



# NEW HORIZON COLLEGE OF ENGINEERING

New Horizon Knowledge Park, Ring Road, Marathalli  
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC  
Accredited by NAAC with 'A' Grade, Accredited by NBA



Name

Aashutosh Gupta

USN

1NH20CS001

Year

2022 - 2023

Program

B.E. in CSE

Semester

5

Section

A

Course

PYTHON PROGRAMMING LAB

Course Code

20CSL56



**NEW HORIZON**  
**COLLEGE OF ENGINEERING**  
New Horizon Knowledge Park, Ring Road, Marathalli  
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC  
Accredited by NAAC with 'A' Grade, Accredited by NBA

# *Laboratory Certificate*

*This is to certify that*

*Mr. Aashutosh Gupta*

*has satisfactorily completed the experiments prescribed by  
New Horizon College of Engineering, Bangalore Affiliated  
to Visvesvaraya Technological University*

*in ...**Python Programming**... Laboratory Course  
for the.....**5<sup>th</sup>**...semester of*

*Computer Science and Engineering Program.*

*Academic Year: 2022 to 2023 (ODD*

*Semester)*

<b>Marks Obtained</b>
<b>Max. Marks</b>

**Student Name: Aashutosh Gupta**

**USN: 1NH20CS001**

**Sem/ Sec: 5 -A**

**Course Code: 20CSL56**

**Signature of Student**

**Signature of the Faculty In-charge**

**Head of the Department**



**LABORATORY PERFORMANCE**  
**EVALUATION SHEET**

**Name of Student:** Aashutosh Gupta

**USN:** 1NH20cs001

**Lab Course:** PYTHON PROGRAMMING LAB

**Course Code:** 20CSL56

**Sem/Sec:** 5-A

**Session:** AFTERNOON

**CIE- PART A- Record and Performance (Max Marks: 10)**

SN	Date of Evaluation	Name of Experiment/ Program	1	2	3	4	Total	Faculty Signature
1		<b>Basic Python Programming</b> 1. Write a Python program to swap two variables. 2. Write a Python Program to Check Prime Numbers. 3. Write a Python Program to Find the Sum of Natural Numbers. 4. Write a Python Program to Find the Largest Among Three Numbers. 5. Write a Python Program to Print the Fibonacci sequence						
2		<b>Python Program Flow Control</b> a) Write a Python Program to Print the Reverse of a given number. b) Write a Python program to find those numbers which are divisible by 7 and multiple of 5, between 1500 and 2700 (both included). c) Write a Python program to print Right Angle Triangle. d) Write a Python Program to Print Pascal Triangles						
3		<b>Function:</b> 1. Illustrate an example program for python functions.						

SN	Date of Evaluation	Name of Experiment/ Program	1	2	3	4	Total	Faculty Signature
4		<b>String Manipulation:</b> Write a Python program to count the number of characters (character frequency) in a string.						
5		<b>Python Data Structures</b> 1. illustrate with a python program to show various insert and delete operations in set, tuple, dictionary and list.						
6		<b>Numpy</b> 1. Python Program to create numpy arrays in different ways and also how to index, Slice and operations on it .						
7		<b>Pandas</b> 1. Python Program to create data frame using pandas in different ways and also how to index, Slice and operations on it 2. Python Program for Data Pre-Processing with statistical measuring using given Dataset.  Python Program how to import different types of datasets.cc						
8		<b>Advanced Python – OOPs Concepts</b> 1. Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number. 2. Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.						

<b>SN</b>	<b>Date of Evaluation</b>	<b>Name of Experiment/ Program</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>Total</b>	<b>Faculty Signature</b>
9		<b>GUI Programming GUI Programming</b> 1. Write a Python GUI program to add a button, Combo box, Check button, three radio buttons widgets, aListbox bar widgets in your application using tkinter module.  2. Write a Python GUI program to create a Text widget using tkinter module. Insert a string at the beginning then insert a string into the current text. Delete the first and last character of the text.						
10		<b>Database Programming</b>  Write a Python program to create a small GUI application for insert, update and delete in a table using a backend and front end for creating form.						
<b>AVG Marks (out of 10 marks)</b>								



**NEW HORIZON**  
**COLLEGE OF ENGINEERING**  
New Horizon Knowledge Park, Ring Road, Marathalli  
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC  
Accredited by NAAC with 'A' Grade, Accredited by NBA

**CIE- PART B - Lab Test (Max Marks: 50)**

	Date of Lab  Test	Procedure  and Write Up (15 Marks)	Conduction  and Results (25 Marks)	Viva Voce  (10 Marks)	Total  (50 Marks)	Faculty  Sign
Test 1						
Test 2						
AVG Marks (out of 10 marks)						

**CIE- Marks Obtained**

CIE-Part A Record and Performance (10 Marks)	CIE-Part B Lab Test (Scaled to 15 Marks)	Total (25 marks)	Faculty Signature

## 1.EXPERIMENT 1 - Basic Python Programming

**1A) AIM:** Write a Python program to swap two variables.

### **THEORY:Method-1**

```
a=int(input("Enter 1st no")  
b= int(input("Enter 2nd no")  
  
temp=a  
a=b  
b=temp  
print("After swapping",a,b)
```

### **Method-2**

```
a=int(input("Enter 1st no")  
b= int(input("Enter 2nd no")  
  
a,b=b,a  
print("After swapping",a,b)
```

### **OUTPUT:**

```
Enter 1st no2  
Enter 2nd no3  
After swapping 3 2
```

**1B) AIM :**Write a Python Program to Check Prime Numbers.

**THEORY:** If the number is divisible by any number between 2 and num/2+1,then it is not a prime number,else it is a prime number.

