

S.Y.B.Sc.(I.T.) Lab Manual Python Programming

Practical No.1

1. Write the program for the following: (by using control statements and control structure) A. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.

Source Code:

```
from datetime import datetime

name=input('Please enter your name')

age=int(input('Please enter your age'))

res=(100-age)+datetime.now().year

print('hello ',name,'you will turn 100 years in ',res)
```

Output:

```
= RESTART: C:\Users\DELL\AppData\Local\Programs\Python\Python39\python.exe
Please enter your nameSYIT
Please enter your age18
hello SYIT you will turn 100 years in 2102
>>> |
```

B. Enter the number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.

Source Code:

```
n=int(input('Please enter a number'))

if(n%2==0):

    print(n,' is even')

else:

    print(n,' is odd')
```

Output:

<pre>= RESTART: C:\Users\DELL\AppData py Please enter a number5 5 is odd \\</pre>	<pre>= RESTART: C:\Users\DELL\AppData py Please enter a number8 8 is even \\</pre>
---	--

C. Write a program to generate the Fibonacci series.

Source Code:

```
n=int(input('Please the length for Fibonnacci Series: '))

a=0

b=1

print(a)

print(b)

for c in range(n-2):

    c=a+b

    print(c)

    a=b

    b=c
```

Output:

```
= RESTART: C:\Users\DELL\AppData\Local\Progr
py
Please the length for Fibonnacci Series: 8
0
1
1
2
3
5
8
13
\\
```

D. Write a function that reverses the user defined value.

Source Code:

```
num= int(input('Please enter a number'))

def reverse(n):
```

```
rev=0

while n!=0:

    rem=n%10

    rev= rev*10+rem

    n=n//10

return rev

print(reverse(num))

print("Reverse of a given number is: " + str(reverse(num)))
```

Output:

```
= RESTART: C:\Users\DELL\AppData\Loc
py
Please enter a number532
235
Reverse of a given number is: 235
>>> |
```

E. Write a function to check the input value is Armstrong and also write the function for Palindrome

Source Code:

```
num=int(input('Please Enter a number'))

def armstrong(n):

    arm=0

    while n!=0:

        rem=n%10

        arm=arm+rem*rem*rem

        n=n//10

    if(num==arm):

        print('no is armstrong')
```

else:

print('No is not a armstrong')

armstrong(num)

def palindrome(n):

rev=0

while n!=0:

rem=n%10

rev=rev*10+rem

n=n//10

if(num==rev):

print('no is palindrome')

else:

print('No is not a palindrome')

palindrome(num)

Output:

```
= RESTART: C:\Users\DELL\AppData\Local\Programs\Python\Python38-32\practle.py
Please Enter a number153
no is armstrong
No is not a palindrome
>>>
= RESTART: C:\Users\DELL\AppData\Local\Programs\Python\Python38-32\practle.py
Please Enter a number1221
No is not a armstrong
no is palindrome
>>> |
```

F. Write a recursive function to print the factorial for a given number.

Source Code:

n= int(input('Enter a number')) #n=6

def fact(a): #a=6

```
if a==1:
    return 1
else:
    return a*fact(a-1) #return-->6*fact(5)--->6*5*fact(4)-->6*5*4*fact(3)-->6*5*4*3*fact(2)---
>6*5*4*3*2*1
```

```
if n<0:
    print('Factorial of negative number does not exists')
elif n==0:
    print('Factorial of 0 is 1')
else:
    print(fact(n))
```

Output:

```
= RESTART: C:\Users\
PY
Enter a number6
720
>>>
```

Practical No.2

Write the program for the following: (by using functions)

A. Write a function that takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise.

Source Code:

```
a=input('Enter a character')

def isVowel(a):

    vowel=['a','e','i','o','u']

    if a in vowel:

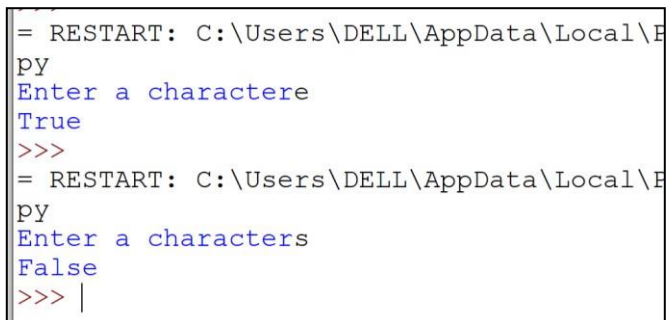
        return True

    else:

        return False

print(isVowel(a))
```

