

INPUT:-

write mode

~~a =~~

```
a = open("apk.txt", "w")
a.write("Deepak" + " Manya")
a.close()
```

INPUT:-

read()

~~b =~~ a = open("apk.txt", "r")

b = a.read()

print("The output of read mode is:", b)

a.close()

OUTPUT:-

```
>>>('The output of read mode is:', 'Deepak Manya')
```

INPUT:-

readline()

a = open("apk.txt", "r")

c = a.readline()

print("The output of readline mode is:", c)

a.close()

OUTPUT:-

```
>>>('The output of readline mode is:', 'Deepak Manya')
```

INPUT:-

readlines()

a = open("apk.txt", "r")

d = a.readlines()

print("The output of readlines is:", d)

a.close()

Output:-

```
>>>('The output of readlines is:', 'Deepak Manya')
```

Aim:- Demonstrate the use of different file accessing modes, different attributes and methods.

Algorithm:-

Step1:- START

Step2:- Create a file object and using write mode, add some content to the file and close it.

Step3:- Now open the file in different read modes like read(), readline(), ~~is~~ readlines() and store the output of these modes in ^{some} variables.

Step4:- Display those value of the variable and then close it.

Step5:- Now, again open the file and find the name, mode, its state/mode and softspace by assigning it to a variable.

Step6:- Display the value of the variable and then close the file.

Step7:- Now open the file in append mode and write some statements to append it to the file.

Step8:- Display the output of the append mode and close the file.

File

Step 9:- Open the file in read mode and perform the tell method and assign it to a variable. Display the value of tell variable.

Step 10:- Close the file.

Step 11:- Now, open the file ^{in read mode} and using seek mode, display the value.

Step 12:- close the file.

Step 13:- END.

INPUT:-

```
#file attributes
a = open("apk.txt", "r")
b = a.name
print("Name of file:", b)
c = a.closed
print(c)
d = a.mode
print("File mode:", d)
e = a.softspace
print("softspace:", e)
```

018

OUTPUT:-

```
>>> Name of file: apk.txt
>>> True
>>> File mode: r
>>> Softspace: 0.
```

INPUT :-

```
#append mode
a = open("apk.txt", "a")
a.write("This is append mode")
a.close()
a = open("apk.txt", "r")
b = a.read()
print("Output of append:", b)
a.close()
```

Jn
3/6/17

OUTPUT:-

```
>>> Output of append: 'Deepak Manya',
>>> This is append mode'
```

INPUT:-

```
#tell #seek
a = open("apk.txt", "r")
b = a.tell()
c = a.read()
print("tell():", b)
a.seek(10, 0)
print("seek:", c)
a.close()
c = a.tell()
print("tell:", c)
```

OUTPUT:-

```
tell(): 0
seek: 0
tell: 10L.
```

INPUT:-

```
#iter()  
#next()
```

```
a = ("Deepak", "Mawya", "1724")
```

```
b = iter(a)
```

```
print(next(b))
```

```
print(next(b))
```

OUTPUT:-

```
>>> Deepak
```

```
>>> Mawya.
```

INPUT:-

```
#iterusingloop
```

```
a = ("Deepak", "Mawya", "1724")
```

```
for b in a:
```

```
    print(b)
```

OUTPUT:-

```
>>> Deepak
```

```
    Mawya
```

```
    1724.
```

INPUT:-

```
#square
```

```
#cube
```

```
def sgr(x):
```

```
    x = x ** 2
```

```
    return x
```

```
def cubl(x):
```

```
    x = x ** 3
```

```
    return x
```

```
a = [sqr, cubl]
```

```
for i in range(3):
```

```
    value = list(map(lambda x: x(i), a))
```

```
    print(value)
```

OUTPUT:-

```
>>> [0, 0]
```

```
[1, 1]
```

```
[4, 8]
```

PRACTICAL-2

Aim:- To demonstrate the use of iterable and
iterators.

Step 0:- START.

Step 1:- Create a tuple with elements that you want to iterate. On every iter and next keyword used, we will get the next iterating element of the tuple.

Step 2:- Instead of using iter and next method repeatedly we can use the iter() and next() method in for loop. A variable has to be declared in for loop for iteration.

Step 3:- Define a function with a parameter for finding the square of the number. Again, define a function which will have a parameter f() to find the cube of the number.

Step 4:- Now call both the functions outside the scope of the functions declared.

Step 5:- ^{Now,} Use the map method in for conditional and declare a lambda function and print the output.

Step 6:- Declare a variable and give some elements to it and then print the variable.

Step 7:- Define a function named having a single parameter.

~~Step~~

Step 8:- Under the scope of this function if the number divided by 2 gives remainder 2, ~~return~~ print the number as even. Else print the number as odd.

Step 9:- Now print the all elements of list and as even or odd using map function.

Step 10:- END

INPUT:-

```
# even  
# odd  
n = [0, 4, 5, 7, 9, 11, 13, 15, 20, 4, 25]
```

Q20

```
n = list(map(lambda x: x % 2 == n))
```

```
print(n)
```

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

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

```
        return "even"
```

```
    else:
```

```
        return "odd"
```

```
d = list(map(even, listnum))
```

```
print(d)
```

OUTPUT:-

```
[0, 4, 0, 2, 4, 1, 3, 0, 0, 4, 0]
```

```
{'even', 'even', 'even', 'even', 'even', 'odd', 'odd', 'even',
```

```
'even', 'even', 'even'}
```

DM
10/12

#1} INPUT:-

try:

```
f = open("dpk.txt", "w")
f.write("Deepak" + " Manya")
f.write("Roll No.: 1724")
f.close()
f = open("dpk.txt", "r")
d = f.readline()
print(d)
f.close()
```

except IOError:

```
    print("File operation successful")
```

else:

```
    print("File operation successful")
```

OUTPUT:-

???

Deepak Manya

Roll No.: 1724

file operation successful.

#2} INPUT:-

try:

```
n = int(input("Enter something:"))
```

except ValueError:

~~print("Please enter an integer value!")~~

else:

~~print("You just entered:", n)~~

OUTPUT:-

???

Enter something: 1

You just entered: 1

???

Enter something: a

Please enter an integer value!

PRACTICAL-3

exception

Aim:- To demonstrate exception handling in python.

Algorithm :-

#1) Step1:- START.

Step2:- In the try block, open a file in write mode. Add some content to it. Now and close the file.

Step3:- Now, again open the file in read mode. Read the content of the file and display it.

Step4:- In the case of IOError, print the file operation as unsuccessful. Else print the above file operation as successful.

Step5:- STOP.

#2) Step1:- START.

Step2:- In the try block, accept an integer value from the user.

Step3:- In case the user gives an string value, throw an ValueError exception and give an appropriate message.

Step4:- Else print the given value.

Step5:- END.

ISS

#3) Step1:- START

Step2:- In the try block, accept an integer value from the user and assign it to a variable. Initialize another variable with any random number.

Step3:- Throw an exception of value Error and print appropriate message. Throw one more exception of zero Division Error and print an appropriate message.

Step4:- Else print the value of the division.

Step5:- END

#4) Step1:- START

Step2:- In the try block, initialise a variable with value 1. Accept an integer value from user and assign it to variable. Print any expression of your choice. Print the value of the division.

Step3:- Throw an zerodivision exception if the user gives 0 as input. Throw another exception of type error for printing invalid expression.

Step4:- END

#3) INPUT:-

*try:

a=1

b = int(input("Enter a number:"))

except ValueError:

 print("Enter an integer number!")

except ZeroDivisionError:

 print("You cannot divide a number by zero!"))

else:

 print(a/b)

OUTPUT:-

>>> Enter a number: a

Enter an integer number.

>>> Enter a number: 0

You cannot divide a number by zero!

>>> Enter a number: 10

0.1

#4) INPUT:-

*try:

a=1

b = int(input("Enter a number:"))

print(a/b)

~~print('10' + 10)~~

except ZeroDivisionError:

 print("You cannot divide a number by zero!"))

try:

 print('10' + 10)

~~except TypeError:~~

~~print("You cannot print such expression!"))~~

OUTPUT:

>>> Enter a number: 0

You cannot divide a number by zero!

You cannot print such expression.

>>> Enter a number: 5

0.2

You cannot print such expression!

022

Q #5) INPUT:-

```
try:  
    a = int(input("Enter a number:"))  
    if a == 0:  
        raise Exception("Enter a valid  
        number")  
    print("You just entered:", a)  
finally:  
    print("This is finally clause!")
```

OUTPUT:-

```
>>>  
Enter a number: 0  
This is finally clause!  
Exception: Enter a valid number.
```

```
>>>  
Enter a number: 45  
You just entered: 45  
this is finally clause!
```

#5) Step1:- START

Step2:- In the try block, accept an integer value from user and assign it to a variable.

Step3:- Then If the given number is less than or equal to zero, It raise an exception and print an appropriate message. Print the given value.

Step4:- Use the finally clause and print appropriate message to show execution of finally clause.

Step5:- END.

Jm
2/11

Topic :- Regular Expressions

Algorithm :-

#1} Step1:- START

Step2:- Import the re module and declare the pattern and the sequence. Declare an argument to the sequence.

Step3:- Match the required word with the sequence. If the match is found, print `match found` else print `match not found`.

Step5:- STOP.

#2} Step1:- START

Step2:- Import re module. Now, declare the pattern with literal and meta character. Declare some string value. Use the `.findall` to print only the integer values.

Step3:- STOP.

#3} Step1:- START

Step2:- Use the above program and by the using `split` method.

Step3:- STOP.

#17 INPUT:-

```
import re  
pattern = r"FYCS"  
seq = "FYCS represents computer science"  
if re.match(pattern, sequence):  
    print("Pattern Matched")  
else:  
    print("Pattern Unmatched")
```

024

OUTPUT:-

Pattern Matched.

#24 INPUT :-

```
import re  
pattern = r'\d+'  
str = 'Deepak 9659, Manya 2065, Dpk 9459'  
otp = re.findall(pattern, str)  
print(otp)
```

OUTPUT:-

{'9659', '2065', '9459'}

#3) INPUT:-

~~```
import re
pattern = r'\d+'
str = 'Deepak 9659, Manya 2065, Dpk 9459'
otp = re.split(pattern, str)
print(otp)
```~~

OUTPUT:-

{'Deepak', 'Manya', 'Dpk'}

#4) INPUT:-

```
import re
str='Deepak . Maurya'
pattern=r'\st'
replace=''
d=re.sub(pattern, replace, str)
print(d)
```

OUTPUT:-

DeepakMaurya.

#5) INPUT:-

```
import re
seq='Python is an interesting programming language.'
v=re.search('A Python', seq)
print(v)v1=v.group()
print(v1)
```

OUTPUT:-

Python.

#6) INPUT:-

```
#import re
l1=['8767280300','9769529393','9892027269']
for value in l1:
 if re.match(r'^[8-9]\d{12}\d{3}$', value or
 len(value) == 10):
 print("Pattern Matched")
 else:
 print("Pattern Unmatched")
```

OUTPUT:-

Pattern Matched

Pattern Matched

Pattern Matched -

#4) Step1:- START.

Step2:- Import re module. Define a string of your choice. Define the pattern and define replace with empty value.

Step3:- Use the sub method to remove the black spaces and assign it to the variable.

Step4:- END.

#5) Step1:- START.

Step2:- Import re module and define a string and assign it to the sequence. Search for any word among the string and print it.

Step3:- END.

#6) Step1:- START

Step2:- Import re module. Declare a list of numbers. Use the if conditional statement to check whether the numbers start with number 8 or 9 using match method.

Step3:- If matched, print appropriate statement else print no matches found. Assign the conditional statement to the loop.

Step4:- END.

CSO

#7} Step 1:- START.

Step 2:- Import re module and write a sentence and assign it to a variable. Use the.findall method to check for the words starting with vowels and print it.

Step 3:- END.

#8} Step 1:- START.

Step 2:- Import re module and write some email addresses and assign it to a variable. Using the.findall method, display the domain name & host name.

Step 3:- END.

#7) INPUT:-

```
import re
sen='Money is everything'
output=re.findall(r'\b[aeiouAEIOU]+\w+', sen)
print(output)
```

026

OUTPUT:-

```
['is', 'everything']
```

#8) INPUT:-

```
import re
seq='dpk9701@gmail.com', 'pyl724attex@gmail.com'
pattern=r'{\w+.-}t{\w+.-}'
output=re.findall(pattern, seq)
print(output)
```

OUTPUT:-

```
['dpk9701', 'gmail.com', 'pyl724attex', 'gmail.com']
```

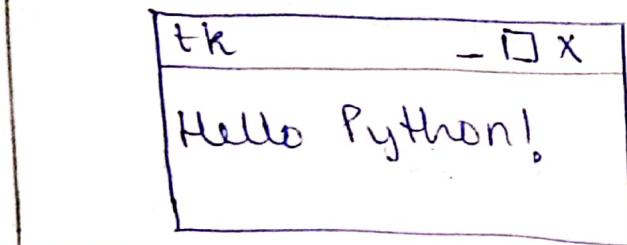
✓ Jan 21/2023

#1&gt;

INPUT:-

```
from tkinter import *
root = Tk()
l = Label(text="Hello Python!")
l.pack(side=LEFT)
root.mainloop()
```

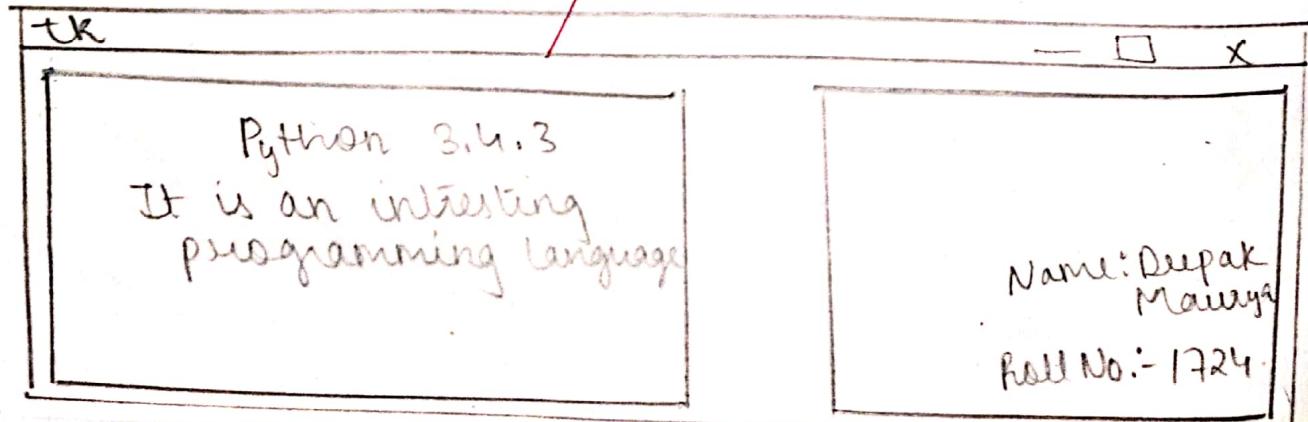
OUTPUT:-



#2&gt; INPUT:-

```
from tkinter import *
root = Tk()
l1 = Label(text="Python 3.4.3 is an interesting
programming language", bg="black", fg="red")
l1.pack(side=LEFT, ipadx=35)
l2 = Label(text="Name: Deepak Maurya in Roll No: 1724",
 bg="black", fg="green")
l2.pack(side=RIGHT, ipady=30)
root.mainloop()
```

OUTPUT:-



PRACTICAL-5

Aim:- Gui components:

Step1:- START

Step2:-

#1) Use the tkinter library for importing the relevant components. Define the parent window.

Step3:- Define the label and enter some text into it and assign it to a variable. Now pack it to the left side. Call the mainloop function.

Step4:- END.

#2) Step1:- START

Dr. M/N

Step2:- Import the tkinter library for importing the relevant components. Define the parent window.

Step3:- Define a label widget. Assign some text to it and give some background and foreground colours. Now pack it to the left side and give some padding to it.

Step4:- Define another label widget. Assign some text to it and give some different background and foreground colours to it. Now pack it to the right side of the parent window and according give some padding to it.

Finally,

Step5:- Call the mainloop method now.

Step6:- END.

PRACTICAL-06

Aim:- GUI components - Label, RadioButton, Button, Scrollbar

Algorithm :-

Step1:- START

Step2:- Import the relevant method `tk` from the Tkinter library. Create an object of the parent window

Step3:- Now define an `select` function which will tell the user the dish ordered by him/her. Use the `label` method to display it.

Step4:- Now using `label` method, give an appropriate title to the listbox.

Step5:- ~~Create~~ Use the an object of `listbox` and use the `insert` method to provide some element to the listbox.

Step6:- Applying the scrollbar to the listbox using `Scrollbar` method.

Step7:- Now create an `radiobutton` for by naming it, give a sufficient value to it, given command as the `select` function defined above so that every time the radio button is triggered, the function must perform its functionality. Give some direction to it in `pack` method every time.

Step8:- Similarly create `radiobuttons` for each element of the listbox.

INPUT:-

028

```
from tkinter import *
root = Tk()
def select():
 selection = "You just ordered " + str(var.get())
 t1 = Label(text=selection, bg="white", fg="green")
 t1.pack(side=TOP)
var = StringVar()
label = Label(text="Please select your favorite dish
you want to order given below!")
label.pack(side=TOP)
ll = Listbox()
ll.insert(1, "1. Pizza")
ll.insert(2, "2. Burger")
ll.insert(3, "3. Samosa Pav")
ll.insert(4, "4. Vada Pav")
ll.insert(5, "5. Noodles")
ll.insert(6, "6. Frankie")
ll.insert(7, "7. Dabeli")
ll.insert(8, "8. Chai")
ll.pack(anchor=N, ipadx=15, ipady=15)
s = Scrollbar(ll)
s.config(command=ll.yview)
ll.config(yscrollcommand=s.set)
r1 = Radiobutton(root, text="Option 1", variable=var,
 value="Pizza", command=select)
r1.pack(anchor=N)
r2 = Radiobutton(root, text="Option 2", variable=var,
 value="Burger", command=select)
r2.pack(anchor=N)
r3 = Radiobutton(root, text="Option 3", variable=var,
 value="Samosa Pav", command=select)
r3.pack(anchor=N)
```

## INPUT:-

```
root = Tk()
r1 = Radiobutton(root, text="Option 1", variable=var, value="Vada Pav", command=select)
r1.pack(anchor=N)
r2 = Radiobutton(root, text="Option 2", variable=var, value="Dosa", command=select)
r2.pack(anchor=N)
r3 = Radiobutton(root, text="Option 3", variable=var, value="Idli", command=select)
r3.pack(anchor=N)
r4 = Radiobutton(root, text="Option 4", variable=var, value="Vada Pan", command=select)
r4.pack(anchor=N)
r5 = Radiobutton(root, text="Option 5", variable=var, value="Noodles", command=select)
r5.pack(anchor=N)
r6 = Radiobutton(root, text="Option 6", variable=var, value="Frankie", command=select)
r6.pack(anchor=N)
r7 = Radiobutton(root, text="Option 7", variable=var, value="Dabeli", command=select)
r7.pack(anchor=N)
r8 = Radiobutton(root, text="Option 8", variable=var, value="Chai", command=select)
r8.pack(anchor=N)

def finish():
 quit()

b1 = Button(text="QUIT", bg="white", fg="green", command=finish)
b1.pack(side=TOP)
root.mainloop()
```

## OUTPUT

Step 9:- define a function `finish` which will perform the quit operation.

Step 10:- Create a button object , give it background and foreground colour to it and call the `finish` function in command to give the button to perform functionality of quit operation . finally call the mainloop method .

Step 11:- END .

OUTPUT:-

|                                                                                                                                                                                                                                                                                         |  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Please select your favourite dish to order given below!                                                                                                                                                                                                                                 |  |
| 1. Pizza<br>2. Burger<br>3. Samosa Pav                                                                                                                                                                                                                                                  |  |
| <input type="radio"/> Option 1<br><input type="radio"/> Option 2<br><input type="radio"/> Option 3<br><input type="radio"/> Option 4<br><input type="radio"/> Option 5<br><input type="radio"/> Option 6<br><input type="radio"/> Option 7<br><input checked="" type="radio"/> Option 8 |  |
| <b>QUIT</b>                                                                                                                                                                                                                                                                             |  |
| You just ordered Pizza                                                                                                                                                                                                                                                                  |  |
| You just ordered Samosa Pav                                                                                                                                                                                                                                                             |  |
| You just ordered Noodles                                                                                                                                                                                                                                                                |  |
| You just ordered Chai                                                                                                                                                                                                                                                                   |  |

Aim:- GUI components - Relief, Messagebox, and traversing over windows.

Algorithm :-

Step 0:- START.

Step 1:- Import the relevant method from Tkinter library. Create an object of the parent window.

Step 2:- Define a finish that will perform quit operation.

Step 3:- Define another function which will consist of the second window and its object.

Step 4:- Create an object for the second window. Define its geometry.

Step 5:- Add a label to the second window and show the label will represent the second window.

Step 6:- Define a for function of withdraw which will withdraw the second window on called.

Step 7:- Define a button of back and add the function of withdraw. Add some attributes to it. Use the relief attribute to give shape to the button and place it on second window.

INPUT:-

from tkinter import \*

root = Tk()

root.geometry("500x500")

def finish():  
 quit()

def w1():  
 s = Tk()  
 s.geometry("500x500")  
 label(s, text="This is the second window!", bg="white", fg="black").pack(side=TOP)  
 def withdraw():  
 s.withdraw()  
 Button(s, text="BACK", bg="white", fg="green", command=withdraw, relief=RIDGE).pack(side=TOP)  
 Button(s, text="QUIT", bg="white", fg="green", command=finish, relief=SUNKEN).pack(side=TOP)  
 s.mainloop()

def msg():  
 messagebox.askyesno("Warning", "Are you sure  
 you want to execute the program?")

m = msg()

label(text="This is the first window", bg="white", fg="black").pack(side=TOP)

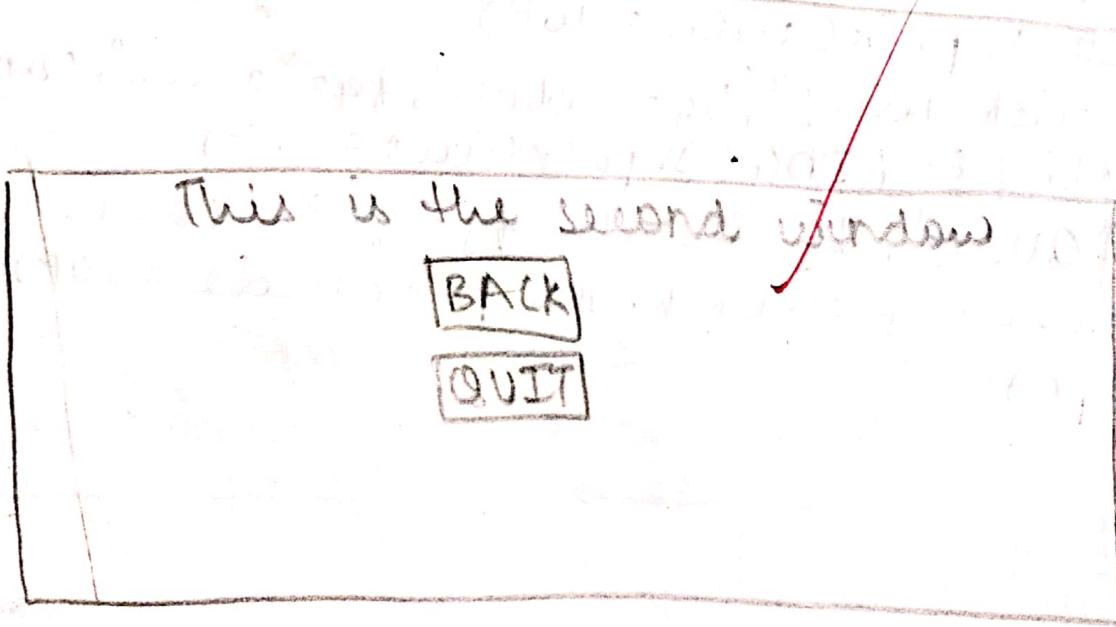
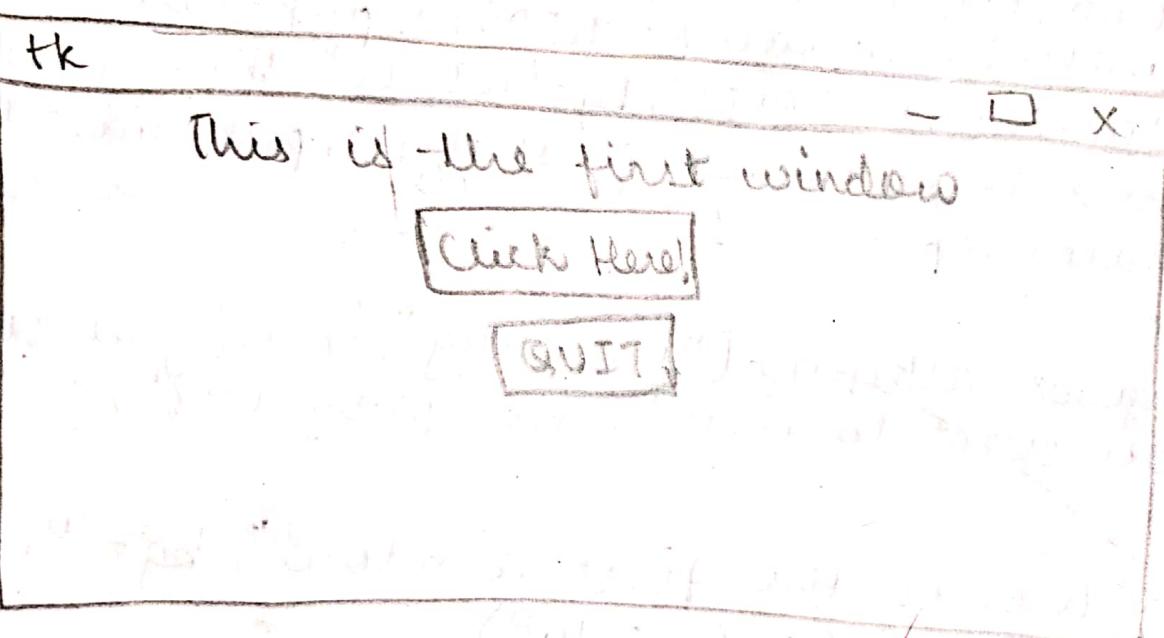
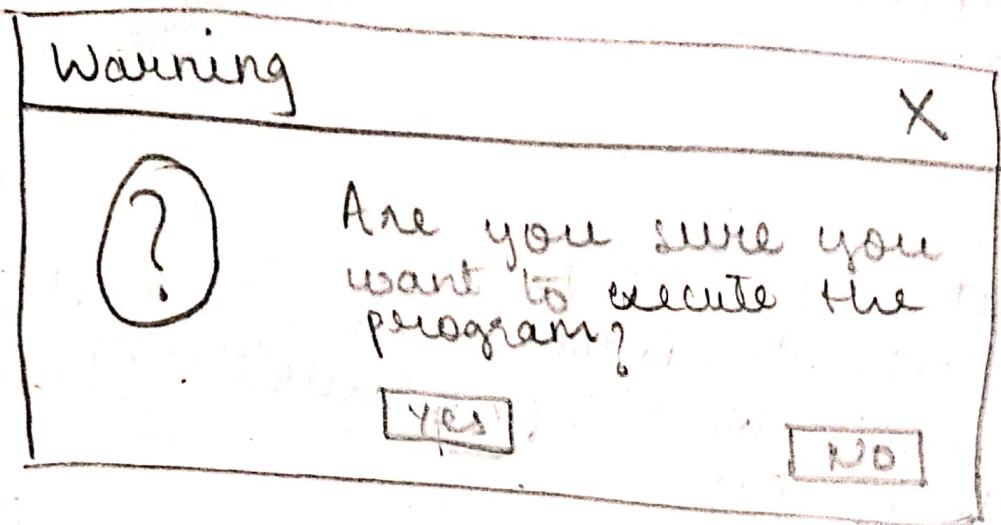
Button(text="Click Here!", bg="white", fg="green", command=w1, relief=RIDGE).pack(side=TOP)

Button(text="QUIT", bg="white", fg="green", command=finish, relief=SUNKEN).pack(side=TOP)

root.mainloop()

## OUTPUT:-

0890



Step 8:- Define another window and place it on the second window. Call the finish() function to quit the program on tapped. Add some attributes to it. Call the mainloop method for second window.

Step 9:- Define a messagebox to ask yes/no and give the title and appropriate message to it.

Step 10:- Call the messagebox function. Now create a label to give heading as the first window.

Step 11:- Create a button such that on tapped it will go onto the second window.

Step 12:- Create another button of quit and again call the finish function created to perform quit operation. Finally call the mainloop method along with the parent window.

Step 13:- END.

JYOTI

Aim:- GUI components:- PhotoImage, Subsample Image

Algorithm :-

Step 1:- START.

Step 2:- Import the relevant method from tkinter library. Create an object of the parent window.

Step 3:- Define the finish function to perform the quit operation.

Step 4:- Define personal info function to give personal information about yourself.

Step 5:- Define the contact function to give contact information about information.

Step 6:- Define a object of photoimage and using the file attribute give the an appropriate image to display.

Step 7:- Define a frame ~~object~~. Give it height and width. Use the grid method and give row as 1 and column as 0.

Step 8:- Create another frame. Give it certain height and width. Use the grid method and give row as 1 and column as 1.

INPUT:-

032

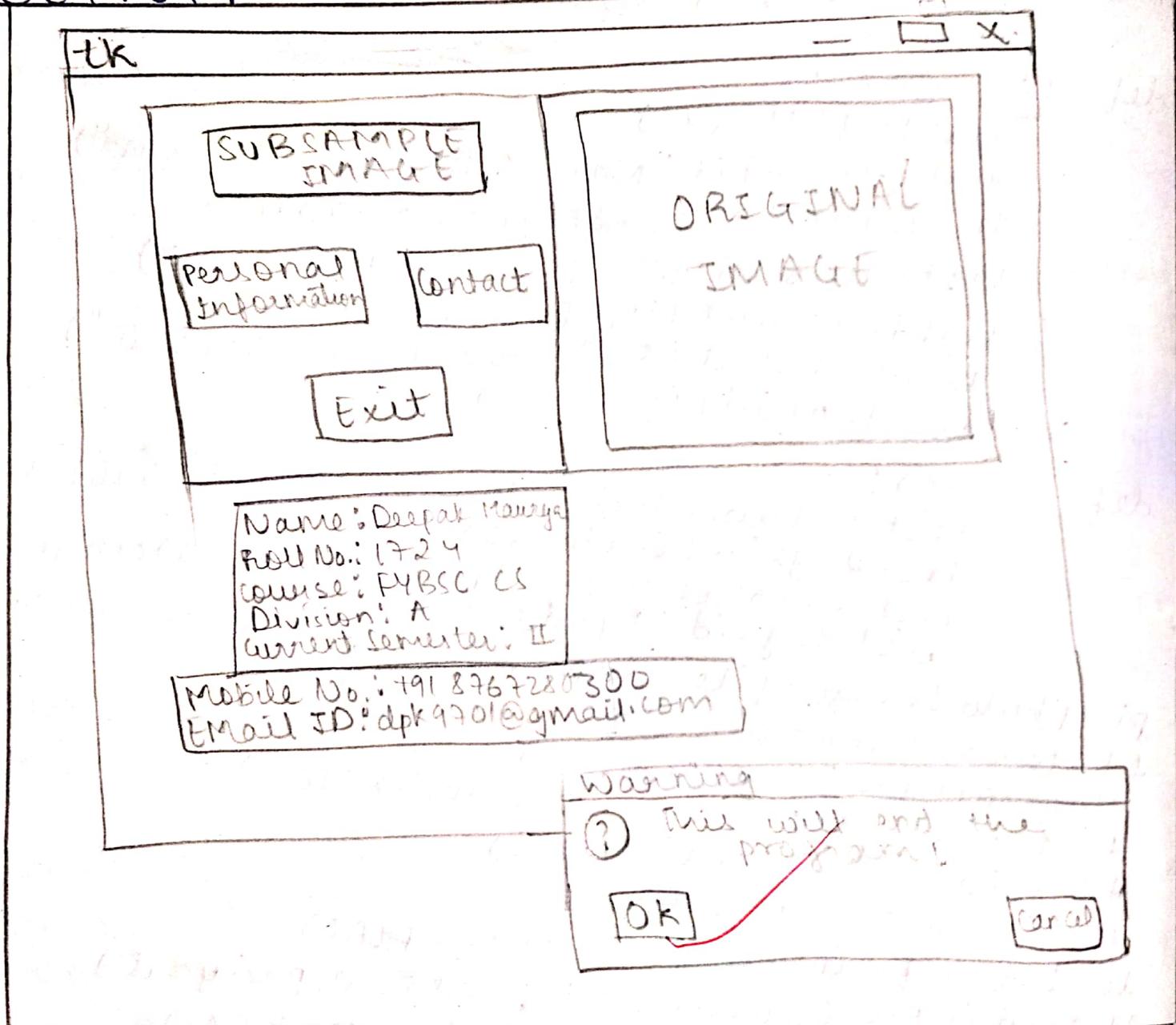
```
from tkinter import *
root = Tk()
root.config(bg="grey")
def finish():
 messagebox.askokcancel("Warning", "This will
 end the program!")
 quit()
def perinfo():
 list1 = Listbox()
 list1.insert(1, "Name: Deepak Manya")
 list1.insert(2, "Ad Roll No.: 1724")
 list1.insert(3, "Course: FYBSC CS")
 list1.insert(4, "Division: A")
 list1.insert(5, "Current semester: II")
 list1.grid(ipadx=30)
def mob():
 list2 = Label(text="Mobile No.: +91 8767280306")
 list2.grid(ipadx=30)
 list3 = Label(text="Email ID: dpk9701@gmail.com")
 list3.grid(ipadx=24)
 p1 = PhotoImage(file="download.gif")
 f1 = Frame(root, height=35, width=5)
 f1.grid(row=1, column=0)
 f1.grid(row=0, column=1, padx=20, pady=15)
 f2 = Frame(root, height=250, width=500)
 f2.grid(row=1, column=1)
 p2 = p1.subsample(8, 4)
 l1 = Label(f1, image=p2, relief=FLAT)
 l1.grid(row=0, column=1, padx=25, pady=10)
 b1 = Button(f1, text="Personal Information",
 relief=SUNKEN, command=perinfo)
 b1.grid(row=1, column=0)
 b2 = Button(f1, text="Contact", relief=SUNKEN,
 command=mob)
 b2.grid(row=1, column=2, padx=5)
```

INPUT:-

\$89  
Exit

```
b3 = Button(f1, text = "Exit", relief=RAISED,
 command = finish)
b3.grid(row=2, column=1, ipadx=15)
root.mainloop()
```

OUTPUT:-



Step 9:- Now create an object of ~~new~~ subsample of the original image.

Step 10:- Create the label ~~as~~ and place it ~~on~~ on the left frame. Use the grid method to give it rows and column and certain padding. Similarly create another label and place it on the right frame. Use the grid method to give it certain paddings.

Step 11:- Now create a button to display personal information. Use the command attribute to call the printo function. Similarly create 2 more button to display contact information and to exit. Call the respective functions on the buttons. Finally call the mainloop method.

Step 12:- finally call the END.

Jan 1-

PRACTICAL-09

Aim :- To demonstrate the use of spinbox, canvas and panned window.

Algorithm :-

Step 1:- START

Step 2:- Import the corresponding library of Tkinter and create the parent window.

Step 3:- Create a panned window <sup>on parent window</sup> and assign it to a variable. Add a label to it firstly.

Create a function p<sup>h</sup> to withdraw the main window.

Step 4:- In the label, insert the text as Panned window 1. Now add this label on the panned window.

Step 5:- Create a p<sup>h</sup> which will contain the second window. Create the parent window of it.

Step 6:- In this p<sup>h</sup>, create another panned window. Create another function to withdraw this window.

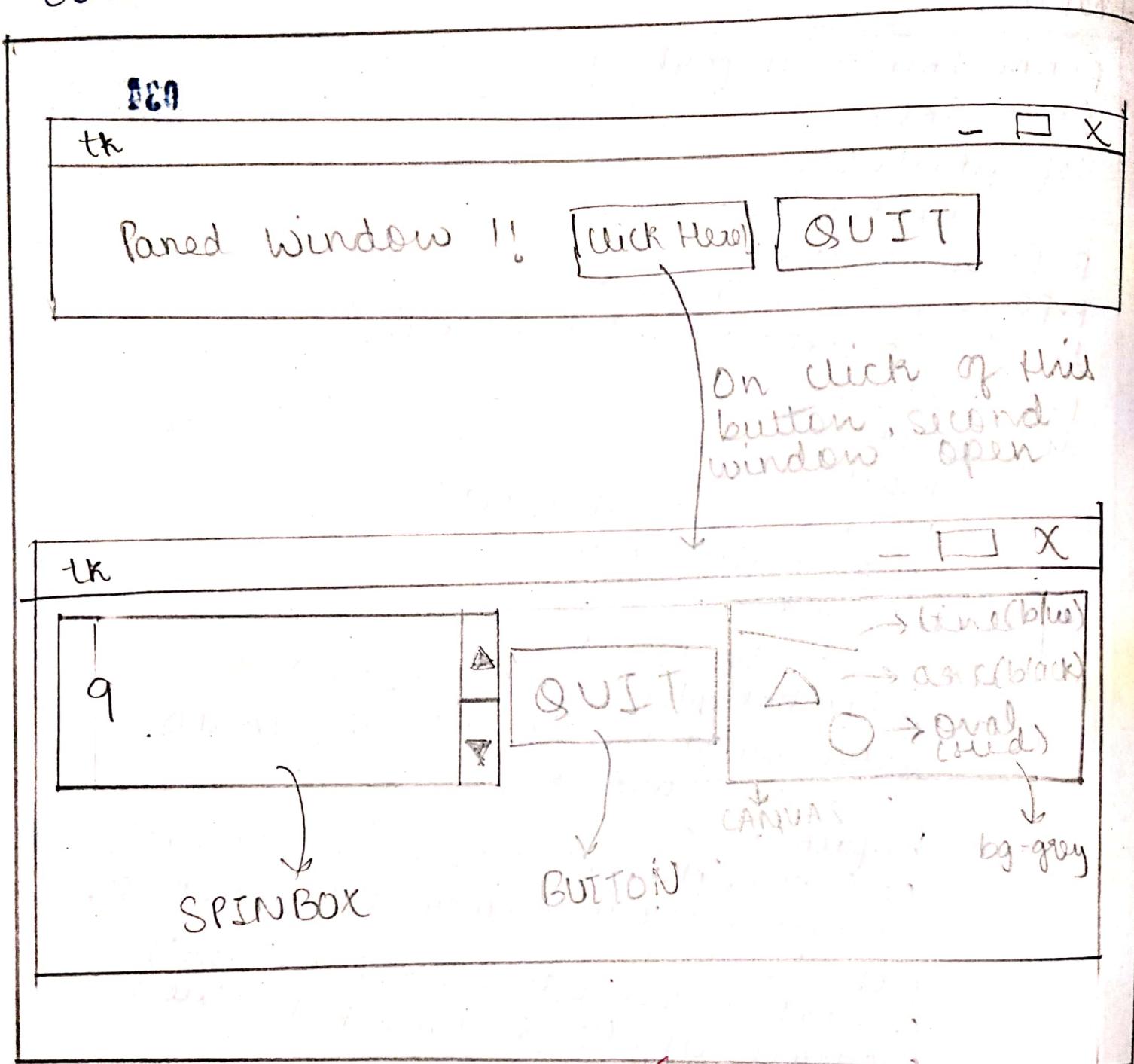
Step 7:- Create a spinbox which will start from 0 and end at 100. Assign it to a variable. Create a button to quit and through command attribute call the withdraw function created in above step.

INPUT:-

```
from Tkinter import *
root = Tk()
def finish():
 root.withdraw()
p = PanedWindow()
p.pack(fill=BOTH, expand=1)
l = Label(p, text="Paned Window 1!")
p.add(l)
def pw2():
 master = Tk()
 p1 = PanedWindow(master)
 p1.pack(fill=BOTH)
 def width():
 master.withdraw()
 s = Spinbox(p1, from_=0, to_=100)
 b1 = Button(p1, text="QUIT!", relief=RAISED,
 command=width)
 b1.pack()
 c = Canvas(p1, height=100, width=200, bg="grey")
 c.create_arc(10, 20, 30, 40, start=10, extent=50,
 fill="black")
 c.create_oval(40, 50, 60, 70, fill="red")
 c.create_line(10, 15, 30, 20, fill="blue")
 c.pack()
 p1.add(s)
 p1.add(b1)
 p1.add(c)
 master.mainloop()
 b = Button(p, text="Click Here!", relief=RAISED,
 command=pw2)
 b.pack()
 b2 = Button(p, text="QUIT", relief=RAISED,
 command=finish)
 b2.pack()
 p.add(b)
 p.add(b2)
root.mainloop()
```

034

OUTPUT:-



Step 8:- Now create a canvas which will contain a arc, line and oval. Use create弧 method to create arc, create线 to create a line and create椭圆 to create oval. Give appropriate coordinates to each one of them.

Now,

Step 9:- Add spinbox, button and canvas to the second panned window created and come outside the function.

Step 10:- Create a button and give it some appropriate text to it. Use relief as raised and call the finish function created in step 3 through command attribute.

Step 11:- Create another button through which we can go to the second window. Call the second window function through command attribute. Add both the buttons to the first panned window. Call the mainloop function lastly.

St

Step 12:- END.

Ans 12

369

## PRACTICAL-10

Aim:- To demonstrate the use of database connectivity.

#1} Step 1:- START

Step 2:- Import the dbm library. Use the open method to create the database by specifying the name and corresponding flag.

Step 3:- Use the obj created for accessing the database and its values. Check whether the value given is not none. If none, print appropriate message. Print appropriate message.

Step 4:- END.

INPUT:-

```
>>>import dbm
>>>db=dbm.open(".database","c")
>>>db["name"]="Deepak Maurya"
>>>if db["name"]==None:
 print("Database creation successful!")
else:
 print("Database creation unsuccessful!")
```

Q36

OUTPUT:-

Database creation successful.

INPUT:-

```
import os,sqlite3
c=sqlite3.connect("studd.db")
cur=c.cursor()
cur.execute("create table stud_details (Name CHAR,
 ROLLNO. INT)")
cur.execute("insert into stud_details('Deepak
 Maurya', 1724), ('Ompal Bhati', 1721), ('Abhang
 Mane', 1727), ('Laxman Choudhary', 1725)")
c.commit()
cur.execute("select * from stud-details")
print(cur.fetchall())
c.close()
```

OUTPUT:-

```
[(('Deepak Maurya', 1724), ('Ompal Bhati', 1721), &
 ('Abhang Mane', 1727), ('Laxman Choudhary', 1725))]
```

#2] Step 1:- START

Step 2:- import the os, sqlite3 library.  
Create a database using connect method.

Step 3:- Create a cursor object. Now use the execute method along with cursor object to create table. Define some attributes of the table.

Step 4:- Again use the execute method along with cursor object to insert values. Now commit the database.

Step 5:- Use the execute method along with cursor object to select all the values from the table created. Use the fetchall method to fetch the values from the table and print it. Use the database finally.

Step 6:- END.

Ans 1

Ans  
07/03

SE

## PROJECT

Aim:- To create python project based on GUI widgets and database.

Description :-

This project is on the basis of college ID. This project contains many GUI widget and attributes such as PhotoImage, Label, Entry, Button and many more attributes.

The project contains 3 windows. The first window ask the user to input the details. The second and third window shows the details entered by the user.

# SOURCE CODE

```
from tkinter import *
global root
root=Tk()
root.title("PyProject")
global e1
global e2
global e3
global e4
global e5
global e6
global e7
global e8
global e9
p1=PhotoImage(file="thakurtrusts.gif")
Label(image=p1,relief=FLAT).grid(row=1,columnspan=2,padx=100)
Label(text="Enter your name :",font="ArialBlack 10 bold").grid(row=2,column=0)
e1=Entry(root,width=40,relief=SUNKEN).grid(row=2,column=1)
Label(text="Enter your class:",font="ArialBlack 10 bold").grid(row=3,column=0)
e2=Entry(width=40,relief=SUNKEN).grid(row=3,column=1)
Label(text="Enter your division",font="ArialBlack 10 bold").grid(row=4,column=0)
e3=Entry(width=40,relief=SUNKEN).grid(row=4,column=1)
Label(text="Enter your roll no.:",font="ArialBlack 10 bold").grid(row=5,column=0)
e4=Entry(width=40,relief=SUNKEN).grid(row=5,column=1)
Label(text="Enter your Date of Birth :",font="ArialBlack 10 bold").grid(row=6,column=0)
e5=Entry(width=40,relief=SUNKEN).grid(row=6,column=1)
Label(text="Enter your age :",font="ArialBlack 10 bold").grid(row=7,column=0)
e6=Entry(width=40,relief=SUNKEN).grid(row=7,column=1)
Label(text="Enter your blood group :",font="ArialBlack 10 bold").grid(row=8,column=0)
e7=Entry(width=40,relief=SUNKEN).grid(row=8,column=1)
Label(text="Enter your residential address :",font="ArialBlack 10 bold").grid(row=9,column=0)
e8=Entry(width=40,relief=SUNKEN).grid(row=9,column=1)
Label(text="Enter your telephone no. :",font="ArialBlack 10 bold").grid(row=10,column=0)
e9=Entry(width=40,relief=SUNKEN).grid(row=10,column=1)
Label(text="*Kindly fill all your details properly!",font="ArialBlack 10 bold").grid(row=11,column=0,pady=15)
def w1():
 global master
 master=Toplevel()
 master.title("College ID")
 p2=p1.subsample(3,3)
 p3=PhotoImage(file="download.gif").subsample(2,2)
 p4=PhotoImage(file="dpk.gif").subsample(3,3)
 p5=PhotoImage(file="science.gif").subsample(1,1)
 p6=PhotoImage(file="signature.gif").subsample(5,5)
 l1=Label(master,text="Thakur Educational Trust's(Regd.)",fg="black",font="Script 22")
 l1.grid(row=0,columnspan=3,ipadx=10)
 Label(master,image=p2,relief=FLAT).grid(row=1,column=0,sticky=E)
 l2=Label(master,text="THAKUR COLLEGE OF \nSCIENCE AND COMMERCE",fg="red",font="TimesNewRomanBaltic 25 bold")
 l2.grid(row=1,column=1)
 Label(master,image=p3,relief=FLAT).grid(row=1,column=3,sticky=W)
 l3=Label(master,text="UGC Recognized & Affiliated to University of Mumbai",relief=FLAT,font="CourierNewBaltic 17 bold")
 l3.grid(row=2,columnspan=3)
```

(NAAC Accredited Grade-A, [3rd Cycle] &  
ISO 9001:2015 Certified)", relief=FLAT, font="CourierNewBaltic 16 bold")

Thakur Village, Kandivali(E), Mumbai-400101.

Tel.: 022-2846 2565/ 2887 0627", relief=FLAT, font="CourierNewBaltic 13")

15.grid(row=4, columnspan=3)  
 Label(master, image=p4, relief=FLAT).grid(row=5, rowspan=5, column=0, sticky=E)  
 Label(master, text="2019-2020", fg="midnightblue", relief=FLAT, font="ArialBlack bold").grid(row=5, column=1)  
 Label(master, text="DEGREE COLLEGE", fg="darkturquoise", relief=FLAT, font="ArialBlack 20 bold").grid(row=6, column=1)  
 Label(master, image=p5, relief=FLAT).grid(row=7, rowspan=3, column=1)  
 Label(master).grid(row=10, columnspan=3)  
 Label(master, text="Name : ", relief=FLAT, font="ArialBlack 15").grid(row=11, column=0, stick=W, padx=10)  
 Label(master, text=e1.get(), font="ArialBlack 15 bold", relief=FLAT).grid(row=11, column=0, columnspan=3, sticky=W, padx=80)  
 Label(master, text="Class : ", relief=FLAT, font="ArialBlack 15").grid(row=12, column=0, stick=W, padx=10)  
 Label(master, text=e2.get(), font="ArialBlack 15 bold", relief=FLAT).grid(row=12, column=0, columnspan=3, sticky=W, padx=80)  
 Label(master, text="Division : ", relief=FLAT, font="ArialBlack 15").grid(row=13, column=0, stick=W, padx=10)  
 Label(master, text=e3.get(), font="ArialBlack 15 bold", relief=FLAT).grid(row=13, column=0, columnspan=3, sticky=W, padx=94)  
 Label(master, text="Roll No. : ", relief=FLAT, font="ArialBlack 15 bold", relief=FLAT).grid(row=14, column=0, stick=W, padx=10)  
 Label(master, text=e4.get(), font="ArialBlack 15 bold", relief=FLAT).grid(row=14, column=0, columnspan=3, sticky=W, padx=96)  
 Label(master, text="Principal's Signature", relief=FLAT, font="ArialBlack 15").grid(row=15, column=1, stick=W, ipadx=55)  
 Label(master, image=p6, relief=FLAT).grid(row=15, column=1, padx=250)

Button(master, text="<<<", relief=RAISED, bg="darkturquoise", fg="midnightblue", font="bold", command=master.withdraw).grid(row=16, column=1, sticky=W)

Button(master, text=">>>", relief=RAISED, bg="darkturquoise", fg="midnightblue", font="bold", command=w2).grid(row=16, column=1, sticky=E)

def w2():  
 master1=Toplevel()  
 master1.title("CollegeID")  
 f1=PhotoImage(file="download.gif")  
 Label(master1, image=f1, relief=FLAT).grid(row=0, column=0, columnspan=3)  
 Label(master1, text="Date of Birth : ", relief=FLAT, font="ArialBlack 15").grid(row=1, column=0, stick=W)  
 Label(master1, text=e5.get(), font="ArialBlack 15 bold", relief=FLAT).grid(row=1, column=0, columnspan=3)  
 Label(master1, text="Age : ", relief=FLAT, font="ArialBlack 15").grid(row=1, column=1, stick=W)  
 Label(master1, text=e6.get(), font="ArialBlack 15 bold", relief=FLAT).grid(row=1, column=1, columnspan=2)  
 Label(master1, text="Blood Group. : ", relief=FLAT, font="ArialBlack 15").grid(row=2, column=0, stick=W)  
 Label(master1, text=e7.get(), font="ArialBlack 15 bold", relief=FLAT).grid(row=2, column=0, columnspan=2)  
 Label(master1, text="Residential Address : ", relief=FLAT, font="ArialBlack 15").grid(row=3, column=0, stick=W)  
 Label(master1, text=e8.get(), font="ArialBlack 15 bold", relief=FLAT).grid(row=3, column=0, columnspan=2)  
 Label(master1, text="Tel. : ", relief=FLAT, font="ArialBlack 15").grid(row=4, column=0, stick=W)  
 Label(master1, text=e9.get(), font="ArialBlack 15 bold", relief=FLAT).grid(row=4, column=0, columnspan=2)  
 Label(master1, text="IF FOUND KINDLY POST TO THE COLLEGE ADDRESS", relief=FLAT, font="ArialBlack 20 bold").grid(row=5, columnspan=3, pady=20)

```
master1.withdraw()
master.withdraw()

Button(master1, text="<<<", relief=RAISED, bg="darkturquoise", fg="midnightblue", command=master1.withdraw).grid(row=6, column=0)

Button(master1, text="QUIT", relief=RAISED, bg="darkturquoise", fg="midnightblue", command=finish).grid(row=6, sticky=E, column=0, columnspan=2)
master1.mainloop()

Button(text="QUIT", relief=RAISED, command=quit, bg="darkturquoise", fg="midnightblue").grid(row=12, column=0)
Button(text=">>>", relief=RAISED, command=w1, bg="darkturquoise", fg="midnightblue").grid(row=12, column=1)
mainloop()
```

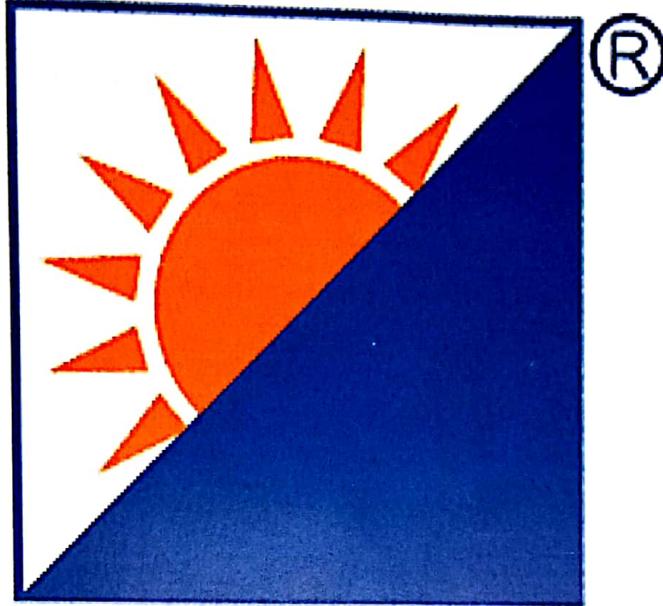
EEG

## OUTPUT

FIRST WINDOW. -



# THAKUR



## TRUSTS

**Enter your name :**

Deepak Maurya

**Enter your class:**

FYBSC CS

**Enter your division**

A

**Enter your roll no.:**

17424

**Enter your Date of Birth :**

09/07/2001

**Enter your age :**

18

**Enter your blood group :**

A-

**Enter your residential address :**

Jay Hanuman Chawl, Shivaji Nagar, Gokuldha

**Enter your telephone no. :**

8767280300

**\*Kindly fill all your details properly!**

**QUIT**

**>>>**

087

SECOND WINDOW:-

College ID



Thakur Educational Trust's(Regd.)

## THAKUR COLLEGE OF SCIENCE AND COMMERCE

tcsc

UGC Recognized & Affiliated to University of Mumbai  
(NAAC Accredited Grade-A,[3rd Cycle] & ISO 9001:2015 Certified)

Thakur Village,Kandivali(E),Mumbai-400101. Tel.:022-2846 2565/ 2887 0627

2019-2020

DEGREE COLLEGE



Name : Deepak Maurya

Class : FYBSC CS

Division : A

Roll No. : 17424

Principal's Signature

<<<

>>>

1A

THIRD WINDOW —

CollegeID



Date of Birth :09/07/2001

Age :18

Blood Group. :A-

Residential Address :Jay Hanuman Chawl, Shivaji Nagar, Gokuldham, Goregaon(East),Mumbai-400063

Tel. :8767280300

**IF FOUND KINDLY POST TO THE COLLEGE ADDRESS**

<<<

QUIT

042