### **PROGRAM:**

```
flag=0  
num=int(input("Enter a no "))  
for i in range(2, int(num/2)+1):  
    if (num % i)==0:  
        flag=1  
        break  
    else:  
        flag=0  
if flag==1:  
    print(num,"is not prime number")  
else  
    print(num, "is a prime number")
```

**OUTPUT:**

Enter a no 5  
 5 is a prime number  
 Enter a no 2  
 2 is a prime number  
 Enter a no 4  
 4 is not prime number

**1C) AIM:** To Write a Python program to Find the Sum of Natural Numbers**THEORY:**

Accept n numbers from user and run a loop from start till end and keep adding the element to a variable initialized to 0.

**PROGRAM:**

```
num=int(input("Enter a no "))
sum = 0
while(num > 0):
    sum += num
    num -= 1
print("The sum is ", sum)
```

**OUTPUT:**

Enter a no 10  
 The sum is 55

**1D) AIM:** To Write a Python Program to Find the Largest Among Three Numbers.**THEORY:**

Start

Read the three numbers to be compared, as A, B and C.  
 Check if A is greater than B.

If true, then check if A is greater than C.  
 If true, print 'A' as the greatest number.  
 If false, print 'C' as the greatest number.

If false, then check if B is greater than C.  
 If true, print 'B' as the greatest number.  
 If false, print 'C' as the greatest number.  
 End



**PROGRAM:**

```

a=int(input("Enter 1st no"))
b=int(input("Enter 2nd no"))
c=int(input("Enter 3rd no"))

def maximum(a, b, c):

    if (a >= b) and (a >= c):
        largest = a

    elif (b >= a) and (b >= c):
        largest = b
    else:
        largest = c

    return largest

print(maximum(a,b,c))

```

**OUTPUT:**

```

Enter 1st no 5
Enter 2nd no 8
Enter 3rd no 1
8

```

**1E) AIM:** To Write a Python Program to Print the Fibonacci sequence

**THEORY:**

The Fibonacci numbers are the numbers in the following integer sequence.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, .....

In mathematical terms, the sequence  $F_n$  of Fibonacci numbers is defined by the recurrence relation

**PROGRAM:**

```

num =int(input("Enter a no "))
n1, n2 = 0, 1
print("Fibonacci Series:", n1, n2, end=" ")
for i in range(2, num):
    n3 = n1 + n2
    n1 = n2
    n2 = n3
    print(n3, end=" ")

print()

```

**OUTPUT:**

```

Enter a no5
Fibonacci Series: 0 1 1 2 3

```

**EXPERIMENT 2 : Python program flow control****2A) AIM:** To Write a Python Program to Print the Reverse of a given number**THEORY:**

Modulo by 10 will allow to access last digit and // 10 will allow the quotient to be accessed.

**PROGRAM:**

```
num =int(input("Enter a no "))
reversed_num = 0

while num != 0:
    digit = num % 10
    reversed_num = reversed_num * 10 + digit
    num //= 10

print("Reversed Number: " + str(reversed_num))
```

**OUTPUT:**

```
Enter a no 345
Reversed Number: 543
```

**2B) AIM:** To Write a Python program to find those numbers which are divisible by 7 and multiple of 5, between 1500 and 2700(both included)**THEORY:**

- i) Use for loop to get the numbers from 1500 to 2700.
  - ii) Select the divisible by 5 and 7 using “%” operator.
  - iii) Print the number if it is divisible, using the if statement.
- Running in Jupyter Notebooks

**PROGRAM:**

```
for i in range(1500,2701):
    if(i%5==0 and i%7==0):
        print(i)
```

**OUTPUT:**

```
1505
1540
1575
1610
1645
1680
1715
1750
1785
```

```

1820
1855
1890
1925
1960
1995
2030
2065
2100
2135
2170
2205
2240
2275
2310
2345
2380
2415
2450
2485
2520
2555
2590
2625
2660
2695

```

```
>>>
```

**2C) AIM:** To Write a Python program to print Right Angle Triangle

**PROGRAM:**

```

rows = int(input("Please Enter the Total Number of Rows : "))

print("Right Angled Triangle Star Pattern")
for i in range(1, rows + 1):
    for j in range(1, i + 1):
        print('*', end = ' ')
    print()

```

**OUTPUT:**

```

Please Enter the Total Number of Rows : 5
Right Angled Triangle Star Pattern
*
* *
* * *
* * * *
* * * * *

```

**2D) AIM:** To Write a Python Program to Print Pascal Triangles**THEORY:**

Pascal's triangle is a pattern of the triangle which is based on  $nCr$ , below is the pictorial representation of Pascal's triangle.

- Take a number of rows to be printed, let's assume it to be  $n$
- Make outer iteration  $i$  from 0 to  $n$  times to print the rows.
- Make inner iteration for  $j$  from 0 to  $(N - 1)$ .
- Print single blank space " ".
- Close inner loop ( $j$  loop) //its needed for left spacing.
- Make inner iteration for  $j$  from 0 to  $i$ .
- Print  $nCr$  of  $i$  and  $j$ .
- Close inner loop.
- Print newline character ( $\backslash n$ ) after each inner iteration.

**PROGRAM:**

```
#Print Pascal's Triangle in Python
from math import factorial
```

```
# input n
n = int(input("enter N"))
for i in range(n):
    for j in range(n-i+1):

        # for left spacing
        print(end=" ")

    for j in range(i+1):

        #  $nCr = n!/((n-r)!*r!)$ 
        print(factorial(i)//(factorial(j)*factorial(i-j)), end=" ")

    # for new line
    print()
```

**OUTPUT:**

```
Enter n 5
 1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
>>>
```

## EXPERIMENT 3- PYTHON FUNCTIONS

**3A) AIM :** Illustrate an example program for python functions.

- i) Import  $\pi$  from the math library.
- ii) Input radius from user.
- iii) Compute radius.
- iv) Print radius.