Output:



```
= RESTART: C:\Users\DELL\AppData\Local\Python
py
Enter a caractere
True
>>>
= RESTART: C:\Users\DELL\AppData\Local\Python
py
Enter a characters
False
>>> |
```

B. Define a function that computes the length of a given list or string.

Source Code:

```
def len_s(s):

    count=0

    for i in s:

        if i != ' ':

            count+=1
```

```
print('The total length of the string:',count)

s='God is great'

len_s(s)
```

Output:

```
>>>
= RESTART: C:/Users/DELL/AppData/Local/
The total length of the string: 10
>>> |
```

C. Define a procedure histogram () that takes a list of integers and prints a histogram to the screen. For example, histogram ([4, 9, 7]) should print the following:

```
****
*****
*****
```

Source Code:

```
l=[]

n=int(input('How many elements you want in list'))

for c in range(0,n):

    a=int(input())

    l.append(a)

def histogram(a):

    for c in range(len(a)):

        print(a[c]*' ')

histogram(l)
```

Output:


```
= RESTART: C:\Users\DELL\AppData\Local
.PY
How many elements you want in list4
2
4
8
3
**
****
*****
***
```

Practical No.-3

Write the program for the following: (by using list)

A. A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not.

Source Code:

```
import string

def ispangram(sentence, alphabet=string.ascii_lowercase):

    alphasets = set(alphabet)

    return alphasets <= set(sentence.lower())

print ( ispangram(input('Sentence: ')) )
```

Output:

```
>>>
= RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python38-32/fjgb.py
Sentence: The quick brown fox jumps over the lazy dog
True
>>> |
```

B. Take a list, say for example this one: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] and write a program that prints out all the elements of the list that are less than 5.

Source Code:

```
l1=[1,1,2,3,5,8,13,21,34,55,89]

l2= []

for i in l1:
```

```
if i < 5:  
    l2.append(i)  
print(l2)
```

Output:

```
= RESTART: C:/Users/DELL/AppDa  
[1, 1, 2, 3]  
>>> |
```

Practical No.4

Write the program for the following: (by using list)

A. Write a program that takes two lists and returns true if they have at least one common member.

Source Code:

```
l1=[1,3,4,5,79,0,8]  
l2=[2,4,6,77,90]  
count=0  
for c in l1:  
    if c in l2:  
        count=count+1  
  
if count>0:  
    print('True')  
else:  
    print('False')
```

Output:-

True

B. Aim :- To write a program to print a specified list after removing the 0th, 2nd, 4th, and 5th element.

Source Code :-

```
l1=[1,3,4,5,79,0,8]
print(l1)
l1.pop(0)
print(l1)
l1.pop(1)
l1.pop(2)
l1.pop(2)
print(l1)
```

Output:-

```
[1, 3, 4, 5, 79, 0, 8]
[3, 4, 5, 79, 0, 8]
[3, 5, 8]
```

C. Aim:- To write a python program to clone or copy a list.

Source Code

```
l1=[1,3,4,5,79,0,8]
print(l1)
l2=[]
l2=l1
print(l2)
print('Program performed by: NEHA A SINGH')
```

Output:-

```
[1, 3, 4, 5, 79, 0, 8]  
[1, 3, 4, 5, 79, 0, 8]
```

PRACTICAL No. 05

A. Aim :- To write a python script to sort (ascending and descending) a dictionary by value.

Source Code

```
a={101:'abc',10:'pqr',3:'xyz',4:'lmn'}  
print('Ascending Order: ' )  
for c,b in sorted(a.items()):  
    print(c,b)  
print('Descending Order: ' )  
for d,e in sorted(a.items(),reverse=True):  
    print(d,e)
```

Output:-

```
Ascending Order:  
3 xyz  
4 lmn  
10 pqr  
101 abc  
Descending Order:  
101 abc  
10 pqr  
4 lmn  
3 xyz
```

Aim : To write a python script to concatenate dictionaries to create a new one.

