

PRACTICAL No. 1

019

Aim: Demonstrate the use of the different file accessing modes, different attributes of the file object and differentiate among the different read methods.

ALGORITHM

1. Create a file object by using open method & use the write access mode followed by writing some contents onto the file & then closing the file.
2. Now open the file in the read mode & use the read method or read lines method & store the output in the variable & finally display the content of the variable.

3. Now use the name of the file, mode in which the file is being open, whether the file is still open or closed & finally the output of the softspace attribute.
4. Now open the fileobj in write mode write some random content close subsequently then again open the fileobj in 'wt' mode that is the update mode & write contents.
5. Open fileobj in read mode, display the update written contents & close open again in 'rt' mode with parameters passed and display the output subsequently.
6. Now open fileobj in append mode open with method write content close the fileobj again open the fileobj in read mode & display the appended output.

program:

```
file obj = open ("Jai.txt", "w")
file obj . write (" My Name is Jai \n 020
                    Jai is my name \n Hello ")
file obj . close ()

file obj = open ("Jai.txt", "r")
str1 = file obj . read ()
print (str1)
file obj . close ()

a = file obj . name
b = file obj . closed
c = file obj . mode
d = file obj . softspace
print (a,b,c,d)
```

OUTPUT:

[My Name is Jai \n
Jai is my name \n
["Jai", 0, "r", 0]]

```
file obj = open ("Jai.txt", "r+")
str1 = file obj . read (10)
print (str1)
pos = file . tell ()
print (pos)
str1 = file . seek (0, 1)
print (str1)
```

```
with open ("Jai.txt", "r") as g:
```

```
    f5(g=1)
```

```
    c=g.read(5)
```

```
    while len(c)>0:
```

```
        print (c, end="*")
```

```
        c=g.read(5)
```

```
file = open ("Jai.txt", "w")
```

```
fileobj.write ("Hello World")
```

```
str1=f1.readlines()
```

```
for line in str1:
```

```
    print (len(line))
```

```
a=input ("Enter the word:")
```

```
f1=open ("Jai.txt", "r")
```

```
str1=f1.readlines()
```

```
c=0
```

```
for line in str1:
```

```
    b=f1.read(5)
```

```
    if b==a:
```

```
        c=c+1
```

```
print (c).
```

✓ OUTPUT

0

48

2

7. Open the fileobj in read mode, declare a variable & perform fileobj.seek tell method & store the output consequently in variable.
8. Use the seek Method with arguments with opening the fileobj in read mode & closing subsequently.
9. Open fileobj with read mode also use the readlines method & store the output consequently in and print the same for counting the length use the 'for' conditional statement & disp the length.

Jan 2012

ISO

PRACTICAL NO. 2

Aim: write a program using iterables method for displaying the set of odd numbers, set of first 10 nos, factorial of first 10 nos.

(A) ALGORITHM :

1. Define a class within that Define an item method which will initialize the first element within the container object.
2. Now use the next method & Define the logic for displaying the odd no values.

Program:

```

class odd:
    def __iter__(self):
        self.num = 1
        return self
    def next(self):
        num = self.num
        self.num += 2
        return num
myobj = odd()
myiter = iter(myobj)
for i in myiter:
    print(next(myiter))

```

OUTPUT:

1
3
5
7.

SSO

Program:

```
class myclass:  
    def __iter__(self):  
        self.a = 1  
        return self  
    def __next__(self):  
        if self.a <= 5:  
            x = self.a  
            self.a += 1  
            return x  
        else:  
            return None  
myobj = myclass()  
myitem = iter(myobj)  
for x in myitem:  
    print(x)
```

OUTPUT:

1
2
3
4
5

(B) ALGORITHM

1. Define a item method with an argument & initialize it to the first value.
2. For extracting the next element from the container use the next method with an argument & compare the number of elements required in the container by using the conditional statement.
3. Now create an object from the given class & pass this object as a argument to the item method.
4. Now using conditional statements display all the values from the given container.

ESO

(C) ALGORITHM:

1. Def function `square` with the arguments & correspondingly return the value.
2. Def function `sacube` with the arguments & correspondingly return the value.
3. Take a variable & list the square & cube in it
4. Use 'for' conditional statement & also make use of the map method
Take the input value from the user & print the output onto the screen

Program:

```
def square(x):  
    return (x * x)  
  
def cube(x):  
    return (x * x * x)  
  
func1 = [square, cube]  
  
for x in range(5):  
    valueout = list(map(lambda n: n(n), func1))  
  
    print(valueout)  
  
x = int(input("Enter:"))
```

024

OUTPUT:

Enter: 5
[16, 64]

↓
for all ()

Program:

```
mytuple = ("Banana", "Oranges", "Apples")
myitem = item(mytuple)
print(next(myitem))
```

Output:

Banana

Oranges

Apple

Program:

```
listnum[10,4,9]
```

```
cistnum= cist (map(lambda x: x%5, listnum))
```

```
print(cistnum)
```

```
def even(x):
```

```
    if x%2==0 :
```

```
        return "even"
```

```
else:
```

```
    return "odd"
```

```
cist(map(even, listnum))
```

Output:

[even, even, odd]

(D) ALGORITHM:

Create a tuple object & assign it the value or element, then create a iterable object to Iter through the tuple with item method, through the tuple now user can use the for loop statement to display all the elements in the iterable objects.

(E) ALGORITHM:

1. Define a variable list & take the value by using tuple.
2. make use of lambda method & print the output using the list method.
3. Define a function even and use if conditional statement & return the output accordingly.
4. Use the list method to print the output onto the main

Jan
16/12/19.

CSII

PRACTICAL No. 3

Exception

(A) ALGORITHM

1. Use the try block to define the normal block of action.
2. Example. Define a file object & open the file in the write mode and write some content onto the file.
3. Use an except block with FD Error as an Environmental Error and convey the appropriate message to the user, else display the message that the operation is carried out successfully.

(2) Program!

```
try:  
    f = open ("Jai.txt", "w")  
    f.write ("My Name Is Jai")  
except IOError:  
    print ("Environment Error")  
else:  
    print ("Successful")
```

026

Output:

Environmental Error

(3) Program!

```
try:  
    a = int (input ("Enter:"))  
except ValueError:  
    print ("Value Arithmetic Error")  
else:  
    print ("Successful")
```

Output:

Enter: 14

Arithmetic Error

Program:

```
list1 = []
list2 = [2, 4, 6, 8]

def a():
    try:
        if len(list1) == 0:
            print("Environmental Error")
    except ValueError:
        print("Environmental Error")

def b():
    try:
        if len(list2) == 0:
            print("Environmental Error")
    except ValueError:
        print(len(list2))

    print(a())
    print(b())
```

Output:

Environmental Error
4

PRACTICAL No- 4

Regular Expression

ALGORITHM:

1. Import re module declare the pattern and declare sequence use match method with declare statements if arguments matched than print the same otherwise print "NOT found!".
2. Import re module declare pattern with literals and meta character \ declare the string value. Use the.findall() with arguments and print the same.

match()

import re

Pattern = r" PYBS"

Sequence = " PYBS represents Computer Science"

If re.match(Pattern, Sequence):

Print ("Match Found")

else:

Print ("Not Found")

Output:

Match Found.

numerical values (Segregation)

import re

Pattern = r"\d+"

Sequence = " Ten 14 , 30 dayu , 129 don "

Output = re.findall(Pattern, Sequence)

Print (Output)

Output:

['14' , '30' , '129']

QSO

split()

```
import re
```

```
Pattern = r"\D+"
```

```
String = "JaiIA, Baayu, 129don"
```

```
Output = re.split(Pattern, String)
```

```
Print Output
```

Output:

```
['Jai', 'aayu', 'don']
```

no-space

```
import re
```

```
Pattern = r"\s+"
```

```
Sequence = "abc def glu"
```

```
replace = ''
```

```
OP = re.sub(Pattern, replace, Sequence)
```

```
Print OP
```

Output:

```
abcdefglu
```

8. Import re module declare pattern with meta character aster use the split() and print the output.
4. import re module declare string and accordingly declare pattern replace the blank space with nb-space , use sub() with 3 arguments & print the string without spaces.

✓ Jyoti

B. 5. import re module declare a sequence use search() for finding subsequently use the group() with dot operator as search() gives memory location using group() it will show up the matched string.

B. import re module declare list with numbers. Use the conditional statements here we have to used up the for conditional statement. Use if condition for checking first number is either 8 or 9 and next number are in orange of 0 to 9 and check whether the entered numbers are equal to 10. if matches print no. is correct otherwise print Invalid Number.

```
# group()
```

```
import re  
sequence = " Python is an interesting lang."  
030  
Pattern j = re.search("IA Python", sequence)  
Print(j)  
OP = j.group()  
Print(OP)
```

Output:

<_sre.SRE_Match object at
0x0281DF00
Python

```
# verifying the given set of phone no.
```

```
import re  
list1 = ['9819492967', '7021181116']  
for values in list1:  
    if re.match(r'[8-9]\d{9} [0-9]\d{9}';  
               len(value) == 10):  
        Print("No. is correct")  
    else:  
        Print("Invalid Number")
```

Output:

No. is correct
No. is correct.

Vowels

```
import re  
str1 = "My Name is Jui"  
Pattern = r"\b[aeiouAEIOU]\w+"  
OP = re.findall(Pattern, str1)  
print(OP)
```

Output
['is']

most & domain

```
import re  
seq = "jai dasuji@gmail.com"  
Pattern = r"[\w\.-]+\.[\w\.-]+"  
OP = re.findall(Pattern, seq)  
print(OP)
```

Output:
['jai dasuji', 'gmail.com']

7. import re module declare a host string and domain name declare pattern for separating the host and domain name. Use the.findall() and then print the output respectively.
8. import re module declare a string use the module with.findall() the vowels in the string and declare the same.

Jai

a. Import re module enter a string use the pattern to display only two elements of the particular string use.findall() declare two variables with initial value as zero use for condition and subsequently use the if condition week count whether condition satisfy add up the or else increment value. And displaying the value subsequently.

counting of first 2 letters! 032

```
import re
s = "mma , ms,b, ms,c, mmt"
pattern = r' [ms\mr]+ '
op = re.findall ( pattern, s)
print (op)
m=0
t=0
for v in op:
    if (v == 'ms'):
        t=t+1
    else:
        m=m+1
print ("no. of Males : ", m)
print ("no. of Females : ", t)
```

Output:

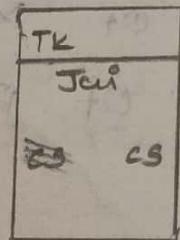
['ms', 'ms' , 'ms' , 'ms']
('no. of Males : 2')
('no. of Females : 2')

Jas

GUI

```
from tkinter import *
root = Tk()
l1 = Label (root, text= "Jui", bg = "red",
            fg = "black")
l1.pack (pady = 20)
l1 = Label (root, text= "CS", bg = "grey",
            fg = "black", font = "10")
l1.pack (side = LEFT, padx = 20)
l2 = Label (root, text= "CS", bg = "blue",
            fg = "black", font = "10")
l2.pack (side = LEFT, pady = 20)
```

```
root.mainloop()
```

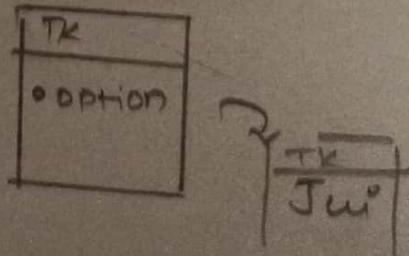


Radio Button

```
from tkinter import *
root = Tk()
def sel():
    print ("Jui")
r1 = Radiobutton (root, text = "OPTION", value=1,
                   command = sel)
```

```
r1.pack()
```

```
root.mainloop()
```



PRACTICAL NO-5

GUI

ALGORITHM:

1. Use the Tkinter library for importing the features of the text widget.
2. Create a variable from text method and position it on to the parent window.
3. Use the pack method along with the object created from the text method & use the parameters.
4. Use the mainloop method for triggering of the corresponding events.
5. Now repeat the above steps with the label method since takes the arguments.
6. Repeat the steps using Radio Button.
7. Repeat the steps

listBox

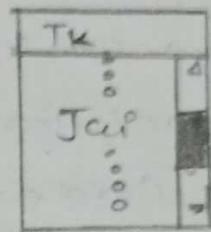
.. Above mentioned all the steps repeat for the listBox using listBox Method.

scrollBar

1. use the tkinter library for importing the features of the text widget.
2. write a para into the new created object.
3. use the Scrollbar method with text & config attribute & also use the pack method.
use text & insert to write the text into scrollbar & make use of it.
4. Finally make use of the mainloop method.

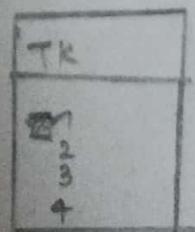
Small Box

```
from tkinter import *
root = Tk()
para = "—"
s = Scrollbar(root)
t = Text(root, height=10, width=10)
s.pack(side=RIGHT, fill="y")
t.pack(side=LEFT, fill="y")
s.config(command=t.yview)
t.config(yscrollcommand=s.set)
s.insert(END, para)
root.mainloop()
```



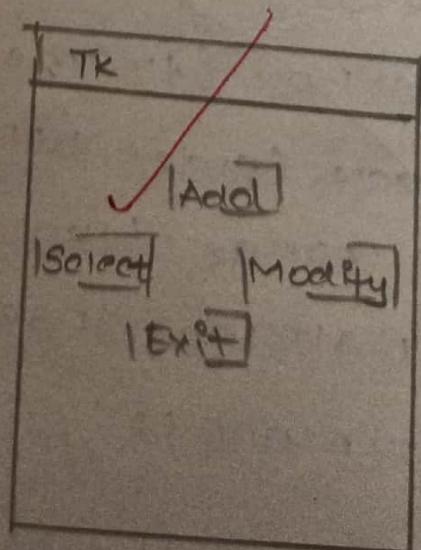
List Box

```
from tkinter import *
window = Tk()
window.geometry("680x500")
label(window, text="numbers:").pack()
frame = Frame(window)
frame.pack()
listnodes = Listbox(frame, font=("Times of Roman", 10))
listnodes.pack(side=LEFT, fill="y")
for x in range(100):
    listnodes.insert(END, str(x))
window.mainloop()
```



Frame & Button

```
from tkinter import *
window = Tk()
window.geometry ("680x500")
frame = Frame (window)
frame.pack()
leftframe = Frame (window)
frame.pack(side=LEFT)
rightframe = Frame (window)
frame.pack(side=RIGHT)
b1 = Button (frame, text="Select")
b1.pack (side=LEFT, padx=50, pady=10)
b2 = Button (frame, text="Modify")
b2.pack (side=RIGHT, padx=60, pady=0)
b3 = Button (frame, text="Add")
b3.pack (side=BOTTOM, pady=70)
b4 = Button (frame, text="Exit")
b4.pack (side=TOP, pady=10)
window.mainloop()
```



ALGORITHM

1. Import the relevant method from tkinter library.
2. Define the object corresponding to the Parent window & define the size of the Parent window in terms of pixels.
3. Use the frame method to create leftframe by rightframe object with text attribute.
4. Create the 8x required buttons by using the button method with the text attributes then use the pack method with side, padx & pady attribute respectively.
5. ~~As Repeat the steps for the desired no. of buttons & then finally use the main method.~~

Message Box

1. import the relevant method from the tkinter library.
2. Define a function and use the message box along with the different methods available.
3. Create an object from the button method and place it onto the parent window, with the title of the button specified & the corresponding event called for triggering.
4. Use the pack method to display the button widget & finally use the mainloop method.
5. If the user want to close the parent window & only the information window should be visible corresponding to the options given above the withdraw method is used.

Message box

```
from tkinter import *
# import messagebox
root = Tk()
def info():
    messagebox.showinfo("Jui", "Python")
def warning():
    messagebox.showwarning("Jui", "Python")
def error():
    messagebox.showerror("Jui", "Python")
def askyesno():
    messagebox.askyesno("Jui", "Python")
def question():
    messagebox.askquestion("Jui", "Python")
def okcancel():
    message.askokcancel("Jui", Python)
b1 = Button(root, text="Info", command=info)
b1.pack()
b2 = Button(root, text="Warning", command=warning)
b2.pack()
b3 = Button(root, text="Error", command=error)
b3.pack()
b4 = Button(root, text="Yes no", command=askyesno)
b4.pack()
b5 = Button(root, text="Question", command=question)
b5.pack()
b6 = Button(root, text="OKcancel", command=okcancel)
b6.pack()
```

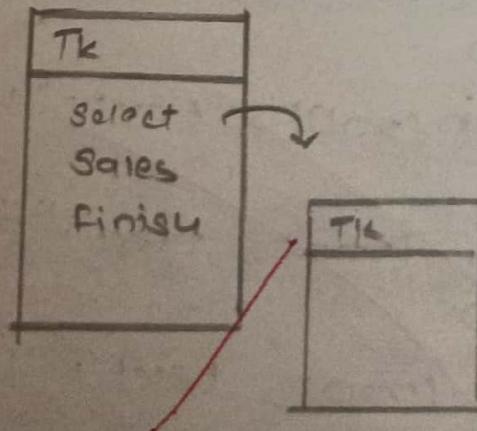
TK
INFO
WARNING
ERROR
YESNO
QUESTION
OKCANCEL

036

Transcending

```
from tkinter import *
root = Tk()
def main():
    root = Tk()
    root.config()
    root.title()
def sales():
    root = Tk()
    root.config()
    root.title()
def finish():
    quit()
```

```
b1 = Button(root, text="Select", command=main)
b1.pack()
b2 = Button(root, text="Sales", command=sales)
b2.pack()
b3 = Button(root, text="Finish", command=finish)
b3.pack()
root.mainloop()
```



Traversing

1. Define a function & create a object of the class window by using the three methods namely config, title & winsize.
2. now create a button object by using the button method with text & command attribute which will trigger the corresponding events.
3. Use the pack method with first button object.
4. Finally use this main
4. now repeat the same steps for the next button objects.
5. Finally use the mainloop method for triggering the corresponding events.

7. image

1. Use the Tkinter library.
2. Create a parent window & use the Geometry widget.
3. Create a frame object & by using the frame method and place it onto the parent window.
4. Now create the leftframe & rightframe object by using the frame method & use the side attribute within the pack method and subsequently use the padx, pady, ipadx & ipady attributes.
5. Create a object by using the Label method & place it onto the parent window.
6. Use the PhotoImage method to place the Photo & subsequently use the Subsample method & give the co-ordinates. And again use the Label method & place the original in the left frame.
7. Finally use the mainloop method for triggering the events.

#Image

038

```
from tkinter import *
root = Tk()
root.geometry ("380x380")

frame= Frame (root)
frame.pack()

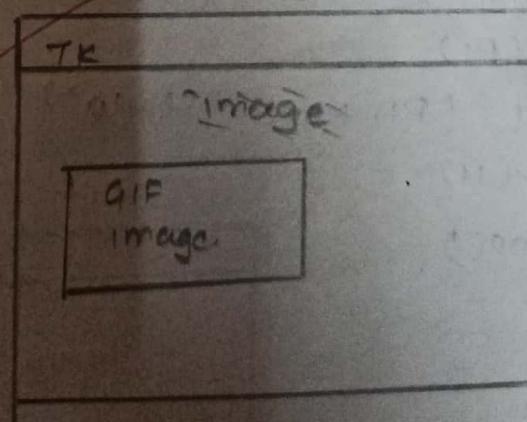
leftframe = Frame (root, bg="blue", height="20",
width="20")
leftframe. Pack (side = LEFT, padx= 20, pady = 20)

rightframe= Frame (root, bg="pink", height="20",
width="20")
rightframe. Pack (side = LEFT, padx= 20, pady = 20)

u= Label (frame, text ="Image", relief = RAISED)
u. Pack ()

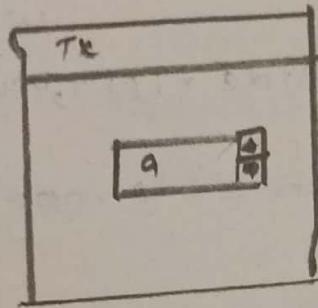
image = PhotoImage (file= "Pen.gif")
originalimage = image. Subsample (3,5)
label (leftframe, image = originalimage). Pack ()

root. mainloop()
```



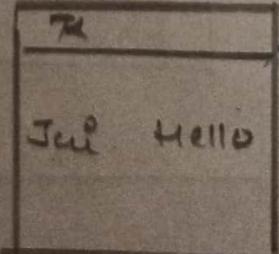
Spinbox

```
from tkinter import *
master = Tk()
s1=Spinbox(master, from_=0, to=10)
s1.pack()
master.mainloop()
```



Paned window

```
from tkinter import *
root = Tk()
p=PanedWindow(root)
p.pack(fill=BOTH, expand=1)
l=Label(p, text="Jui")
p.add(l)
p1=PanedWindow(root, orient=VERTICAL)
p.add(p1)
u=Label(p1, text="Hello")
p1.add(u)
mainloop()
```



Jui
Hello

spinbox

1. Create object from `tk()` by subsequently create from `spinbox` method.
2. Make new object so created like the parent window & trigger the corresponding events.

Paged window

1. Create an object from `Paged window` method by use `tkel` `pack` method & `fill` & expand
2. Create an object from the `label` method and put it on to the `Paged window` code like `text` attribute & use `add` method to imbed the new object.
3. now create another `label` object and pack it onto the `2nd` `Paged window` object and add `pk` on to the `2nd` `page`.
4. Trigger the `mainloop` method.

Q80

canvas

1. Create an object from the canvas widget & use the attribute width, background colour & the parent window object.
2. Use the method createoval, createoval & createoval along with the canvas object so created and use the coordinate values.
3. Similarly, use the other methods and call the pack method in the mainloop method.

Canvas

```

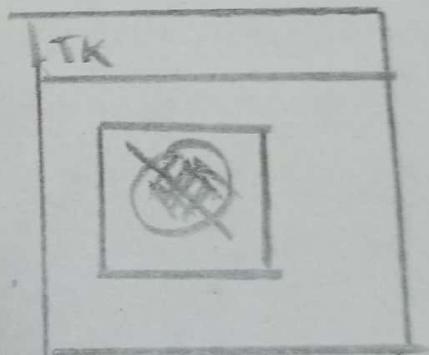
from tkinter import *
root = Tk()

c = Canvas(root)
coord = 10, 50, 240, 210
arc = c.create_arc(coord, start=0, extent=150,
                   fill="red")

line = c.create_line(coord, fill="black")
oval = c.create_oval(coord, fill="red")

c.pack()
root.mainloop()

```



Jordan

dbm

```
import dbm  
db = dbm.open ("database", "c")  
if db["https://www.google.com"] != None:  
    print ("URL found")  
else:  
    print ("URL not found")  
db.close()
```

Output!

URL found .

PRACTICAL NO-6

Aim: Demonstrate the use of Database connectivity.

dbm

ALGORITHM

1. import the dbm library & use the open method for creating the database by specifying the name of the database along with the corresponding flag.
2. Assume three string values C, N & W where C & N are used for creating a new database & while W flag is used for writing onto the existing database
3. Use the object so created for accessing the given website and the corresponding & regular name for the website.
4. check whether the given URL address with the regular name of the page is not equal to none. then print the URL found else print not found.
5. Use the close method to terminate the database library.

SQLite3

1. Import the corresponding library for the database.
2. Make a database and connect it to recognise the connect method.
3. Define a cursor object and make use of cursor method.
4. Use execute method & create the table with the database & make the structure of the table.
5. Use the execute method with the cursor object created & insert the value corresponding to the structure of the table created above.
6. Use commit method & then use the execute to now run the command.
7. Finally, use the fetchall method to display the output.

SQLite3

042

```
import os, sqlite3  
connection = sqlite3.connect("employee.db")  
cursor = connection.cursor()  
cursor.execute('create table des (name CHAR,  
                    Roll INTEGER)')
```

<sqlite3.Cursor object at 0x02E50520>

```
cursor.execute('insert into des values("Jai", 120)')
```

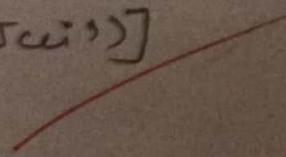
<sqlite3.Cursor object at 0x02E50520>

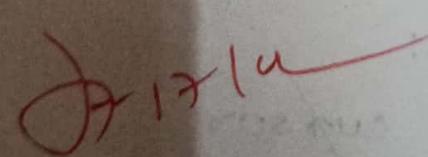
```
connection.commit()
```

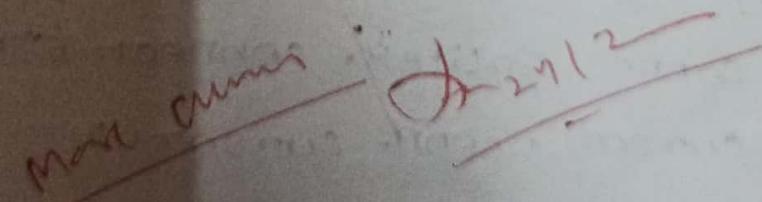
```
cursor.execute('alter table des ADD Surname TEXT')  
cursor.execute('Select Name from des')
```

```
cursor.execute('Update des Set Name = "Aayu" where  
                    Roll = 120)')
```

<sqlite3.Cursor object at 0x02E50520>

[('Jai')] 

Jai 

Maa Anna 

Project LOGIN FORM.

Summary:

1. Here GUI & Database concept is combined here together for the purpose of the admin and user login individually.
2. Here, from Database it is checking the values, if right then showing the corresponding output, else "Error".
3. GUI's widget is also used namely
 - (i) Label
 - (ii) Entry
 - (iii) Frame
 - (iv) PhotoImage
 - (v) Buttonwith both the methods pack & grid respectively.

Program:

```
from tkinter import *
import sqlite3
root = Tk()

root.title("Login Form")
width = 400
height = 280

screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()

x = (screen_width / 2) - (width / 2)
y = (screen_height / 2) - (height / 2)

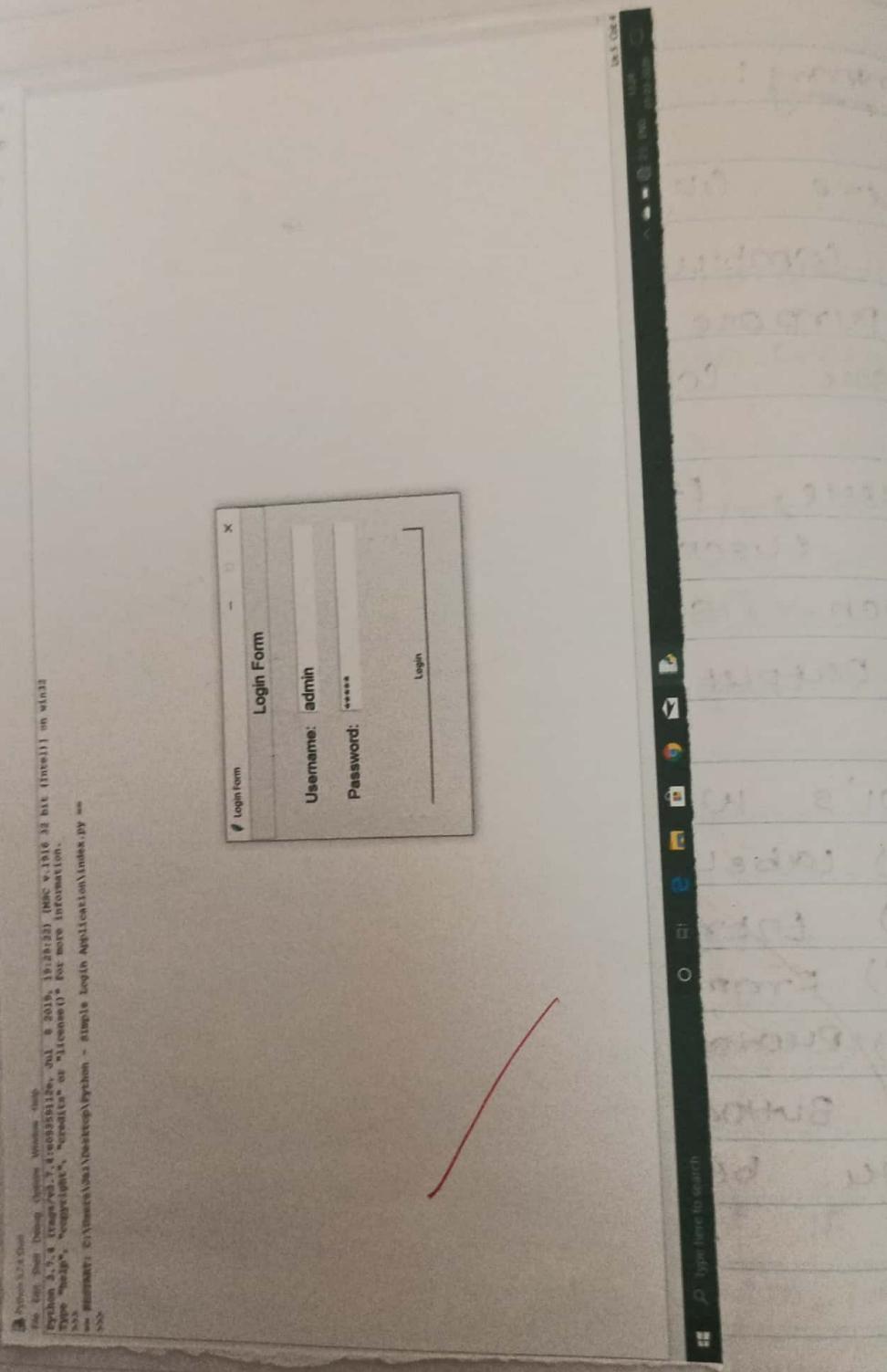
root.geometry(f'{width}x{height}+{x}+{y}')

root.resizable(0, 0)

def Database():
    global conn, cursor
    conn = sqlite3.connect("PythonProject.db")
    cursor = conn.cursor()

    cursor.execute("CREATE TABLE IF NOT EXISTS member(id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT, Username TEXT, Password TEXT)")

    conn.commit()
```



```
cursor.execute ("SELECT * FROM 'member' WHERE 'username' = 'admin' AND 'password' = 'admin'")  
if cursor.fetchone() is None:  
    cursor.execute ("INSERT INTO 'member' ('username', password) values ('admin', 'admin')")  
    conn.commit()  
  
def login (event = None):  
    Database()  
    if USERNAME.get() == "" or PASSWORD.get() == "":  
        lbl_text.config (text="Please complete the required Field!", fg = "red")  
    else:  
        cursor.execute ("Select * from 'member' where 'username' = ? AND 'password' = ?", (USERNAME.get(), PASSWORD.get()))  
    if cursor.fetchone() is not None:  
        HoweWindow()  
        USERNAME.set("")  
        PASSWORD.set("")  
        lbl_text.config (text="")
```

```
;10
else:
    if text == contig(text + "Invalid Username  
or Password", fg="red")
        USERNAME.set("")
        PASSWORD.set("")

    cursor.close()
    conn.close()

def HaveWindow():
    global home
    root.withdraw()
    home = Toplevel()
    home.title("Login Form")
    width = 600
    height = 500
    screen_width = root.winfo_screenwidth()
    screen_height = root.winfo_screenheight()
    xc = (screen_width / 2) - (width / 2)
    yc = (screen_height / 2) - (height / 2)
    root.resizable(0, 0)
    home.geometry("%dx%d+%d+%d" % (width, height, xc, yc))
```

```
lbl_home = Label (Home, text="Welcome !!!",
font = ('times new roman', 20)).pack()
```

```
img = PhotoImage (file= "Pic.gif")
label (Home, image = img).pack (Padx=50, Pady=10)
```

```
btn_back = Button (Home, text="Back To Main
Page", command = Back).pack (
Pady=20, fill=x)
```

```
def Back():
    Home.destroy()
    root.deiconify()
```

USERNAME = StringVar()

PASSWORD = StringVar()

TOP = Frame (root, bd=2, relief=RIDGE)

TOP.pack (side=TOP, fill=x)

form = Frame (root, height=200)

form.pack (side=TOP, Pady=20)

lbl_title = Label (TOP, text= "Login Form",
font = ('arial', 15))

lbl_title . pack (fill = X)

lbl_username = Label (Form, text = "Username",
font = ('arial', 14), bd = 15)

lbl_username . grid (row = 0, sticky = E)

lbl_password = Label (Form, text = "Password",
font = ('arial', 14), bd = 15)

lbl_password . grid (row = 1, sticky = E)

lbl_text = Label (Form)

lbl_text . grid (row = 2, column = 2, rowspan = 2)

username = Entry (Form, textvariable = USERNAME,
font = (14))

username . grid (row = 0, column = 1)

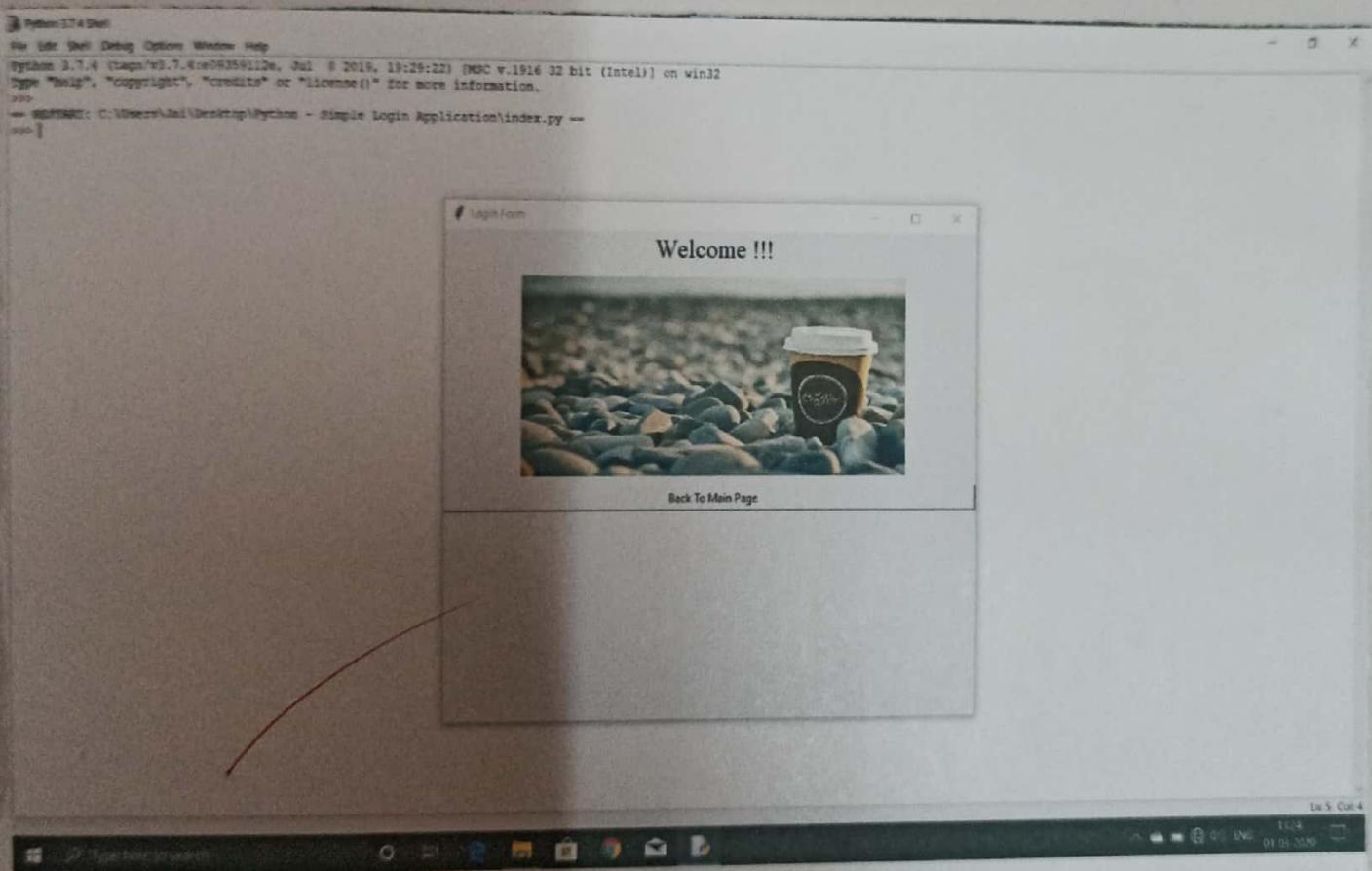
password = Entry (Form, textvariable = PASSWORD,
show = "*", font = (14))

password . grid (row = 1, column = 1)

bth_login = Button (Form, text = "Login" width = 45
command = login)

bth_login . grid (padx = 25, row = 3, columnspan = 2)

bth_login . bind ('<Return>', login)



DR 01/3