Running in Jupyter Notebooks

### PROGRAM:

```
#area of a circle.  
from math import pi  
radius=int(input("Enter a radius : "))  
print("Area = ",(pi*radius*radius))
```

### OUTPUT:

```
Enter a radius: 5  
Area = 78.5398163397448
```

**EXPERIMENT 4 STRING MANIPULATION:**

**AIM:** Write a Python program which accepts the a string from the user and count the number of characters in that string

**PROGRAM:**

```
def char_frequency(str1):  
    dict = {}  
    for n in str1:  
        keys = dict.keys()  
        if n in keys: dict[n] += 1  
        else:  
            dict[n]=1  
    return dict print(char_frequency('welcometopythonprogramminglab'))
```

**OUTPUT:**

```
{w:1,e:2,c:1,o:4,m:1,t:2,p:2,y:1,h:1,n:2,r:2,g:2,a:2,m:2,i:1,b:1}
```

**EXPERIMENT 5 PYTHON DATA STRUCTURES**

**AIM:** To write a python program to show various insert and delete operations in set, tuple, dictionary and list.

**PROGRAM:****#LISTS**

#program to demonstrate list operations

```
list1=[]
print(list1)
```

**OUTPUT :** []

```
list1=[1,2.6,3.9,"ram"]
print(list1)
```

**OUTPUT:** [1, 2.6, 3.9, 'ram']

```
list1=[1,2,3,4,4]
list2=["welcome",2,["python","programming","class"]]
print(list1)
print("Minimum of mylist is",min(list1))
print("Maximum of mylist is",max(list1))
print(list1[-1])
```

```
print(list2[2][2])
i=-1
while i>=-4:
    print(list1[i])
    i=i+1
```

```
#using for loop to print list-method1
length=len(list1)
for i in range(length+1):
    print(i)
```

```
#using for loop to print list-method2
for i in list1:
    print(i)
```

**OUTPUT :**

```
[1, 2, 3, 4, 4]
Minimum of mylist is 1
Maximum of mylist is 4
4
class
4
4
3
2
```

1  
1  
2  
3  
4  
4

### **TUPLE :**

```
t=(1.5,2,3,"ram",1.5)
print(t[2:4])
print(t[0])
print(t)
print(len(t))
print(t.count(1.5))
print(t.index(1.5))
t1=(2,3)
t2=(2,3)
print(t1==t2)
t1=tuple((3,4,5))
print(t1)
#access
t1=(2,3,4)
for i in t1:
    print(i)
```

### **OUTPUT:**

```
(3, 'ram')
1.5
(1.5, 2, 3, 'ram', 1.5)
5
2
0
True
(3, 4, 5)
2
3
4
```

### **#SETS**

```
set1={"Apple","Banana","Cranberry","Apple"}
set2={"mango","Tender Coconut","PineApple","Apple","Banana","Cranberry"}
print(set1)
set1.add("ORANGE")
print(set1)
set1.update(set2)
print(set1)
print(set2)
set1.remove("Apple")
```



```

print(set1)
set1.discard("Banana")
print(set1)
set1.discard("Banana")
print(set1)
set2.pop()
print(set2)
set1.update(set2)
print(set1)
set1.union(set2)
print(set1)

set3=set1.copy()
print(set3)
print(set2.difference(set3))
print(set2.intersection(set3))
print(set1.isdisjoint(set2))
print(set1.issuperset(set2))
print(set1.issubset(set2))

```

### OUTPUT:

```

{'Banana', 'Cranberry', 'Apple'}
{'Banana', 'ORANGE', 'Cranberry', 'Apple'}
{'Tender Coconut', 'mango', 'Apple', 'Banana', 'ORANGE', 'PineApple', 'Cranberry'}
{'Banana', 'mango', 'Cranberry', 'PineApple', 'Tender Coconut', 'Apple'}
{'Tender Coconut', 'mango', 'Banana', 'ORANGE', 'PineApple', 'Cranberry'}
{'Tender Coconut', 'mango', 'ORANGE', 'PineApple', 'Cranberry'}
{'Tender Coconut', 'mango', 'ORANGE', 'PineApple', 'Cranberry'}
{'mango', 'Cranberry', 'PineApple', 'Tender Coconut', 'Apple'}
{'Tender Coconut', 'mango', 'Apple', 'ORANGE', 'PineApple', 'Cranberry'}
{'Tender Coconut', 'mango', 'Apple', 'ORANGE', 'PineApple', 'Cranberry'}
{'ORANGE', 'mango', 'Cranberry', 'PineApple', 'Tender Coconut', 'Apple'}
set()
{'mango', 'Cranberry', 'Apple', 'PineApple', 'Tender Coconut'}
False
True
False

```

### #DICTIONARY

```

d={"name": "uma", "age": 40, "sec": '5c'}
print(d["name"])
d["age"] = 41
print(d.get("name"))
print(d.keys())
d["height"] = "short"
print(d)
print(d.values())
print(d.items())
if "name" in d:
    print("name is a key in dict")

```

```
d.update({"age":42})  
print(d)  
d.pop("age")
```

```
print(d)  
d.popitem()  
print(d)  
del d["name"]  
print(d)  
d.clear()  
print(d)
```

**OUTPUT :**

```
uma  
uma  
dict_keys(['name', 'age', 'sec'])  
{'name': 'uma', 'age': 41, 'sec': '5c', 'height': 'short'}  
dict_values(['uma', 41, '5c', 'short'])  
dict_items([('name', 'uma'), ('age', 41), ('sec', '5c'), ('height', 'short')])  
name is a key in dict  
{'name': 'uma', 'age': 42, 'sec': '5c', 'height': 'short'}  
{'name': 'uma', 'sec': '5c', 'height': 'short'}  
{'name': 'uma', 'sec': '5c'}  
{'sec': '5c'}  
{}
```