Source Code :

```
a={101:'abc',102:'pqr',103:'xyz'}  
b={'name':'Snehal','surname':'Kajaniya'}  
c={'department':'IT','Year':'SY'}  
d={}  
  
for c in a,b,c:  
    d.update(c)  
print(d)
```

Output:-

```
{101: 'abc', 102: 'pqr', 103: 'xyz', 'name': 'Snehal', 'surname': 'Kajaniya', 'department': 'IT', 'Year': 'SY'}
```

C. Aim : To write a python program to sum all the items in a dictionary.

Source Code

```
a={101:12,103:3,103:4,104:16}  
print(sum(a.values()))
```

Output:-

```
32
```

PRACTICAL No. :- 06(A)

AIM :- To write a program to read an entire text file.

Source Code :-

```
f=open('t1.txt','r')  
print(f.read())
```

Output:-

HELLO SYIT

B. Aim :- To write a program to append text to a file and display the text

Source Code :

```
f=open('t1.txt','a+')  
f.write('\n Welcome to PP Practicals')  
f.close()
```

```
f=open('t1.txt','r')  
print(f.read())
```

Output:-

```
HELLO SYIT  
Welcome to PP Practicals
```

C. AIM :- To write a python program to read last n lines of a file.

Source Code :-

```
f=open('t1.txt','r')  
a=f.readlines()  
print(a[-1])
```

Output:-

```
Welcome to PP Practicals
```

PRACTICAL No. 07

A.Aim :- To design a class that store the information of student and display the same.

Source Code

```
class Student:
    def getdata(self):
        self.Name=input('Please Enter Student Name: ')
        self.PID=int(input('Please Enter PID Number: '))
        self.Course=input('Please Course Name: ')
    def display(self):
        print('Student Details: ')
        print('Student Name: '+self.Name +' PID number: '+str(self.PID) +'
Course: ' +self.Course)
obj=Student()
obj.getdata()
```



```
obj.display()
```

Output:-

```
Please Enter Student Name: NEHA SINGH
Please Enter PID Number: 1001
Please Course Name: Bsc IT
Student Details:
Student Name: NEHA SINGH PID number: 1001 Course: Bsc IT
```

B. AIM : To implement the concept of inheritance using python.

Source Code :

Single Inheritance

```
class Phone:
    def call(self):
        print('Calling...')
class Mobile(Phone):
    def Camera(self):
        print('Photo Clicked...')
    def WIFI(self):
        print('WIFI Connected')

obj=Mobile()
obj.Camera()
obj.WIFI()
```

```
obj.call()
```

Multiple Inheritance

```
class marks1:
    def m1(self,st1):
        self.st1=st1
class marks2:
    def m2(self,st2):
        self.st2=st2
class result(marks1,marks2):

    def r(self,st1,st2):
        self.res=st1+st2
        print(self.res)
obj=result()
obj.r(72,77)
```

Multilevel Inheritance

```
class Defence:
    def type(self):
        print('Security')
class Army(Defence):
    def head(self):
        print('Chief of Army Staff')
```

```
class General(Army):  
    def commands(self):  
        print('Armed force')  
obj=General()  
obj.type()  
obj.head()  
obj.commands()
```

Output:-

Single Inheritance

```
Photo Clicked...  
WIFI Connected  
Calling...
```

Multiple Inheritance

```
149
```

Multilevel Inheritance

```
Security  
Chief of Army Staff  
Armed force
```

PRACTICAL No. :- 07(C)

C.Aim : To create a class called Numbers, which has single class attribute called Multiplier, and a constructor which takes the parameters x and y.

