PYTHON PROGRAMMING AND DATA SCIENCE

**Assignment:1**

**1. Write a program to print "Hello World" using function.**

def hw():

return "Hello world"

print(hw())



**2. Write a program to  add  two numbers  and   print the  result**

a=10

b=20

print("sum of",a," + ",b,"=",(a+b))



**3. Write a program to  add   two  numbers  and print the    result using function.**

def sum(a,b):

print("sum =",a+b)

sum(10,50)



**4. Write  a   program to      add     two    numbers   and return the result using function.**

def sum(a,b):

return "sum =",(a+b)

print(sum(10,34))



**5. Write a program to add, subtract, multiply and divide two numbers and print the result.**

a=int(input("Enter your 1st no."))

b=int(input("Enter your 2nd no."))

c=a+b

d=a-b

e=a\*b

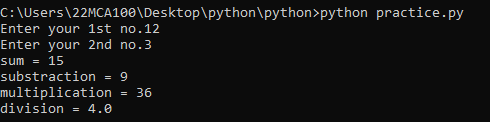
f=a/b

print("sum =",c)

print("substraction =",d)

print("multiplication =",e)

print("division =",f)



**6. Write a program to add, subtract, multiply and divide two numbers using function and return the result.**

def cal(a,b):

sum=a+b

sub=a-b

mul=a\*b

div=a/b

return sum,sub,mul,div

print(cal(20,5))



**Assignment 2:**

**1.Write a function to add, subtract, multiply and divide two numbers using function and return the result in list.**

def cal(a,b):

res=[]

sum=a+b

sub=a-b

mul=a\*b

div=a/b

res=sum,sub,mul,div

return res

print(list(cal(20,5)))



**2. Write a function to find even numbers and return a list.**

def even(lst):

l=[]

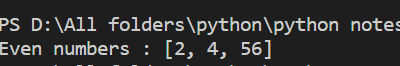
for i in lst:

if i%2==0:

l.append(i)

return l

print("Even numbers :",even([2,4,3,56,43,77]))



**3. Write a function to find odd numbers and return a list.**

def even(lst):

l=[]

for i in lst:

if i%2==1:

l.append(i)

return l

print("Even numbers :",even([2,4,3,56,43,77])) 

**4. Write a function to find prime numbers and return a list.**

def max(lst):

l=[]

for i in lst:

if i==0 or i==1:

continue

for j in range(2,i//2+1):

if i%j==0:

break

else:

l.append(i)

if len(l):

print("Prime number list :",l)

else:

print("No prime number in the list .")

max([1,2,5,23,22])



**5. Write a function inside another function.**

def fun(s):

def fun1():

return "Hy budy"

return fun1()+"How are you ,"+s

print(fun("Bikash"))



**6. Write a program to pass function as a parameter. Hint write a function, which returns a string ‘How are you?’ Pass this function, as a parameter to another function that print Hi, How are you**

def fun(fun):

def fun1():

return "How are you ?"

return fun+","+fun1()

print(fun("Hi"))



**Assignment:3**

**1.Write a Python function to find the maximum of three numbers.**

def find(a,b,c):

if a>b and a>c:

print(a,"a is greater .")

if b>c:

print(c," is smaller .")

else:

print(a," is smaller .")

elif b>c and b>a:

print(b," is greater .")

if a>c:

print(c," is smaller .")

else:

print(a," is smaller .")

elif c>a and c>b:

print(c," is greater .")

if b>a:

print(a," is smaller .")

else:

print(b," is smaller .")

else :

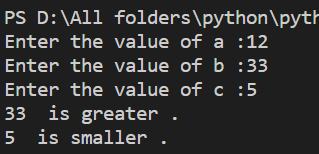
print("all are same .")

a=int(input("Enter the value of a :"))

b=int(input("Enter the value of b :"))

c=int(input("Enter the value of c :"))

find(a,b,c)



**2. Write a Python function to multiply all the numbers in a list. (Numbers can be negative, positive or zero)**

def fun(lst):

mul=1

for i in lst:

mul=mul\*i

return mul

print(fun([10,22,3,16,5]))



**3. Write a Python function to calculate the factorial of a number. The function accepts the number as an argument.**

def fun(n):

mul=1

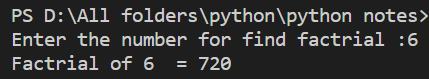
for i in range(1,n+1):

mul=mul\*i

return mul

n=int(input("Enter the number for find factrial :"))

print("Factrial of",n," =",fun(n))



**4. Write a Python function that takes a list and returns a new list with distinct elements from the first list**

def unique(lst):

l=[]

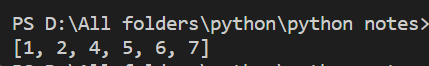
for i in lst:

if i not in l:

l.append(i)

return l

print(unique([1,2,2,4,5,5,1,6,7,7]))



**5. Write a python function to find the largest item from a given list.**

def max(lst):

a=lst[0]

for i in range(len(lst)):

if lst[i]>a:

a=lst[i]

return a

print("Greatest number form the list is =",max([1,2,2,4,5,555,1,6,7,77]))



**Assignment:4**

**1.Write a function to add up all the numbers in a list.**

def sum(lst):

sum=0

for i in lst:

sum=sum+i

return sum

print("Sum of elements in the list :",sum([23,10,45,50]))



**2. Write a function takes a list of strings and returns a new list that contains capitalized strings**.

def capatalize(lst):

l=[]

for i in lst:

l.append(i.capitalize())

return l

print("Capitalize elements in the list :",capatalize(["bikash","prabhat","sachin"]))



**3. Write a function called middle that takes a list and returns a new list that contains all but the first and last elements should be removed. So middle([1,2,3,4]) should return [2,3].**

def remove\_ele(lst):

lst.pop(0)

lst.pop(-1)

return lst

print("Middle element list :",remove\_ele([1,3,5,75,7]))



**4. Write a function which breaks a string into individual letters.**

def indi(lst):

l=[]

for i in lst:

l.append(i)

return l

print("Indivdual element of string :",indi("Jagannath"))



5. Write a function which takes a list of strings and concatenates the elements.

def concatinate(lst):

sum=""

for i in lst:

sum=sum+i

return sum

print("Concatinate element of the list:",concatinate(["jay","Jagannath","Ram"]))



**6. Write a function that takes a list and returns a new list with distinct elements from the first list**

def unique(lst):

l=[]

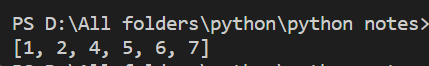
for i in lst:

if i not in l:

l.append(i)

return l

print(unique([1,2,2,4,5,5,1,6,7,7]))



**Assignment 5:-**

**1. Write lambda function to find maximum value from two numbers.**

f=lambda a,b:a if a>b else b

max=f(10,5)

print("Greater number is :",max)



**2. Write a lambda function to find modulo of given number.**

mode=lambda x,y:x%y

a,b=[int(n) for n in input("Enter two number : ").split(',')]

print("module :",mode(a,b))



**3. Use filter to find odd values from a list. Use lambda function to**

**Write function.**

lst=[1,5,23,6,64,7,8]

odd=lambda x:True if x%2==1 else False

result=list(filter(odd,lst))

print("Odd numbers list is :",result)



**4. Use filter function to find values greater than 10 and less than 500 function.**

seq=[28,90,1,54,600,23,65,800,50]

f=filter(lambda x:True if x>10 and x<500 else False,seq)

print(list("Numbers are :",f))



**5. Use map to find square for each values of the list. Use lambda function to write function.**

def sqr(n):

return n\*n

lst=[3,2,5,9]

#map(function,iterationList)

res=map(sqr,lst)

print("Square of all the elements of the list is = ",list(res))



**6. Use filter function to extract vowels from given list of alphabets**

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

seq=['f','j','u','k','i']

vow=filter(lambda x: True if (x in v) else False,seq)

print("Vowel letters is the list :",list(vow))



**7. Use map function to increase salary by 25% of all employees.**

def sal(n):

return (n+n\*0.25)

se=[100,200,300,400]

x=map(sal,se)

print("25% increased salary :",list(x))



**8. Write a Python program to convert all the characters into uppercase. Use map**

def up(n):

return n.upper()

lst=['f','e','u']

res=map(up,lst)

print("Upper case letter are : ",list(res))



**Assignment 6:-**

**1.Using List Comprehension to Iterate through a String.**

str="bikash pradhan"

lst=[i for i in str]

print("List :",lst)



**2. Please check in the range from 0 – 9 if the item’s value is divisible by 2.**

obj=[]

check=[i for i in range(10) if i%2==0 ]

print(list(check))



**3. Check the five numbers from 0 to 9. If y is divisible by 2, then even is appended to the obj list.If not, odd is appended.**

lst=[2,3,5,7,9]

obj=[]

odd=[]

for i in lst:

if i%2==0:

obj.append(i)

else:

odd.append(i)

print("Even list :",obj)

print("Even list :",odd)



**4. Finding the elements in a list in which elements are ended with the letter ‘b’ and the length of that element is greater than 2.**

lst=["bik","bbb","ikb","iik","jaieb"]

k=[x for x in lst if x[-1]=="b" and len(x)>2]

print(k)



**5. Add two list X & Y and display the result.**

l=[1,2,3,5]

l1=[10,23,34,44]

l2=[] #3rd list

for i in range(len(l)):

sum=l[i]+l1[i] #sum of list elements

l2.append(sum)

print("sum of two list elements :",l2)



**6. Lets take two list L1 & L2 with numbers and create another list L3 with numbers present in L1 but not in L2**

l1=eval(input("Enter your 1st list :"))

l2=eval(input("Enter your 2nd list :"))

l3=[]

print("Enter your 3rd list :")

for i in range(len(l1)):

n=int(input())

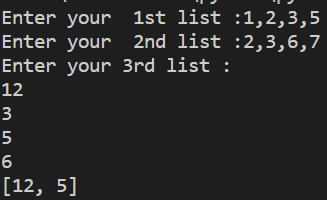
if n in l2:

continue

else:

l3.append(n)

print(l3)

****

**Assignment 7:-**

**1)Write a python program to create a dictionary with the employee details and retrieve the value upon giving the keys.**

emp={1:"nasi",2:"bei",5:"kaesi"}

print(emp[1])

print(emp[2])

print(emp[5])



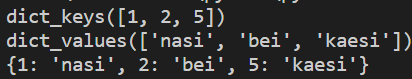
**2) Write a python program to retrieve keys, values and key-value pair from a dictionary.**

emp={1:"nasi",2:"bei",5:"kaesi"}

print(emp.keys())

print(emp.values())

print(emp)



**3) Write a python program to create a dictionary and find the sum of values.(Use eval & sinput method).**

dic={"A":1,"h":5,"t":10}

sum=0

for i in dic.values():

sum+=i

print("sum of the values of dictionary :",sum)



**4) Write a python program to create a dictionary from keyboard and display the elements.**

dic={}

n=int(input("Enter the size of the dictinary :"))

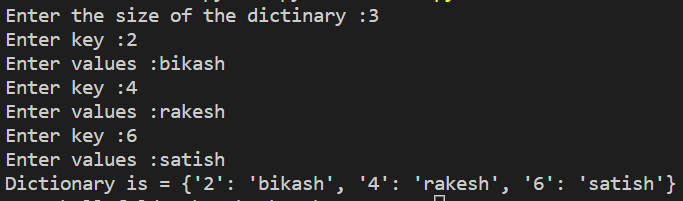
for i in range(n):

key=input("Enter key :")

val=input("Enter values :")

dic.update({key:val})

print("Dictionary is =",dic)



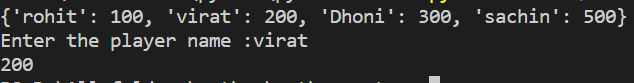
**5) Write a python program to create a dictionary with cricket player’s names and scores in a match.Also we are retrieving runs by entering the player’s name.**

cir={"rohit":100,"virat":200,"Dhoni":300,"sachin":500}

print(cir)

x=input("Enter the player name :")

print(cir[x])

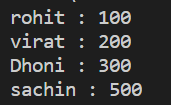


**6) Write a python program to show the usage of for loop to retrieve elements of dictionaries.**

cir={"rohit":100,"virat":200,"Dhoni":300,"sachin":500}

for i in cir:

print(i,":",cir[i])



**7) Write a python program to find the number of occurrences of each letter in a string using**

**Dictionary**

var="JayJagannath"

dic={}

for i in var:

key=i

val=var.count(i)

dic.update({key:val})

print("Occurance of letter :",dic)

****

**8) Write a python program to sort the elements of a dictionary based on a key or values.(use lambda function)**

dic={1:"zai",3:"eine",5:"ien"}

key=lambda i:dic.keys()

dict\_sort=sorted(dic.items(),key=lambda x:x[0])

print(dict\_sort)

****

**9) Write a python program to convert the elements of two lists into key-value pairs of dictionary.**

l1=[1,2,4,5]

l2=['ake','eie','bei','3wue']

dic={}

for i in range(len(l1)):

key=l1[i]

val=l2[i]

dic.update({key:val})

print(dic)



**10) Write a python program to convert string into key value pair and store them into a dictionary**

var="RadhaSyam"

dic={}

for i in var:

key=i

val=var.count(i)

dic.update({key:val})

print("Occurance of letter :",dic)



**Assignment 8:-**

**1.Write a function to return an integer, write one decorator to increment the value by returned by function, write another decorator to multiply the value by 2. Print the results, then change the order of decorator applied and print the result**.

def decor(fun):

def inner():

val=fun()

return val+10

return inner

def decor1(fun):

def inner():

val=fun()

return val\*2

return inner

def num():

return 30

#@decor

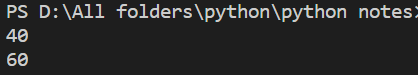
#@decor1

result1=decor(num)

result2=decor1(num)

print(result1())

print(result2())



**2. Define a module containing math functions like a. Add b. Subtract c. Multiply d. Divide Create a module to create a decor to print name of the math function, import math and decor in third file to use the function defined in math module and print name using decor module.**

**Math.py**

def Add(\*args):

total=0

for i in args:

total += i

print("Add : ",total)

def Subtract(\*args):

total=0

for i in args:

total -= i

print("Subtract : ",total)

def Multiply(\*args):

total=1

for i in args:

total \*= i

print("Multiply : ",total)

def Division(\*args):

total=1

for i in args:

total /= i

print("Division : ",total)

**Decor.py**

from Math import \*

def func\_name(func):

def wrapper(\*args):

print("Function Name : "+func.\_\_name\_\_)

func(\*args)

return wrapper

**Result.py**

import Math

import Decor

e = func\_name(Add)

f= func\_name(Subtract)

g= func\_name(Multiply)

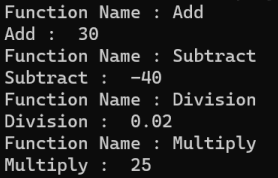
h= func\_name(Division)

e(10,20)

f(10,30)

h(10,5)

g(5,5)



**3. Define a module geometric containing functions to calculate area of a. Square b. Rectangle c. Circle d. Triangle e. IsSquare Create a module to create a decor to print name of the Geometirc function, import Geometirc and decor in third file to use the function defined in Geometirc module and print name using decor module.**

**Geom.py**

def Square(r):

print(r\*r)

def Rectangle(h,w):

print(h\*w)

def Circle(r):

print(3.14\*r\*r)

def Triangle(h,b):

print(h\*b/2)

def IsSquare():

print("True")

**Decor.py**

from Geom import \*

def func\_name(func):

def wrapper(\*args):

print("Function Name : "+func.\_\_name\_\_)

func(\*args)

return wrapper

**Area.py**

import Geom

import Decor

s = func\_name(Square)

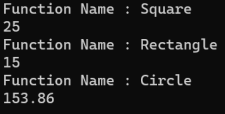
r = func\_name(Rectangle)

c = func\_name(Circle)

s(5)

r(5,3)

c(7)



**4. Write a generator to returns a sequence in given range (Hint. values between x and y say 5 and 10)**

def rang(x,y):

while x<=y:

yield x

x+=1

g=rang(5,10)

lis=[]

for i in g:

lis.append(i)

print(lis)



**5. Write generator to returns a sequence in given range (Hint. values between x and y say 10 and 1)**

def fun(a,b):

while a>b:

yield a

a-=1

val=fun(10,1)

l=[]

for i in val:

l.append(i)

print(l)



**6. Write a program to display the source of execution of a program using name variable.**

def fun(func):

def wrapper(\*args):

print("Function that started running ",func.\_\_name\_\_)

func(\*args)

return wrapper

def add(\*args):

sum=0

for i in args:

sum+=i

print("Total =",sum)

k=fun(add)

k(20,34,55)



**9.write a python program to convert celsius to Fahrenheit and vice-a-versa.**

def c\_f(cel):

fer=(1.8\*cel)+32

return fer

def f\_c(fer):

cel=(fer-32)/1.8

return cel

ch=input("Enter c or f to convert temperature (c=celsius and f=faherheit):")

val=int(input("Enter the value :"))

if ch=='f':

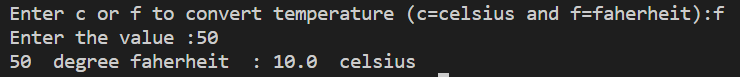
print(val," degree faherheit :",f\_c(val)," celsius ")

elif ch=='c':

print(val,"degree celsius :",c\_f(val)," faherheit")

else:

print("Invalid input! ")



**10.Write a Program to Convert Decimal to Binay ,Octal,Hexadecimal.**

def de\_by(val,x):

l=[]

while(val!=0):

rem=val%x

l.append(rem)

val=val//x

l.reverse()

for i in l:

print(i,end=" ")

a=int(input("Enter value to convert :"))

print("Binary is :",end="")

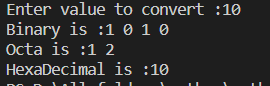
de\_by(a,2)

print("\nOcta is :",end="")

de\_by(a,8)

print("\nHexaDecimal is :",end="")

de\_by(a,16)



**11.Write a program to make a simple calculator (using funcion).**

def add(a,b):

return "Addition :",(a+b)

def sub(a,b):

return "Substraction :",(a-b)

def mul(a,b):

return "Multiplication :",(a\*b)

def div(a,b):

return "Divison:",(a/b)

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

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

x=input("Enter operation(+,-,\*,/) :")

if x=="+":

print(add(a,b))

elif x=="-":

print(sub(a,b))

elif x=="\*":

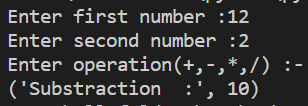
print(mul(a,b))

elif x=="/":

print(div(a,b))

else:

print("Enter valid input !")



**12.Write a program in python to find the maximun and minimun numbers oout of threee use-entered numbers .**

def find(a,b,c):

if a>b and a>c:

print(a,"a is greater .")

if b>c:

print(c," is smaller .")

else:

print(a," is smaller .")

elif b>c and b>a:

print(b," is greater .")

if a>c:

print(c," is smaller .")

else:

print(a," is smaller .")

elif c>a and c>b:

print(c," is greater .")

if b>a:

print(a," is smaller .")

else:

print(b," is smaller .")

else :

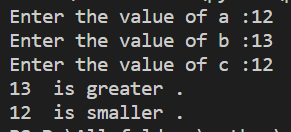
print("all are same .")

a=int(input("Enter the value of a :"))

b=int(input("Enter the value of b :"))

c=int(input("Enter the value of c :"))

find(a,b,c)



**13.Write a program that will allow the user to enter 10 numbers and display the largest odd number from them.It will display an appropriate message in case.no odd number is found.**

def odd():

print("Enter you 10 numbers :")

l=[]

l2=[]

for i in range(1,11):

x=int(input())

l.append(x)

for i in l:

if i%2==1:

l2.append(i)

print("Odd and lagrest number :",max(l2))

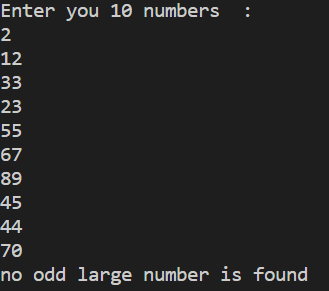
break

else:

print("no odd large number is found ")

break

odd()

****

**14.Write a python program to check if the number is Amstrong number .**

x=int(input("Enter you number :"))

lst=str(x)

sum=0

for i in lst:

sum+=int(i)\*\*len(lst)

if sum==x:

print( x," is an armstrong number .")

else:

print( x," is not armstrong number .")



**15.Write a Python program to perform the following operation on give string input :**

**a)Count Number of Vowel in a given string**

def vowel(lst):

l=["a","e","i","o","u"]

count=0

for i in lst:

if i.lower() in l:

count+=1

return count

print("Number of vowels in the string :",vowel("Kaushitamam"))



**b)Count Length of String(do not use len())**

def length(s):

count=0

for i in s:

count+=1

return count

print("Lenght of word is =",length("Jayanamam"))



**c)Reverse String**

print("PrasantSir"[::-1])



**d)Find and Replace operation**

l="Bikash pradhan"

x=l.replace("Bikash","Akash")

print(x)



l="Hy , My name is Bikash Pradhan "

print(l.find("is"))



**e)Check whether Stirng entered is a palindrom or not :**

s=str(input("Enter a string :"))

rev=s[::-1]

if s==rev:

print(" String is palindrome ")

else:

print("String is not a palindrome ")



**16.Write a program in python to implement the Fibonnacci series up to user enterd number.(use recursive function).**

def fab(n):

if n<=1:

return n

else:

return(fab(n-1)+fab(n-2))

x=int(input("Enter a range for find fabonacci series : "))

if x<=0:

print("Enter a positive number !")

else:

print("Fabonacci sequense is :",end="")

for i in range(x):

print(fab(i)," ",end="")



**17.Write a program in python to implement the Factorial series up to user-enterd number.(use recursive function).**

def fun(n):

if n==0:

return 1

return n\*fun(n-1)

val=int(input("Enter the value for find factorial :"))

print("Factorial of ",val,"is :",fun(val))



**Assingment 9:**

**1. Write a program to generate TypeError Exception and handle it.**

x=1

y='hello'

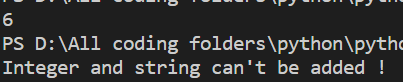
try:

result=x+y

print(result)

except TypeError :

print("Integer and string can't be added !")

****

**2. Write a program to generate ZeroDivisionError Exception and handle it.**

a=10

b=0

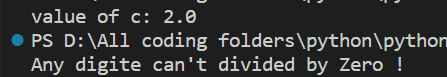
try:

c=a/b

print("value of c:",c)

except ZeroDivisionError:

print("Any digite can't divided by Zero !")



**3. Write a program to generate SyntaxError Exception and handle it.**

try:

eval("x===x")

except SyntaxError:

print("There is a syntatical error !")



**4. Write a program to generate IOError Exception and handle it.**

try:

f=open("P9.txt","r")

content=f.readline()

f.close

except IOError:

print("IOError exception !")



**5. Write a program to generate NameError Exception and handle it.**

try:

name="Kausika pal mam"

print(nam)

except NameError:

print("This is nameError Exception ! ")



**Assignment 10:-**

**1.Write a program to find city of given person, have some user with their city in dictionary, the user will enter name to find his/her city, the program should print name of the city and if the user is not found it should use user-defined exception and give message that the user doesn’t exist.**

class Myexc(Exception):

pass

persion\_det={'ram':'surat','syam':'ahemdabad','rakesh':'bhubneswar'}

person\_name=input("Enter you name :")

try:

if person\_name.lower() in persion\_det:

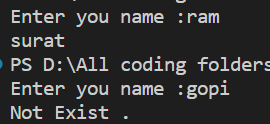
print(persion\_det[person\_name.lower()])

else:

raise Myexc("Not Exist .")

except Myexc as e:

print(e)

****

**2. Write a program to enter a specific number say n, the program should not terminate until user enters specific number n. It should generate exception Too SmallValue if the value < n and generate exception TooLargeValue if value > n.**

class Myexception(Exception):

pass

n=50

while(True):

guess=int(input("Enter guess values :"))

try:

if(guess==n):

print("You win !")

break

elif(guess>=n):

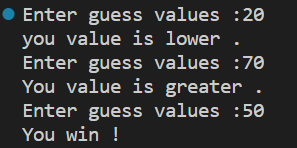
raise Myexception("You value is greater .")

else:

raise Myexception("you value is lower .")

except Myexception as e:

print(e)



**3.Write a program to accept positive digit. Define a user-defined exception to check whether the value given is a valid positive digit or not. If the value is not numeric the exception Not\_Suitable\_value\_exceptions should be triggered and handled.**

class Myexc(Exception):

pass

while(True):

try:

n=int(input("Enter a positive number :"))

if(n>=0):

print("perfect number .")

break

else:

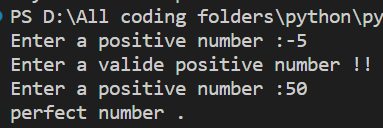
raise Myexc("Enter a valide positive number !!")

except Myexc as e:

print(e)

except ValueError:

print("Not\_Suitable\_value")



**4. Write a user-defined exception to generate message if the balance of the given user is less than 2000INR in his bank account. Use dictionary to maintain account data with username and balance.**

class Myexception(Exception):

pass

dic={"rakesh":20000,'gopi':3000,'pratik':4000,'bikash':1000}

while(True):

name=input("Enter the name of the user :")

try:

if dic[name] <2000:

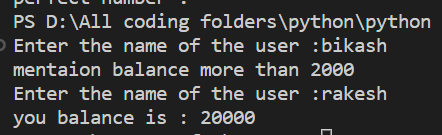
raise Myexception("mentaion balance more than 2000")

else:

print("you balance is :",dic[name])

except Myexception as e:

print(e)



**5. Write a program, which accepts marks of a student (between 0 to 100) and checks whether it is within the range or not. If it is within the range then it displays “marks entered successfully”, if not then it throws the exception of user defined class “MarksOutOfRangeException” .**

class Myexc(Exception):

pass

while(True):

try:

marks=int(input("Enter you markes :"))

if(marks>=0 and marks<=100 ):

print("marks entered successfully")

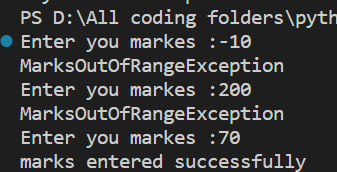
break

else:

raise Myexc("MarksOutOfRangeException")

except Myexc as e:

print(e)



1. **Write an assert statement to accept a value in range of 10 and 20.**

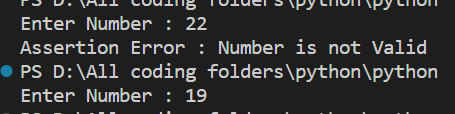
try:

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

assert x>=10 and x<=20 , "Number is Valid."

except AssertionError:

print("Assertion Error : Number is not Valid")



1. **Write an assert statement to accept a positive number.**

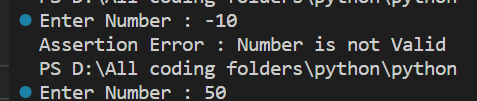
try:

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

assert x>=0 , "Number is Valid."

except AssertionError:

print("Assertion Error : Number is not Valid")



1. **Write a program to generate TypeError, NameError, and ZeroDivision error and handle** .

try:

a=10

# c=a/0

# print(d)

b = "hello"

e = a+b

except ZeroDivisionError as e:

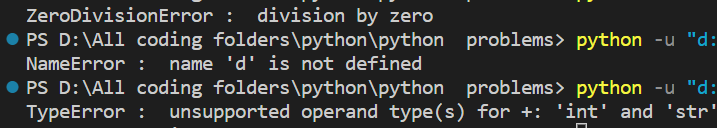
print("ZeroDivisionError : ",e)

except NameError as e:

print("NameError : ",e)

except TypeError as e:

print("TypeError : ",e)



**9.Write a program to generate and handle ValueError.**

try:

s = int(input("Enter Value : "))

print(s)

except ValueError as e:

print("ValueError : ",e)



**10.Write a program to generate and handle KeyError.**

dic = {"himansu":"surat","jagdis":"delhi","bimal":"varoda","mukesh":"ahmedabad"}

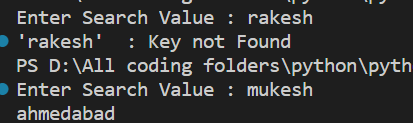
try:

s = input("Enter Search Value : ")

print(dic[s])

except KeyError as e:

print(e," : Key not Found")



**Assignment 11(file handling):**

**1. Write a program to open a file content.txt in write mode and write data, the file should take input until it finds @ character.**

f=open("context.txt",'w')

x=" "

while x!="@":

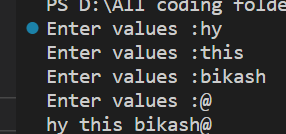
x=input("Enter values :")

f.write(x)

f=open("context.txt",'r')

print(f.read())

f.close()



**2. Write a program to open a file content.txt in read mode and read the content of the file.**

f=open("context.txt",'r')

print(f.read())

f.close()



**3. Open and read a content of given file, check whether file exists or not, if file doesn’t exist give message. (Hint. Use os.path.isfile(filename) to check file existence).**

import os.path

path='./the.txt'

check\_file=os.path.exists(path)

print("File is not exits =",check\_file)

****

**4. Write a program to copy image in a new file. (Binary file).**

f1=open('xyz.jpg','rb')

f2=open("new.jpg",'wb')

x=f1.read()

f2.write(x)

print("writer succesfully .")

f1.close()



**5. Read the content of the file. Use with method to open a file.**

f=open('context.txt','r')

print(f.read())

f.close



**6.Consider a student object with id, name, and percentage and create a class. Import this class and use pickle to dump several objects in a file named Student.dat. No. of objects should be taken as input.**

import student

Import pickel as p

f = open("Student.dat","wb")

n = int(input("How many students : "))

for i in range(n):

sid = int(input("Enter Id : "))

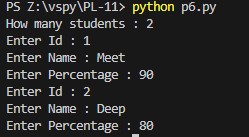
name = input("Enter Name : ")

percentage = float(input("Enter Percentage : "))

s = student.student(sid,name,percentage)

p.dump(s,f)

f.close()



**7. Read content from Student.dat using pickle.**

import student

import pickle as p

f = open("Student.dat","rb")

while True:

try:

obj = p.load(f)

obj.display()

except Exception as e:

break

f.close()



**8 .Write a program to search for city name in the binary file named cities.bin and display the record number that contains the city name.**

recordLength = 20

with open("cities.bin","wb") as f:

n = int(input("Enter Number Of cities : "))

for i in range(n):

city = input("Enter City Name : ")

l = len(city)

city = city + (recordLength - l)\*' '

l = len(city)

city = city.encode()

f.write(city)

reclen = 20

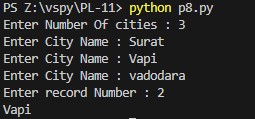
with open("cities.bin","rb") as f:

n = int(input("Enter record Number : "))

f.seek(reclen\*(n-1))

str = f.read(reclen)

print(str.decode())



**9. Write a program to update or modify a record in a binary file(cities.bin).**

import os

reclen = 20

size = os.path.getsize("cities.bin") n = int(size/reclen)

with open("cities.bin","r+b") as f:

city = input("Enter City Name : ") city = city.encode()

newCity = input("Enter New Name : ")

ln = len(newCity) newCity = newCity + (20-ln)\*' ' newCity = newCity.encode() position = 0

found = False

for i in range(n):

f.seek(position) st = f.read(20) if city in st:

print("Updated Record Number : ",(i+1))

found = True f.seek(-20,1)

f.write(newCity)

position += reclen

if not found:

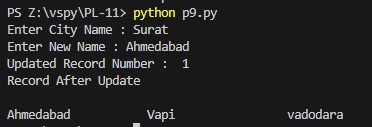
print("City Not Found")

with open("cities.bin","r+b") as f:

s = f.read()

print("Record After Update\n")

print(s.decode())



1. **Write a program to randomly access a record from a binary file(cities.bin).**

reclen = 20

with open("cities.bin","rb") as f:

* 1. = int(input("Enter Record Number : ")) f.seek(reclen \* (n-1)) st = f.read(reclen)

print(st.decode())



**12.Write a program to create a phone book with names and phone numbers. Store data in binary file named Phonebook.dat.**

with open("PhoneBook.dat","wb") as f:

n= int(input("How many PhoneNo Add ? : "))

for i in range(n):

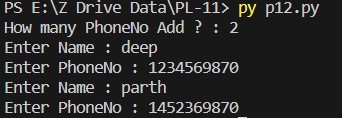
name = input("Enter Name : ")

phoneno = input("Enter PhoneNo : ")

name = name.encode()

phoneno = phoneno.encode()

f.write(name+phoneno)



import sys,mmap

while True: print("1-To display all the entries\n2-To display Phone numbers\n3-Modify an entry\n4-Exit")

ch = int(input("\nEnter Your Choice :"))

with open("phonebook.dat","r+b") as f:

mm = mmap.mmap(f.fileno(), 0)

if ch == 1:

print(mm.read().decode())

elif ch == 2:

name = input('Enter name:')

n = mm.find(name.encode())

n1 = n+len(name)

ph = mm[n1: n1+10]

print('Phone no: ', ph.decode())

elif ch == 3:

name = input('Enter name:')

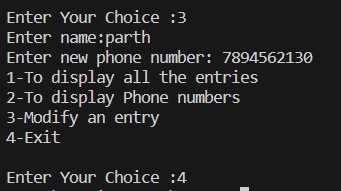
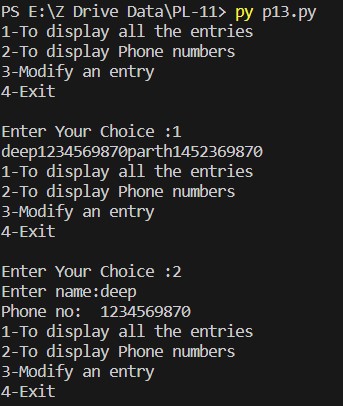
n = mm.find(name.encode())

n1 = n+len(name)

sys.exit()

else:

print("Invalid Choice.")



**Assignment -12(Data Science):**

**1. Consider data shown in table I in student.csv file, and write listed programs in python**

**a**. **A program to display Id on X-axis and Stipend on Y-axis in the form of bar graph.**

import pandas as pd

import matplotlib.pyplot as plt

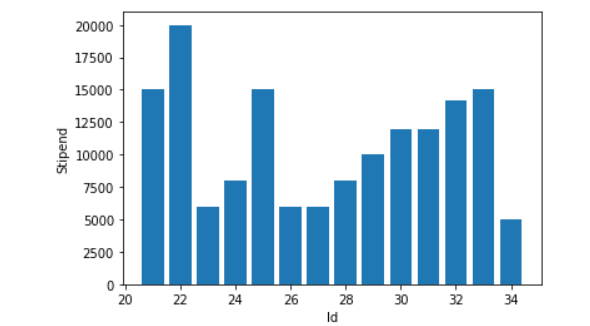
df=pd.read\_csv("student.csv")

plt.bar(df['Id'] ,df["Stipend"])

plt.xlabel('Id')

plt.ylabel('Stipend')

plt.show()



**b. A program to display histogram showing the percentage of Students in specific age group.**

import pandas as pd

import matplotlib.pyplot as plt

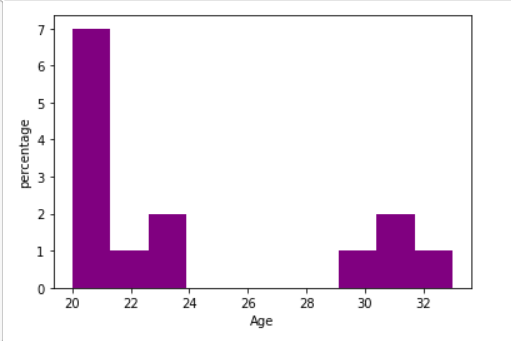
df=pd.read\_csv("student.csv")

plt.hist(df['Age'],bins=10,color='purple')

plt.xlabel('Age')

plt.ylabel('percentage')

plt.show()



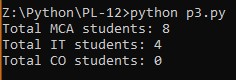
**C .Write program to find total number of students in Branch MCA, IT and CO departments.**

import pandas as pd

data = pd.read\_csv("student.csv")

mca\_students = data[data['Branch'] == 'MCA'] it\_students = data[data['Branch'] == 'IT'] co\_students = data[data['Branch'] == 'CO'] mca\_count = len(mca\_students) it\_count = len(it\_students) co\_count = len(co\_students)

print(f"Total MCA students: {mca\_count}") print(f"Total IT students: {it\_count}") print(f"Total CO students: {co\_count}")



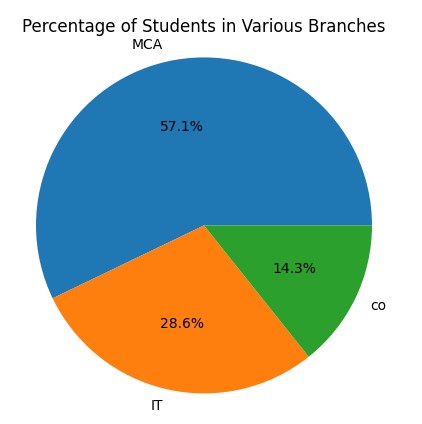
**D - A program to display pie charts showing the percentage of students in various branches.**

import pandas as pd import matplotlib.pyplot as plt

data = pd.read\_csv('student.csv')

branch\_counts = data['Branch'].value\_counts() branches = branch\_counts.index.tolist() student\_counts = branch\_counts.values.tolist() total\_students = sum(student\_counts)

percentages = [(count / total\_students) \* 100 for count in student\_counts] plt.pie(percentages, labels=branches, autopct='%1.1f%%') plt.title('Percentage of Students in Various Branches') plt.axis('equal') plt.show()

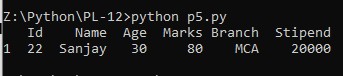


1. **- Display name of student with maximum stipend.**

import pandas as pd

data = pd.read\_csv('student.csv')

max\_stipend = data['Stipend'].max() name = data[data['Stipend'] == max\_stipend] print(name)



**E. Display names of student who’s age is 20.**

import pandas as pd

data = pd.read\_csv('student.csv')

filtered\_data = data[data['Age'] == 20]

if not filtered\_data.empty: student\_names = filtered\_data['Name'].values.tolist() print(f"Student(s) with age 20 : {student\_names}") else:

print(f"No student found with age 20")



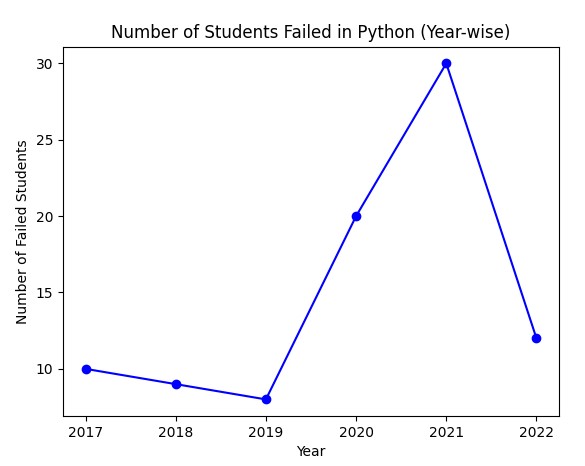
**2. Write a program to create line graph to show the number of students failed in python yearwise. Consider years =[‘2017’, ‘2018’, ‘2019’, ‘2020’, ‘2021’, ‘2022’] and number of students failed as Failed\_Students = [10, 9, 8, 20, 30, 12].**

import matplotlib.pyplot as plt

years = ['2017', '2018', '2019', '2020', '2021', '2022'] failed\_students = [10, 9, 8, 20, 30, 12]

plt.plot(years, failed\_students, marker='o', linestyle='-', color='b') plt.title('Number of Students Failed in Python (Year-wise)') plt.xlabel('Year')

plt.ylabel('Number of Failed Students') plt.show()



**Program – 2**

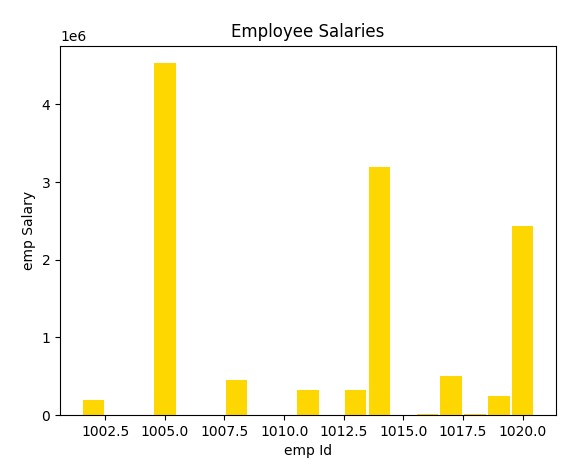
**1.To display employee id numbers on X-axis and their salaries on Y-axis in the form of a bar graph**

import matplotlib.pyplot as plt import pandas as pd

Emp = pd.read\_csv("EmployeeProduction.csv")

Emp

x = Emp['Id'] y = Emp['Salary'] plt.xlabel('emp Id') plt.ylabel('emp Salary') plt.title('Employee Salaries')

plt.bar(x, y, label='Employee data',width=0.9,color='gold') plt.show()

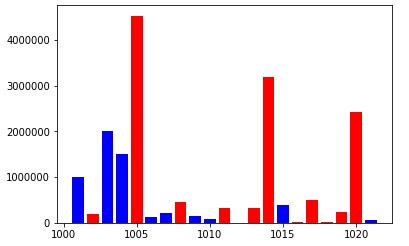
**2. To display employee id numbers on X-axis and their salaries on Y-axis in the form of a bar graph for two departments of a company.**

import matplotlib.pyplot as plt import pandas as pd

Emp = pd.read\_csv("EmployeeProduction.csv")

Emp

x = Emp['Id'] y = Emp['Salary'] x1 = Emp['Id'] y1 = Emp['Salary'] plt.xlabel('emp Id') plt.ylabel('emp Salary') plt.title('Employee Salaries') plt.bar(x, y, label='Sales Department', color='blue') plt.bar(x1, y1, label='Production Department', color='red') plt.show()



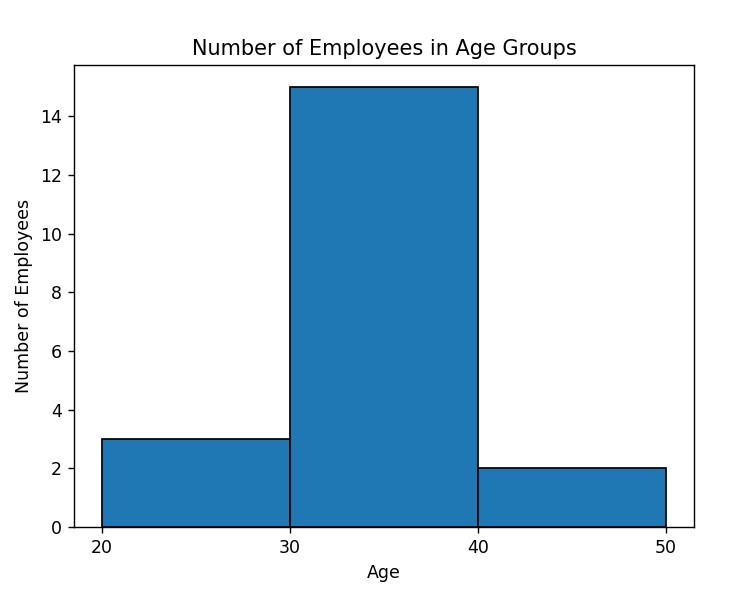
**3.To display a histogram showing the number of employees in a specific age group.**

import matplotlib.pyplot as plt

ages = [28, 35, 42, 39, 33, 30, 32, 28, 37, 40, 35, 32, 31, 29, 30, 35, 38, 31, 35, 36] age\_groups = [20, 30, 40, 50]

plt.hist(ages, bins=age\_groups, edgecolor='black') plt.xlabel('Age')

plt.ylabel('Number of Employees') plt.title('Number of Employees in Age Groups') plt.xticks(age\_groups) plt.show()

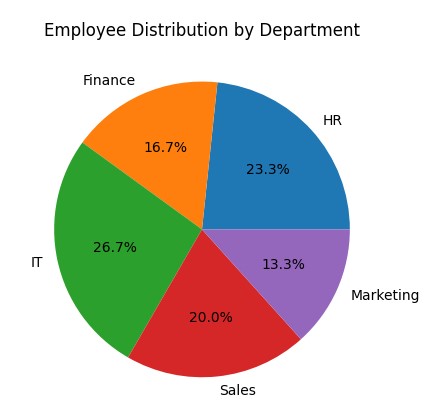


1. **To display a pie chart showing the percentage of employees in each department of a company**

import matplotlib.pyplot as plt

departments = ['HR', 'Finance', 'IT', 'Sales', 'Marketing'] employee\_counts = [35, 25, 40, 30, 20]

plt.pie(employee\_counts, labels=departments, autopct='%1.1f%%') plt.title('Employee Distribution by Department') plt.show()



1. **To create a line graph to show the profits of a company in various years.**

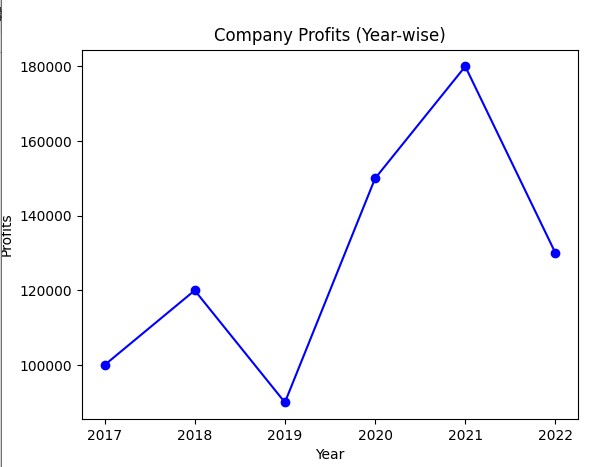
import matplotlib.pyplot as plt

years = ['2017', '2018', '2019', '2020', '2021', '2022'] profits = [100000, 120000, 90000, 150000, 180000, 130000]

plt.plot(years, profits, marker='o', linestyle='-', color='b') plt.title('Company Profits (Year-wise)')

plt.xlabel('Year') plt.ylabel('Profits')

plt.show()



**Assignment 13:-**

**1.A python program to create a regular expression to retrieve all words starting with a in a given string.**

import re

str = "Hello, How are you,jyoti"

result = re.findall(r'H[\w]\*',str)

print(result)

****

**2.A python program to create a regular expression to retrieve all words starting with a numeric digit.**

import re

str="Hello, my name is 23rakesh,90jhunjhun"

result = re.findall(r'\d[\w]\*',str)

print(result)

****

**3.A python program to create a regular expression to retrieve all words having 5 characters length.**

import re

str = "Hello, How are you,prbhat"

result = re.findall(r'\b\w{5}\b',str)

print(result)

****

**4.A python program to create a regular expression to retrieve all words having 5 characters length using search().**

import re

str = "Hello, How are you,krishna"

result = re.search(r'\b\w{5}\b',str)

print(result))

****

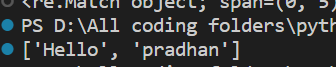
**5.A python program to create a regular expression to retrieve all words having length of at least 4 characters.**

import re

str = "Hello, How are you,pradhan"

result = re.findall(r'\b\w{4,}\b',str)

print(result)

****

**6.A python program to create a regular expression to retrieve all words having length of 3 or 4 or 5 characters.**

import re

str = "Hello, How are you,harish"

result = re.findall(r'\b\w{3,5}\b',str)

print(result)

****

**7.A python program to create a regular expression to retrieve only single digits from a string.**

import re

str = "Hello, 78 How 8are you, Ranoliya"

result = re.findall(r'\d+',str) print(result)



**8.A python program to create a regular expression to retrieve the last word of a string, if it starts with t.**

import re

str = "tiktak twist topper tomato"

result = re.findall(r't(\w+)$',str)

print(result)

****

**9.A python program to create a regular expression to retrieve the phone number of a person from string.**

import re

str = "pratik - 28394820383"

result = re.findall(r'\d+',str)

print(result)

****

**10.A python program to create a regular expression to retrieve only name but not the phone number of a person from string.**

import re

str = "pratik - 28394820383"

result = re.findall(r'^\w+',str)

print(result)

****

**11.A python program to create a regular expression to retrieve birth date from a string.**

import re

str = "My name is Deep. I was born on 16-09-2001 in surat city." result = re.findall(r'\d{2}[/-]\d{2}[/-]\d{4}',str) print(result)



**12.A python program to create a regular expression to search whether a given string is starting with ‘He’ or not.**

import re

str = input("Enter String : ") #search using regex

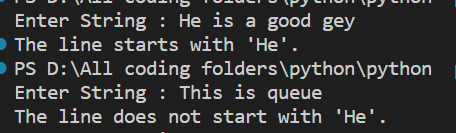
x = re.search(r'^He', str)

if(x!=None):

print('The line starts with \'He\'.')

else:

print('The line does not start with \'He\'.')

****

**13. A python program to create a regular expression to search for a word at the ending of a string.**

import re

str = input("Enter String : ")

x = re.search(r'\w+\S\*$', str)

print(x)

****

**14 .A python program to create a regular expression to search for a word at the ending of a string by ignoring the case.**

import re

str = input("Enter String : ")

word = input("Enter Search word :")

pattern = re.compile(fr'\b{word}\b$', re.IGNORECASE)

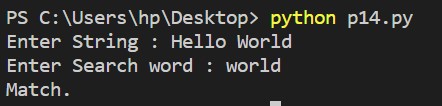
x = pattern.search(str)

if(x!=None):

print('Match.')

else:

print('Not Match.')



**15. A python program to create a regular expression to retrieve marks and names from a given string.**

import re

def getNameAndMarks(str):

matches = re.findall(r'(\w+)\s\*-\s\*(\d+)', str)

result = []

for match in matches:

name = match[0]

marks = int(match[1])

result.append((name,marks))

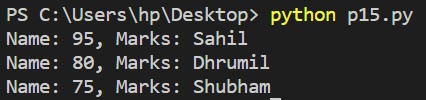
return result

str = "Sahil - 95, Dhrumil - 80, Shubham - 75"

result = getNameAndMarks(str)

for marks, name in result:

print("Name: {0}, Marks: {1}".format(name,marks))

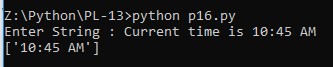


**16. A python program to create a regular expression to retrieve the timing s either ‘am’ or ‘pm’.**

import re

str = input("Enter String : ")

x=re.findall(r'\d{2}[:]\d{2}\s?(?:am|pm|AM|PM)$',str)

print(x) 

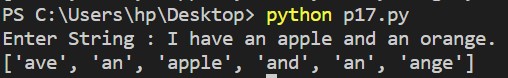
**17. A python program to create a regular expression to find all words starting with ’an’ or ’ak’ .**

import re

str = input("Enter String : ")

result = re.findall(r'[an|ak]\w+',str)

print(result)



# Assignment – 14 :

**1. With data given in CancerData.csv create a classification model to predict breast tumour data into malignant breast tumour or benign breast tumour and obtain its accuracy level.**

import pandas as pd

from sklearn.model\_selection import train\_test\_split from sklearn.ensemble import RandomForestClassifier from sklearn.metrics import accuracy\_score

data = pd.read\_csv('CancerData.csv')

X = data.drop('id', axis=1) # Features y = data['id'] # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42) classifier = RandomForestClassifier(random\_state=42) classifier.fit(X\_train, y\_train) y\_pred = classifier.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)

**2. With data given in iris.csv create a classification model to classify iris data into setosa, virginica, versicolor using iris dataset and obtain its accuracy level.**

import pandas as pd

from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LogisticRegression from sklearn.metrics import accuracy\_score

data = pd.read\_csv('iris.csv')

X = data.drop('species', axis=1) y = data['species']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42) model = LogisticRegression() model.fit(X\_train, y\_train) y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)

**Output :**

**3. Implement a python program that takes internal marks (x), finds the equation that best fits the data and is able to predict External marks using the data given in q9.csv. (Use linear regression)**

import pandas as pd from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression from sklearn.metrics import mean\_squared\_error

data = pd.read\_csv('q9.csv')

X = data['InteralMarks'].values.reshape(-1, 1) # Features y = data['ExternalMarks'].values # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42) regression = LinearRegression() regression.fit(X\_train, y\_train) y\_pred = regression.predict(X\_test) mse = mean\_squared\_error(y\_test, y\_pred)

print("Mean Squared Error:", mse)



**4. Implement a python program that takes Population (x), finds the equation that best fits the data and is able to predict Profit using the data given in**

PopulationProfit.csv. (Use linear regression)

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

data = pd.read\_csv('PopulationProfit.csv')

X = data['Population(in lakhs)'].values.reshape(-1, 1) # Features y = data['Profit (in Lakhs)'].values # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

regression = LinearRegression()

regression.fit(X\_train, y\_train)

y\_pred = regression.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

print("Mean Squared Error:", mse)



**Assignment – 15**

**1. Consider a student object; write a student class with attributes id, name,marks1, marks2 and marks3 and methods display and calculate\_Percentage. Define constructor. Use pickle to dump and load the object student in and from a binary file.**

import pickle

class Student: def \_\_init\_\_(self, student\_id, name, marks1, marks2, marks3):

self.id = student\_id self.name = name self.marks1 = marks1 self.marks2 = marks2 self.marks3 = marks3

def display(self):

print("Student ID:", self.id)

print("Name:", self.name)

print("Marks 1:", self.marks1)

print("Marks 2:", self.marks2)

print("Marks 3:", self.marks3)

def calculate\_percentage(self): total\_marks = self.marks1 + self.marks2 + self.marks3 percentage = (total\_marks / 300) \* 100 return percentage

student = Student(1, "John Doe", 80, 75, 90)

with open("student.bin", "wb") as file:

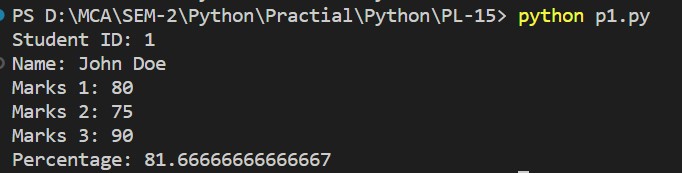
pickle.dump(student, file)

with open("student.bin", "rb") as file:

student = pickle.load(file)

student.display()

percentage = student.calculate\_percentage() print("Percentage:", percentage)



**2. Consider bank object with attributes BankName, BranchCode, BranchName, and Sales. Define Constructor and methods display (Displaying total Sales of a bank with bank name) and TotalSalesofBank to find total sales of the bank. Use pickle to dump and load the object student in and from a binary file.**

**Output should be “Sales of SBI is Rs. 100000”**

import pickle

class Bank:

def \_\_init\_\_(self, bank\_name, branch\_code, branch\_name, sales):

self.bank\_name = bank\_name

self.branch\_code = branch\_code

self.branch\_name = branch\_name

self.sales = sales

def display(self):

print("Sales of", self.bank\_name, "is Rs.", self.sales)

def TotalSalesofBank(self,bank):

total\_sales = sum(bank.sales)

return total\_sales

bank = Bank("SBI", "1234", "Main Branch", 100000)

with open("bank.bin", "wb") as file:

pickle.dump(bank, file)

with open("bank.bin", "rb") as file:

bank = pickle.load(file)

bank.display() 

**18. Write a program in python to implement Salary printing file read operation. (File format: EmployeeNo, name, deptno, basic, DA, HRA, and Conveyance) should perform below operations.**

1. **Print Salary Slip for given Employee Number**

def print\_salary\_slip(employee\_number):

with open('salary\_file.txt', 'r') as file:

found = False

for line in file:

values = line.strip().split(',')

if values[0] == employee\_number:

found = True

name = values[1]

deptno = values[2]

basic = float(values[3])

da = float(values[4])

hra = float(values[5])

conveyance = float(values[6])

gross\_salary = basic + da + hra + conveyance

print("Salary Slip")

print("Employee Number:", employee\_number)

print("Name:", name)

print("Department Number:", deptno)

print("Basic Salary:", basic)

print("DA:", da)

print("HRA:", hra)

print("Conveyance:", conveyance)

print("Gross Salary:", gross\_salary)

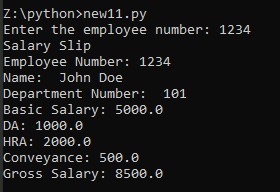
break

if not found:

print("Employee number not found in the file.")

employee\_no = input("Enter the employee number: ")

print\_salary\_slip(employee\_no)



**b) Print Employee List for Given Department Number**

def print\_employee\_list(department\_number):

with open('salary\_file.txt', 'r') as file:

found = False

for line in file:

values = line.strip().split(',')

if values[2] == department\_number:

found = True

employee\_number = values[0]

name = values[1]

print("Employee Number:", employee\_number)

print("Name:", name)

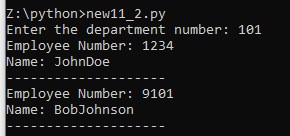
print("--------------------")

if not found:

print("No employees found for the given department number.")

department\_no = input("Enter the department number: ")

print\_employee\_list(department\_no)



**19. Write a program to create a regular expression to search for strings starting with m and having a total of 3 characters.**

import re

def search\_strings(strings):

pattern = r'^m\w{2}$'

matching\_strings = []

for string in strings:

if re.match(pattern, string):

matching\_strings.append(string)

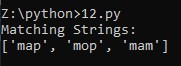
return matching\_strings

input\_strings = ['map', 'mouse', 'car', 'mop', 'dog', 'moon', 'mam', 'mars']

result = search\_strings(input\_strings)

print("Matching Strings:")

print(result)



**20. Write a program to create a regular expression to retrieve phone number, date of birth and email address of a person.**

import re

def extract\_information(text):

phone\_pattern = r'\b\d{3}-\d{3}-\d{4}\b'

dob\_pattern = r'\b\d{2}/\d{2}/\d{4}\b'

email\_pattern = r'\b[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\b'

phone\_numbers = re.findall(phone\_pattern, text)

dobs = re.findall(dob\_pattern, text)

emails = re.findall(email\_pattern, text)

return phone\_numbers, dobs, emails

input\_text = "John Doe's phone number is 123-456-7890. His date of birth is 01/15/1985. You can reach him at john.doe@example.com."

phone\_numbers, dobs, emails = extract\_information(input\_text)

print("Phone Numbers:")

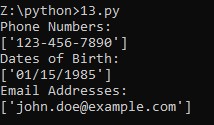
print(phone\_numbers)

print("Dates of Birth:")

print(dobs)

print("Email Addresses:")

print(emails)



**21. Implement a program to demonstrate classification problems.**

from sklearn.datasets import load\_iris from sklearn.model\_selection import train\_test\_split from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

iris = load\_iris()

X = iris.data

y = iris.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

classifier = DecisionTreeClassifier()

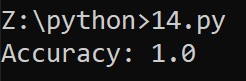
classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)

**Output :**



**22. Implement a program to demonstrate regression problem.**

from sklearn.datasets import load\_boston from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression from sklearn.metrics import mean\_squared\_error, r2\_score

boston = load\_boston()

X = boston.data

y = boston.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print("Mean Squared Error:", mse)

print("R-squared Score:", r2)

**Output :**

|  |  |
| --- | --- |
| Mean Squared Error: 24.29111947497351 | |
| R-squared Score: 0.6687594935356308 |  |

**END**