Exam Seat No._____

THAKUR COLLEGE OF SCIENCE & COMMERCE

NAAC Accredited with Grade "A" (3rd Cycle)

THAKUR TRUSTS

ISO 9001 : 2015 Certified

Degree College

Computer Journal

CERTIFICATE

SEMESTER II UID No. _____

Class FYBSC Roll No. 1758 Year 2019-20

This is to certify that the work entered in this journal is the work of Mst. / ~~Ms.~~ CHAUHAN HIMANSHU.K

who has worked for the year 2020 in the Computer Laboratory.

Teacher In-Charge

Head of Department

Date : _____

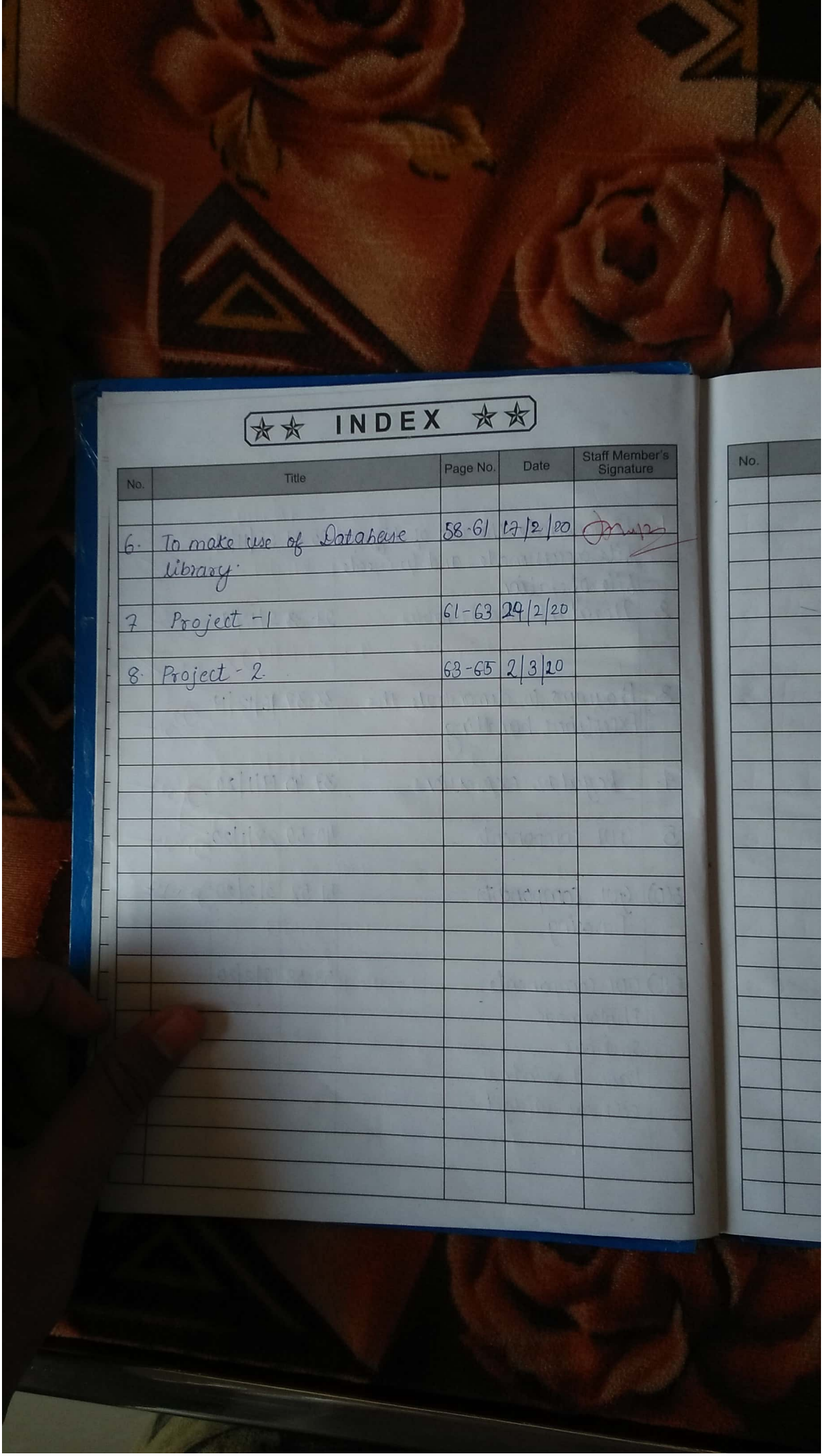
Examiner

Scanned by CamScanner

☆☆ INDEX ☆☆

No.	Title	Page No.	Date	Staff Member's Signature
1.	Demonstrate the use of different file access mode and to create file operation	25-29	25/11/19	
2.	Iterators and Iterables	29-33	2/12/19	
3.	Programs to demonstrate the Exceptions handling	33-37	16/12/19	
4.	Regular expressions.	37-40	19/1/20	
5	GUI components	40-50	27/1/20	
5(D)	GUI components Traversing	51-52	3/2/20	
5(E)	GUI components Photoimage Spinbox Panned window canvas widget	53-58	10/2/20	

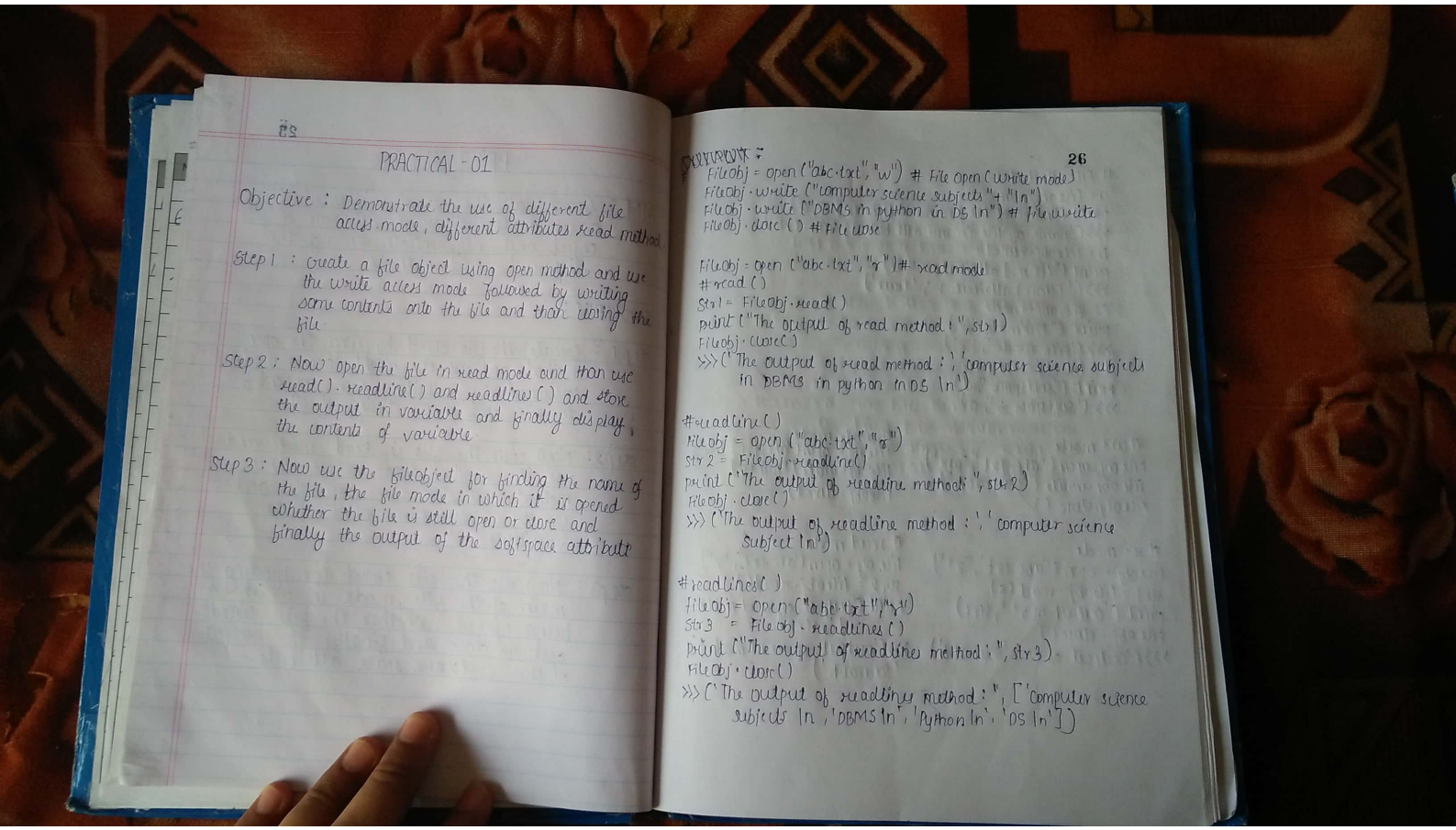
Scanned by CamScanner



★★ INDEX ★★

No.	Title	Page No.	Date	Staff Member's Signature
6.	To make use of Database library.	58-61	17/2/20	
7	Project - 1	61-63	24/2/20	
8.	Project - 2.	63-65	2/3/20	

Scanned by CamScanner



PS

PRACTICAL - 01

Objective : Demonstrate the use of different file access mode, different attributes read method

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

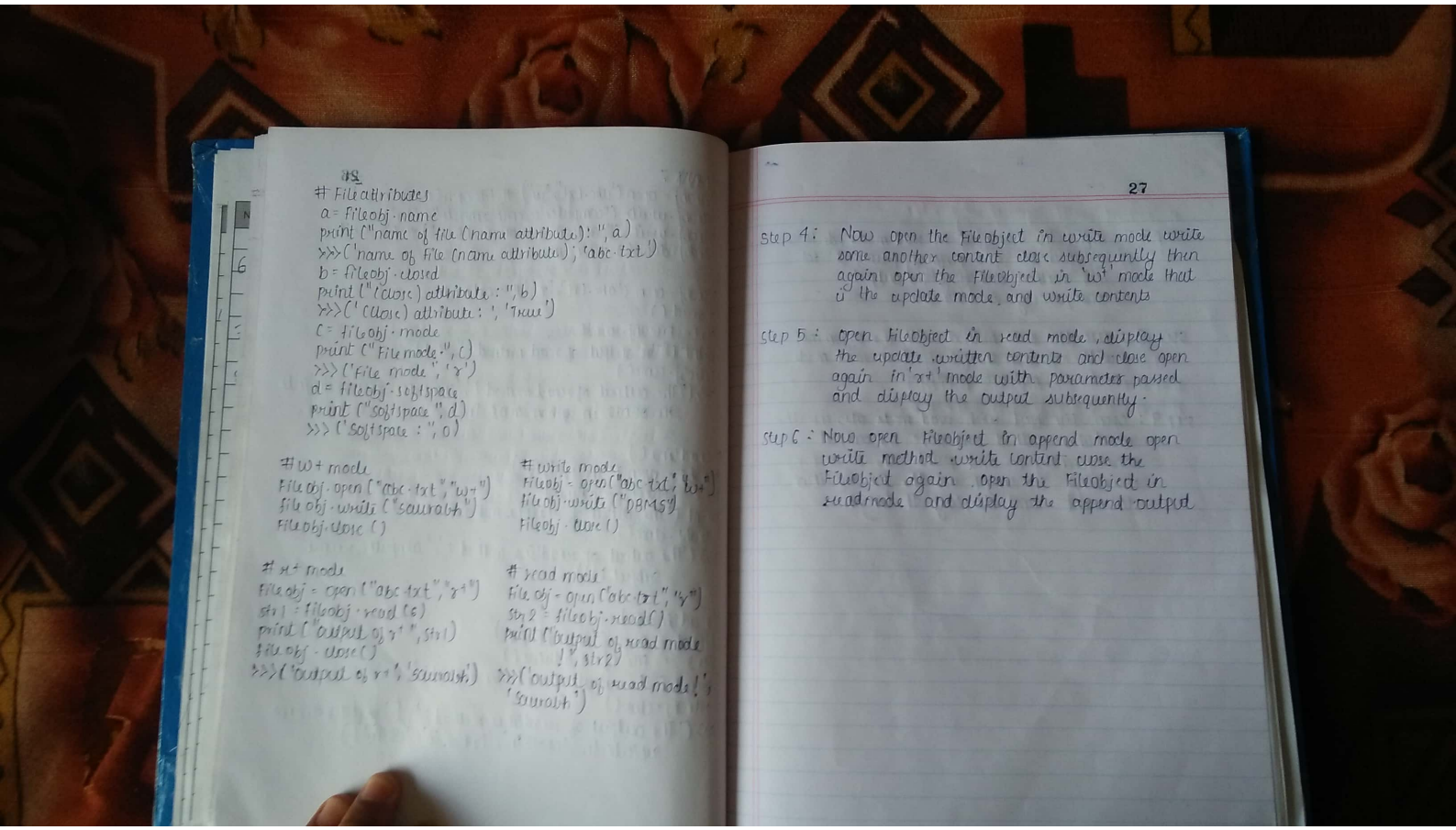
Step 2 : Now open the file in read mode and then use read(), readline() and readlines() and store the output in variable and finally display the contents of variable

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

PROGRAM : 26

```
FileObj = open("abc.txt", "w") # File open (write mode)
FileObj.write("computer science subjects"+"\n")
FileObj.write("DBMS in python in DS \n") # file write
FileObj.close() # File close

FileObj = open("abc.txt", "r")# read mode
#read()
Str1 = FileObj.read()
print("The output of read method :",Str1)
FileObj.close()
>>> ("The output of read method :', 'computer science subjects
    in DBMS in python in DS \n')

#readline()
FileObj = open("abc.txt", "r")
Str2 = FileObj.readline()
print("The output of readline method:", Str2)
FileObj.close()
>>> ("The output of readline method :', 'computer science
    Subject \n")

#readlines()
FileObj = open("abc.txt","r")
Str3 = FileObj.readlines()
print("The output of readline method:", Str3)
FileObj.close()
>>> ("The output of readline method:", ['computer science
    subjects \n', 'DBMS \n', 'Python \n', 'DS \n'])
```

Scanned by CamScanner



```
# File attributes
a = fileobj.name
print ("name of file (name attribute): ", a)
>>> ('name of file (name attribute): ', 'abc.txt')
b = fileobj.closed
print ("(close) attribute: ", b)
>>> ('(close) attribute: ', True)
c = fileobj.mode
print ("File mode:", c)
>>> ('File mode:', 'r')
d = fileobj.softspace
print ("softspace", d)
>>> ('softspace: ', 0)
```

```
#w+ mode
Fileobj = open ("abc.txt", "w+")
fileobj.write ("saurabh")
Fileobj.close ()
```

```
# write mode
Fileobj = open ("abc.txt", 'w')
Fileobj.write ("python")
Fileobj.close ()
```

```
# r+ mode
Fileobj = open ("abc.txt", "r+")
str1 = fileobj.read (6)
print ("output of r+", str1)
fileobj.close()
>>> ('output of r+', 'saurab')
```

```
# read mode
File_obj = open ("abc.txt", "r")
str2 = fileobj.read()
print ("output of read mode !", str2)
>>> ('output of read mode !', 'saurabh')
```

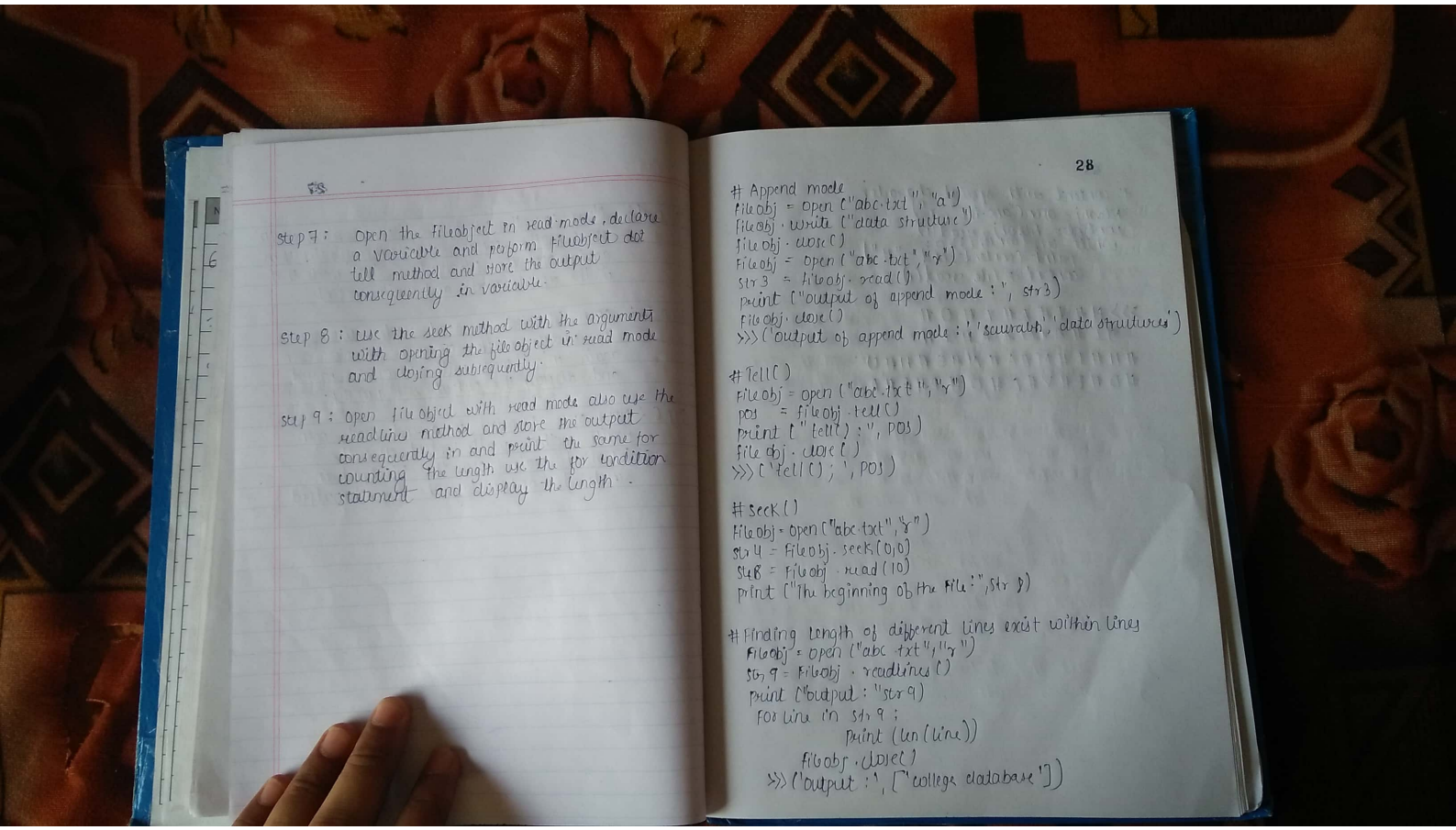
27

step 4: Now open the Fileobject in write mode write some another content close subsequently then again open the Fileobject in 'w+' mode that is the update mode and write contents

step 5: open Fileobject in read mode, display the update written contents and close open again in 'r+' mode with parameter passed and display the output subsequently.

step 6: Now open Fileobject in append mode open write method write content close the Fileobject again open the Fileobject in readmode and display the append output

Scanned by CamScanner



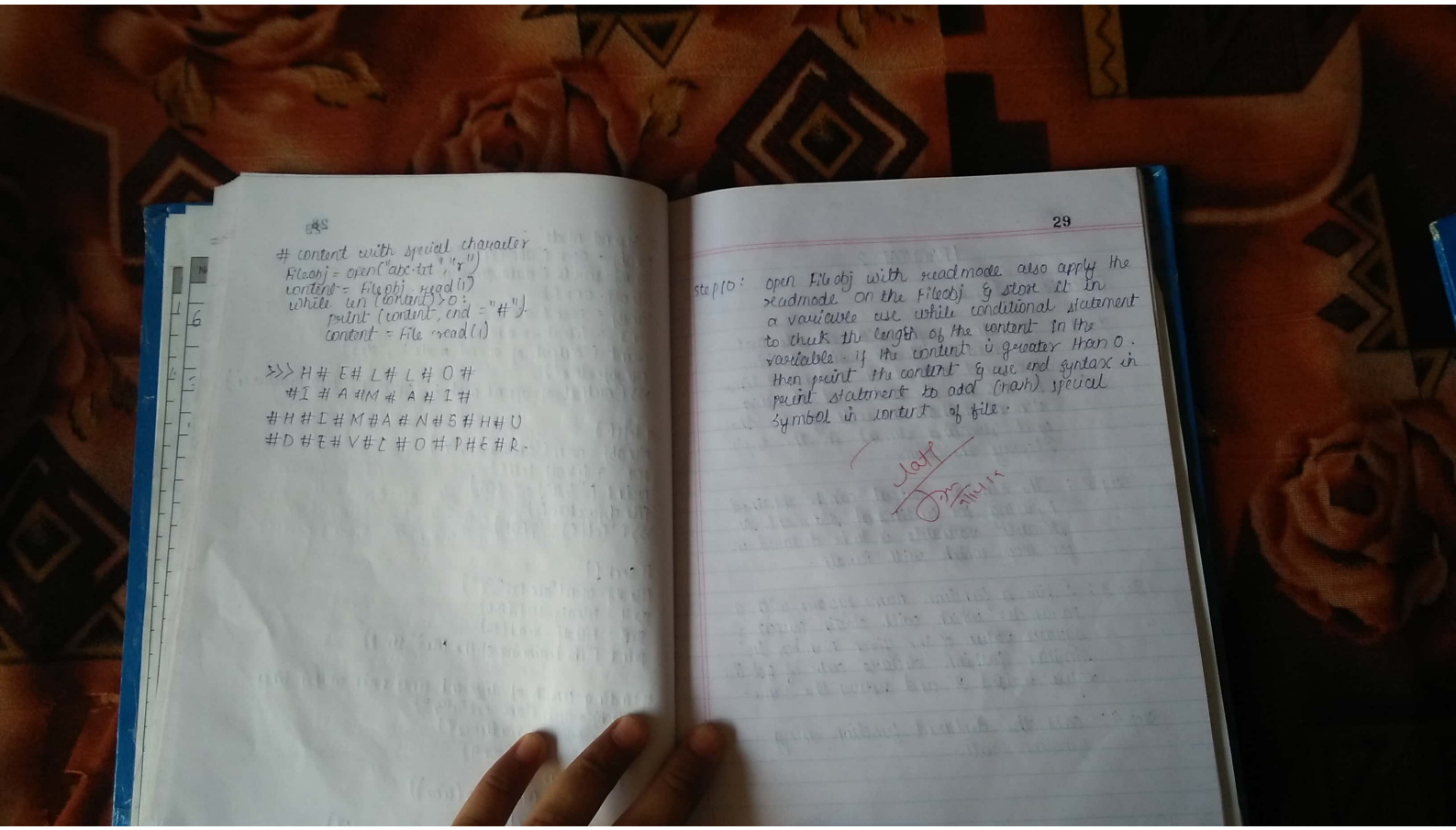
Step 7: Open the fileobject in read mode, declare a variable and perform fileobject.tell method and store the output consequently in variable.

Step 8: use the seek method with the arguments with opening the file object in read mode and closing subsequently.

Step 9: open file object with read mode also use the readline method and store the output consequently in and print the same for counting the length use the for condition statement and display the length.


```
# Append mode
File obj = open ("abc.txt", "a")
File obj . write ("data structure")
file obj . close()
File obj = open ("abc.txt", "r")
str 3 = File obj . read()
print ("output of append mode :", str 3)
File obj . close()
>>> ('output of append mode :', 'saurabh', 'data structures')

# Tell()
File obj = open ("abc.txt", "r")
pos = File obj . tell()
print ("tell() :", pos)
file obj . close()
>>> ('tell() ; ', pos)

# seek()
File obj . open ("abc.txt", "r")
str 4 = File obj . seek(0,0)
str 8 = File obj . read(10)
print ("The beginning of the file:", str 8)

# Finding length of different lines exist within lines
File obj = open ("abc.txt", "r")
str 9 = File obj . readlines()
print ("Output : "str 9)
for line in str 9 :
        print (len(line))
    File obj . close()
>>> ('output :', ['college database'])
```

Scanned by CamScanner



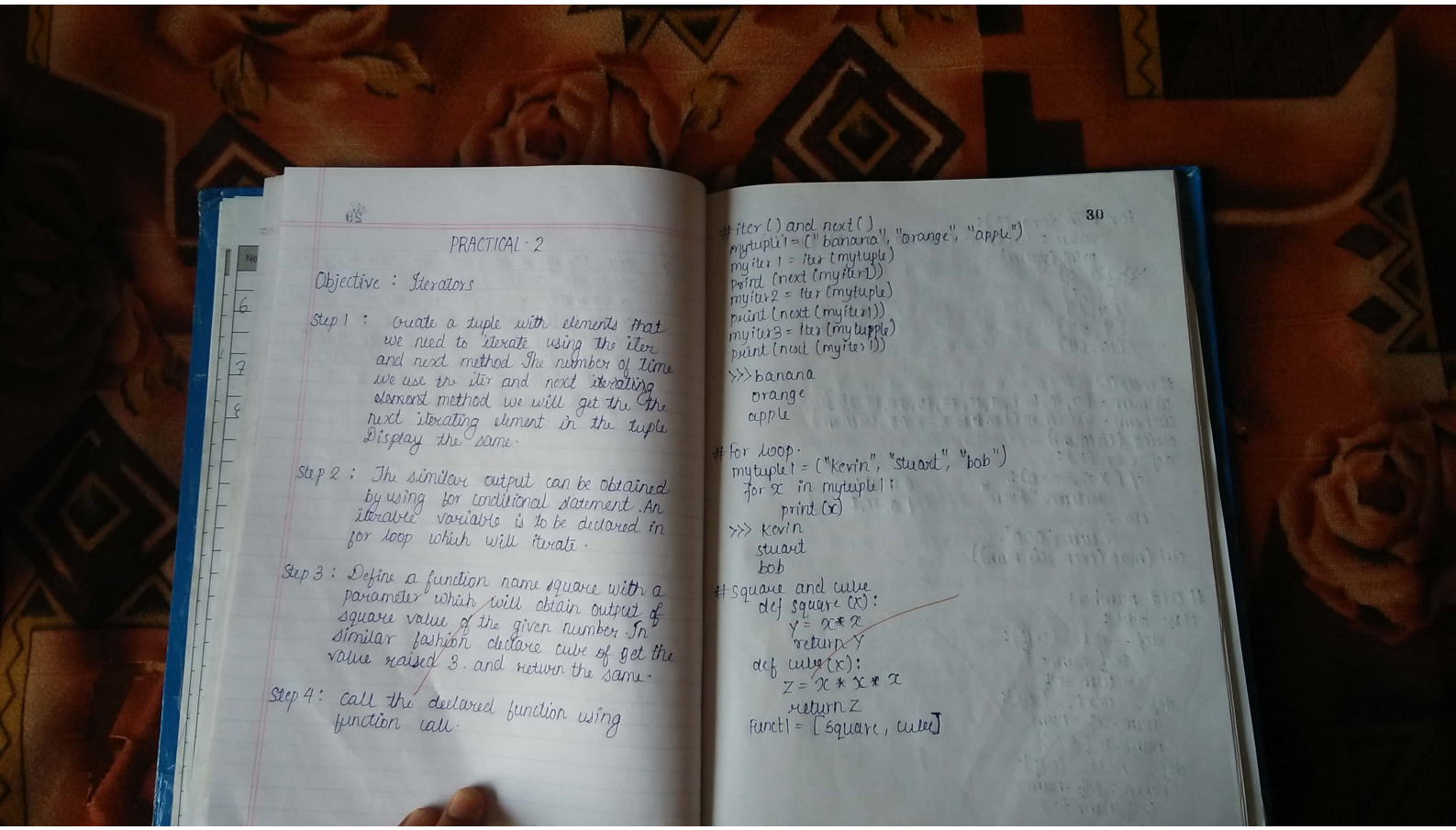
# content with special character
Fileobj = open("abc.txt","r")
content = Fileobj.read(1)
while len(content)>0:
    print(content, end="#")
    content = File.read(1)

>>> H# E# L# L# O#
#I #A #M # A # I #
#H #I #M #A #N #S #H #U
#D #E #V #L #O #P #E #R.

29

step10: open Fileobj with readmode also apply the readmode on the Fileobj & store it in a variable use while conditional statement to check the length of the content in the variable. if the content is greater than 0. then print the content & use end syntax in print statement to add (hash) special symbol in content of file.

Scanned by CamScanner



PRACTICAL - 2

Objective : Iterators

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

Step 2 : The similar output can be obtained by using for conditional statement. An iterable variable is to be declared in for loop which will iterate.

Step 3 : Define a function name square with a parameter which will obtain output of square value of the given number. In similar fashion declare cube of get the value raised 3. and return the same.

Step 4 : call the declared function using function call.

30

```
#iter() and next()
mytuple1 = ("banana", "orange", "apple")
myiter1 = iter(mytuple)
print(next(myiter1))
myiter2 = iter(mytuple)
print(next(myiter))
myiter3 = iter(mytuple)
print(next(myiter))
```

```
>>> banana
    orange
    apple
```

```
# for loop.
mytuple1 = ("Kevin", "stuart", "bob")
  for x in mytuple1:
      print(x)
```

```
>>> Kevin
    stuart
    bob
```

```
#square and cube
    def square(x):
        y = x*x
        return y
    def cube(x):
        z = x*x*x
        return z
Funct1 = [square, cube]
```

Scanned by CamScanner

```
for i in range(5):
    value = list(map(lambda x: x(i), func1))
    print(value)
>>> [0,0]
    [1,1]
    [4,8]
    [9,27]
    [16,64]

#map()
listnum = [0,4,5,7,9,11,13,15,20,19,25]
listnum = list(map(lambda x: x*5, listnum))
print(listnum)
def even(x):
    if(x%2==0):
        return "Even"
    else:
        return "ODD".
list(map(even, listnum))

#Odd numbers
class odd:
    def __iter__(self):
        self.num = 1
        return self
    def __next__(self):
        num.self = num
        num.self += 2
    def __next__(self):
        num = self.num
        self.num += 2
        return num.
```

OUTPUT:
[0, 4, 1, 2]

Step 5 : using for conditional statement specifying the range use the list type casting with map method declare a lambda i.e annonymus function and print the same.

Step 6 : Declare a listnum variable and declare some elements then use the map method with help of lambda function give two argument display the output.

Step 7 : Define a function even with a parameter then using conditional statement do check whether the number is even and odd and return respectively.

Step 8 : Define a class and within that define the iter() method which will initialise the first element within the container object

Step 9 : Now use the next() and define the logic for displaying odd value.

Scanned by CamScanner

Step 10 : Define an object of a class.

Step 11 : Accept an number from the user till which we want to display the odd number.

Step 12 : Define a iter method with a argument and initialize it to a first value.

Step 13 : For extracting the next element from the container use the next method with an argument and compare no of elements reserved in a container.

Step 14 : Now create an object from the given class and pass the object as an argument to the iter method.

Step 15 : Now using the conditional statement display all the value.


```
myobj = odd()
myitr = iter(myobj)
x = int(input("Enter a number : "))
for i in myitr:
    if (i< x):
        print (i)
```

```
>>> Enter a number : 15
1
3
5
7
9
11
13
```

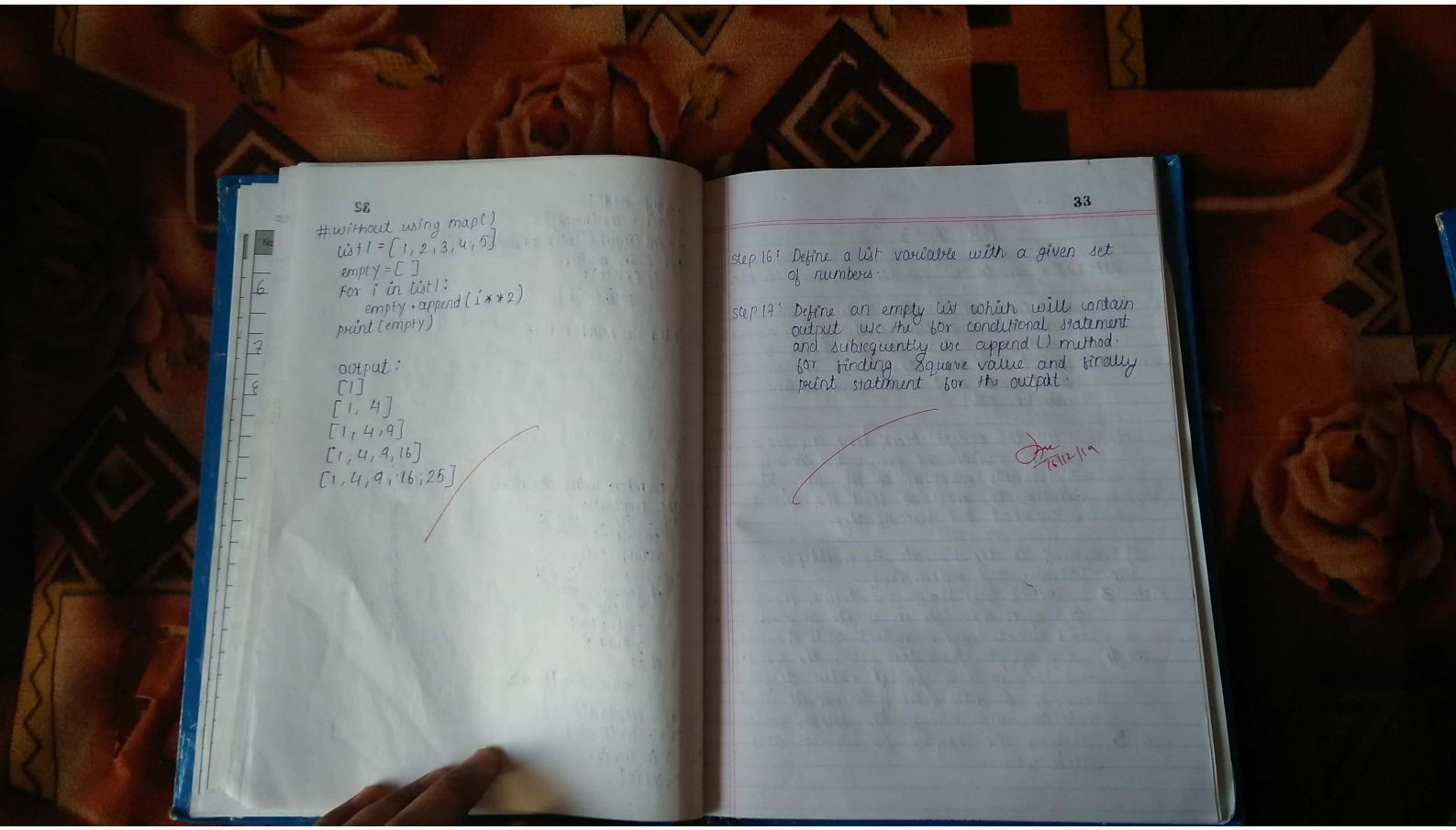
Odd numbers using iter method.

```
class myclass :
  def __iter__(self):
    self.a = 1
    return self
  def __next__(self):
    if self.a <= 10:
      x = self.a
      self.a += 1
      return x
    else:
      raise StopIteration

myobj = myclass()
myitr = iter(myobj)
for x in myitr:
  print(x)
```

output :

```
1
2
3
4
5
6
7
8
9
10
```

Scanned by CamScanner



32

```
#without using map()
list1 = [1,2,3,4,5]
empty = []
for i in list1:
    empty.append(i**2)
print(empty)
```

output:

```
[1]
[1, 4]
[1, 4, 9]
[1, 4, 9, 16]
[1, 4, 9, 16, 25]
```

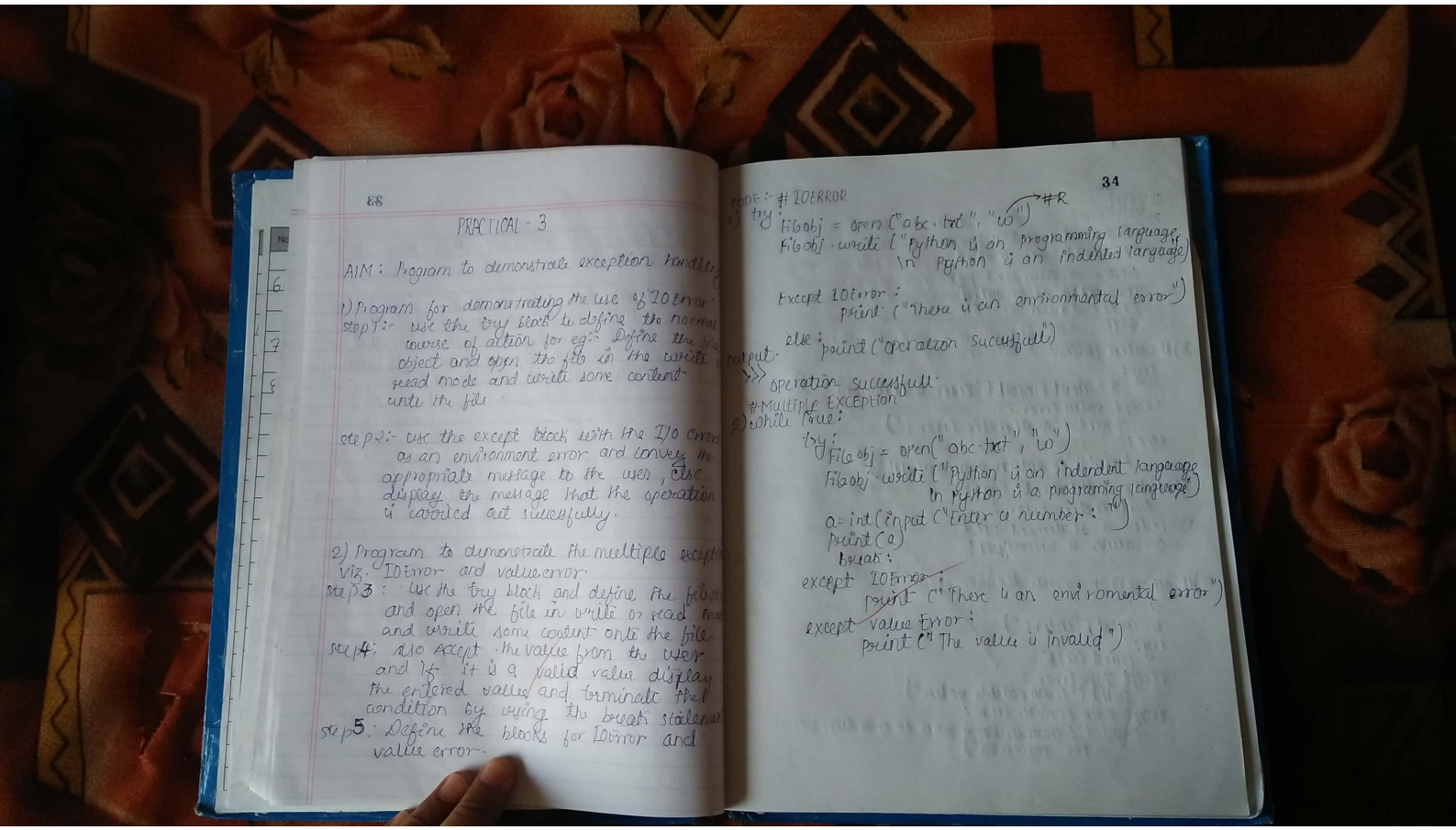
33

Step 16: Define a list variable with a given set of numbers.

Step 17: Define an empty list which will contain output use the for condutional statement and subsequently use append() method. for finding Square value and finally print statement for the output.

16/12/19

Scanned by CamScanner



PRACTICAL - 3

AIM: Program to demonstrate exception handlin

1) Program for demonstrating the use of IO Error

step1:- use the try block to define the normal course of action for eg:- Define the file object and open the file in the write read mode and write some content onto the file

step2:- use the except block with the I/O error as an environment error and convey the appropriate message to the user, also display the message that the operation is carried out successfully.

2) Program to demonstrate the multiple exceptio viz. IOError and value error.

step3: Use the try block and define the file and open the file in write or read mo and write some content onto the file.

step4: Also accept the value from the user and if it is a valid value display the entered value and terminate the condition by using the break statement

step5: Define the blocks for IOError and value error.