Source Code :

```
class Numbers:  
    multiplier=5  
    def __init__(self,x,y):  
        self.x=x  
        self.y=y  
    def add(self):
```

```
        return self.x+self.y
```

```
@classmethod
```

```
def multiply(self,a):
```

```
    return a*self.multiplier
```

```
@staticmethod
```

```
def subtract(b,c):
```

```
    return b-c
```

```
obj=Numbers(5,5)
```

```
print(obj.add())
```

```
print(obj.multiply(5))
```

```
print(obj.subtract(9,2))
```

Output:-

```
10  
25  
7
```

Practical No. 8

A. Write the program for the following: (IDLE and exception handling) A. Open a new file in IDLE ("New Window" in the "File" menu) and save it as geometry.py in the directory where you keep the files you create for this course. Then copy the functions you wrote for calculating volumes and areas in the "Control Flow and Functions" exercise into this file and save it. Now open a new file and save it in the same directory. You should now be able to import your own module like this: import geometry 16 Try and add print dir(geometry) to the file and run it. Now write a function pointyShapeVolume(x, y, squareBase) that calculates the volume of a square pyramid if squareBase is True and of a right circular cone if squareBase is False. x is the length of an edge on a square if squareBase is True and the radius of a circle when squareBase is False. y is the height of the object. First use squareBase to distinguish the cases. Use the circleArea and squareArea from the geometry module to calculate the base areas.

Source Code:

geometry.py

```
import math
```

```
def sphereArea(r):
```

```
    return 4*math.pi*r**2
```

```
def sphereVolume(r):
```

```
    return 4*math.pi*r**3/3
```

```
def sphereMetrics(r):
```

```
    return sphereArea(r),sphereVolume(r)
```

```
def circleArea(r):
```

```
    return math.pi*r**2
```

```
def squareArea(x):
```

```
return x**2
```

demo.py

```
import geometry
```

```
def pointyShapeVolume(x,h,square):
```

```
    if square:
```

```
        base=geometry.squareArea(x)
```

```
    else:
```

```
        base=geometry.circleArea(x)
```

```
    return h*base/3.0
```

```
print(dir(geometry))
```

```
print(pointyShapeVolume(4,2.6,True))
```

```
print(pointyShapeVolume(4,2.6,False))
```

Output:

```
= RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python38-32/demo.py
['_builtins_', '__cached__', '__doc__', '__file__', '__loader__', '__name__'
, '__package__', '__spec__', 'circleArea', 'math', 'sphereArea', 'sphere
Metrics', 'sphereVolume', 'squareArea']
13.866666666666667
43.56341812977846
>>> |
```

C. Aim: Write a program to implement exception handling

Source Code:

```
try:
```

```
    num=int(input('Enter a number: '))
```

```
    re=100/num
```

```
except(ValueError,ZeroDivisionError):
```

```
    print('Something is wrong')
```

```
else:
```

```
print('Result is ',re)
```

Output:

```
= RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python38-32/fgi.py =  
Enter a number: 12  
Result is  8.333333333333334  
>>>  
= RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python38-32/fgi.py =  
Enter a number: 0  
Something is wrong  
|
```

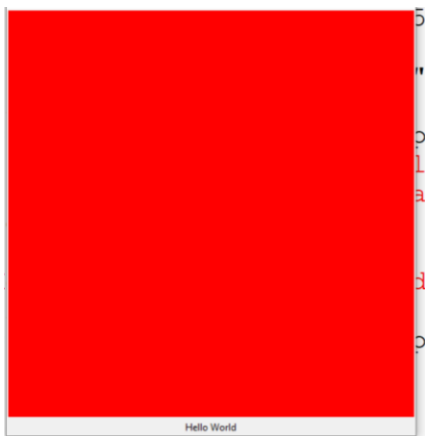

Practical No.9

A. Write the program for the following: (Widget - GUI) A. Try to configure the widget with various options like: bg="red", family="times", size=18

Source Code:

```
import Tkinter
from Tkinter import *
root=Tk()
O=Canvas(root,bg="red",width=500,height=500)
O.pack()
n = Label(root,text="Hello World")
n.pack()
root.mainloop()
```

Output:



B. Try to change the widget type and configuration options to experiment with other widget types like Message, Button, Entry, Checkbutton, Radiobutton, Scale etc

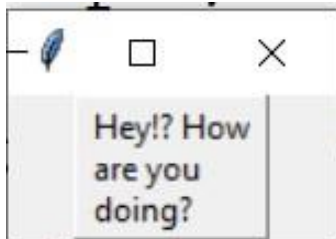
Source Code:

```
import tkinter
from tkinter import *

root = Tk()
```

```
var = StringVar()  
label = Message( root, textvariable=var, relief=RAISED )  
var.set("Hey!? How are you doing?")  
label.pack()  
root.mainloop()
```