**EXPERIMENT 6 NUMPY**

a) **Python Program to create numpy arrays in different ways and also how to index, Slice and operations on it .**

**AIM:** Write a Program to create numpy arrays in different ways and also how to index, Slice and operations on it.

```
pip install numpy
import numpy
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(arr)
print(type(arr))
```

**OUTPUT :**

```
[1 2 3 4 5]
<class 'numpy.ndarray'>
```

**"""0-D Arrays"""**

```
import numpy as np
arr = np.array(12)
print(arr)
```

**OUTPUT : 12**

**"""1-D Arrays"""**

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(arr)
```

**OUTPUT: [1, 2, 3, 4, 5]**

**"""2-D Arrays"""**

```
import numpy as np
arr = np.array([[1, 2, 3], [4, 5, 6]])
print(arr)
```

**OUTPUT : [[1 2 3]  
[4 5 6]]**

```
import numpy as np
a = np.array(12)

b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
print(a.ndim)

print(b.ndim)
print(c.ndim)
print(d.ndim)
```

```
import numpy as np
arr = np.array([1, 2, 3, 4], ndmin=5)
print(arr)

print('number of dimensions :', arr.ndim)
```

**OUTPUT :**

```
0
1
2
3
[[[[[1 2 3 4]]]]]
number of dimensions : 5
```

**""Access Array Elements""**

```
import numpy as np
arr = np.array([1, 2, 3, 4])
print(arr[0])
print(arr[1])
print(arr[2] + arr[3])
```

**OUTPUT : 7**

```
import numpy as np
arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])
print('2nd element on 1st row: ', arr[0, 1])
print('5th element on 2nd row: ', arr[1, 4])
```

**OUTPUT :** 2<sup>nd</sup> element on 1<sup>st</sup> row : 2  
5<sup>th</sup> element on 2<sup>nd</sup> row : 10

```
import numpy as np
arr = np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])
print(arr[0, 1, 2])
```

**OUTPUT : 6**

```
import numpy as np
arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])
print('Last element from 2nd dim: ', arr[1, -1])
```

**OUTPUT :**

Last element from 2nd dim: 10

**""Slicing arrays""**

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[1:5])
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[2:])
arr = np.array([1, 2, 3, 4, 5, 6, 7])
```

```

print(arr[:4])
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[-3:-1])
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[1:5:2])
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[:5])
arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
print(arr[1, 1:4])
arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
print(arr[0:2, 2])
arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
print(arr[0:2, 1:4])

```

**OUTPUT:**

```

[2 3 4 5]
[3 4 5 6 7]
[1 2 3 4]
[5 6]

```

```

[2 4]
[1 6]
[7 8 9]
[3 8]

```

```

[[2 3 4]
 [7 8 9]]

```

**""Checking the Data Type of an Array""**

```

import numpy as np
arr = np.array([1, 2, 3, 4])
print(arr.dtype)
arr = np.array(['apple', 'banana', 'cherry'])
print(arr.dtype)
arr = np.array([1, 2, 3, 4], dtype='S')
print(arr)
print(arr.dtype)
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])
arr = np.concatenate((arr1, arr2))
print(arr)
arr1 = np.array([[1, 2], [3, 4]])
arr2 = np.array([[5, 6], [7, 8]])
arr = np.concatenate((arr1, arr2), axis=1)
print(arr)

arr = np.array([1, 2, 3, 4, 5, 6])
newarr = np.array_split(arr, 8)
print(newarr)

```

**OUTPUT:**

```

int64

```

```
<U6
```

```
[b'1' b'2' b'3' b'4']
```

```
|S1
```

```
[1 2 3 4 5 6]
```

```
[[1 2 5 6]
```

```
[3 4 7 8]]
```

```
[array([1]), array([2]), array([3]), array([4]), array([5]), array([6]), array([], dtype=int64),  
array([], dtype=int64)]
```

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4, 5, 6])
```

```
newarr = np.array_split(arr, 4)
```

```
print(newarr)
```

**OUTPUT :** [array([1, 2]), array([3, 4]), array([5]), array([6])]

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4, 5, 4, 4])
```

```
x = np.where(arr == 4)
```

```
print(x)
```

**OUTPUT:** (array([3, 5, 6]),)

```
import numpy as np
```

```
arr = np.array([3, 2, 0, 1])
```

```
print(np.sort(arr))
```

**OUTPUT :** [0 1 2 3]

## EXPERIMENT 7

### 7)Pandas

- Python Program to create data frame using pandas in different ways and also how to index, Slice and operations on it
- Python Program for Data Pre-Processing with statistical measuring using given Dataset.
- Python Program how to import different types of datasets.cc

**7A) AIM:**To write python program to create dataframes using pandas ,to index and do slice operation on it

#### PROGRAM:

```
import pandas as pd
df = pd.DataFrame()
print(df)
```

#### ""Method #1: Creating Dataframe from Lists""

```
# Import pandas library
import pandas as pd
data = [100,200,300,400,500,600]
# Create the pandas DataFrame with column name is provided explicitly
df = pd.DataFrame(data, columns=['Numbers'])
# print dataframe.
Df
```

	Numbers
0	100
1	200
2	300
3	400
4	500
5	600

#### ""Method #2: Creating Pandas DataFrame from lists of lists.""

```
# Import pandas library
import pandas as pd
# initialize list of lists
data = [['Monday', 100], ['Tuesday', 150], ['Wednesday', 200]]
# Create the pandas DataFrame

df = pd.DataFrame(data, columns=['Day', 'Sales Unit'])
# print dataframe.
Df
```

	Day	Sales Unit
0	Monday	100
1	Tuesday	150
2	Wednesday	200

### ""Method #3: Creating DataFrame from dict of narray/lists""

```
import pandas as pd
# initialize data of lists.

data = {'Name': ['Arun', 'Ajay', 'Abi', 'Ajith'],
'Age': [20, 21, 29, 28]}
# Create DataFrame

df = pd.DataFrame(data)
# Print the output.
Df
```

