**NAME: ALLEN FAMY**

**STD: SE COMPS(B)**

**SUBJECT: OSTL**

**ROLL NO: 02**

**ASSIGNMENT NO : 03**

**ASSIGNMENT NO: 03**

1.a) A Python program to create a text file to store individual characters.

Ans)

# write data in a file.

file1 = open("myfile.txt","w")

L = ["This is Delhi \n","This is Paris \n","This is London \n"]

# \n is placed to indicate EOL (End of Line)

file1.write("Hello \n")

file1.writelines(L)

file1.close() #to change file access modes

1.b) A Python program to read all the strings from the text file and display them.

Ans)

# Program to show various ways to read and

# write data in a file.

file1 = open("myfile.txt","w")

L = ["This is Delhi \n","This is Paris \n","This is London \n"]

# \n is placed to indicate EOL (End of Line)

file1.write("Hello \n")

file1.writelines(L)

file1.close() #to change file access modes

file1 = open("myfile.txt","r+")

print "Output of Read function is "

print file1.read()

print

# seek(n) takes the file handle to the nth

# bite from the beginning.

file1.seek(0)

print "Output of Readline function is "

print file1.readline()

print

file1.seek(0)

# To show difference between read and readline

print "Output of Read(9) function is "

print file1.read(9)

print

file1.seek(0)

print "Output of Readline(9) function is "

print file1.readline(9)

file1.seek(0)

# readlines function

print "Output of Readlines function is "

print file1.readlines()

print

file1.close()

1.c) A Python program to know whether a file exists or not.

Ans)

import os.path

if os.path.isfile('filename.txt'):

print ("File exist")

else:

print ("File not exist")

2.a) A Python program to copy an image file into another file.

Ans)

with open(“hello.jpeg”) as f :

with open (“copy.txt”, “w”) as f1:

for line in f:

f1.write(line)

2.b) A Python program to use ‘with’ to open a file and write some strings into the file.

Ans)

# Sample code(2) Write using with statement

with open("sample\_log.txt", "w") as fh:

fh.write("I love Python even more!!")

2.c) A Python program to create phone book with names and phone numbers.

Ans)

def print\_menu():

print('1. Print Phone Numbers')

print('2. Add a Phone Number')

print('3. Remove a Phone Number')

print('4. Lookup a Phone Number')

print('5. Quit')

print()

numbers = {}

menu\_choice = 0

print\_menu()

while menu\_choice != 5:

menu\_choice = int(input("Type in a number (1-5): "))

if menu\_choice == 1:

print("Telephone Numbers:")

for x in numbers.keys():

print("Name: ", x, "\tNumber:", numbers[x])

print()

elif menu\_choice == 2:

print("Add Name and Number")

name = input("Name: ")

phone = input("Number: ")

numbers[name] = phone

elif menu\_choice == 3:

print("Remove Name and Number")

name = input("Name: ")

if name in numbers:

del numbers[name]

else:

print(name, "was not found")

elif menu\_choice == 4:

print("Lookup Number")

name = input("Name: ")

if name in numbers:

print("The number is", numbers[name])

else:

print(name, "was not found")

elif menu\_choice != 5:

print\_menu()

3.a) A Python program to rename a directory.

Ans)

# Pythono3 code to rename multiple

# files in a directory or folder

# importing os module

import os

# Function to rename multiple files

def main():

    for count, filename in enumerate(os.listdir("xyz")):

        dst ="Hostel" + str(count) + ".jpg"

        src ='xyz'+ filename

        dst ='xyz'+ dst

        # rename() function will

        # rename all the files

        os.rename(src, dst)

# Driver Code

if \_\_name\_\_ == '\_\_main\_\_':

    # Calling main() function

    main()

3.b) A Python program to display all contents of the current directory.

Ans)

from os import listdir

from os.path import isfile, join

files\_list = [f for f in listdir('/home/students') if isfile(join('/home/students', f))]

print(files\_list);

4.a) A Python program to create a regular expression to search for strings starting with m and having total 3 characters using the search() method.

import re

str="mat man women"

res=re.search(r'm\w\w',str)

if res:

print(res.group())

OUTPUT

mat

4.b) A Python program to create a regular expression to search for strings starting with m and having total 3 characters using the findall() method.

Ans)

import re

str='mat sun man'

res=re.findall(r'm\w\w',str)

print(res)

OUTPUT

['mat', 'man']

5) a) A Python program to create a regular expression to retrieve the phone number of a person.

Ans)

import re

str='Raju :9706612345'

res=re.search(r'\d+',str)

if res:

print(res.group())

OUTPUT

9706612345

5) b) A Python program to create a regular expression to extract only name but not number from a string.

Ans)

import re

str='Raju :9706612345'

res=re.search(r'\D+',str)

if res:

print(res.group())

OUTPUT

Raju :

|  |
| --- |
| 6) A Python Program to Get IP Address  import socket  hostname = socket.gethostname()  IPAddr = socket.gethostbyname(hostname)  print("Your Computer Name is:" + hostname)  print("Your Computer IP Address is:" + IPAddr) |

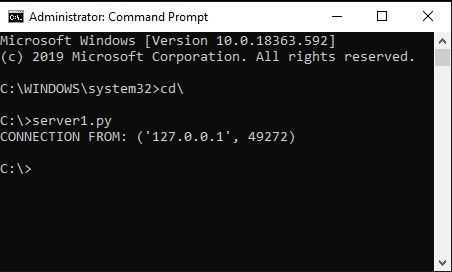
Output:

Your Computer Name is:pppContainer

Your Computer IP Address is:10.98.162.168

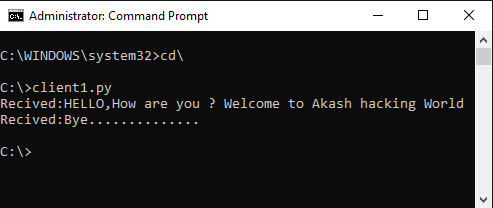
6) B Python program to create a TCP/IP client program that receives messages from the server.

|  |
| --- |
| import socket  # take the server name and port name  host = 'local host'  port = 5000  # create a socket at server side  # using TCP / IP protocol  s = socket.socket(socket.AF\_INET,  socket.SOCK\_STREAM)  # bind the socket with server  # and port number  s.bind(('', port))  # allow maximum 1 connection to  # the socket  s.listen(1)  # wait till a client accept  # connection  c, addr = s.accept()  # display client address  print("CONNECTION FROM:", str(addr))  # send message to the client after  # encoding into binary string  c.send(b"HELLO, How are you ? \  Welcome to Akash hacking World")  msg = "Bye.............."  c.send(msg.encode())  # disconnect the server  c.close() |



|  |
| --- |
| 6) C Python program to create a TCP/IP server program that sends message to a client.  **CODE:**  import socket  # take the server name and port name  host = 'local host'  port = 5000  # create a socket at client side  # using TCP / IP protocol  s = socket.socket(socket.AF\_INET,  socket.SOCK\_STREAM)  # connect it to server and port  # number on local computer.  s.connect(('127.0.0.1', port))  # receive message string from  # server, at a time 1024 B  msg = s.recv(1024)  # repeat as long as message  # string are not empty  while msg:  print('Recived:' + msg.decode())  msg = s.recv(1024)  # disconnect the client  s.close() |

**Output of Server:**

**Output of Client:**  


7) A Python program to create a UDP server that sends messages to the client.

**CODE:**

import socket

localIP = "127.0.0.1"

localPort = 20001

bufferSize = 1024

msgFromServer = "Hello UDP Client"

bytesToSend = str.encode(msgFromServer)

# Create a datagram socket

UDPServerSocket = socket.socket(family=socket.AF\_INET, type=socket.SOCK\_DGRAM

# Bind to address and ip

UDPServerSocket.bind((localIP, localPort))

print("UDP server up and listening")

# Listen for incoming datagrams

while(True):

bytesAddressPair = UDPServerSocket.recvfrom(bufferSize)

message = bytesAddressPair[0]

address = bytesAddressPair[1]

clientMsg = "Message from Client:{}".format(message)

clientIP = "Client IP Address:{}".format(address)

print(clientMsg)

print(clientIP)

# Sending a reply to client

UDPServerSocket.sendto(bytesToSend, address)

**Output:**

UDP server up and listening

Message from Client:b"Hello UDP Server"

Client IP Address:("127.0.0.1", 51696)

7) B A Python program to create a UDP client that receives messages from the server.

Example: UDP Client using Python

**CODE:**

import socket

msgFromClient = "Hello UDP Server"

bytesToSend = str.encode(msgFromClient)

serverAddressPort = ("127.0.0.1", 20001)

bufferSize = 1024

# Create a UDP socket at client side

UDPClientSocket = socket.socket(family=socket.AF\_INET, type=socket.SOCK\_DGRAM

# Send to server using created UDP socket

UDPClientSocket.sendto(bytesToSend, serverAddressPort)

msgFromServer = UDPClientSocket.recvfrom(bufferSize)

msg = "Message from Server {}".format(msgFromServer[0])

print(msg)

**Output:**

Message from Server b"Hello UDP Client"

8) A python program to perform operations on a queue given below:

a) Add a new element

b) Delete an element

c) Search the position of a given element

**CODE:**

class Queue:

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

self.qu=[]

def isempty(self):

return self.qu==[]

def add(self,ele):

self.qu.append(ele)

def delete(self):

if self.isempty():

return -1

else:

return self.qu.pop(0)

def search(self,ele):

if self.isempty():

return -1

else:

try:

n=self.qu.index(ele)

return n+1

except ValueError:

return -2

def display(self):

return self.qu

q=Queue()

choice=0

while choice<4:

print('Queue Operations')

print('1 Add element')

print('2 Delete element')

print('3 Search for element')

print('4. Exit')

choice=int(input('your choice: '))

if choice==1:

ele=float(input('Enter element: '))

q.add(ele)

elif choice==2:

ele=q.delete()

if ele==-1:

print("The queue is empty")

else:

print("Remove element",ele)

elif choice==3:

ele=float(input('Enter element: '))

pos=q.search(ele)

if pos==-1:

print("The queue is empty")

elif pos==-2:

print("Element not found in the queue")

else:

print("Element found at position: ",pos)

else:

break

print('queue= ',q.display())

**Output:**

Queue Operations

1 Add element

2 Delete element

3 Search for element

4. Exit

your choice: 1

Enter element: 20

queue= [20.0]

Queue Operations

1 Add element

2 Delete element

3 Search for element

4. Exit

your choice: 1

Enter element: 30

queue= [20.0, 30.0]

Queue Operations

1 Add element

2 Delete element

3 Search for element

4. Exit

your choice: 1

Enter element: 40

queue= [20.0, 30.0, 40.0]

Queue Operations

1 Add element

2 Delete element

3 Search for element

4. Exit

your choice: 2

Remove element 20.0

queue= [30.0, 40.0]

Queue Operations

1 Add element

2 Delete element

3 Search for element

4. Exit

your choice: 3

Enter element: 30

Element found at position: 1

queue= [30.0, 40.0]

Queue Operations

1 Add element

2 Delete element

3 Search for element

4. Exit

your choice: 4

9) A python program to create and use dequeue and perform the following operation:

a) Add an element at the front

b) Delete element at the front

c) Add element at the rear

d) Delete an element at the rear

e) Delete an element at any place

f) Search for an element in the dequeue

g) Reverse the dequeue

**CODE:**

#import deque

from collections import deque

#create an empty deque

d=deque()

# Adding element at the front

element=input("Enter element: ")

d.appendleft(element)

# Deleting element at the front

if len(d)==0:

print("Deque is empty")

else:

d.popleft()

# Adding element at the rear

element=input("Enter element: ")

d.appendleft(element)

# Deleting element at the rear

if len(d)==0:

print("Deque is empty")

else:

d.pop()

# Deleting element at any place

element=input("Enter element: ")

try:

d.remove(element)

except ValueError:

print("Element not found")

# Searching for an element in the deque

element=input("Enter element: ")

c=d.count(element)

print("No. of times the element found: ",c)

**OUTPUT:**

Enter element: 10 20 30 40 50

Enter element: 10

Enter element: 60

Element not found

Enter element: 20

No. of times the element found: 0

10) A python to create three push buttons and change the background of the frame according to the button clicked by the user.

**CODE:**

from tkinter import \*

def action1():

frame=Frame(top, width=500, height=500, background="red").place(x=00,y=100)

def action2():

frame=Frame(top, width=500, height=500, background="blue").place(x=00,y=100)

def action3():

frame=Frame(top, width=500, height=500, background="yellow").place(x=00,y=100)

top = Tk()

top.geometry("550x500")

b1 = Button(top,height=6,width=20,text = "RED",bg='red',command=lambda:action1()).place(x=0, y=0)

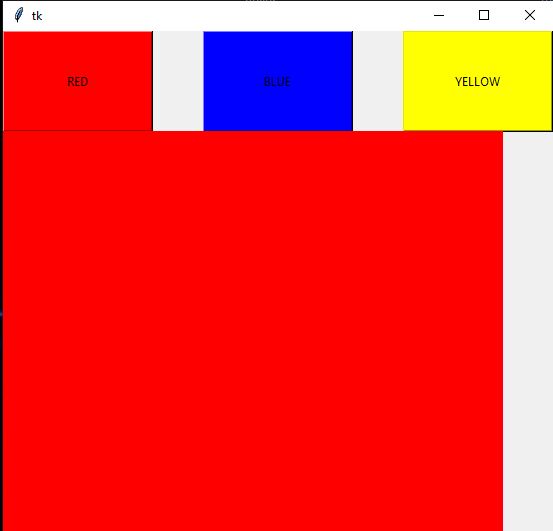
b2 = Button(top,height=6,width=20,text = "BLUE",bg='blue',command=lambda:action2()).place(x=200, y=0)

b3 = Button(top,height=6,width=20,text = "YELLOW",bg='yellow',command=lambda:action3()).place(x=400, y=0)

#b.pack()

top.mainloop()

**OUTPUT:**



11 A Python program using GUI to retrieve a row from a MySQL database table.

**CODE:**

import mysql.connector

from mysql.connector import Error

try:

connection = mysql.connector.connect(host='localhost',

database='Electronics',

user='pynative',

password='pynative@#29')

sql\_select\_Query = "select \* from Laptop"

cursor = connection.cursor()

cursor.execute(sql\_select\_Query)

records = cursor.fetchall()

print("Total number of rows in Laptop is: ", cursor.rowcount)

print("\nPrinting each laptop record")

for row in records:

print("Id = ", row[0], )

print("Name = ", row[1])

print("Price = ", row[2])

print("Purchase date = ", row[3], "\n")

except Error as e:

print("Error reading data from MySQL table", e)

finally:

if (connection.is\_connected()):

connection.close()

cursor.close()

print("MySQL connection is closed")

**OUTPUT:**

Total number of rows in Laptop is: 7

Printing each laptop record

Id = 1

Name = Lenovo ThinkPad P71

Price = 6459.0

Purchase date = 2019-08-14

Id = 2

Name = Area 51M

Price = 6999.0

Purchase date = 2019-04-14

Id = 3

Name = MacBook Pro

Price = 2499.0

Purchase date = 2019-06-20

Id = 4

Name = HP Pavilion Power

Price = 1999.0

Purchase date = 2019-01-11

Id = 5

Name = MSI WS75 9TL-496

Price = 5799.0

Purchase date = 2019-02-27

Id = 6

Name = Microsoft Surface

Price = 2330.0

Purchase date = 2019-07-23

Id = 7

Name = Acer Predator Triton

Price = 2435.0

Purchase date = 2019-08-15

MySQL connection is closed