

Exam Seat No. \_\_\_\_\_

# THAKUR COLLEGE OF SCIENCE & COMMERCE

NAAC  
Accredited  
with Grade "A"  
(3<sup>rd</sup> Cycle)



ISO  
9001 : 2015  
Certified

## Degree College Computer Journal CERTIFICATE

SEMESTER 2 UID No. \_\_\_\_\_

Class FYBSC CSA Roll No. 1705 Year 2019-20

This is to certify that the work entered in this journal  
is the work of Mst. / Ms. Ayushmaan Kanwarkar.

who has worked for the year 2019 in the Computer  
Laboratory.

~~Teacher In Charge~~

~~Head of Department~~

Date : \_\_\_\_\_

Examiner

Semester

# INDEX

No.	Title	Page No.	Date	Staff Member's Signature
1.	Practical 1: Using different Access modes.	19	26-12-19	Jy 31/12/19
2.	Practical 2: Threads	21	26-12-19	Jy 10/12/19
3.	Practical 3: Exception	23	10-12-19	Jy 12/12/19
4.	Practical 4: Re Expressions	25	17-12-19	Jy 24/12/19
5.	Practical 5: GUI in Python. (Label, frame, TK, Text)	27	7-1-20	Jy 14/01/20
5.1	More with Tkinter. (Label, frame, TK, Text, Scrollbar, Radio Button, Geometry, Listbox)	29	14-1-20	Jy 21/1
5.2.	GUI - Tkinter - (Messagebox, relief, hovering through window)	31	21-1-20	Jy 22/1
5.3	GUI - Tkinter - (Grid, Photoimage) (Canvas)	33	28-1-20	Jy 11/2
5.4	GUI - Tkinter (Canvas, Spinbox, Toplevel Window)	35	11-2-20	Jy 11/2
6.	Database Management with Python	37	14-2-20	Jy 12/2

# ★ INDEX ★

① CODE

```
fileobj = open ("abc.txt", "w") # Write Mode  
fileobj.write ("WATASHI IN WATASHI")  
fileobj.write ("New in New")  
fileobj.close()  
file1 = open ("abc.txt", "r") # Read Mode.  
SR1 = file1.read()  
print ("output: ", SR1)  
file1.close() OUTPUT  
» New  
New
```

```
file1 = open ("abc.txt", "r")  
SR2 = file1.readlines () # Readlines  
print ("readlines: ", SR2)  
file1.close () OUTPUT  
» readlines : New  
  
file2 = open ("abc.txt", "r")  
SR3 = file2.readlines () # Readline  
print (SR3)  
file2.close () OUTPUT  
» ['New in New']
```

```
a = file1.name  
print (a)  
» "abc.txt"  
  
b = file1.closed  
print (b)  
» True
```

✓

```
c = file1.mode  
print (c)  
» r  
  
de = file1.softspace  
print (a)  
» 0
```

Practical 1:  
Objective: Demonstrate the use of different file accessing modes & different attributes read mode.

Step 1: Create a file object using open method and use write access mode followed by writing some contents onto the file & then closing the file.

Step 2: Now open the file in read mode & use read(), readline() & readlines() & store the output in variables & finally display the contents of the variable.

Step 3: Now use the fileobject for finding the name of the file, the file mode in which its opened whether the file is still open or close & finally the output of the softspace attribute.

Step 4: Now open the file obj in write mode. Write some another contents & close it, then again 'wt' that is update & write contents.

Step 5: Open fileobj in read mode display the update written contents close the & open again in 'r+' & update.

210

**Step 6:** Now open fileobj is append mode, open write method write contents, close the file obj & again open the file obj in read mode & display the appending output.

Step 7: Open the file in read mode, declare a variable & perform file obj do tell method & store the output cons in variable.

Step 8: Use the seek method with opening the file obj in read mode & closing sub.

Step 9: Open the file(s) with read mode also use readlines method & store the output consequently & print the same for counting the length use the statement to display length.

```
fileObj = open("abc")
fileObj.write("Ne")
fileObj.write("N")
fileObj.close()

file1 = open("abc")
file1.read(2)
print(a)

b. file1.read()
print(b)          out

>> New
>> NewNew
```

```
#append mode  
file1 = open("abc.txt", "a")  
file1.write("New")  
file1.close()  
file1 = open("abc.txt", "r")  
print(file1.read())  
print("New")  
> NewNewNew1
```

```
# tell 1)  
fileObj = open ("c  
pos = fileObj.tell  
print (pos)  
=> 10
```

```
open  
file obj  
mode  
  
declare  
dot  
  
using the  
ubs.  
  
also  
output  
  
length.  
  
#append mode  
file = open("abc.txt", "a")  
file.write("New1")  
file.close()  
file = open("abc.txt", "a")  
a = file.read()  
print(a)  
file.close()  
print(a) OUTPUT  
=> New  
=> NewNew
```

```
#tell()  
fileObj = open("abc.txt", "w+") # w+  
fileObj.write("New")  
fileObj.write("New")  
fileObj.close()  
file = open("abc.txt", "r") # r+  
a = file.read(2)  
print(a)  
b = file.read()  
print(b) OUTPUT  
=> New  
=> NewNew
```

```
#seek()  
fileObj = open("abc.txt", "w+")  
pos = fileObj.tell()  
print(pos)  
=> 10
```

020  
#finding length of  
different lines

```
file1 = open("abc.txt", "r")  
sk1 = file1.readlines()  
for q in sk1:  
    print(len(q))  
file1.close()  
=> 10
```

# seek  
file1 = open("abc.txt", "r")  
st4 = fileObj.seek(2)  
print(st4)  
=> None

2/10/19

<code>

```
# func() & next()  
table 1 = ("banana", "orange", "apple")
```

```
ite 1 = ite (table 1)  
print (next (ite 1))  
print (next (ite 1))  
print (next (ite 1))
```

Output:  
» banana  
» orange  
» apple

# for loop

```
table 1 = ("banana", "orange", "apple")
```

```
for i in table 1:  
    print (i)
```

Output:

» banana  
» orange  
» apple

# square & cube

```
def square (x):  
    y = x**2
```

```
def cube (x):  
    z = x**3
```

return z

```
table = [square, cube]
```

```
for i in range (5):
```

```
    value = list (map (lambda n: n (r), func 1))
```

print (value)

\* Code:  
Step 1  
ite

Step 2

Step 3

Step 4

### ? Practical 2:

#### Iterators.

\* Code:

Step 1: Create a tuple with elements that we need to iterate using iter and next method. The number of times we use the iter & next method will get the next iterating element in the tuple. Display the same.

Step 2: Create a tuple with element then using a for loop with range upto the tuple, display each element.

Step 3: Define a user-defined function square with arithmetic eqn to square a given variable, using another user defined fn create an eqn to s cube of the given variable. Create a list to enter ~~a~~ the fu names, using loop and a lambda function, print appropriate value.

Step 4: Create a list elements to have values on it. using lambda function to call elements from it use if fu to check if its divisible by 2 and give appro. output.

output

```
[0, 0]  
[1, 2]  
[4, 8]  
[9, 27]  
[16, 64]
```

#map

```
listnum = [0, 2, 7, 9, 11, 13, 15, 17, 19, 21, 25]
```

```
listnum = list(map(lambda n: n**2, listnum))
```

```
if (n % 2 == 0):
```

```
    return "Even"
```

else:

```
    return "odd"
```

```
print(*map(lambda n: n if n % 2 == 0 else "Even", listnum))
```

#odd\_numbers using class iter..

```
class mody:
```

```
    def __iter__(self):  
        self.a = 1  
        return self
```

```
    def __next__(self):  
        if (self.a <= 20):  
            num = self.a  
            self.a += 2  
        else:  
            raise StopIteration  
        return num
```

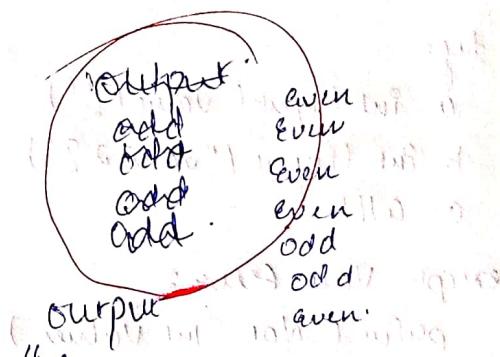
```
myobj = mody()
```

```
myitee = iter(myobj)
```

```
for i in myitee:  
    print(i)
```

odd numbers using class iter

022



Output

1	odd
3	odd
5	odd
7	odd
9	odd
11	odd
13	odd
15	odd
17	odd
19	odd

Output

odd

Code # Environment Error

```
try:  
    fo = open("abc.txt", "w")  
    fo.write ("writings here in okay")  
except IOError:  
    print ("failed")
```

Output  
> success

else:  
 print ("success")

# Arithmetic Error

```
try:  
    a = int(input("value1 ?"))  
    b = int(input("value2 ?"))  
    c = a/b
```

```
except ValueError:  
    print ("Not Int Values")  
else:  
    print (c)
```

Output  
> value1 ? 2.2  
value2 ?  
> failed Not Int Values

# Arithmetic Errors

```
try:  
    a = 1  
    b = 0  
    result = a/b
```

```
except ZeroDivisionError:  
    print ("Zero Division")  
else:  
    print ("success")
```

Output  
Zero Division

\* code  
Step

Step

Step

Step

## Practical 3:

## Exception:

\* code:

Step 1: Use try block to enter file operation action and give 'w' attribute to it. Using write operation enter some strings; take Except block with IOError, display error if error or using else block print appropriate output.

Step 2: Use try block to take 2 values from the user using type-casting as int. Then divide the values. Using Except block with ValueError, display the appropriate error, or using else block display the output of the division.

Step 3: Use try block and assign variable a value and assign another value to a second variable, perform operations on the division, use except block with ZeroDivisionError to display any error or with else block display appropriate output of result.

Step 4: Create a list with a set amount of elements in it. Using try block print value from the list within index and then print value outside of index, using the except block with IndexError display appropriate error.

# lookup tree

A = [1, 2, 3]

try:  
print(A[1])  
print(A[4])

except IndexError:

print("Out of bounds")

→  
<code>

<code>

z = print(  
 print("value ! = 1")  
)

try:  
if (z == 1):

raise ValueError

finally:

print(z)

021

Output:

2

Out of bounds

→  
<code>

Output:

value ! = 1 2

ValueError

1

① <code>

```
import re
par = r'f4CS'
seq = 'f4CS is CS Stream'
if re.match(par, seq):
    print("found")
else:
    print('not found')
```

\*OUTPUT  
found

② <code>

```
import re
# \d → numerics in a string
# \D → alphabets in a string
par = r'\d+'
par1 = r'\D+'
seq = 'hellow123, howdy69, 12Yellow'
print(re.findall(par, seq))
print(re.findall(par1, seq))
```

\*OUTPUT  
['123', '69', '12']  
['hellow', 'howdy', 'Yellow']

③ <code>

```
import re
print(re.split(r'\w+', 'Words, words, Words'))
print(re.split(r'\W+', 'Word's, words, word'))
```

# \w → non alphanumeric  
# \W → alphanumeric

</code>

\*OUTPUT

```
[['Words', 'words', 'Words']]
```

## Practical 4:

## Re Expression:

~~Step 1: Importing re module, initialize a variable with "F4CS" and a sequence to go with it. Using conditional statements, use ~~match~~ function to check the given 2 variable and display appropriate output.~~

~~Step 2: Using re module, create a variable for Id to get numbers and another 1D string alphabets from a string; initialize a sequence of alphanumeric characters to a variable, using function.findall, check the variables with the sequence and display appropriate output.~~

~~Step 3: Importing re module, use split function to get alphanumeric and non-alphanumeric characters of a sequence of string and display appropriate output.~~

~~Step 4: Importing re module, create a variable to store an empty string; create a variable to store part to get whitespace character and initialize a sequence of string, using 'sub' function now initialize all these variable in sequence and display appropriate value.~~

PSO

Step 5: Create a variable to have alphanumeric with one or more as a set of character using symbol + and @ for emails and the set, using.findall function find the email from a set of variable containing strings and with the display appropriate output.

Step 6: Create a list of elements containing random phone numbers. Using for loop for the list of elements, use match function to check if first two elements does 8 or 9 and is 10 digits long, the display appropriate output with phone no.

Step 7: Importing re module, create a sequence variable containing strings, using.findall functions, insert vowels in set, having alphanumeric set and another also with 16 to get for start of string! display appropriate output.

(1) code  
import  
pat = "  
# 8 →  
# 3 →  
rep = ""  
string = a  
Output  
print()  
</code>  
(2) code  
pat = "  
seq = ""  
print()  
</code>  
\* OUTPUT  
['abc@i

(3) code  
seq = [ ]  
for val

\* OUTPUT  
cell 91  
cell 8

026

(1) `<code>`  
import re  
pat = re.compile(r'')

# S → whitespace character

# S → non-white ---

rep = "

string: 'abc def ghi'

Output = re.sub(pat, rep, string)

print(output)

`</code>`

\* OUTPUT

'abcdefghi'

(2) `<code>`

pat = re.compile(r'[\w\.-]+@[ \w\.-]+\t')

seq = 'log on to abc@edu.in and waka@gogo.go'

print(re.findall(pat, seq))

`</code>`

\* OUTPUT

['abc@edu.in', 'waka@gogo.go']

(3) `<code>`

seq = ['9142618200', '8216218000', '123467']

for value in seq:

if re.match(r'[8-9]\d{1}[0-9]\d{2}', value):  
 print('cell', value)

\* OUTPUT

cell 9142618200

cell 8216218000

`<node>` `import` `ue`  
`seq = seqplay fat fellow on go apple`  
`print (ue. findall (u'`b[aeiou]wr', seq))`  
`print (ue. findall (u'[aeiou]wr', seq))`

`<1 code>`

## \* OUTPUT

[‘on’, ‘apple’]

[on, apple] [omeday, at, 'ellow, on, go, 'apple']

8

`<code>`

import re

seq = 'peru pan bear on the wall'

point 14.  
# DUSTPOT.

[ 'pe', 'pa', 'sa', 'on', 'th', 'wa' ]

$\langle \omega_d \rangle$

## Importance

• Seg = 'm. telangana m. wataka m. ngumpali'  
mal = 're. hindati / y / m. 7821 , - ?'

$\text{max} = \text{true} \cdot \text{findall}([4], [\text{me.}], \{2\}, 1.., \text{seq})$

```
final = (u.findall('mes.', seq))
```

print (len(mal))

plant (her (female))

## \* OUTPUT

1

2

## Practical 5:

## GUI in Python - Tkinter:

Step 1: Use tkinter library to import the necessary widgets.

Step 2: Create an object using TK() module.

Step 3: Create a variable and put Label for widget in it and add some text and pack it.

Step 4: Using mainloop() on the object run the program.

Step 5: Using tkinter library import.

Step 6: Create a variable from the text method & position it on parent window.

Step 7: Using pack on the variable use the side parameter and padding on it.

Step 8: Use mainloop on the object & run the program.

Step 9: Now repeating the above steps on a new program, make use of bg → background, fg → foreground → name of parent window, the use pack to add relevant attributes.

```
(node)
from tkinter import *
```

```
tk = TK()
```

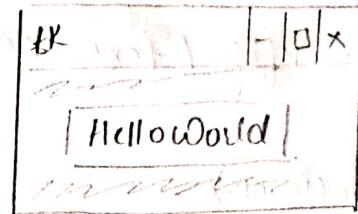
```
frame = Frame(tk)
frame.pack()
```

```
label = Label(frame, text = "Hello World", bg = "red", fg = "black", font = "Arial",
height = 5, width = 40)
```

```
label.pack(side = TOP, padx = 70)
tk.mainloop()
```

```
(node)
```

028



```
# Basic Hello World
```

```
import tkinter as gui
```

```
root = gui.TK()
```

```
frame = gui.Frame(root).pack(expand = 1)
```

```
label1 = gui.Label(frame, text = "Hello, myself")
```

```
label1.pack()
```

```
root.mainloop()
```

```
# Complex
```

```
from tkinter import *
```

```
root = TK()
```

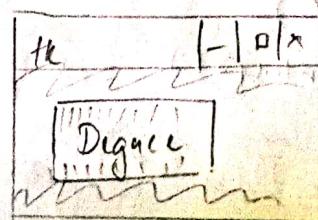
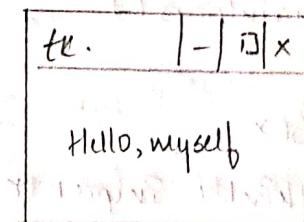
```
frame = Frame(root).pack(fill = True, expand = 1)
```

*Jan 14/19*

```
l1 = Label(frame, text = "Degree", fg = "yellow", bg = "red", font = "20")
```

```
l1.pack(side = BOTTOM, ipadx = 20, ipady = 40)
```

```
root.mainloop()
```



<code>

```
# Radio Button
import tkinter
from tkinter import *

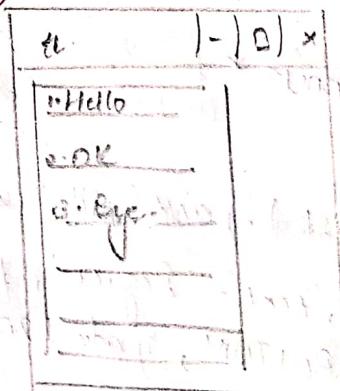
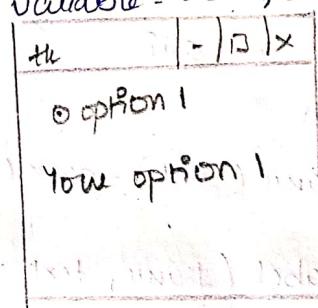
def sel():
    select = "Your option " + str(var.get())
    lab1.config(text=select, justify=RIGHT)

root = Tk()
var = IntVar(root)
frame = Frame(root).pack(expand=1)
q1 = Radiobutton(frame, text="Option 1", variable=var, value=1,
                  command=sel)
q1.pack(anchor=S)
lab1 = Label(frame, text='Hello')
lab1.pack()
root.mainloop()
```

# Listbox

```
from tkinter import *
```

```
root = Tk()
frame = Frame(root).pack()
ls = Listbox()
ls.insert(1, "Hello")
ls.insert(2, "OK")
ls.insert(3, "Bye")
ls.pack()
root.mainloop()
```



More GUI using Tkinter  
(Radiobutton, ScrolledListbox, Geometry)

- # Step 1: Import necessary Tkinter modules and initialize the Tkinter frame.
- Step 2: Create a function sel, and creating variable select, input or function with the variable (using label.config) input select variable as text.
- Step 3: Using the mainloop, create variable var and use interval() passing root as frame variable.
- Step 4: Create Radiobutton in a variable and add the necessary attributes and pack it.
- Step 5: Create label attribute and give necessary attributes finally run mainloop to run the program.
- # Step 6: Using the above method to create a Tkinter window.
- Step 7: Use the listbox function to create a listbox.
- Step 8: Insert various strings to list using the insert function and passing all arguments. Use mainloop & run the program.

PSQ

# Step 9: Using the previous methods, create a Tkinter window.

Step 10: Using the Text widget pass the root variable to it.

Step 11: Create ScrolledText widget & pass the necessary attributes, thereby linking it to the Text widget.

Step 12: Using insert, insert paragraphs of text into the text widget and pack all widget.  
Finally using mainloop, run the program.

# Step 13: Create a tkinter window with appropriate tkinter libraries.

Step 14: Create variable top and use Ttk() and using top as fn, use geometry and pass resolution as string.

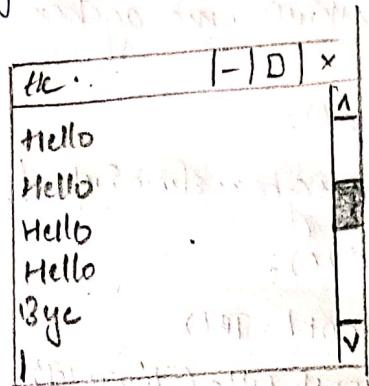
Step 15: Create , frame , leftframe & rightframe and pack with the necessary attributes.

Step 16: Create a button widget , using rightframe as parent, appropriate attributes and pack it.

Step 17: Finally use mainloop and run the program.

## # Scrollbar

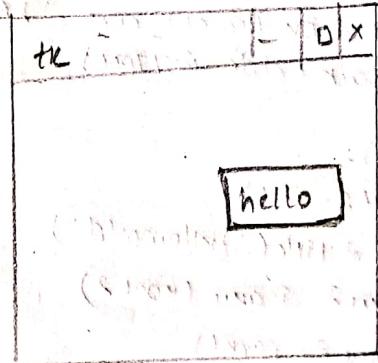
```
from tkinter import *  
t1 = "Hello\nHello\nHello\nHello\nHello\nHello\nHello\nHello\nBye"  
root = Tk()  
frame = Frame(root).pack()  
T = Text(root)  
S = Scrollbar()  
S.pack(side=RIGHT, fill=Y)  
S.config(command=T.yview)  
T.config(yscrollcommand=S.set)  
T.insert(INSERT, t1)  
T.pack(side=BOTTOM)  
root.mainloop()
```



Output

## # Geometry

```
from tkinter import *  
top = Tk()  
top.geometry("500x200")  
frame = Frame(top)  
leftframe = Frame(top)  
rightframe = Frame(top)  
frame.pack()  
leftframe.pack(side=LEFT)  
rightframe.pack(side=RIGHT)  
B1 = Button(rightframe, text="Hello", activebackground="green", bg="green")  
B1.pack()  
mainloop()
```



Output

Jr  
111

## # Messagebox & Others

```
1 wader 186
from tkinter import *
import tkinter.messagebox as msgbox

def msg():
    msgbox.showinfo("hello", "helloworld")

def wind1():
    root1 = Tk()
    root1.title("helloworld")
    frame1 = Frame(root1)
    frame1.pack()
    t1 = Text(root1)
    t1.insert(INSERT, "new window")
    t1.pack()
    b3 = Button(root1, text="LAST window", command=wind1, relief=GROOVE)
    b3.pack(side=BOTTOM)

def wind2():
    root2 = Tk()
    root2.title("helloworld")
    frame2 = Frame(root2)
    frame2.pack()
    b4 = Button(root2, text="First window", command=main, relief=RIDGE)
    b4.pack()

def finish():
    quit()

def main():
    root = Tk()
    root.wm_title("cyan")
    root.minsize(400, 400)
    root.title("5,2")
    frame = Frame(root)
    frame.pack()
    l1 = Label(frame, text="traverse")
    b0 = Button(frame, text="new-win", command=wind1)
```

GUI

Step 1

Step 2:

Step 3

Step

Step

Step

Step

## Practical 5.2:

GUI → MessageBox, Relief, Traversing through window

Step 1: Import the necessary tkinter library & messagebox

Step 2: Define function msgbox & create an object msgbox with some info.

Step 3: Define function wind1, create a new root tkinter window and add title, & create object frame widget, finally also create text widget & button widget with command being wind1 & pack.

Step 4: Define function wind2. Create a new tkinter window, add frame widget, & add a button widget to traverse through the first window, as command as main.

Step 5: Create a function finish, with keyword quit() to quit program.

Step 6: Finally create function main, with root as default parent window, we config to add bg color to root, message & title, create frame widget, label & button with command as wind1, create another button with command as msgbox finally last button with command as finish, pack all widget accordingly and we main loop function.

Step 7: Finally call main() function outside.

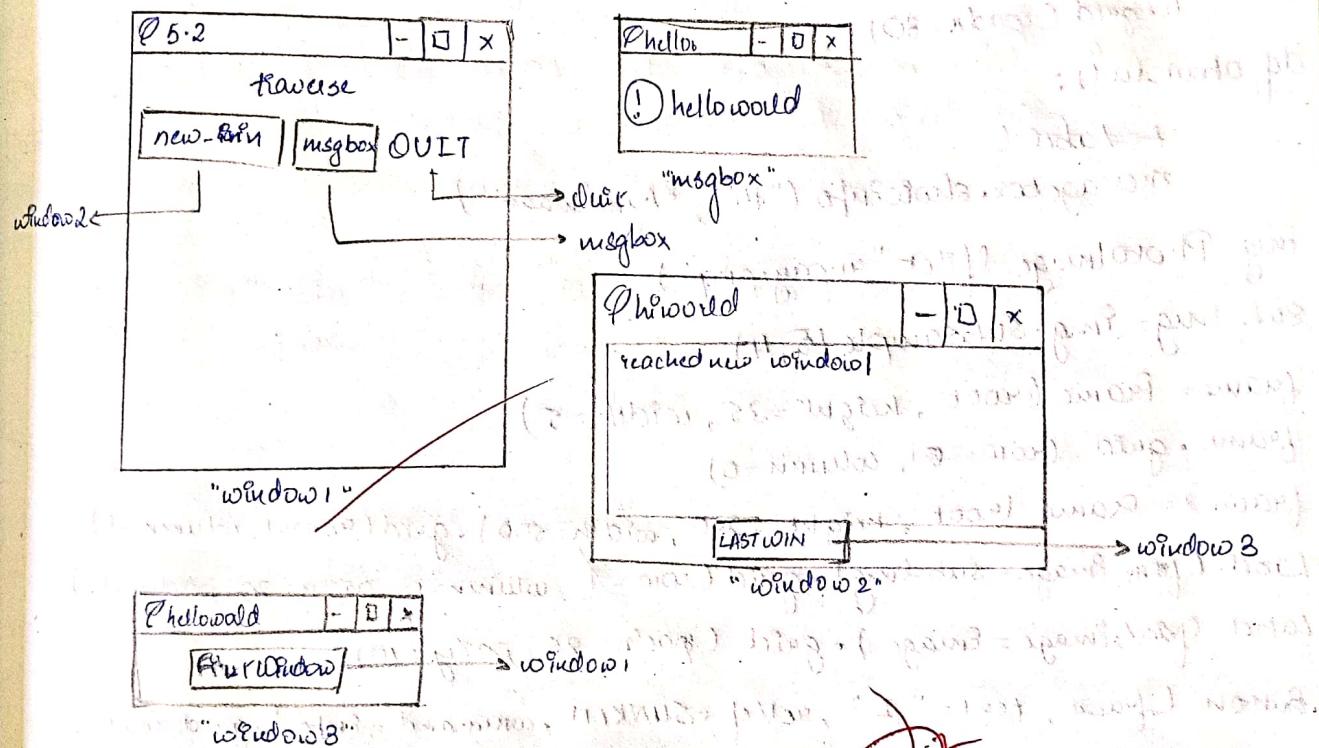
```

B1 = Button(frame, text = "msgbox", command=msg, relief=RAISED)
B2 = Button(frame, text = "QUIT", command=finish, relief=FLAT)
B0 = pack(side=LEFT, ipadx=10)
B1 = pack(side=LEFT)
B2 = pack(side=LEFT)
mainloop()

```

main()

### OUTPUT:



```

    <code> from tkinter import *
root = Tk()
root.title("Hello")
root.minsize(400, 500)
def end():
    messagebox.askokcancel("Exit", "End Program")
    quit()
def info():
    li = Listbox()
    li.insert(1, "Android")
    li.insert(2, "Hello")
    li.grid(ipadx=80)
def about_us():
    text =
    messagebox.showinfo("Hi", "Ayushmaan")
img = PhotoImage(file="aupay.png")
sub_img = img.subsample(5, 4)
frame = Frame(root, height=35, width=5)
frame.grid(row=0, column=0)
frame2 = Frame(root, height=250, width=500).grid(row=1, column=1)
Label(frame, image=sub_img).grid(row=1, column=0, padx=20, pady=15)
Label(frame, image=img).grid(padx=25, pady=10)
Button(frame, text="I", relief=SUNKEN, command=info).grid(row=1, column=0)
Button(frame, text="A", relief=RIDGE, command=about_us)
    .grid(row=1, column=2, padx=5)
B3 = Button(frame, text="Q", command=end)
B3.grid(row=2, column=1, ipadx=15)
mainloop()
# END Program

```

## Practical B-3:

GUITKINTER → GRID, PHOTOIMAGE

Step 1: Import the necessary tkinter modules and create a parent window root and give title and message attributes to it.

Step 2: Create a function end to quit program and add a messagebox into it.

Step 3: Create another function info and add listbox with some data insert to it.

Step 4: Finally create our function for aboutus panel and create appropriate messagebox for it.

Step 5: Using object image, create image and make subsample of it and store in another variable.

Step 6: Create two frame for parent window with some attribute of height & width to it and use grid to place them.

Step 7: Create two label and store both the object image one in each respectively, finally use grid.

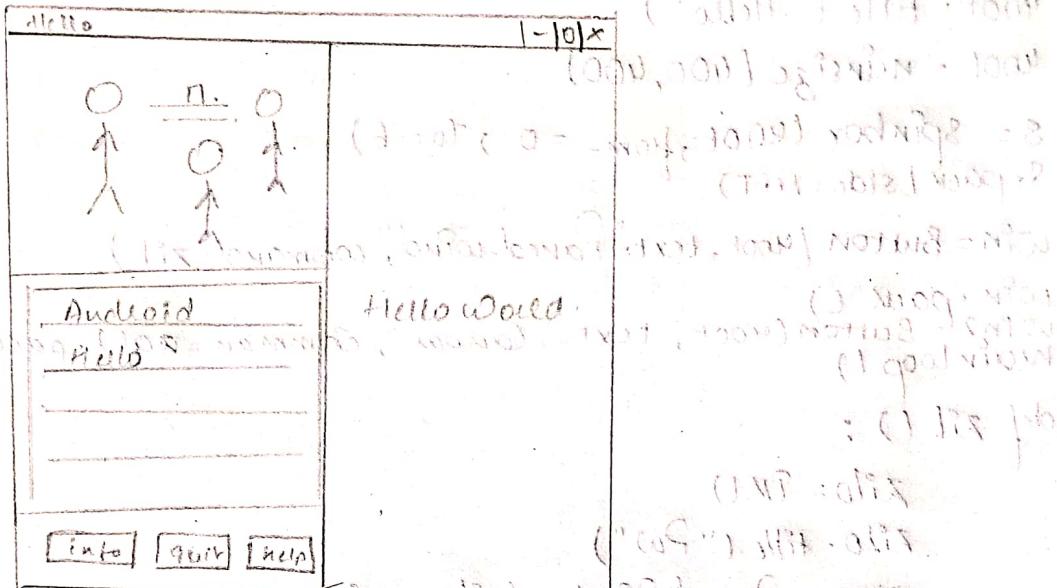
100

i&Q

Step 8: Create buttons and add some attributes to it's and assign command function to each to call the user created functions from the class respectively and finally used grid to place them.

Step 9: Finally trigger the event by adding mainloop to it.

OUTPUT



```

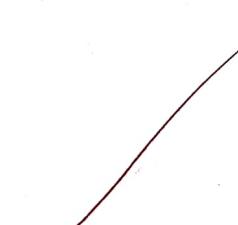
from tkinter import *
root = Tk()
root.title("Hello")
root.minsize(400, 400)
s = Spinbox(root, from_ = 0, to = 6)
s.pack(side = LEFT)
w1 = Button(root, text = "Paned-Window", command = zil)
w1.pack()
w2 = Button(root, text = "Canvas", command = zal)
w2.pack()
mainloop()

def zil():
    zilo = Toplevel()
    zilo.title("PW")
    pane = PanedWindow(zilo, orient = VERTICAL)
    pane.pack()
    label = Label(pane, text = "Pane", fg = "green")
    label.pack()
    pane.add(label)

    cb = Checkbutton(pane, text = "lol")
    cb.pack()
    pane.add(cb)

def zal():
    zalo = Tk()
    c = Canvas(zalo, height = 400, width = 400)
    c.create_oval(10, 80, 80, 180, fill = "red")
    c.create_oval(10, 180, 80, 250, fill = "green")
    c.create_line(10, start = 0, end = 1)
    c.pack()

```



TKinter GUI → Canvas, Spinbox, Parent Window:

Step 1: Import tkinter library and create a parent window.

Step 2: Create object Spinbox and add the necessary attributes and create two button to call two functions respectively and pack all widgets and trigger mainloop event.

Step 3: Create function zii, create new parent window, create object parent window and pack.

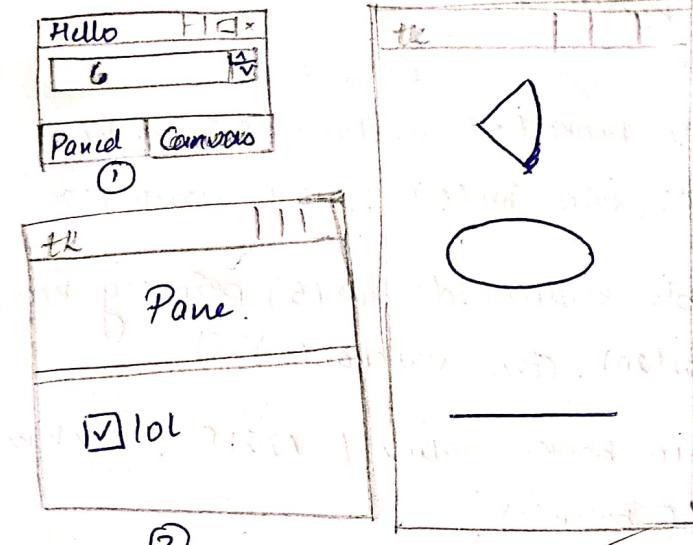
Step 4: Create a label and checkbox with parent window being the panel window object & pack, used pane.add(object) to add the widget.

Step 5. Clear another window on a new function definition, create variable to store 4 coordinates, create canvas object and add needed attributes to it, finally draw arc, oval, line with the required attributes and stored coordinate and pack canvas.

Done

homework

036



Collect the selected word with the first letter of the word.

(C) (A) (B) (D) (E) (F) (G) (H) (I) (J) (K) (L) (M) (N) (O) (P) (Q) (R) (S) (T) (U) (V) (W) (X) (Y) (Z)

(W) (X) (Y) (Z)

1 code

```
import dbm, os, sqlite3  
con = sqlite3.connect("book1b.db")  
cur = con.cursor()  
cur.execute("Create table books (id varchar(5) primary key,  
name varchar(20), price int(5), author varchar(20))")  
cur.execute("create table readers (ids int(5) primary key,  
name varchar(20), class varchar(5))")  
cur.execute("insert into books values ('AZ21P', 'Alchemist',  
400, 'O-brien'),  
('J21CA', 'Python-B', 700, 'JamesC'),  
('ZZZ11', 'C#-Unity', 1820, 'Rudolf')")
```

con.commit()

cur.execute("select \* from books where id= ZZZ11")

lot =

cur.fetchall()

print(lot)

con.close()

→ output

[('ZZZ11', 'C#-Unity', 1820, 'Rudolf')]

→ 1 code

import dbm

db = dbm.open("lot", "c")

db["lot"] = "Write lot"

if db["lot"] != None:

print("Yes")

else:

print("No")

→ output:

Yes.

## Practical 6:

037

### : Database Management with Python:

- ① Import necessary modules to work with the database and create a connection with the database.
- ② Now create object to execute the database cursor.
- ③ Finally using the cursor execute command , create table with appropriate dbms command syntaxes .
- ④ Insert data as to the table created and use commit to perform transaction to the database.
- ⑤ Finally , using select dbms command get data , and using fetchall , store it in a variable .
- ⑥ Lastly , print the variable & close the database .
- ⑦ Now for dbm module , import , using open , initialize database , store some string value to the database .
- ⑧ Finally using if check if database has the string value & display appropriate data .

Ans 12

## Project-01: DBMS

01

file 1:

File - G:\PG\backup\_for project\database\01.initialize - database.py

```
File - G:\\GBackup_for project\\database1.py

1 ##
2 # this is just going to create a table, following
3 # procedures soon.
4 # also while inserting some random data for instant use.
5 #
6 import os,sqlite3
7 import dbm
8 import time
9 com = sqlite3.connect("muta.db")
10 cur = com.cursor()
11 #
12 cur.execute("create table customer (id int(5),name varchar
    (50),credit varchar(19),room_id int(4),comments varchar(
    100),grade varchar(2))")
13 cur.execute("create table employee (emp_id int(2),emp_name
    varchar(20),emp_comments varchar(20),emp_pass int(5))")
14 #
15 print("Tables Created Feeding Data..")
16 time.sleep(1)
17 cur.execute("""insert into customer(ID,name,credit,room_id
    ,comments,grade) values (10001,"Ravi P","4567 8901 2345
    6789",3101,"likes roses","B-"),(10002,"Bridgette B","4588
    8505 2444 6782",2105,"always keep extra champagne","A+"),(
    10003,"Rena L","4564 8301 2545 6769",3102,"raque is the
    soap","A"),(10004,"Lisa P","4547 8341 5334 6484",5101,"new
    customer","B+"),(10005,"Jennie C","8909 6546 4324 6432",
    9101,"only give higher floors","C")""")
18 #
19 print("Table Customer is feeded with 5 data..")
20 time.sleep(1)
21 print("Feeding data to employee")
22 cur.execute("""insert into employee(emp_id,emp_name,
    emp_comments,emp_pass) values (12,"Farhan C","null",3245
    ),(13,"Jaque D","null",4455),(14,"Grisa M","null",9900)""")
23 print("Table emp is feeded with 3 data")
24 time.sleep(1)
25 print("line executed, data not saved")
26 time.sleep(1)
27 print("printing all entered data")
28 # cur.execute("select name from customer and emp")
29 id_store = 10001
30 print("select id,name,room_id,comments,grade from customer
    where id = {0}".format(id_store))
```

```
File - G:\PGbase  
31 cur.  
32 cust  
33 a =  
33 for  
34  
35 pri  
36 cur  
37 b =  
38 pri  
39 pri  
40 #cu  
41 com  
42 com  
43  
44  
45  
46 ##  
47 ##  
48  
49 ##  
50 ##  
51 ''  
52 db  
53  
54 if  
55  
56 el  
57  
58  
59 pr  
60 #  
61  
62
```

040

(02)

File - G:\PG\backup\_for project\database\01\_initialize - database.py

```
31 cur.execute("select id,name,room_id,comments,grade from
customer where id = {0}".format(id_store))
32 a = cur.fetchall()
33 for i in range(0,1): # print a single data
34     print(a[i])
35 print(a)
36 cur.execute("select * from employee")
37 b = cur.fetchall()
38 print(b)
39 print('\n\n done..!')
40 #cur.fetchall()
41 com.commit()
42 com.close()
43
44
45
46 ##
47 ##
48
49 ###
50 ###
51 '''db = dbm.open("lol","c")
52 db["lol"] = "WRITE OPE"
53
54 if db["lol"]!=None:
55     print("lasun")
56 else:
57     print("wata")
58
59 print(db["lol"])
60 # FTW
61
62
```

01

file 2:

File - G:\PGI\backup\_for project\database\02. work - with the database.py

```
1 ## program to work on already created database.
2 # test on taking password from employee, and say their
3 # name on the write id n pass
4 # finally give emp data on the customer they want,
5 # except credit, unless specified to quit.
6 ##
7 import os,sqlite3
8 import dbm
9 import time
10 import sys
11 com = sqlite3.connect("muta.db")
12 cur = com.cursor()
13
14 def cust(com,cur):
15     que = input("Do you want customer data? or LOGOUT
16     ? (Y/N)")
17     que.upper()
18     if que == "Y":
19         # print("What customer data do you want?")
20         info(com,cur)
21     else:
22         time.sleep(1)
23         print("LOG OUT.")
24         sys.exit()
25
26 def info(com,cur):
27     while True:
28         inc = input("What would you like to search FOR? (ID
29         /NAME/ALL/ROOT")
30         inc.upper()
31
32         if (inc == "ROOT"):
33             print("root environment not allowed\n you have
34             been reported")
35             sys.exit()
36
37         elif (inc == "ALL"):
38             cur.execute("select id from customer")
39             id_store = cur.fetchall()
40             # id_cust = cur.fetchall()
41             take = int(input("Enter ID: "))
42             print(id_store)
43             for id_search in (0,5):
44                 if (take == id_store[id_search][0]):
```

041

(02)

File - G:\PG\backup\_for project\database\02.work - with the database.py

```
42         print ("data for customer - ",take)
43         cur.execute("select id,name,room_id,
44                     comments,grade from customer where id = {0}".format(take))
45         print_data = cur.fetchall()
46         if print_data!=None:
47             print("Customer DATA = ",
48                  print_data[0])
49         else:
50             print("Record not Found")
51             break
52
53         elif (inc == "ID"):
54             take = input ("Enter Name: ")
55             print(take)
56             cur.execute('select id from customer where
57                         name = "{0}"'.format(take))
58             data_p = cur.fetchall()
59             if (data_p!=None):
60                 print("Customer ID = ",print_data[0])
61             else:
62                 print("Record not Found")
63             elif (inc == "NAME"):
64                 print("NAME")
65             else:
66                 choici = input("invalid value. Try again?(Y/N
67 ")
68             if ( choici == "N"):
69                 print("System Exit\n\nThank You.")
70                 sys.exit()
71
72             cur.execute("select emp_id from employee")
73             idkeys = cur.fetchall()
74             cur.execute("select emp_pass from employee")
75             passkeys = cur.fetchall()
76
77             print("System Ready.")
78
79             print("\n\nMURAGAPA HOTEL AND SYSTEMS LTD: ")
80             ide = int(input("Employee ID: "))
81             passe = int(input("Password: "))
82
83             for ids in range(0,3):
84                 if (ide == idkeys[ids][0]):
85                     idf = ids
86                     break
87
88             if (passe == passkeys[idf][0]):
```

40

03

File - G:\PG\backup\_for project\database\02. work - with the database.py

```
84     print("Success taking you to customer data")
85     cust(com,cur)
86     #reak
87 else:
88     print("invalid pass")
89 # else:
90 #     print("hmm somethings wrong")
91
92
93 # extracting emp data from db
94
```

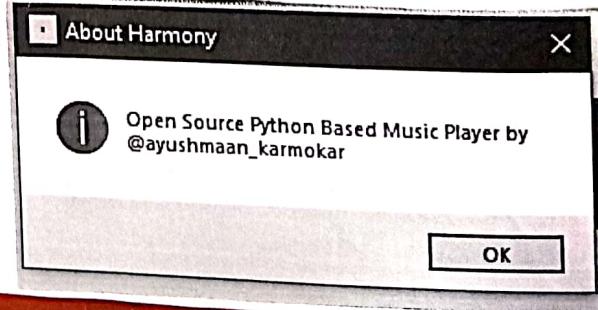
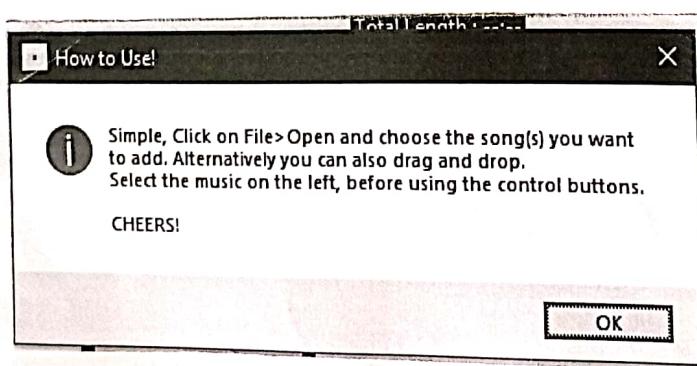
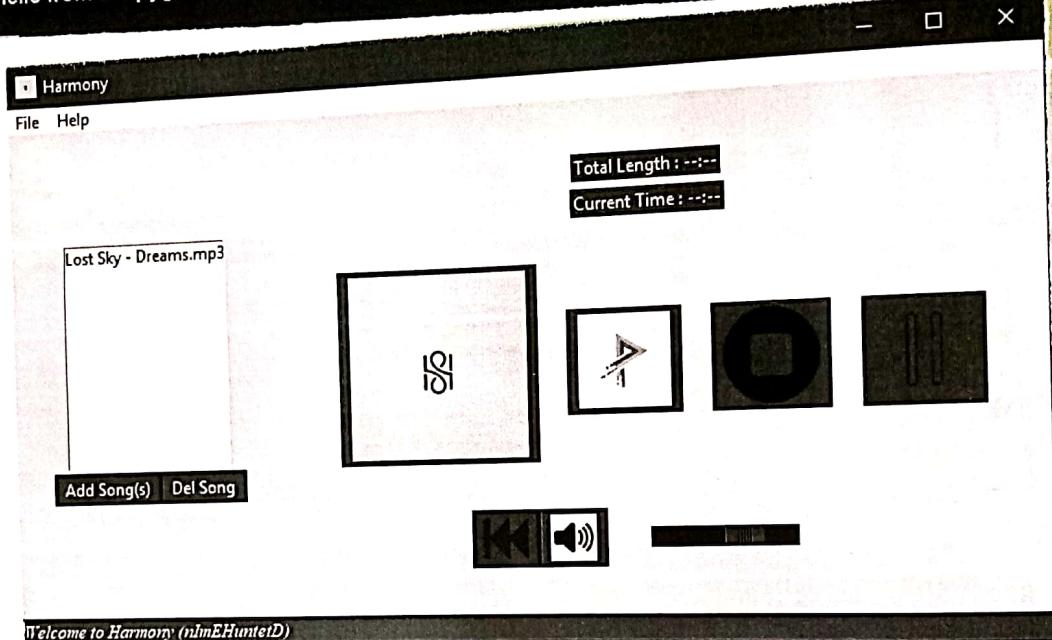
## Output :

File - 02. work - with the database

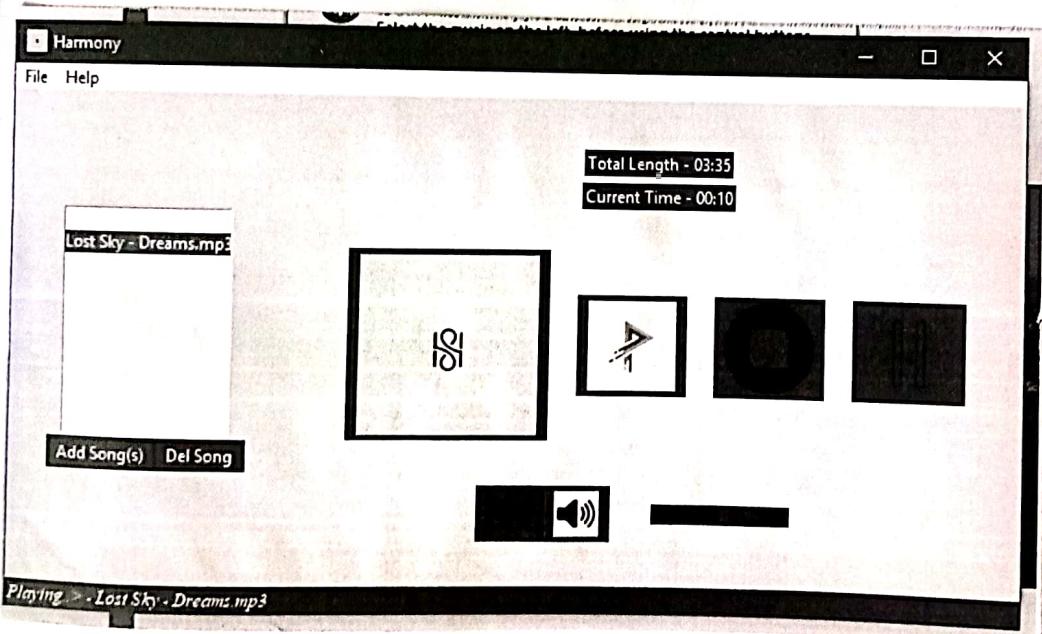
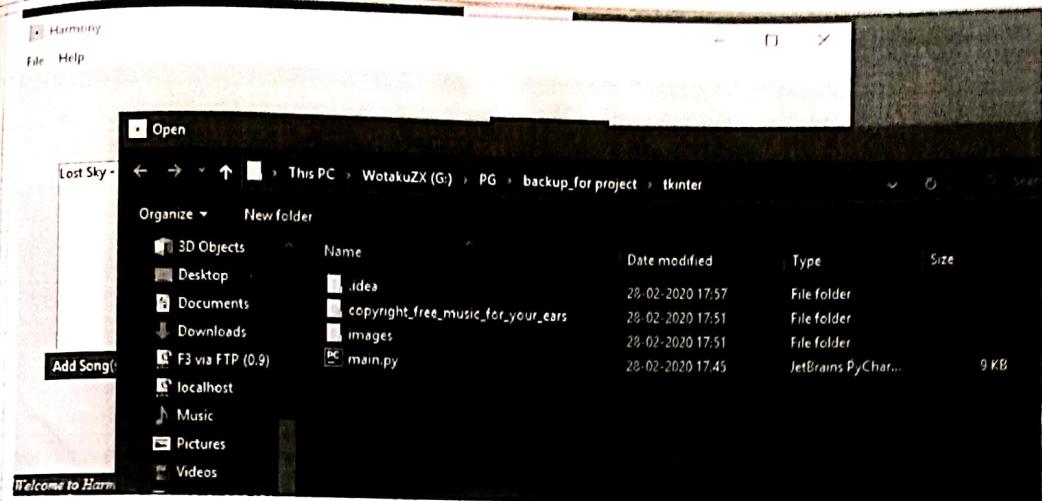
```
1 C:\Users\ayush\AppData\Local\Programs\Python\Python37\  
python.exe "G:/PG/backup_for project/database/02. work -  
with the database.py"  
2 System Ready.  
3  
4  
5 MURAGAPA HOTEL AND SYSTEMS LTD:  
6 Employee ID: 12  
7 Password: 3245  
8 Success taking you to customer data  
9 Do you want customer data? or LOGOUT? (Y/N)Y  
10 What would you like to seach FOR? (ID/NAME/ALL/ROOT)ID  
11 Enter Name: Ravi P  
12 Ravi P  
13 Customer ID = 10001  
14 What would you like to seach FOR? (ID/NAME/ALL/ROOT)ALL  
15 Enter ID: 10001  
16 [(10001,), (10002,), (10003,), (10004,), (10005,)]  
17 data for customer - 10001  
18 Customer DATA = (10001, 'Ravi P', 3101, 'likes roses', 'B  
-')  
19 What would you like to seach FOR? (ID/NAME/ALL/ROOT)ROOT  
20 root environment no allowed  
21 you have been reported  
22  
23 Process finished with exit code 0  
24
```

# Output -

```
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Python38/test.py =====
>>> ===== RESTART: G:/PGIbackup_for project/tkinter/main.py =====
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
```



047



047