	Name	Age
0	Arun	20
1	Ajay	21
2	Abi	29
3	Ajith	28

### ""Method #4: Creating a DataFrame by proving index label explicitly.""

```
# Python code demonstrate creating
# pandas DataFrame with indexed by
# DataFrame using arrays.
import pandas as pd
data = {'Name': ['Abhi', 'Babu', 'Raj', 'Ram'],
'marks': [99, 98, 95, 90]}

# Creates pandas DataFrame.
df = pd.DataFrame(data, index=['rank1',

                                'rank2',
                                'rank3',
                                'rank4'])

# print the data
Df
```



:

	Name	marks
rank1	Abhi	99
rank2	Babu	98
rank3	Raj	95
rank4	Ram	90

"""**Method #5 : Creating DataFrame using zip() function.**"""

# Python program to demonstrate creating  
# pandas Datadaframe from lists using zip.

```
import pandas as pd
# List1
Name = ['tom', 'krish', 'nick', 'juli']
# List2
Age = [25, 30, 26, 22]

# get the list of tuples from two lists.
# and merge them by using zip().
list_of_tuples = list(zip(Name, Age))

# Assign data to tuples.
list_of_tuples

# Converting lists of tuples into
# pandas Dataframe.
df = pd.DataFrame(list_of_tuples,
                  columns=['Name', 'Age'])

# Print data.
Df list_of_tuples
```

**OUTPUT :**[( 'tom', 25), ('krish', 30), ('nick', 26), ('juli', 22)]

"""**Method#6: Creating dataframe from series**"""

```
import pandas as pd
# Initialize data to series.
d = pd.Series([10, 20, 30, 40])
# creates Dataframe.
df = pd.DataFrame(d)
df
```

"""**Slicing** """

```
import pandas as pd
player_list = [['M.S.Dhoni', 36, 75, 5428000],
               ['A.B.D Villers', 38, 74, 3428000],
```

```

['V.Kholi', 31, 70, 8428000],
['S.Smith', 34, 80, 4428000],
['C.Gayle', 40, 100, 4528000],
['J.Root', 33, 72, 7028000],
['K.Peterson', 42, 85, 2528000]]
df = pd.DataFrame(player_list, columns=['Name', 'Age', 'Weight', 'Salary'])
df
df1 = df.iloc[:,0:2]
df1

```

	Name	Age	Weight	Salary
0	M.S.Dhoni	36	75	5428000
1	A.B.D Villers	38	74	3428000
2	V.Kholi	31	70	8428000
3	S.Smith	34	80	4428000
4	C.Gayle	40	100	4528000
5	J.Root	33	72	7028000
6	K.Peterson	42	85	2528000

### # Slicing columnss in data frame

```

df1 = df.iloc[:, 0:2]
df1

```

	Name	Age
0	M.S.Dhoni	36
1	A.B.D Villers	38
2	V.Kholi	31
3	S.Smith	34
4	C.Gayle	40
5	J.Root	33
6	K.Peterson	42

```

# importing pandas library
import pandas as pd

```

```

# creating and initializing a list

```

```

values = [['Rohan', 455], ['Elvish', 250], ['Deepak', 495],
          ['Sai', 400], ['Radha', 350], ['Vansh', 450]]
df = pd.DataFrame(values, columns=['Name', 'Univ_Marks'])
# displaying the data frame
print('Data frame before calculating percentage\n')
print(df)

```

```
print("\nData frame with Percentage Column\n")
df["Percentage"] = df["Univ_Marks"]/500*100
print(df)
```

	Name	Age	Weight	Salary
A	M.S.Dhoni	36	75	5428000
B	A.B.D Villers	38	74	3428000
C	V.Kholi	31	70	8428000
D	S.Smith	34	80	4428000
E	C.Gayle	40	100	4528000
F	J.Root	33	72	7028000
G	K.Peterson	42	85	2528000

**7B) AIM:**To write program for Data preprocessing and to import different types of datasets

## DATA PREPROCESSING

```
# importing pandas library
import pandas as pd
values = [['Rohan', 455], ['Elvish', 250], ['Deepak', 495],
          ['Sai', 400], ['Radha', 350], ['Vansh', 450]]
df = pd.DataFrame(values, columns=['Name', 'Univ_Marks'])
print('Data frame before calculating percentage\n')
print(df)
print("\nData frame with Percentage Column\n")
df["Percentage"] = df["Univ_Marks"]/500*100
print(df)
```

Data frame before calculating percentage

	Name	Univ_Marks
0	Rohan	455
1	Elvish	250
2	Deepak	495
3	Sai	400
4	Radha	350
5	Vansh	450

Data frame with Percentage Column

	Name	Univ_Marks	Percentage
0	Rohan	455	91.0
1	Elvish	250	50.0
2	Deepak	495	99.0
3	Sai	400	80.0
4	Radha	350	70.0
5	Vansh	450	90.0

```
# importing pandas library
import pandas as pd
```

```
# Initializing the nested list with Data set
player_list = [['M.S.Dhoni', 36, 75, 5428000],
```

```
[ 'A.B.D Villers', 38, 74, 3428000],
[ 'V.Kholi', 31, 70, 8428000],
[ 'S.Smith', 34, 80, 4428000],
[ 'C.Gayle', 40, 100, 4528000],
[ 'J.Root', 33, 72, 7028000],
[ 'K.Peterson', 42, 85, 2528000]]
```

```
# creating a pandas dataframe
df = pd.DataFrame(player_list, columns=['Name', 'Age', 'Weight', 'Salary'])
df.sort_values(by=['Weight'])
```

	Name	Age	Weight	Salary
6	K.Peterson	42	85	2528000
1	A.B.D Villers	38	74	3428000
3	S.Smith	34	80	4428000
4	C.Gayle	40	100	4528000
0	M.S.Dhoni	36	75	5428000
5	J.Root	33	72	7028000
2	V.Kholi	31	70	8428000

```
import pandas as pd
df = pd.read_csv(List.xl)
df.head()
```

	S N	USN	NAME	SEM/SEC
0	1	1NH18CS053	D VINDHYA SREE	7A
1	2	1NH18CS056	DEEPTHI N	7A
2	3	1NH18CS057	DEVENDRA DESAI	7A
3	4	1NH18CS058	DHARSHANA PANDIYAN	7A
4	5	1NH18CS059	DIPAK YADAV	7A

```
df.isna().sum()
```

```

age          0
sex          0
steroid      1
antivirals   0
fatigue      1
malaise      1
anorexia     1
liver_big    10
liver_firm   11
spleen_palpable 5
spiders      5
ascites      5
varices      5
bilirubin    6
alk_phosphate 29
sgot         4
albumin      16
protime      67
histology    0
class        0
dtype: int64

```

```
df.dropna(axis=1)
```

```
df.dropna(axis=0)
```

```

age          0.000000
sex          0.000000
steroid      0.645161
antivirals   0.000000
fatigue      0.645161
malaise      0.645161
anorexia     0.645161
liver_big    6.451613
liver_firm   7.096774
spleen_palpable 3.225806
spiders      3.225806
ascites      3.225806
varices      3.225806
bilirubin    3.870968
alk_phosphate 18.709677
sgot         2.580645
albumin      10.322581
protime      43.225806
histology    0.000000
class        0.000000
dtype: float64

```

```
df.dtypes
```

```
: age          int64
sex           object
steroid       object
antivirals    bool
fatigue       object
malaise       object
anorexia      object
liver_big     object
liver_firm    object
spleen_palpable object
spiders       object
ascites       object
varices       object
bilirubin     float64
alk_phosphate float64
sgot          float64
albumin       float64
protime       float64
histology     bool
class         object
dtype: object
```

## EXPERIMENT 8 OOPs CONCEPTS

- a) Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number.
- b) Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.

**AIM:** Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number

### STEPS

- i) Define a python class.
- ii) Define a function that accepts a array of numbers and a target number.
- iii) Check for the condition using for loop and if statement, in the dictionary defined.
- iv) If the condition is true for any 2 numbers in the array, then return the 2 numbers.
- v) Call the function using the class name, and by passing the array of numbers and a target number.

### PROGRAM:

```
class py_solution:
    def twoSum(self, nums, target):
        lookup = {}
        for i, num in enumerate(nums):
            if target - num in lookup:
                return (lookup[target - num], i )
            lookup[num] = i
print("index1=%d, index2=%d" % py_solution().twoSum((10,20,10,40,50,60,70),50))
```

**8b) AIM:** Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle

### PROGRAM:

```
class rectangle():
    def __init__(self, breadth, length):
self.breadth=breadth
self.length=length
    def area(self):
        return self.breadth*self.length
a=int(input("Enter length of rectangle: "))
b=int(input("Enter breadth of rectangle: "))

obj=rectangle(a,b)
print("Area of rectangle: ",obj.area())
```

### OUTPUT:

```
Enter the width 5
Enter a length 66
area of the rectangle is 330
```

**EXPERIMENT 9 GUI PROGRAMMING**

- a) Write a Python GUI program to add a button, Combo box, Check button, three radio buttons widgets, a Listbox bar widgets in your application using tkinter module.
- b) Write a Python GUI program to create a Text widget using tkinter module. Insert a string at the beginning then insert a string into the current text. Delete the first and last character of the text.

**9(A) AIM:** Write a Python GUI program to add a button, Combo box, Check button, three radio buttons widgets, a Listbox bar widgets in your application using tkinter module.

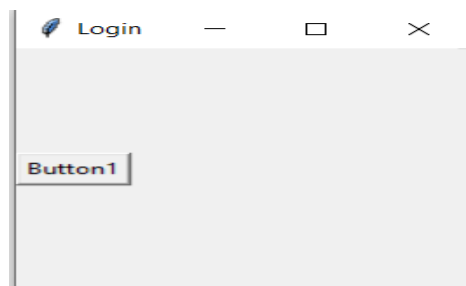
**STEPS:**

- i) import the functions from the tkinter library.
- ii) Define the root variable, and using the root variable configure the tkinter window.
- iii) Define the buttons, combobox and radio buttons using the 'Buttons', 'Combobox', 'Radiobutton' function and by passing the root variable, the text of the radio button and its value.
- iv) Position the buttons and box in the tkinter window using the grid function.
- v) Define a label using the 'Label' function and by passing the root and the text of the label.
- vi) Define the list box, using the 'Listbox' function and by passing the values using the insert function.
- vii) Call the 'mainloop' function to run the window created.

Running in Jupyter Notebooks

**PROGRAM:****Python program to add button**

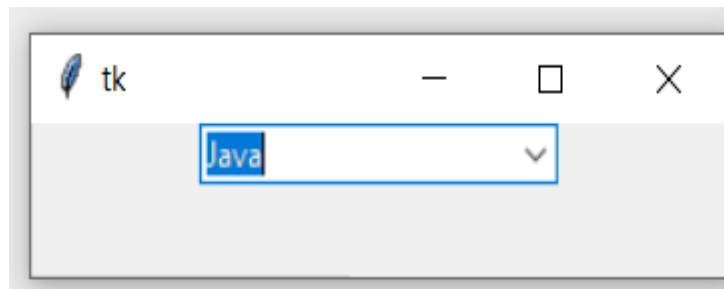
```
from tkinter import*
root=Tk()
root.title("Login")
root.geometry("500x300")
b1=Button(root,text="Button1")
b1.pack(side=LEFT)
root.mainloop()
```

**OUTPUT:****Python program to add Combo box**

```
import tkinter as tk
from tkinter import ttk
root = tk.Tk()
my_str_var = tk.StringVar()
my_combobox = ttk.Combobox(
    root, textvariable = my_str_var,
    values=["PHP", "Java", "Python"])
my_combobox.pack()
root.mainloop()
```

**OUTPUT:**

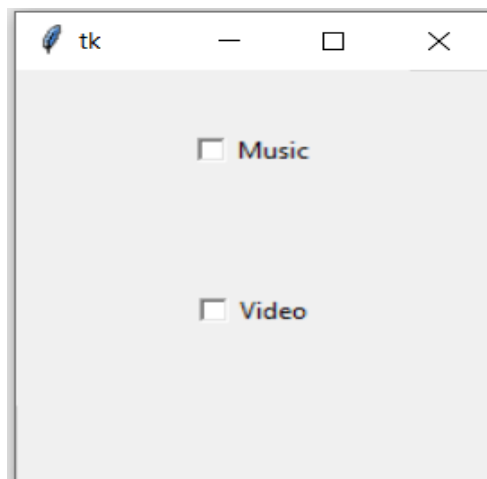




### Python Code for Checkbutton Widget

```
from tkinter import*
import tkinter
root = Tk()
CheckVar1 = IntVar()
CheckVar1.set(False) #set check state
CheckVar2 = IntVar()
CheckVar2.set(True) #set check state
C1 = Checkbutton(root, text = "Music", variable = CheckVar1, onvalue = 1, offvalue = 0, height=5, width = 20)
C2 = Checkbutton(root, text = "Video", variable = CheckVar2, onvalue = 2, offvalue = 0, height=5, width = 20)
C1.pack()
C2.pack()
root.mainloop()
```

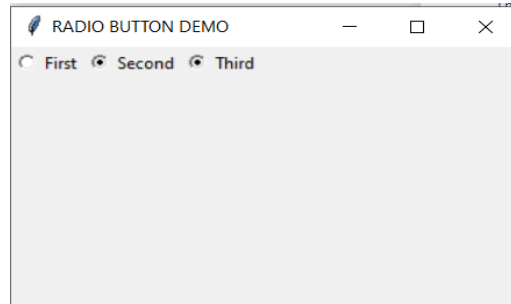
### **OUTPUT:**



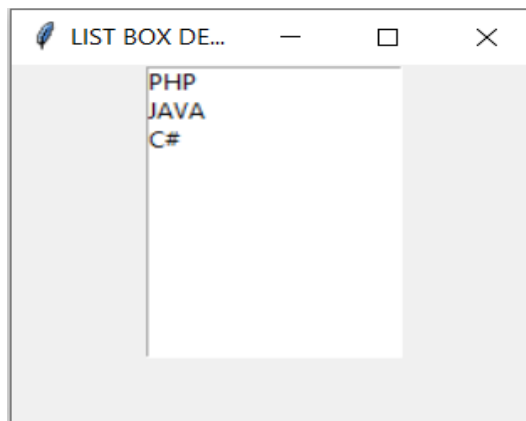
### Python Code for three Radiobutton widget

```
from tkinter import *
root = Tk()
root.title("RADIO BUTTON DEMO")
root.geometry('350x200')
#creating radiobuttons
rad1 = Radiobutton(root, text='First', value=1)
rad2 = Radiobutton(root, text='Second', value=2)
```

```
rad3 = Radiobutton(root,text='Third',value=3)
rad1.grid(column=0, row=0)
rad2.grid(column=1, row=0)
rad3.grid(column=2, row=0)
root.mainloop()
```

**OUTPUT:****Python Code for Listbox widget**

```
from tkinter import *
root = Tk()
root.title("LIST BOX DEMO")
root.geometry("250x200")
label1 = Label(root,text = " Choose your favourite programming language...").pack
listbox = Listbox(root)
listbox.insert(1,"PHP")
listbox.insert(2,"JAVA")
listbox.insert(4,"C#")
listbox.pack()
root.mainloop()
```

**OUTPUT**

**9(B) AIM:** Write a Python GUI program to create a Text widget using tkinter module. Insert a string at the beginning then insert a string into the current text. Delete the first and last character of the text.

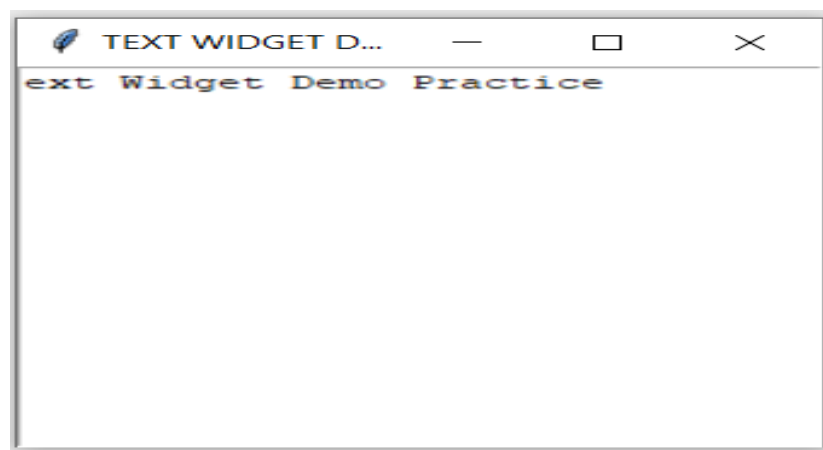
**STEPS:**

- i) import the functions from the tkinter library.
  - ii) Define the root variable, and using the root variable configure the tkinter window.
  - iii) Define the text widget using the 'Text' function.
  - iv) Use the insert function to insert values into the text widget, and position it in the tkinter window using the pack function.
  - v) Call the 'mainloop' function to run the window created.
- Running in Jupyter Notebooks

**PROGRAM:**

```
from tkinter import *
root = Tk()
root.title("TEXT WIDGET DEMO")
text = Text(root)
# insert a string at the beginning
text.insert('1.0', "Text Widget Demo")
# insert a string into the current text
text.insert('1.19', ' Practice,')
# delete the first and last character (including a newline character)
text.delete('1.0')
text.delete('end - 2 chars')
text.pack()
root.mainloop()
```

**OUTPUT:**



**EXPERIMENT 10 : DATABASE PROGRAMMING**

**AIM:** Write a Python program to create a small GUI application for insert, update and delete in a table using a backend and front end for creating form.

**THEORY:**

- i) import the functions from the tkinter library, and import the sqlite3 library.
- ii) Define the root variable, and using the root variable configure the tkinter window.
- iii) Connect the program to the database, using the connect function and by passing the path of the database file.
- iv) Create the cursor object and call the execute function to execute the various sql commands.
- v) Call the commit function to save the changes done after executing all the sql commands.
- v) Call the 'mainloop' function to run the window created. Running in Jupyter Notebooks

**PROGRAM:**

```

from tkinter import *
import sqlite3
#=====
#CREATING ROOT WINDOW
root=Tk()
root.geometry('410x450')
root.title("DATABASE USING SQLITE3 AND TKINTER")
root.config(background="red")
#=====
#CREATING DB CONNECTION,CREATING TABLE
db=sqlite3.connect("C:/Users/AMAN/Dropbox/My PC (LAPTOP-K3QKQ8QO)
/Downloads/SQLiteStudio-3.2.1/SQLiteStudio/fire_forest.sqlite")
cursor=db.cursor()
cursor.execute("CREATE TABLE IF NOT EXISTS 'people' (name TEXT,phone TEXT)")
db.commit()
#=====
textin=StringVar()
textinn=StringVar()

def insert():
    name1=textin.get()
    phone1=textinn.get()
    conn=sqlite3.connect("C:/Users/AMAN/Dropbox/My PC (LAPTOP-K3QKQ8QO)
/Downloads/SQLiteStudio-3.2.1/SQLiteStudio/fire_forest.sqlite")
    with conn:
        cursor = conn.cursor()
    cursor.execute("INSERT INTO 'people'(name,phone) VALUES(?,?)",(name1,phone1))
    db.close()

def show():
    conn=sqlite3.connect("C:/Users/AMAN/Dropbox/My PC (LAPTOP-K3QKQ8QO)
/Downloads/SQLiteStudio-3.2.1/SQLiteStudio/fire_forest.sqlite")
    cursor = conn.cursor()

```

```

cursor.execute("SELECT * FROM people")
    for row in cursor.fetchall():

```

```

print(row)

```

```

lab=Label(root,text="Name")
lab.place(x=0,y=0)
entname=Entry(root,width=20,textvar=textin)
entname.place(x=80,y=0)
labl=Label(root,text="Phone")
labl.place(x=0,y=40)
entphone=Entry(root,width=20,textvar=textinn)
entphone.place(x=80,y=40)
but=Button(root,padx=2,pady=2,text="Submit",command=insert)
but.place(x=60,y=100)
res=Button(root,padx=2,pady=2,text="Show",command=show)
res.place(x=160,y=100)
#=====
#UPDATE THE TABLE
name=StringVar()
phone=StringVar()
def updateContact():
    nam=name.get()
    #print(name)
    pho=phone.get()
    #print(pho)
    conn=sqlite3.connect("C:/Users/AMAN/Dropbox/My PC (LAPTOP-K3QKQ8QO)
/Downloads/SQLiteStudio-3.2.1/SQLiteStudio/fire_forest.sqlite")
    cursor = conn.cursor()
    cursor.execute("UPDATE people SET name = ? WHERE phone = ?",(nam,pho))
    conn.commit()
labunname=Label(root,text="Update Name:")
labunname.place(x=0,y=200)
enttupdatename=Entry(root,width=20,textvar=name)
enttupdatename.place(x=160,y=200)
labuphone= Label(root,text="Provide Phno Number:")
labuphone.place(x=0,y=240)
entupdatephone=Entry(root,width=20,textvar=phone)
entupdatephone.place(x=210,y=240)
buttupdate=Button(root,padx=2,pady=2,text="Update",command=updateContact)
buttupdate.place(x=80,y=280)
#=====
#DELETE A RECORD FROM THE TABLE
del1=StringVar()
def det():
    dee=del1.get()
    conn=sqlite3.connect("C:/Users/AMAN/Dropbox/My PC (LAPTOP-K3QKQ8QO)
/Downloads/SQLiteStudio-3.2.1/SQLiteStudio/fire_forest.sqlite")
    cursor = conn.cursor()
    cursor.execute("DELETE FROM 'people' WHERE name = ?", (dee,))
    conn.commit()

```

```

labdelete=Label(root,text="Delete")
labdelete.place(x=0,y=340)
endelete=Entry(root,width=20,textvar=del1)
endelete.place(x=90,y=340)

```

```

butdel=Button(root,padx=2,pady=2,text="Delete",command=det)
butdel.place(x=90,y=100)

```

```

#=====
#DROP TABLE
def drop():
    conn=sqlite3.connect("C:/Users/AMAN/Dropbox/My PC (LAPTOP-K3QKQ8QO)
/Downloads/SQLiteStudio-3.2.1/SQLiteStudio/fire_forest.sqlite")
    cursor= conn.cursor()
    cursor.execute("DROP table people")
    conn.commit()
    buttdrop=Button(root,padx=2,pady=2,text="Drop Table",command=drop)
    buttdrop.place(x=180,y=380)
#=====
root.mainloop()

```

### OUTPUT:

