PYTHON PROGRAMMING

LAB MANUAL

PARUL UNIVERSITY
FACULTY OF ENGINEERING AND TECHNOLOGY
COMPUTER SCIENCE & ENGINEERING
WAGHODIA, VADODARA

PARUL UNIVERSITY

FACULTY OF ENGINEERING AND TECHNOLOGY

PARUL INSTITUTE OF TECHNOLOGY

Department of Computer Science & Engineering

LAB MANNUAL

(EVEN SEMESTER)

SUBJECT CODE: 03105255

SUBJECT NAME: Python Programming

YEAR/SEM: II/IV

DEPARTMENT: CSE

PREFACE

It gives us immense pleasure to present the first edition of *Python Programming Practical Book* for the B.Tech. 2nd year students for PARUL UNIVERSITY.

The Python programming theory and laboratory courses at **PARUL UNIVERSITY, WAGHODIA, VADODARA** are designed in such a way that students develop the basic understanding of the subject in the theory classes and then try their hands on the experiments to realize the various logical phenomena learnt during the theoretical sessions. The main objective of the Python Programming laboratory course is: **Learning Python Programming**.

The objective of this ***Python Programming*** is to provide a comprehensive source for all the experiments included in the Python Programming course. It explains all the aspects related to control structure, exception handling etc. It also gives sufficient information on how to interpret and discuss the obtained results.

We acknowledge the authors and publishers of all the books which we have consulted while developing this Practical book. Hopefully this *Python Programming Book* will serve the purpose for which it has been developed.

Instructions to students

1. The main objective of the Python Programming laboratory is: ***Learning concepts of Python Language***
2. Be prompt in arriving to the laboratory and always come well prepared for the experiment.
3. Every student should have his/her individual copy of the *Python Programming Practical Book*.
4. Every student have to prepare the notebooks specifically reserved for the Python Programming practical work: *"Python Programming Book"*
5. Every student has to necessarily bring his/her *Python Programming Practical Book*, *Python Programming Practical Class Notebook* and *Python Programming Practical Final Notebook*, when he/she comes to the Practical to perform the experiment.
6. Do not forget to get the information of your next allotment (the experiment which is to be performed by you in the next laboratory session) before leaving the laboratory from the Technical Assistant.
7. The grades for the Python Programming practical course work will be awarded based on your performance in the laboratory, regularity, recording of experiments in the *Python Programming Practical Final Notebook*, lab quiz, regular viva-voce and end-term examination

CERTIFICATE

This is to certify that

Mr./Ms.…… with

enrolment no. ……has successfully completed his/her

laboratory experiments

In Python Programming laboratory during the academic year……

Date: *Signature of lab teacher:*

Signature of HOD:

CONTENTS OF INDEX

Sr. no.	Title	No. of Hours	Page No.	Performance Date	Assessment Date	Marks out of 10	Sign
1	WAP to read and display the following information. Name, Address, Phone no.	2	7				
2	WAP to read two numbers from the keyboard and display the larger one on the screen.	2	8				
3	WAP to find, a given number is PRIME or NOT.	2	9				
4	Write a Function to swap values of a pair of integers.	2	10				
5	WAP to find N! Using function.	2	11				
6	WAP to print Fibonacci series of n numbers, where n is given by the programmer.	2	12				
7	WAP to read a set of numbers in an array & to find the largest of them.	2	13				
8	WAP to sort a list of names in ascending order.	2	14				
9	WAP to read a set of numbers from keyboard & to find the sum of all elements of the given array using a function.	2	15				
10	Calculate area of different geometrical figures (circle, rectangle, square, and triangle).	2	16-17				
11	WAP to increment the employee salaries on the basis of their designation. Use employee name, id, designation and salary as data member and inc_sal as member function	4	18				

12	Create two classes namely Employee and Qualification. Using multiple inheritance derive two classes Scientist and Manager. Take suitable attributes & operations. WAP to implement this class hierarchy.	4	19-20				
13	WAP to read data from keyboard & write it to the file. After writing is completed, the file is closed. The program again opens the same file and reads it.	4	21				
14	Case Study of Security Mechanism in Python	2	22				
15	Case Study of Graphics in Python	2	22				

Experiment 1

WAP to read and display the following information. Name, Address, Phone no.

Code:

```
Name=input("Enter Name:")
Address=input("Enter Address:")
Phone=input("Enter Phone no.:")
print("Name:",Name,"\n Address:",Address,"\nPhoneno.:",Phone)
```

Output:

```
Enter Name: "Smith"
Enter Address:"Baroda,Gujrat"
Enter Phone no.:999999999
Name:Smith
Address:Baroda,Gujrat
Phone no.: 999999999
```

Experiment 2

WAP to read two numbers from the keyboard and display the larger one on the screen.

Code:

```
A=input("Enter First number")
B=input("Enter Second number")
if(A>B):
    Print("A is Greater")
else:
    Print("B is Greater")
```

Output:

```
Enter First number3
Enter Second number5
B is Greater
```

Experiment 3

WAP to find, a given number is PRIME or NOT.

Code:

```
# Python program to check if the input number is prime or not

# take input from the user

# num =int(input("Enter a number: "))

# prime numbers are greater than 1

ifnum> 1:

  # check for factors

for i in range(2,num):

        if (num % i) == 0:

                print(num,"is not a prime number")

                print(i,"times",num//i,"is",num)  break

else:

print(num,"is a prime number")

# if input number is less than

# or equal to 1, it is not prime

else:

print(num,"is not a prime number")
```

Output:

```
Enter a number:400

400, is not a prime number
```

Experiment 4

Write a Function to swap values of a pair of integers.

Code:

```
def swap(s1, s2):
    s1 = 9
    s2 = 8
    return s2, s1


s1, s2 = swap(s1, s2)
print s1, s2
```

Output:

```
8 9
```

Experiment 5

WAP to find N! Using function.

Code:

```
# Python program to find the factorial of a number provided by the user.
# change the value for a different result
def Fact()
    num = int(input("Enter a number: "))
    factorial = 1
    # check if the number is negative, positive or zero
    if num< 0:
        print("Sorry, factorial does not exist for negative numbers")
    elif num == 0:
        print("The factorial of 0 is 1")
    else:
        for i in range(1,num + 1):
        factorial = factorial*i
        print("The factorial of",num,"is",factorial)
Fact()
```

Output:

```
Enter a number: 5
The factorial of 5 is 120
```

Experiment 6

WAP to print Fibonacci series of n numbers, where n is given by the programmer.

Code:

```
def recur_fibo(n):
   """Recursive function to  print Fibonacci sequence"""
if n <= 1:
        return n
else:
        return(recur_fibo(n-1) + recur_fibo(n-2))
# Change this value for a different result
nterms = int(input("How many terms? "))
# check if the number of terms is valid
if nterms<= 0:
        print("Plese enter a positive integer")
else:
        print("Fibonacci sequence:")
for i in range(nterms):
        print(recur_fibo(i))
```

Output:

```
How many terms? 4
Fibonacci sequence: 0 1 1 2
```

Experiment 7

WAP to read a set of numbers in an array & to find the largest of them.

Code:

```
alist=[-45,0,3,10,90,5,-2,4,18,45,100,1,-266,706]
largest=alist[0]
for large in alist:
    if large > largest:
        largest=large
print(largest)
```

Output: 706

Experiment 8

WAP to sort a list of names in ascending order.

Code:

```
# Program to sort alphabetically the words form a string provided by the user

# change this value for a different result

my_str = "Hello World welcome in python"

# uncomment to take input from the user

#my_str = input("Enter a string: ")

# breakdown the string into a list of words

words = my_str.split()

# sort the list

words.sort()

# display the sorted words

print("The sorted words are:")

for word in words:

        print(word)
```

Output:

```
The sorted words are:  Hello in python World welcome
```

Experiment 9

WAP to read a set of numbers from keyboard & to find the sum of all elements of the given array using a function.

Code:

```
deflistsum(numList):
    theSum = 0
    for i in numList:
        theSum = theSum + i
    returntheSum
list1=[]
for i in range(5):
    list1.append(int(input("Enter value:")))
print(list1)
```

Output:
```
Enter value: 1
Enter value: 11
Enter value: 3
Enter value: 9
Enter value: 7
31
```

Experiment 10

Calculate area of different geometrical figures (circle, rectangle, square, and triangle).

Code:

```
import math
##formulas for each geometric figure
def calc_square(a_side):
    square_area = a_side ** 2
    return square_area
def calc_rectangle(w_side, l_side):
    rect_area = l_side * w_side
    return rect_area


def calc_triangle(base, height):
    triangle_area = (base * height) / 2
    return triangle_area
def calc_circle(radius):
    circle_area = math.pi * radius ** 2
    return circle_area
##function determining which formula to calculate
def area_calc_logic(user_calc):
    if user_calc == "square":
        a_side = float(raw_input("Give length of side: "))
```

```
        print calc_square(a_side)
    elif user_calc == "rectangle":
        l_side = float(raw_input("Give the length: "))
        w_side = float(raw_input("Give the width: "))
        print calc_rectangle(w_side, l_side)
    elif user_calc == "triangle":
        base = float(raw_input("Give the length of base: "))
        height = float(raw_input("Give the height: "))
        print calc_triangle(base, height)
    elif user_calc == "circle":
        radius = float(raw_input("Give the radius: "))
        print calc_circle(radius)
    else:
        area_calc_logic(raw_input("Error, Re-enter input: "))
        print "This program will calculate/narea of some geometric shapes for you"
        print "Shapes available are squares, triangles, circles, and trapezoid"
        print "Enter square, rectangle, triangle, circle, or trapezoid"
area_calc_logic(raw_input("What area would you like to calculate? "))
```

Output:

What area would you like to calculate?" square"

Give length of side:4

Experiment 11

WAP to increment the employee salaries on the basis of their designation. Use employee name, id, designation and salary as data member and inc_sal as member function

Code:

```
class Employee(object):
    def __init__(self,name,designation,salary):
        self.name = name
        self.designation = designation
        self.salary = salary
    def inc_sal(self):
        if(self.designation == "Manager"):
        self.salary += self.salary * 0.15
    elif(self.designation == "Senior Manager"):
        self.salary += self.salary * 0.20
    else:
        print("Incorrect Designation");
emp = Employee("Manni","Manager",100000)
print(emp.salary)
emp.inc_sal()
print(emp.salary)
```

Output:

```
100000
115000.0
```

Experiment 12

Create two classes namely Employee and Qualification. Using multiple inheritance derive two classes Scientist and Manager. Take suitable attributes & operations. WAP to implement this class hierarchy.

Code:

```
class Employee(object):
        def __init__(self):
                #super(Employee, self).__init__()
                print("Employee ")


class Qualification (object):
        def __init__(self):
                super(Qualification, self).__init__()
                print("Qualification ")


class Scientist (Qualification,Employee):
        def __init__(self):
                super(Scientist, self).__init__()
                print("Scientist ")


Scientist();
class Manager (Qualification,Employee):
        def __init__(self):
```

```
        super(Manager, self).__init__()
        print("Manager ")
Manager();
```

Output:

```
Employee
Qualification
Scientist
Employee
Qualification
Manager
```

Experiment 13

WAP to read data from keyboard & write it to the file. After writing is completed, the file is closed. The program again opens the same file and reads it.

Code:

```
text_file = open("input.txt",'w')
text_file.write("My name is Mohit")
text_file.write("\nI am a C|EH")
text_file.write("\nI am also E|CSA")
text_file.close()
file_input = open("input.txt ",'r')
print(file_input.readline())
print(file_input.readline())
file_input.close()
```

Output

```
My name is Mohit
I am a C|EH
I am also E|CSA
```

Experiment 14

Case study of Security Mechanism in python

Provide brief overview of security mechanism in python with examples.

https://www.python.org/dev/peps/pep-0466/

Experiment 15

Case Study of Graphics in Python

http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/graphics.html