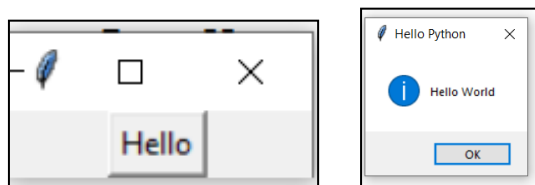
Output:



Button.py

```
import tkinter  
from tkinter import *  
from tkinter import messagebox  
  
root = Tk()  
def helloCallBack():  
    messagebox.showinfo( "Hello Python", "Hello World")  
B = Button(root, text ="Hello", command = helloCallBack)  
B.pack()  
root.mainloop()
```

Output:



Entry.py

```
import tkinter

from tkinter import *

root = Tk()

L1 = Label(root, text="User Name")

L1.pack( side = LEFT)

E1 = Entry(root, bd =5)

E1.pack(side = RIGHT)

root.mainloop()
```

Output:



Checkbutton.py

```
import tkinter

from tkinter import *

root = Tk()

CheckVar1 = IntVar()

CheckVar2 = IntVar()

C1 = Checkbutton(root, text = "Music", variable = CheckVar1,onvalue = 1, offvalue = 0,
height=5, width = 20)

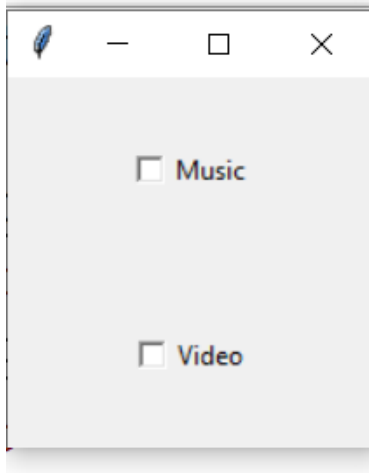
C2 = Checkbutton(root, text = "Video", variable = CheckVar2,onvalue = 1, offvalue = 0, height=5,
width = 20)

C1.pack()

C2.pack()
```

```
root.mainloop()
```

Output:



Radiobutton.py

```
import tkinter
from tkinter import *
root = Tk()
def sel():
    selection = "You selected the option " + str(var.get())
    label.config(text = selection)

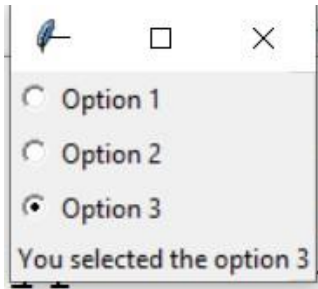
var = IntVar()
R1 = Radiobutton(root, text="Option 1", variable=var,value=1, command=sel)
R1.pack( anchor = W )
R2 = Radiobutton(root, text="Option 2", variable=var, value=2,command=sel)
R2.pack( anchor = W )
R3 = Radiobutton(root, text="Option 3", variable=var,value=3, command=sel)
R3.pack( anchor = W )
```

```
label = Label(root)
```

```
label.pack()
```

```
root.mainloop()
```

Output:



Scale.py

```
import tkinter
```

```
from tkinter import *
```

```
root = Tk()
```

```
def sel():
```

```
    selection = "Value = " + str(var.get())
```

```
    label.config(text = selection)
```

```
var = DoubleVar()
```

```
scale = Scale( root, variable = var )
```

```
scale.pack(anchor=CENTER)
```

```
button = Button(root, text="Get Scale Value", command=sel)
```

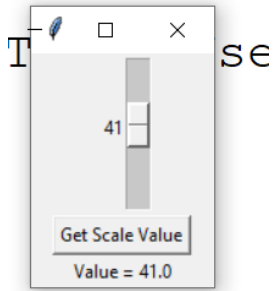
```
button.pack(anchor=CENTER)
```

```
label = Label(root)
```

```
label.pack()
```

```
root.mainloop()
```

Output:



Practical No.10

A. Design a simple database application that stores the records and retrieve the same.

Source Code: