MAR GREGORIOS COLLEGE OF ARTS & SCIENCE

(Approved by the Govt. of Tamil Nadu and Affiliated to University of Madras)

Block No.8, Mogappair West, Chennai-600037.

DEPARTMENT OF COMPUTER SCIENCE

LAB RECORD

ADVANCED PYTHON PROGRAMMING LAB

November - 2023

Name :

Register Number :

MASTER OF COMPUTER SCIENCE , SHIFT - I

MAR GREGORIOS COLLEGE OF ARTS & SCIENCE

(Approved by the Govt.of Tamil Nadu and Affiliated to University of Madras)

Block No.8, Mogappair West, Chennai-600037.

DEPARTMENT OF COMPUTER SCIENCE

LAB RECORD

ADVANCED PYTHON PROGRAMMING LAB

This is certified to be bonafide record of practical work done by _____Register No._____of I Year Master of Computer Science in_____subject during NOVEMBER-2023.

Head of the Department **Lecturer In-Charge**

Submitted for the University Examination held in_____at Mar Gregorios College.

Date:_____ Examiners

1)

2)

MAR GREGORIOS COLLEGE OF ARTS & SCIENCE

Block No.8, Mogappair West, Chennai-600037.

DEPARTMENT OF COMPUTER SCIENCE

ADVANCED PYTHON PROGRAMMING LAB

LAB RECORD

NOVEMBER-2023.

COMPLETED

No. of Experiments : ..... Signature

Date : .....

SL.NO	DATE	NAME OF EXPERIMENT	PAGE NO	INITIAL
1		EXCEPTION A) PROGRAM TO CATCH FOLLOWING EXCEPTION I) VALUE ERROR II) INDEX ERROR III) NAME ERROR IV) TYPE ERROR V) DIVIDEZERO ERROR B)PYTHON ROGRAM TO CREATE USER DEFINED EXCEPTION C)PYTHON PROGRAM TO UNDERSTAND THE USE OF ELSE AND FINALLY BLOCK WITH TRY BLOCK D)PYTHON PROGRAM THAT USES RAISE AND EXCEPTION CLASS TO THROW AN EXCEDPTION		
2		NUMBER LIBRARY A)CREATE A NUMPY ARRAY FROM IST TUPLE WITH FLOAT DATE B)PYTHON PROGRAM TO DEMONSTRATE SLICING INTEGER AND BOOLEAN ARRAY INDEXING C)PYTHON PROGRAM TO FIND NIN MAX SUM CUMULATIVE SUM OF ARRAY D)PYTHON PROGRMA TO DEMONSTRATE USE OF NDIM SHAPE SIZE DTYPE		
3		NUMBER LIBRARY: LINEAR ALZEBRA A)PYTHON PROGRAM TO FIND RANK DETERMINANT AND TRACE OF AN ARRAY B)PYTHON PROGRMA TO FIND EIGEN VALUES OF MATRICES C)PYTHON PROGRAM TO FIND MATRIX AND VECTOR PRODUCTS (DOT INNER OUTER PRODUCT) MATRIX EXPONENTIATION D)PYTHON PROGRAM TO SOLVE A		

		LINEAR MATRIX EQUATION OR SYSTEM OR LINEAR SCALAR EQUATIONS		
		E)CREATE A WHITE IMAGE USING NUMPY IN PYTHON		
		F)CONVERT A NUMPY ARRAY TO AN IMAGE AND CONVERT IMAGE TO NUMPY ARRAY		
		G)PERFORM SORTING SEARCHING AND COUNTING USING NUMPY METHODS		
		H)PYTHON PROGRAM TO DEMONSTRATE THE USE OF THE RESHAPE() METHOD		
4		PANDAS LIBRARY		
		A)PYTHON PROGRAM TO IMPLEMENT PANDAS SERIES WITH LABELS		
		B)CREATE A PANDAS SERIES FROM A DICTIONARY		
		C)CREATING A PANDAS DATAFRAME		
		D)PROGRAM WHICH MAKE USE OF FOLLOWING PANDAS METHODS – DESCRIBE( ), TAIL( ), HEAD( )		
		E)CONVERT DATAFRAME TO NUMPY ARRAY		
		F)PROGRAM THAT DEMONSTRATE THE COLUMN SELECTION COLUMN ADDITION AND COLUMN DELETION		
		G)PROGRAM THAT DEMONSTRATES THE ROW SELECTION ROW ADDITION AND ROW DELETION		
		H)GET N-LARGEST AN N-SMALLEST VALUES FROM A PARTICULAR COLUMN IN PANDAS DATA FRAME		
5		VISUALIZATION		
		A)WRITE A PROGRAM TO DEMONSTRATE USE OF GROUPBY( ) METHOD		
		B)PROGRAM TO DEMONSTRATE PANDAS MERGING JOINING CONCATENATION		
		C)CREATING DATA FRAME FROM CSV AND EXEL FILE		
6		OBJECT ORIENTED		

		PROGRAMMING A)BMI RESULT B)WRITE A PROGRAM TO DEMONSTRATE VARIOUS KINDS OF INHERITANCE C)WRITE A PROGRAM TO DEMONSTRATE OPERATOR OVERLOADING		
7		MULTITHREADING PYTHON PROGRAM TO CREATE TWO THREADS TO KEEP A COUNT OF NUMBER OF EVEN NUMBERS ENTERED BY THE USER		

EXCEPTION

(a) Program to catch following exception

(1) VALUE ERROR

```
import math
x = int(input("Enter positive number :"))
try
print('square root of',x,'is',math,squrt(x))
except value Error as ve:
print("Please enter the positive number")
```

OUTPUT

Enter positive number :2

square root of 2 is 1.414

Enter positive number :-2

Please enter the positive number

RESULT

Thus,the program was executed successfully.

(2) INDEX ERROR

```
import sys
try
my_list=[3,7,9,4,6]
   print(my_list[6])
except IndexError as e:
    print(e)
print(sys.executable)
```

OUTPUT

IndexError : List index out of range

RESULT

Thus,the program was executed successfully.

(3) NAME ERROR

```
def geek_message():
    try
        geek = “Geeks For Geeks”
        return geeksforgeeks
except NameError:
        return”NameError occurred same variable isn’t defined”
print(geek_message())
```

OUTPUT

NameError occurred same variable isn’t defined

RESULT

Thus,the program was executed successfully.

(4) TYPE ERROR

```
geeky_list=[“Geeky”,”GeeksforGeeks”,”SuperGeek”,”Gee
k”]
indices=[0,1,”2”,3]
for i in range(len(indices)):
    try:
        print(geeky_list[indices[i]])
        except TypeError:
print(“TypeError:check,list of indices”)
```

OUTPUT

Geeky

GeeksforGeeks

TypeError:check list of indices

Geek

RESULT

Thus,the program was executed successfully.

(5) DIVIDE ZERO ERROR

```
def sample(num)
try:
   div=1/num
except ZeroDiisionError:
print("We cannot divide by zero")
else:
print(div)
num=int(input("Enter the number :"))
sample(num)
```

OUTPUT

Enter the number : 0

We cannot divide by zero

RESULT

Thus,the program was executed successfully.

(b) Python program to create user defined exception

```
class PercentageError(Exception):
    pass
class InvalidPercentageError(Percentage Error):
    def_init_(self):
super()._init_("percentage is invalid")
class LessPercentageError(PercentageError)
    def_init_(self):
super()._init_("The percentage is lesser then the cut-off,Please try again:")
    class checkPercentage(PercentageError):
        def_init_(self.per):
        if per<80:
            raise LessPercentageError
        if per>100:
            raise InvalidPercentageError
       print("Congrats you're Enrolled")
```

```
try:
        print(“For percentage:93”)
       checkPercentage(93)
       except Exception as e:
                print(e)
try:
      print(“\n For Percentage: 102”)
      checkPercentage(102)
      except Exception as e:
                print(e)
try:
        print(“\n For percentage:58”)
        checkPercentage(58)
        except Exception as e:
                 print(e)
```

OUTPUT

For percentage:93

Congrats you’re Enrolled

For percentage:102

Percentage is invalid

For percentage:58

The percentage is lesser then the cut-off,please try again

RESULT

Thus,the program was executed successfully.

(c) Python program to understand the use of else and finally block with try block

```
try:
X=int(input("Enter a number :"))
except:
print("An error occurred!")
else
print("The number entered is",X)
finally:
print("Program ended")
```

OUTPUT

```
Enter a number : a
An error occurred!
Program ended
Enter a number : 20
The number entered is 20
Program ended
```

RESULT

Thus,the program was executed successfully.

(d) Python program that user raise and exception class to throw an exception

```
try:
  a = int(input("Enter a positive integer :")
  if a<=0:
      raise ValueError("This is not a positive integer!")
except ValueError as e:
      print(e)
```

OUTPUT

```
Enter a positive integer : -1
This is not a positive integer!
```

RESULT

Thus,the program was executed successfully.

NUMPY LIBRARY

(a) Create a numpy array rom list,tuple with float type

```
import numpy as np
npArray=np.array([1.3,2.6,3.4,4.3,5.4,6.8,7.2,8.5,9.3]dtype=floa
t)
print("Contents of the npArray:",npArray)
print(npArray[2:5])
print(npArray[:4])
print(npArray[5:])
print(npArray[:])
#tuple
tuple1=([8.5,4.2.6],[1,2,3.5])
array2=np.asarray(tuple1)
print(array2)
s=np.array([0,1,2,3,4,5,6,7,8,9])
print(s[2:5])
print(s[:4])
print([6:1])
print([:])
```

OUTPUT

Contents of the
npArray:[1.3,2.6,3.4,4.3,5.4,6.8,7.2,8.5,9.3]

[3.4,4.3,5.4]

[1.3,2.6,3.4,4.3]

[7.2,8.5,9.3]

[1.3,2.6,3.4,4.3,5.4,6.3,7.2,8.5,9.3]

[3,4,5]

[0,1,2,3,4]

[7,8,9]

[0,1,2,3,4,5,6,7,8,9]

RESULT

Thus,the program was executed successfully**.**

(b) Python program to demonstrate slicing integer and Boolean array indexing

```
import numpy as np
arr=np.array([[1,2,0,-4],[4,6,7,0],[3,-7,8,5]])
#Slicing array
temp=arr[:2,::2]
print("Array with first 2 rows and alternate columns (0and2):\n",temp)
#indexing array
temp=arr[[0,1,2,1],[3,2,1,0]]
print("\n Elements at indices(0,2),(1,2),(2,1),(2,0):\n",temp)
```

OUTPUT

Array with first 2 rows and alternate columns (0and2):[[1 0] [4 7]]

Elements at indices(0,2),(1,2),(2,1),(2,0): [-4 7 -7 4]

RESULT

Thus,the program was executed successfully**.**

(c) Python program to find min,max,sum,cumulative sum of array

```
import numpy as np
x=np.array([[0,1],[2,3]])
print("Original array:")
print(x)
print("Minimum element of an array:")
print(np.min(x))
print("Maximum element of an array:")
print(np.max(x))
print("Sum of all element:")
print(np.sum(x))
print("Sum of each column:")
print(np.sum(x,axis=0))
print("Sum of each row:")
print(np.sum(x,axis=1))
print("Cumulative sum of an array:")
print(np.cumsum(x))
```

OUTPUT

Original array:[[0 1] [2 3]]

Minimum element of an array: 0

Maximum element of an array: 3

Sum of all element: 6

Sum of each column: [2 4]

Sum of each row: [1 5]

Cumulative sum of an array: [0 1 3 6]

RESULT

Thus,the program was executed successfully.

(d) Python program to demonstate use of ndim,shape,size,dtype

```
import numpy as np
a=np.array([[[1,2,3],[4,5,6],[7,8,9],[10,11,12]]])
print(“Dimension of the given Ndarray=”,a.ndim)
print(“Shape of the given Ndarray=”,a.shape)
print(“Size of the given Ndarray=“,a.size)
print(“Data type of the given Ndarray=”,a.dtype)
print(“Size of the each element in Ndarray=”,a.itemsize)
```

OUTPUT

Dimension of the given Ndarray=3

Shape of the given Ndarray=(1,4,3)

Size of the given Ndarray=12

Data type of the given Ndarray=int32

Size of the each element in Ndarray=4

RESULT

Thus,the program was executed successfully.

NUMBER LIBRARY-LINEAR ALGEBRA

(a) Python program to find rank,determinant and trace of an array

```
import numpy as np
mat=np.array([[50,29],[30,44]])
print("Numpy matrix is:")
print(mat)
det=np.linalg.det(mat)
print("\n Determinant of given 2*2 matrix:")
print(int(det))
rank=np.linalg.matrix_rank(mat)
print("Rank of the given matrix is:",rank)
print("Trace of the matrix:",mat,trace())
```

OUTPUT

Numpy matrix is:[[50 29] [30 44]]

Determinant of given 2*2 matrix:1330

Rank of the given matrix is:2

Trace of the matrix:94

RESULT

Thus,the program was executed successfully.

(b) Python program to find eigan values of matrices

```
import numpy as np
a=np.array([[3,1],[2,]])
w=np.linalg.eig(a)
print(w)
```

OUTPUT

Array([3.73205081,0.26794919]

RESULT

Thus,the program was executed successfully.

(c) Python program to find matrix and vector products matrix exponentiation

```
import numpy as np
from numpy.linalg import matrix_power
a=np.array([2,6])
b=np.array([3,10])
print("Vector:")
print("a=",b)
print("\n Inner product of vector a and b=")
print(np.inner(a,b))
print("\n Outer product of vector a and b=\n")
print("Matrix Exponentiation")
i=np.array([[0,1].[-1,0]])
print(matrix_power(i,3))
```

OUTPUT

Vector:a=[2 6] b=[3 10]

Inner product of vector a and b= 66

Outer product of vector a and b= [[6 20] [18 60]]

Matrix Exponentiation [[0 -1] [1 0]]

RESULT

Thus,the program was executed successfully.

(d) Python program to solve a linear matrix equation or system of linear scalar equation

```
import numpy as np
arr1=np.array([[1,2],[3,5]])
arr2=np.array([1,2])
print("Result…",np.linalg.solve(arr1,arr2))
```

OUTPUT

Result…[-1,1]

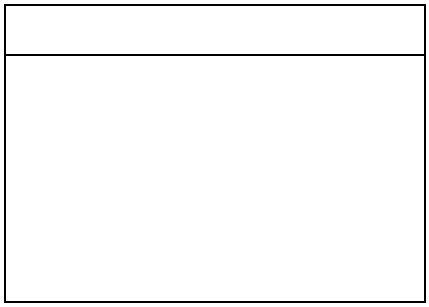
RESULT

Thus,the program was executed successfully.

(e) Create a white image using Numpy in python

```
#importing the libraries
import cv2
import numpy as np
#creating an array using np.full
#255 is code for white color
array_created=np.full((500,500,3),255,dtype=np.uint8)
#displaying the image
cv2.imshow(“image”,array_created)
```

OUTPUT



RESULT

Thus,the program was executed successfully.

(f) Convert a numpy array to an image and convert image to Numpy array

```
import numpy as np
from PIL import Image as im
def main():
        array=np.arange(0,737280,1,np.uint8)
        print(type(array))
        print(array.shape)
        array=np.reshape(array,(1024,720))
        print(array.shape)
        print(array)
        data=im.fromarray(array)
        data.save('gfg_dummy_pic.png')
if_name_==”_main_”:
main()
```

OUTPUT

```
<class 'numpy.ndarray'>
(737280,)
(1024,720)
[[  0     1     2   … 205 206 207]
 [208 209 210 … 157 158 159]
 [160 161 162 … 109 110 111]
  …
 [144 145 146 … 93  94  95]
 [ 96  97  98 … 45 46 47]
 [ 48  49  50 … 253 254 255]]
```

RESULT

Thus,the program was executed successfully.

(g) Perform sorting,searching and counting using numpy method

```
import numpy as np
Arr=np.array([[1,20,5],[21,4,3],[11,5,50]])
SortedArr=np.sort(Arr)
print(“Hello world”)
print(“Original Array:”)
print(“\n Sorted Array:”)
print(SortedArr)
CountingArr=np.sum(Arr)
Print(CountingArr)
```

OUTPUT

Hello world

Original Array:[[1 20 5] [21 4 3][11 5 50]]

Sorted Array:[[1 5 20] [3 4 21] [5 11 50]]

120

RESULT

Thus,the program was executed successfully.

h) Python program to demonstrate the use of reshape() method

```
import numpy as geek
array1 = geek.arange(4)
print(“Original array : \n”,array1)
array2 = geek.arange(4).reshape(2,2)
print(“\n array reshaped with 2 rows and 4 columns : \n”, array2)
```

OUTPUT :

Original array :

[0 1 2 3]

Array reshaped with 2 rows and 4 columns :

[[0 1]

[2 3]]

RESULT :

Thus the program was executed successfully.

PANDAS LIBRARY

a) Python program to implement pandas series with labels

```
import pandas as pd
import numpy as np
data = np.array(['g','e','e','k','f','o','r','g','e','e','k','s'])
ser = pd.series(data,index = [10,11,12,13,14,15,16,17,18,19,20,21,22])
print(ser[18])
```

OUTPUT :

g

RESULT :

Thus the program was executed successfully.

b) Create a pandas series from a dictionary

```
import pandas as pd
dictionary = {‘A’:10,’B’:63,’C’:32}
s = pd.series(dictionary)
print(s)
```

OUTPUT :

A 10

B 63

C 32

RESULT :

Thus the program was executed successfully.

c) Creating a pandas data frame

```
import pandas as pd

data = {‘key’:[‘k0’,’k1’,’k2’,’k3’,’k4’],
’Name’:[‘ram’,’uma’,’tom’,’tim’,’jim’] ,‘Age’:[22,23,21,24,32];
‘Qualification’:[‘mca’,’msc’,’ma’,’mba’,’phd’]}

df = pd.DataFrame(data)

print(df)
```

OUTPUT :

	Key	Name	Age	Qualification
0	k0	ram	22	mca
1	k1	uma	23	msc
2	k2	tom	21	ma
3	k3	tim	24	mba
4	k4	jim	32	phd

RESULT :

Thus the program was executed successfully.

d) Program which make use of following pandas methods

i) describe( ) tail ( )

```
import pandas as pd
#define a dictionary containing employee data
data = {‘Name’:’jai’,’princi’,’gaurav’,’anuj’],’Age’:’27’,’24’,’22’,23’],
‘Qualification’:’Msc’,’MA’,’MCA’,’Phd’]}
#convert the dictionary into dateframe
df = pd.DtaeFrame(data)
address = [‘Delhi’,’Bangalore’,’Chennai’,’Patna’]
df[‘Address’]=address
df.drop([“Qualification”], axis=1, inplace=True)
print(df.describe( ))
df.tail(2)
```

OUTPUT :

	Name	Age	Address
2	Gaurav	22	Chennai
3	Anuj	32	Patna

RESULT :

Thus the program was executed successfully.

ii) head( )

```
import pandas as pd

#define a dictionary containing employee data

data =
{‘Name’:[‘jai’,’princi’,’gaurav’,’anuj’],’Age’:[27,24,22,32],’Qualification’:[‘
Msc’,’MA’,’MCA’,’Phd’]}

#convert the dictionary into DataFrame

df=pd.DataFrame(data)

address=[‘Delhi’,’Bangalore’,’Chennai’,’Panta’]

df[‘Address’]=address

df.drop([“Qualification”],axis=1, inplace=True)

df.head(2)
```

OUTPUT :

	Name	Age	Address
0	Jai	27	Delhi
1	Princi	24	Bangalore

RESULT :

Thus the program was executed successfully.

e) Convert data frame to numpy array

```
#initialize a dataframe
df=pd.DataFrame([[1,2,3],[4,5,6],[7,8,9,],[10,11,12]], columns=['a','b','c'])
#convert dataframe to numpy array
arr = df.to_numpy( )
print('\nNumpy Array \n----------\n', arr)
print(type(arr))
```

OUTPUT :

```
Numpy Array
[ [ 1  2  3]
  [4  5   6]
  [7  8  9]
  [10  11  12]]
<class ‘numpy.ndarray’>
```

RESULT :

Thus the program was executed successfully.

f) Python that demonstrate the column selection, column addition and column deletion

```
#import pandas package

import pandas as pd

#Define a dictionary containing employee data

data = {‘Name’:’jai’,’princi’,’gaurav’,’anuj’],’Age’:’27’,’24’,’22’,23’],
‘Qualification’:’Msc’,’MA’,’MCA’,’Phd’]}

#convert the dictionary into dateframe

df = pd.DtaeFrame(data)

address = [‘Delhi’,’Bangalore’,’Chennai’,’Patna’]

df[‘Address’]=address

df.drop([“Qualification”], axis=1, inplace=True)

#select two columns

print(df[[‘Name’,’Address’]])
```

OUTPUT :

	Name	Address
0	Jai	Delhi
1	Princi	Bangalore
2	Gaurav	Chennai
3	Anuj	Patna

RESULT :

Thus the program was executed successfully.

g) Program that demonstrate the row selection, row deletion and deletion

```
import pandas as pd

import numpy as np

dict={‘name’:[‘sneha’,’logi’,’vini’,’sri’,’vidhya’],’address’:[‘kanpur’,’luckno
w’,’chennai’,’ponneri’,’redhills’],’tamil’:[90,88,76,44,22],’english’:[88,90,65,
44,35],’maths’:[45,67,89,98,67]}

df=pd.DataFrame(dict)

print(df)

print(“after selecting 4th row”)

print(df.iloc[3])

df2={‘name’:’kala’,’address’:’lucknow’,’tamil’:88,’english’:90,’maths’:67}

df=df.append(df2,ignore_index=True)

print(df)

print(“after deleting”)

df=df.drop(df.index[1])

print(df)
```

OUTPUT :

	Name	Address	tamil	english	maths
0	sneha	Kanpur	90	88	45
1	logi	lucknow	88	90	67
2	vini	Chennai	76	65	89
3	sri	ponneri	44	44	98
4	vidhya	redhills	22	35	67

After selecting 4th row

Name sri

Address ponneri

Tamil 44

English 44

Maths 98

Name	Address	tamil	english	maths
0 sneha	Kanpur	90	88	45
1 logi	lucknow	88	90	67
2 vini	Chennai	76	65	89
3 sri	ponneri	44	44	98
4 vidhya	redhills	22	35	67
5 kala	lucknow	88	90	67

After deleting

0	sneha	Kanpur	90	88	45
2	vini	Chennai	76	65	89
3	sri	ponneri	44	44	98
4	vidhya	redhills	22	35	67
5	kala	lucknow	88	90	67

RESULT :

Thus the program was executed successfully.

h) Get n-largest and n- smallest values from a particular columns

```
import pandas as pd

df=pd.DataFrame([[10,20,30,40],[7,14,21,28],[55,15,8,12],[15,14,1,8],[7,1,1,
8],[95,4,9,2]],
columns=['apple','orange','banana','pear'],index=['basket1','basket2','baske
t3','basket4','basket5','basket6'])

print("\n--------nsmallest----------\n")

print(df.nsmallest(2,['aple']))

print("\n--------nlargest----------\n")

print(df.nlargest(2,['aple']))
```

OUTPUT :

--------nsmallest----------

	Apple	Orange	Banana	Pear
Basket 6	5	4	9	2
Basket 2	7	14	21	20

----------nlargest----------

	Apple	Orange	Banana	Pear
Basket 3	55	15	8	12
Basket 4	15	14	1	8

RESULT :

Thus the program was executed successfully.

VISUALIZATION

a) Write a program to demonstrate us of group by ( ) method

```
import pandas as pd
#creating the dataframe
df=pd.read_csv(nba.csv")
#first grouping based on "Team"
#within each team we are grouping base on "Position"
gkk = df.groupby(['Team','Position'])
#print the first value in each group
gkk.first( )
```

OUTPUT :

Team	Position	Name	Number	Age	Height	Weight	College	Salary
Atlanta Hawks	C	AJ Horford	15.0	30	6-10	245.0	Florida	12000000.0D
	PF	Kris Humphriess	43.0	31	6-9	235.0	Minnesota	1000000.0D
	PG	Dennis Schroder	17.0	22	6-1	172.0	Wage forest	1763400.0D
	SF	Kent Bazemore	24.0	26	6-5	201.0	Old dominion	2000000.0D
	SG	Tim Hardaway Jr	10.0	24	6-6	205.0	Michigan	1304520.0D
Boston Cettics	C	Kelly Olynyk	41.0	25	7-0	238.0	Gonzage	2165160.0D
	PF	Jones Jerebko	8.0	29	6-10	231.0	LSU	5000000.0D
	PG	Avery Bradley	0.0	25	6-2	180.0	Texas	7730337.0D
	SF	Jae Crowder	99.0	25	6-6	235.0	Marquette	6796117.0D
	SG	John Holland	30.0	27	6-5	205.0	Boton university	1148640.0D
Brooklyn Nets	C	Brook Lopez	11.0	28	7-0	275.0	Stanford	19689000.0D
	PF	Chris McCullough	1.0	21	6-11	200.0	Syracuse	1140240.0D
	PG	Jarret Jack	2.0	32	6-3	200.0	Georgia tech	6300000.0D
	SG	Bojin Bogdanobic	44.0	27	6-8	216.0	Oklahoma state	3425510.0D
Charlote Horne	C	AJ Jefferson	25.0	31	6-10	289.0	Wisconsin	13500000.0D
	PF	Taylor Hansbrough	50.0	30	6-9	250.0	North Carolina	947276.0D
	PG	Jorge Gutierrtz	12.0	27	6-3	189.0	California	189455.0D
	SFf	Michael Kidd-Gilchr	14.0	22	6-7	232.0	Kentucky	6331404.0D

RESULT :

Thus the program was executed successfully.

b) Program to demonstrate pandas merging joining and concatenting

A) Merging :

```
#importing panda module

import  pandas as pd

#define a dictionary containing employee  data

data1={‘key’:[‘K0’,’K1’,’K2’,’K3’],‘Name’:[‘Jai’,’Princi’,’Gaurav’,’Anuj’], ‘Age’:[‘27’,’24’,’22’,32’]}

#define a dictionary containing employee  data

data1={‘key’:[‘K0’,’K1’,’K2’,’K3’],‘Address’:[‘Nagpur’,’Kanpur’,’Allahabad’,’Kannuaj’], ‘Qualification’:[‘Btech’,’BA’,’Bcom’,Bhons’]}

#convert the dictionary into DataFrame

df=pd.DataFrame(data1)

#convert the dictionary into DataFrame

df1=pd.DataFrame(data2)

print(df,”\n\n”,df1)
```

Run on IDE

Now we are using .merge() with one unique key combination

```
#using .merge( ) function
```

```
res=pd.merge(df, df1, on='key')
res
```

OUTPUT :

```
   Key  Name   Age
0  K0   Jai     27
1  K1   Princi  24
2  K2   Gaurav  22
3  K3   Anuj    32
```

```
   Key  Address    Qaulification
0  K0   Nagpur     Btech
1  K1   Kanpur     BA
2  K2   Allahabad  BCom
3  K3   Kannuaj    Bhons
```

	Key	Name	Age	Address	Qualification
0	K0	Jai	27	Nagpur	Btech
1	K1	Princi	24	Kanpur	BA
2	K2	Gaurav	22	Allahabad	Bcom
3	K3	Anuj	32	Kanuaj	Bhons

RESULT :

Thus the program was executed successfully

B) Ccncatenation:

```
#importing panda module

import  pandas as pd

#define a dictionary containing employee  data

data1={‘Name’:[‘Jai’,’Princi’,’Gaurav’,’Anuj’], ‘Age’:[‘27’,’24’,’22’,32’],
‘Address’:[‘Nagpur’,’Kanpur’,’Allahabad’,’Kannuaj’],
‘Qualification’:[‘Msc’,’MA’,’MCAaaaa’,Phd’]}

#define a dictionary containing employee  data

data2={‘Name’:[‘Abhi’,’Ayushi’,’Dhiraj’,’Hitesh’],
‘Age’:[‘17’,’14’,’12’,52’],
‘Address’:[‘Nagpur’,’Kanpur’,’Allahabad’,’Kannuaj’],
‘Qualification’:[‘Btech’,’BA’,’Bcom’,Bhons’]}

#convert the dictionary into DataFrame

df=pd.DateFrame(data1, index=[0,1,2,3])

#convert the dictionary into DataFrame

df=pd.DateFrame(data1, index=[4,5,6,7])

print (df, “\n\n”, df1)

#using a. concat( ) method

frames=[df, df1]

res1=pd.concat(frames)

res1
```

OUTPUT :

	Name	Age	Address	Qaulification
0	Jai	27	Nagpur	Msc
1	Princi	24	Kanpur	MA
2	Gaurav	22	Allahabad	MCA
3	Anuj	32	Kannuaj	Phd
4	Abhi	17	Nagpur	Btech
5	Ayusthi	14	Kanpur	BA
6	Dhiraj	12	Allahabad	BCom
7	Hitesh	52	Kannuaj	Bhons

	Name	Age	Address	Qaulification
0	Jai	27	Nagpur	Msc
1	Princi	24	Kanpur	MA
2	Guarav	22	Allahabad	MCA
3	Anuj	32	Kannuaj	Phd
4	Abhi	17	Nagpur	Btech
5	Ayusthi	14	Kanpur	BA
6	Dhiraj	12	Allahabad	BCom
7	Hitesh	52	Kannuaj	Bhons

RESULT :

Thus the program was executed successfully

C) Joining:

```
#importing padas module

Import pandas as pd

#define a dictionary containing employee details

data1={‘Name’:[’Jai’,’Princi’,’Gaurav’,’Anuj’], ‘Address’:
[‘Nagpur’,’Kanpur’,’Allahabad’,’Kanniaj’],
‘Qualification’:[‘Msc’,’MA’,’MCA’,’Phd’], ‘Mobile no’:[97,91,58,76]}

#define a dictionary containing employee details

data2={‘Name’:[’Gaurav’,’Anuj’,’Dhiraj’,’Hitesh’], ‘Age’:[22,32,12,52],
‘Address’: [‘Allahabad’,’Kannauj’,’Allahabad’,’Kannuaj’],
‘Qualification’:[’MCA’,’Phd’,’Bcom’,’Bhons’],
‘Salary’:[1000,2000,3000,4000]}

#convert the dictionary into DataFrame

df=pd.DataFrame(data1, index=[0,1,2,3]

#convert the dictionary into DataFrame

df=pd.DataFramedata2, index=[4,5,6,7]

print(df, “\n\n”, df1)

Run on IDE

Now we set axes join = inner for intersection of dataframe

#applying concat with axes

#join = ‘inner’

res2=pd.concat([df,df1],axis=1,join=’inner’)
```

OUTPUT :

	Name	Age	Address	Qualification	Mobile no
0	Jai	27	Nagpur	Msc	97
1	Princi	24	Kanpur	MA	91
2	Gaurav	22	Allahabad	MCA	58
3	Anuj	32	Kannuaj	Phd	76

	Name	Age	Address	Qualification	Salary
2	Gaurav	22	Allahabad	MCA	1000
3	Anuj	32	Kannuaj	Phd	2000
6	Dhiraj	12	Allahabad	Bcom	3000
7	Hitesh	52	Kannuaj	Bhons	4000

	Name	Age	Address	Qualification	Mobileno	Name	Age	Address	Qualification	Salary
2	Gaurav	22	Allahabad	MCA	58	Gaurav	22	Allahabad	MCA	1000
3	Anuj	32	Kannuaj	Phd	76	Anuj	32	Kannuaj	Phd	2000

RESULT :

Thus the program was executed successfully

c) Creating data frame from CSV and exel file

```
#importing pandas as pd
import pandas as pd
#creating the dataframe
df=pd.read_csv(“nba.csv)
#print the dataframe
df
```

OUTPUT :

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Cellics	0.0	PG	25.0	6-2	180.0	Texas	
1	Jae Crowder	Boston Cellics	99.0	SF	25.0	6-6	235.0	Maruette	
2	John Holland	Boston Cellics	30.0	SG	27.0	6-5	185.0	Boston	
3	RJ Hunter	Boston Cellics	28.0	SG	22.0	6-5	231.0	Georgia state	
4	Jonas Jerebko	Boston Cellics	8.0	PF	29.0	6-10	240.0	NaN	
5	Amir Johnson	Boston Cellics	90.0	PF	29.0	6-9	235.0	NaN	
6	Jordan mickey	Boston Cellics	55.0	PF	21.0	6-8	238.0	LSU	
7	Kelly Chynyk	Boston Cellics	41..0	C	25.0	7-0	190.0	Gonzaga	
8	Terry Rozler	Boston Cellics	12.0	PG	22.0	6-2	220.0	Louisville	
9	Marcus smart	Boston Cellics	36.0	PG	22.0	6-4	260.0	Okshamo state	
10	Jared sullinger	Boston Cellics	7.0	C	24.0	6-9	185.0	Ohio state	
11	Isaiah Thomas	Boston Cellics	4.0	PG	27.0	5-9	220.0	Washington	
12	Evan tumer	Boston Cellics	11.0	SG	27.0	6-7	215.0	Ohio state	
13	James young	Boston Cellics	13.0	SG	20.0	6-6	253.0	Kentucky	
14	Tyler zeller	Boston Cellics	44.0	C	26.0	7-0	216.0	North Carolina	
15	Bojan Bogdano	Boston Cellics	44.0	SG	27.0	6-8	251.0		

RESULT :

Thus the program was executed successfully

OBJECT ORIENTED PROGRAMMING

a) Python program for BMI result

```
class person:
  def init (self,name=’’ age=0,height=0,weight=0):
self.name=’name’
self.age=age
self.height=height
self.weight=weight
def BMI_result(self):
self.BMI=self.weight/(self.height/100)**2
print(“your body mass index is”,self.BMI)
if self.BMI<=18.5:
print(“oops! You are under weight”)
elif self.BMI<=24.9:
print(“awesome you are healthy”)
elif self.BMI<=29.9:
```

```
print(“Eee! you are over weight”)
else:
print(“seesh! You are obese”)
p1=person()
p1.init(‘priya’,21,159,45)
p1.BMI_result( )
```

OUTPUT:

Your body mass index is 17.799928800284796

Oops! You are under weight

RESULT :

Thus the program was executed successfully

b)Write a python program to to demonstrate various kind of inheritance

```
class calculation1:
 def summation(self, a, b):
return a=b;
class calculation2:
 def multiplication(self, a, b):
return a*b
class derived(calculation1, calculation2):
def divide(self, a, b):
return a/b;
d=derived( )
print(d.summation(10,20))
print(d.multiplication(10,20))
print(d.divide(10,20))
```

OUTPUT :

30

200

0.5

RESULT :

Thus the program was executed successfully

c) Write a program to demonstrate operator overloading

```
class bubble:
def__init__(self,volume):
self.volume=volume
def__add__(self, other):
volume=self.volume + other.volume
return volume
b1=bubble(20)
b2=bubble(30)
b3=bubble(“hai”)
b4=bubble(“welcome”)
print(b1+b2)
print(b3+b4)
```

OUTPUT :

50

haiwelcome

RESULT :

Thus the program was executed successfully

MULTITHREADING

Python program to create two threads to keep count of numbers

```
from time import sleep, perf_counter
from threading import Thread
numbers=(1,2,3,4,5,6,7,8,9)
count_odd=0
count_even=0
for x in numbers:
if not x%2:
count_even+=1
else:
count_odd+=1
def task( 0):
print(‘starting a task…’)
print(“number of even numbers:”, count_even)
sleep(1 9)
```

```
print(“number of odd numbers:”, count_odd)
start_time=perf_counter( )
t1=Thread(target=task)
t1.start( )
t1.join( )
end_time=perf_counter( )
print(f‘it  took {end_time-start_time:0.2f} second(s) to complete.’)
```

OUTPUT :

Starting a task….

Number of even numbers : 4

Number of odd numbers : 5

It took 1.03 second(s) to complete.

RESULT :

Thus the program was executed successfully