, a variable in a python program gives data to the computer for processing.

Every value in Python has a datatype. Different data types in Python are Numbers, List, Tuple, Strings, Dictionary, etc. Variables can be declared by any name or even alphabets like a, aa, abc, etc.

How to Declare and use a Variable

Let see an example. We will declare variable "a" and print it.

a=100

print a

Types of Operator

Python language supports the following types of operators.

* Arithmetic Operators
* Comparison (Relational) Operators
* Assignment Operators
* Logical Operators
* Bitwise Operators
* Membership Operators

• Identity Operators.

Assignment 2

Aim: To study strings in Python

Theory:

String Literals

String literals in python are surrounded by either single quotation marks, or double quotation marks.

‘hello' is the same as "hello".

You can display a string literal with the print() function:

Example

print("Hello")

print('Hello')

Assign String to a Variable

Assigning a string to a variable is done with the variable name followed by an equal sign and the string:

Example

a = "Hello"

print(a)

## PROGRAM

## Aim:

Write a program to demonstrate different number data types in Python.

a=10; #Integer Datatype

b=11.5; #Float Datatype

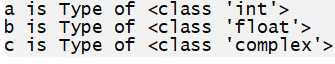
c=2.05j; #Complex Number

print("a is Type of",type(a)); #prints type of variable a

print("b is Type of",type(b)); #prints type of variable b

print("c is Type of",type(c)); #prints type of variable c

## Output:



## Aim:

Write a program to perform different Arithmetic Operations on numbers in Python.

## Source Code:

a=int(input("Enter a value")); #input() takes data from console at runtime as string.

b=int(input("Enter b value")); #typecast the input string to int.

print("Addition of a and b ",a+b);

print("Subtraction of a and b ",a-b);

print("Multiplication of a and b ",a\*b);

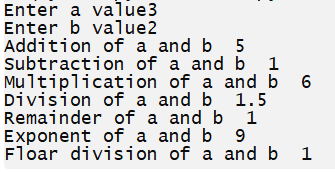
print("Division of a and b ",a/b);

print("Remainder of a and b ",a%b);

print("Exponent of a and b ",a\*\*b); #exponent operator (a^b)

print("Floar division of a and b ",a//b); # floar division

## Output:



## Aim:

Write a program to create, concatenate and print a string and accessing sub-string from a given string.

## Source Code:

s1=input("Enter first String : ");

s2=input("Enter second String : ");

print("First string is : ",s1);

print("Second string is : ",s2);

print("concatenations of two strings :",s1+s2);

print("Substring of given string :",s1[1:4]);

## Output:

## 

## Aim:

Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST 2017”

## Source Code:

import time;

ltime=time.localtime();

print(time.strftime("%a %b %d %H:%M:%S %Z %Y",ltime)); #returns the formatted time

'''

%a : Abbreviated weekday name.

%b : Abbreviated month name.

%d : Day of the month as a decimal number [01,31].

%H : Hour (24-hour clock) as a decimal number [00,23].

%M : Minute as a decimal number [00,59].

%S : Second as a decimal number [00,61].

%Z : Time zone name (no characters if no time zone exists).

%Y : Year with century as a decimal number.'''

## Output:

## 

## Aim:

Write a program to create, append, and remove lists in python.

## Source Code:

pets = ['cat', 'dog', 'rat', 'pig', 'tiger']

snakes=['python','anaconda','fish','cobra','mamba']

print("Pets are :",pets)

print("Snakes are :",snakes)

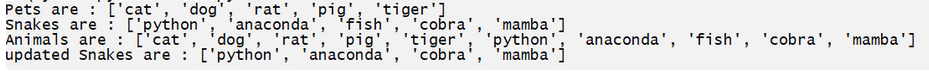
animals=pets+snakes

print("Animals are :",animals)

snakes.remove("fish")

print("updated Snakes are :",snakes)

## Output:



## Aim:

Write a program to demonstrate working with tuples in python.

## Source Code:

T = ("apple", "banana", "cherry","mango","grape","orange")

print("\n Created tuple is :",T)

print("\n Second fruit is :",T[1])

print("\n From 3-6 fruits are :",T[3:6])

print("\n List of all items in Tuple :")

for x in T:

print(x)

if "apple" in T:

print("\n Yes, 'apple' is in the fruits tuple")

print("\n Length of Tuple is :",len(T))

## Output:

## 

## Aim:

Write a python program to find largest of three numbers.

## Source Code:

num1 = int(input("Enter first number: "))

num2 = int(input("Enter second number: "))

num3 = int(input("Enter third number: "))

if (num1 > num2) and (num1 > num3):

largest = num1

elif (num2 > num1) and (num2 > num3):

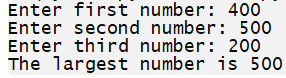
largest = num2

else:

largest = num3

print("The largest number is",largest)

## Output:



## Aim:

Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula: c/5 = f-32/9]

## Source Code:

print("Options are \n")

print("1.Convert temperatures from Celsius to Fahrenheit \n")

print("2.Convert temperatures from Fahrenheit to Celsius \n")

opt=int(input("Choose any Option(1 or 2) : "))

if opt == 1:

print("Convert temperatures from Celsius to Fahrenheit \n")

cel = float(input("Enter Temperature in Celsius: "))

fahr = (cel\*9/5)+32

print("Temperature in Fahrenheit =",fahr)

elif opt == 2:

print("Convert temperatures from Fahrenheit to Celsius \n")

fahr = float(input("Enter Temperature in Fahrenheit: "))

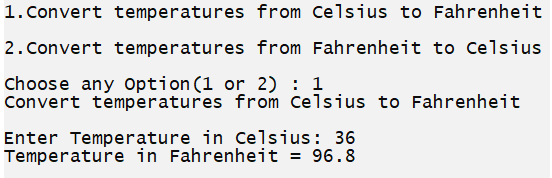
cel=(fahr-32)\*5/9;

print("Temperature in Celsius =",cel)

else:

print("Invalid Option")

## Output:



## Aim:

Write a Python program to construct the stars(\*) pattern, using a nested for loop

## Source Code:

'''Write a Python program to construct the following pattern, using a nested for loop

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

'''

n=5;

for i in range(n):

for j in range(i):

print ('\* ', end="")

print('')

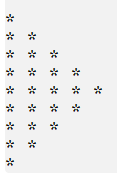
for i in range(n,0,-1):

for j in range(i):

print('\* ', end="")

print('')

## Output:



## Aim:

Write a Python script that prints prime numbers less than 20.

## Source Code:

print("Prime numbers between 1 and 20 are:")

ulmt=20;

for num in range(ulmt):

# prime numbers are greater than 1

if num > 1:

for i in range(2,num):

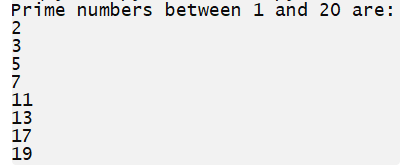
if (num % i) == 0:

break

else:

print(num)

## Output:



## Aim:

Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).

## Source Code:

base=float(input("Enter length of Base : "))

perp=float(input("Enter length of Perpendicular : "))

hypo=float(input("Enter length of Hypotenuse : "))

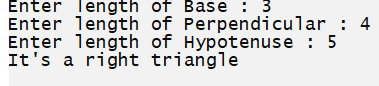
if hypo\*\*2==((base\*\*2)+(perp\*\*2)):

print("It's a right triangle")

else:

print("It's not a right triangle")

## Output:



## Aim:

Write a python program to define a module to find Fibonacci Numbers and import the module to another program.

## Source Code:

### fibonacci.py

# Fibonacci numbers module

def fib(n): # write Fibonacci series up to n

a, b = 0, 1

while b < n:

print(b, end =" ")

a, b = b, a+b

### week14.py

'''Write a python program to define a module to find Fibonacci Numbers and import the

module to another program'''

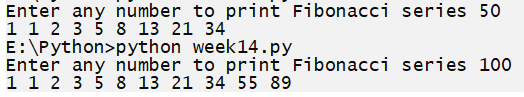
#import fibonacci module

import fibonacci

num=int(input("Enter any number to print Fibonacci series "))

fibonacci.fib(num)

## Output:



## Aim:

Write a python program to define a module and import a specific function in that module to another program.

## Source Code:

### arth.py

''' Arithmetic Operations Module with Multiple functions'''

def Add(a,b):

c=a+b

return c

def Sub(a,b):

c=a-b

return c

def Mul(a,b):

c=a\*b

return c

### week.py

'''Write a python program to define a module and import a specific function in that

module to another program.'''

from arth import Add

num1=float(input("Enter first Number : "))

num2=float(input("Enter second Number : "))

print("Addition is : ",Add(num1,num2))

print("Subtraction is : ",Sub(num1,num2)) #gives error:Not importing Sub function from arth Module

## Output:



## Aim:

Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.

## Source Code:

### file1.txt

This is python program

welcome to python

### week16.py

'''Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second

file'''

file1=input("Enter First Filename : ")

file2=input("Enter Second Filename : ")

# open file in read mode

fn1 = open(file1, 'r')

# open other file in write mode

fn2 = open(file2, 'w')

# read the content of the file line by line

cont = fn1.readlines()

#type(cont)

for i in range(0, len(cont)):

fn2.write(cont[i])

# close the file

fn2.close()

print("Content of first file copied to second file ")

# open file in read mode

fn2 = open(file2, 'r')

# read the content of the file

cont1 = fn2.read()

# print the content of the file

print("Content of Second file :")

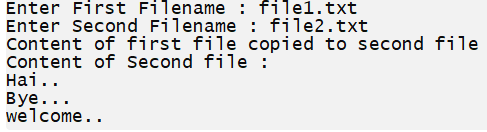
print(cont1)

# close all files

fn1.close()

fn2.close()

## Output:



## Aim:

Write a Python class to convert an integer to a roman numeral.

## Source Code:

'''Write a Python class to convert an integer to a roman numeral.'''

class irconvert:

num\_map = [(1000, 'M'), (900, 'CM'), (500, 'D'), (400, 'CD'), (100, 'C'), (90, 'XC'),(50, 'L'), (40, 'XL'), (10, 'X'), (9, 'IX'), (5, 'V'), (4, 'IV'), (1, 'I')]

def num2roman(self,num):

roman = ''

while num > 0:

for i, r in self.num\_map:

while num >= i:

roman += r

num -= i

return roman

num=int(input("Enter any Number :"))

print("Roman Number is : ",irconvert().num2roman(num))

## Output:





## Aim:

Write a Python class to implement pow(x, n)

## Source Code:

class py\_pow:

def powr(self, x, n):

if x==0 or x==1 or n==1:

return x

if x==-1:

if n%2 ==0:

return 1

else:

return -1

if n==0:

return 1

if n<0:

return 1/self.powr(x,-n)

val = self.powr(x,n//2)

if n%2 ==0:

return val\*val

return val\*val\*x

x=int(input("Enter x value :"))

n=int(input("Enter n value :"))

print("pow(x,n) value is :",py\_pow().powr(x,n));

## Output:



## Aim:

Write a Python class to reverse a string word by word.

## Source Code:

class py\_reverse:

def revr(self, strs):

sp=strs.split()

sp.reverse()

res=" ".join(sp)

return res

str1=input("Enter a string with 2 or more words : ")

print("Reverse of string word by word: \n",py\_reverse().revr(str1));

## Output:

