# Exp.no 1 a) Implementation of data types in python

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

b=float(input("Enter b value:"))

c=a+b

print("sum of int & floart=", c)

# addition of two complex numbers

p=3+2j

q=5+3j

r=p+q

print("Addition of complex numbers=", r)

# binary addition

a=0b101 # decimal 5

b=0b001 # decimal 1

c=a+b # decimal 6

print("addition of binary numbers=", c) # output 6

# hexa decimal addition

m=0x12 # decimal equal 18

n=0x10 # decimal equal 16

p=m+n # output 34

print("Addition of hexa decimal=", p)

#octal addition

x=0o12 #decimal equal 10

y=0o10 #decimal equal 8

z=x+y

print("Addition of octal numbers=",z) # output 18

#Exp1. b) Implementation of python operators

a = 5 # Initialize the value of a

b = 2 # Initialize the value of b

#Arithmatic operators

print('Addition of two numbers:',a+b)

print('Subtraction of two numbers:',a-b)

print('Multiplication of two numbers:',a\*b)

print('Division of two numbers:',a/b)

print('Reminder of two numbers:',a%b)

print('Exponent of two numbers:',a\*\*b)

print('Floor division of two numbers:',a//b)

#Relational operators

print('Two numbers are equal or not:',a==b)

print('Two numbers are not equal or not:',a!=b)

print('a is less than or equal to b:',a<=b)

print('a is greater than or equal to b:',a>=b)

print('a is greater b:',a>b)

print('a is less than b:',a<b)

# logical operators

a=5

print('Is this statement true?:',a > 3 and a < 5)

print('Any one statement is true?:',a > 3 or a < 5)

print('Each statement is true then return False and vice-versa:', (not(a>3 and a<5)))

# EXP2A CREATE LIST, ACCESSING USING INDEX & SLICING METHODS

# creation of empty list

list=[]

print(list)

print(type(list))

# create list of numbers and access elements using indexing method

a=[10,20,30,40,50]

# list accessing using indexing method

#DISPLAY 2ND ELEMENT

print("list second element=", a[1])

print("ALL forward indexing elements")

for i in range(0, len(a)):

print(a[i])

print("display list element direct access")

for x in a:

print(x)

# list accessing using negative indexing

print("Backward access using negative indexing")

for i in range(-1, -len(a)-1,-1):

print(a[i])

# List access using slicing method

#list accessing using slicing method

a=[10,20,30,40,50]

b=a[0:3]

print("New list b elements")

print(b)

print("elements from a ")

print(a[0:4])

print("selecting elements in even position")

print(a[0:5:2])

print("selecting elements in odd position")

print(a[1:5:2])

print("select all elements")

print(a[::])

print("select all elements in reverse order")

print(a[::-1])

print("negative selection")

print(a[-1:-3:-1])

# EXP2B LIST METHODS append, extend, remove, pop, index, count, insert, reverse, sort, max, min,sum

a=[10,20,30]

print("list elements")

print(a)

# append method

a.append(40)

print("list elements after append")

print(a)

# append 5 elements using input method

for i in range(5):

x=int(input("Enter new element to append"))

a.append(x)

print("list elements after append")

print(a)

#extend method

b=[100,110,120]

a.extend(b)

print("list elements after extend")

print(a)

#remove method

a.remove(30)

print("list elements after remove")

print(a)

#pop method

print("Last element delete=", a.pop())

print(a)

# display index of the given element

print("index of 100=", a.index(100))

#insert new element

#insert element 5 into 2 position

a.insert(2,5)

print("list elements after insert")

print(a)

print("reverse order list")

a.sort(reverse=True)

print(a)

print("Ascending order")

a.sort()

print(a)

print("Length =", len(a))

# count method is used to count number of time given element exists

print("Count of element 1 =", a.count(1))

print("maximum value =", max(a))

print("minimum value =", min(a))

print("Sum of all values =", sum(a))

# EXP3 demonstrate working with tuples in Python

# Different types of tuples

# Empty tuple

tup1 = ()

print(tup1)

# Tuple having integers

tup2 = (1, 2, 3,4,5,2)

print(tup2)

# tuple with mixed datatypes

tup3 = (1, "PYTHON", "JAVA", 3.4)

print(tup3)

# nested tuple

tup4 = ("mouse", [8, 4, 6], (1, 2, 3))

print(tup4)

#ACCESSING TUPLES USING INDEXING

print("tuple access using indexing")

for x in tup3:

print(x)

#ACCESSING TUPLES USING slicing

print("tuple access using slicing")

print(tup3[0:2])

print(tup2[::-1])

print(tup4[0:2])

print(tup4[0][0])

print(tup4[1][2])

#tuple methods count, index

print("count of 2 =", tup2.count(2))

print("index of 5=", tup2.index(5))

#membership operator in tuple

if "PYTHON" in tup3:

print("PYTHON COUSE AVAILABLE")

else:

print("PYTHON COUSE not AVAILABLE")

# EXP4 demonstrate working with dictionaries in Python

d1={1:"python", 2:"java", 3:"c++", 4:".net"}

print(type(d1))

print(d1)

# copy method

d2=d1.copy()

print(d2)

#clear method

d2.clear()

print(d2)

#Accessing the dictionary values

print("accessing dict values using index")

print(d1[1])

print(d1[3])

#adding,updating & delete dictionary

d1[5]="DBMS"

print("After adding new key and value")

print(d1)

#update key 4 value .net with "html"

d1[4]="html"

print("After updation the dictionary given below")

print(d1)

# dicrionary with update method

d1.update({4:"Web Technology"})

print(d1)

# using pop() method

pop\_key = d1.pop(2)

print("deleted key value=", pop\_key)

print(d1)

#dictionary Access using keys & items methods

print("Dictionary keys ")

for x in d1:

print(x)

print("Dictionary using keys() method")

for x in d1.keys():

print(x)

print("Dictionary using items() method")

for x in d1.items():

print(x)

print("Dictionary using values() method")

for x in d1.values():

print(x)

# using popitem() method

pop\_key = d1.popitem()

print("deleted key value=", pop\_key)

print(d1)

# dictionary methods

print("using get method")

dict = {1: "Hcl", 2: "TCS", 3: "Facebook", 4: "Amazon", 5: "Flipkart"}

# get() method ->print(dict.get(3))

Exp-5 :based on array numpy

import numpy as np

a=np.array([])

print("Empty array & type", type(a))

a=np.array([10,20,30,40,50])

print("one dimensional Array elements", a)

b=np.array([[1,2,3],[4,5,6],[7,8,9]])

print("two dimensional Array elements")

print(b)

print("1-D Array accessing using slicing ")

print(a[0:4]) # accessing 0th elt to 3rd

print(a[::]) # accessing all elts

print(a[::-1]) # accessing reserve

print("2-D Array accessing using slicing ")

print(b[:, 0:2]) # accessing all rows and 0 and 1 column

print(b[1:3, 0:3]) # accessing 1 & 2 rows and 0,1 & 2 columns

print(b[::-1, ::-1]) # accessing rows and columns in reversed order

print("Integer array indexing")

print(a[2]) # print 2nd element

print("2D array indexing")

temp=b[[1,2,0],[0,1,2]]

print(temp)

print("Boolean Array indexing")

print(b[b>3])

print("A transpose")

print(b.T)

print("sum row wise", b.sum(axis=1))

print("sum col wise", b.sum(axis=0))

print("sorting elements")

print(np.sort(b,axis=None))

c=np.array([[1,1,3],[4,5,6],[7,5,5]])

print("array addition of a+b")

d=b+c

print(d)

print("array subtraction of a-b")

d=b-c

print(d)

print("array multiplication of a\*b")

d=b\*c

print(d)

#LAB EXPNO: 6, Implement mean, median, mode , standard deviation and variance

from scipy import stats as st

import numpy as np

a=np.array([1,2,3,4,5,6,5,7])

m=np.mean(a)

print("Mean=", m)

md=np.median(a)

print("Median=", md)

mde=st.mode(a)

print("Mode =", mde)

sd=np.std(a)

print("Standard deviation=", sd)

var1=np.var(a)

print("Variance =", var1)

#LAB EXPNO. 7 COPY ONE FILE INTO ANOTHER FILE

f1 = open('list1.txt‘, “r”)

f2 = open('output.txt', ‘w')

for line in f.readlines():

f2.write(line)

f1.close()

f2.close()