# Table of Contents

[**Table of Contents**](#_s9rsng9a64vl) **1**

[**Experiment 3: Lists**](#_6auzy7g39pin) **2**

[**Experiment 4: Tuple**](#_oo9dkq3qshv6) **4**

[**Experiment 5:Sets**](#_9kc7965umqll) **5**

[**Experiment 6: Dictionaries**](#_itg83pmfrrln) **6**

[**Experiment 7: Classes and Objects**](#_ptgz45ek1fny) **7**

[**Experiment 8: Inheritance**](#_o5dtalxozqdp) **8**

[**Experiment 9: Package and Module**](#_h8fc4cqra18i) **10**

[**Experiment 10: Exception Handling**](#_qnd4napvtu1n) **11**

[**Experiment 11: File Handling**](#_dqy7zbeqz176) **12**

[**Experiment 12: Tkinter and Sqlite3**](#_jk73v7ldw5wy) **13**

# **Experiment 3:** Lists

lst1 = [int(x) for x in input("Enter list 1: ").split()]

lst2 = [int(x) for x in input("Enter list 1: ").split()]

print("Orignal List 1: ",lst1, "Orignal List 2: ",lst2)

lst1.sort()

lst2.sort()

print("Sorted list 1: ", lst1,"\nSorted list 2:", lst2)

l1 = len(lst1)

l2 = len(lst2)

lst3=list()

i, j = 0, 0

while i<l1 and j<l2:

if lst1[i] <=lst2[j]:

lst3.append(lst1[i])

i+=1

else:

lst3.append(lst2[j])

j+=1

while i<l1:

lst3.append(lst1[i])

i+=1

while j<l2:

lst3.append(lst2[j])

j+=1

print("Final List is:", lst3)

print("Updating the first value: as 23")

lst3[0]= 23

print("New list:", lst3)

print("deleting the middle element")

mid = len(lst3)//2

lst3.remove(lst3[mid])

print("list after deleting" , mid , " element is :", lst3)

print("Min is ", min(lst3), " max is ", max(lst3))

nlist = [x for x in input("Enter three strings: ").split()]

for n in nlist:

lst3.append(n)

print("list is : ", lst3)

try:

if lst3.index("python"):

print("python present")

except ValueError:

print("Python absent")

'''

if lst3.index("python") is ValueError:

print("python present")

else:

print("Python absent")'''

# Experiment 4: Tuple

emp = ((10, "vijay", 9000.90), (20, "bijoy", 5500.50), (30, "vanaja", 9800.00), (40, "kapoor", 5000.00))

for e in emp:

if "bijoy" in e:

print(e[1],e[2])

print(sorted(emp, key = lambda x: x[1])) # sort on name

emp = ((10, "vijay", 9000.90), (20,"bijoy", 5500.50), (30, "vanaja", 9800.00), (40, "kapoor", 5000.00))

tpl1=(90,"kadh", 9247)

i=0

list3=list()

list1 = emp[:2]

list2 = emp[2:]

for e in list1:

list3.append(e)

list3.append(tpl1)

for e in list2:

list3.append(e)

list3 = tuple(list3)

print(list3)

# Experiment **5:Sets**

set1 = {x for x in input("enter string 1")}

set2 = {x for x in input("enter string 2")}

print(set1,"::", set2)

print(set1.intersection(set2))

print(set1.difference(set2))

print(set1.union(set2))

print(set1.symmetric\_difference(set2))

# Experiment 6: Dictionaries

countries = ["USA", "India", "Germany", "France"]

cities = ["washington", "New Delhi", "Berlin", "Paris"]

z= zip(countries, cities) # Convert to zip object

d=dict(z) # convert to dictionary

print(d)

d.update({"USA":"NYC"})

print(d)

d.pop("USA")

print(d)

keys = d.keys()

if "India" in keys:

print(d["India"])

# Experiment 7: Classes and Objects

class emp():

def \_\_init\_\_(self, id=0, name=''):

self.id = id

self.name= name

def display(self):

print(self.id)

print(self.name)

def modify(self, id, name):

self.id = id

self.name= name

e = emp(10, 'ninad')

e.display()

e.modify(name='nin', id = 90)

e.display()

l1 = list()

l1.append(e)

for i in l1:

if i.id == 90:

print(i.name)

# Experiment 8: Inheritance

# defining class A

class A:

def \_\_init\_\_(self, a):

self.var\_a= a

def pr(self):

print(self.var\_a, 'I am in A Class')

def mod(self, a):

self.var\_a= a

# B class inheriting A

class B:

def \_\_init\_\_(self, b):

self.var\_b= b

def pr(self):

print(self.var\_b, 'I am in B class')

def mod(self, b):

self.var\_b= b

# C class inheriting B

class C(A, B):

def \_\_init\_\_(self, a, b, c):

self.var\_c= c

A.\_\_init\_\_(self, a)

B.\_\_init\_\_(self, b)

def pr(self):

A.pr(self)

B.pr(self)

print(self.var\_c, 'I am in C class')

def mod(self,a, b, c):

self.var\_c= c

A.mod(self,a)

B.mod(self,b)

x=C("Hi", "Hello", "Hola")

x.pr()

x.mod("ih", "aha","nah")

x.pr()



# Experiment 9: Package and Module

# demo.py

from x.demo2 import addition

print(addition(1,2))

print(addition(1,3,2))

# x ⇒ demo2.py

def addition(a=None, b=None, c=None):

if (a!=None and b!=None and c!=None):

return a+b+c

elif (a!=None and b!=None):

return a+b

else:

return a

# Experiment 10: Exception Handling

import random

class ValueTooLargeError(Exception):

def \_\_init\_\_(self, msg):

self.msg = msg

class ValueTooSmallError(Exception):

def \_\_init\_\_(self, msg):

self.msg = msg

i\_num = random.randint(1,100)

count = 0

while True:

count += 1

try:

u\_num= int(input("Enter a number between 1 and 100"))

if u\_num >i\_num:

raise ValueTooLargeError("Enter small value")

elif u\_num< i\_num:

raise ValueTooSmallError("Enter Larger value")

else:

break

except ValueTooLargeError as me:

print(me)

except ValueTooSmallError as me:

print(me)

print("Guessed correctly in {} attempts".format(count))

# Experiment 11: File Handling

import os

f= open("demotxt.txt", "w")

ch = 1

while ch == 1:

stmt = input("Enter statement")

f.write(stmt+"\n")

ch = int(input("enter 1 to continue"))

f.close()

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

print(f.read())

f.seek(0,0)

nw=0

nc=0

for l in f:

nw=len(l.split())

nc=len(l)

print(nw,"::",nc)

f.close()

for dirpath, dirnames, filenames in os.walk("."):

print(dirnames)

# Experiment 12: Create a login application using Tkinter and SQLite

'''

Create a login application using Tkinter and SQLite

'''

from tkinter import \*

import sqlite3

with sqlite3.connect("details.db") as db:

cursor = db.cursor()

cursor.execute(""" CREATE TABLE IF NOT EXISTS users(username text PRIMARY KEY, password text NOT NULL); """)

def add\_new\_user():

newuser=username.get()

newpass= password.get()

cursor.execute("SELECT COUNT(\*) FROM users WHERE username='"+newuser+"'")

result = cursor.fetchone()

if int(result[0])>0:

error["text"] = "Error: User already exists"

else:

error["text"] = "Added new user"

cursor.execute("INSERT INTO users(username, password) VALUES(?,?)",(newuser, newpass))

db.commit()

window =Tk()

window.geometry("450x180")

error = Message(text="", width=160)

error.place(x=30, y=10)

label1 = Label(text = "Enter Username:")

label1.place(x = 30, y = 40)

username = Entry(text = "")

username.place(x=150, y=40, width=200, height= 25)

label1 = Label(text = "Enter Password:")

label1.place(x = 30, y = 80)

password = Entry(text = "", show="\*")

password.place(x=150, y=80, width=200, height= 25)

button = Button(text="Add", command= add\_new\_user)

button.place(x=150, y=120, width=75, height= 35)

window.mainloop()