CODE: #IOERROR

```
try:
    Fileobj = open("abc.txt", "w")
    Fileobj.write("Python is an programming language
                   In Python is an indented language")
except IOError:
    print("There is an environmental error")
else:
    print("operation successfull")
```

output:

>>> operation successfull

#Multiple EXCEPTION

```
while True:
    try:
        Fileobj = open("abc.txt", "w")
        Fileobj.write("Python is an indendent language
                       In Python is a programing language")
        a=int(input("Enter a number:"))
        print(a)
        break:
except IOError:
        print("There is an environmental error")
except ValueError:
        print("The value is invalid")
```

Scanned by CamScanner

OUTPUT:

```
>>> Enter a number : abc
The value is invalid.
Enter a number : pqr
The value is invalid.
Enter a number : 64
64.
```

3) # value Error:

```
try:
    x = input(input("Enter a statement : "))
except value Error :
    Print ("Arithmetic Error ! ")
else:
    Print ("Operation is successfull ! ")
```

```
>>> Enter a statement : abc.
Arithmetic Error !
>>> Enter a statement : 145
Operation is successfull !
```

4) # Type error, zero division error

```
a = 1
b = input ("Enter: ")
try:
    print (a/b)
except Type Error :
    Print ("Incompatible value")
except Zero Division Error :
    Print ("Denominator is zero so operation cannot
        be performed")
```


3) Program to demonstrate the use of value Error:

Step 6 : In try block accept an input from the user

Step 7 : In except block use value error and print the same message

Step 8 : Else display the operation is successfull !

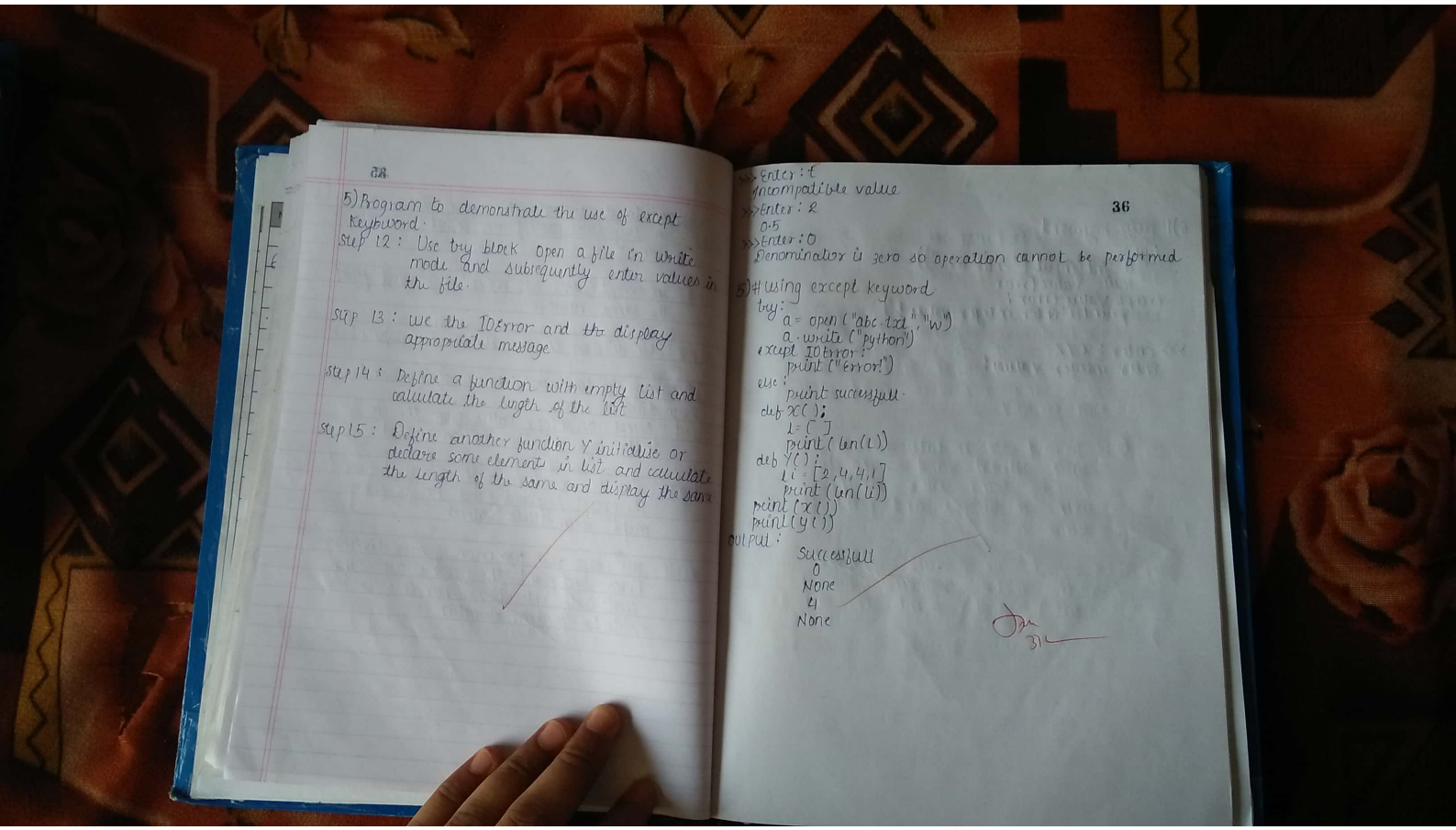
4) Program to demonstrate the use of Type and Zero division error.

Step 9 : Accept an integer value from the user. In the try use the division method.

Step 10 : For the exception to be raised use the except keyword (and) i.e Typeerror print Incompatible values.

Step 11 : use except with type error of zero division Error and print the message according that is if entered number is zero not able to perform operation !

Scanned by CamScanner



35

5) Program to demonstrate the use of except keyword.

Step 12 : Use try block open a file in write mode and subsequently enter values in the file.

Step 13 : We the IOError and the display appropriate message

Step 14 : Define a function with empty list and calculate the length of the list

Step 15 : Define another function Y initialise or declare some elements in list and calculate the length of the same and display the same

>>Enter : t
Incompatible value
>>Enter : 2
0.5
>>Enter : 0
Denominator is zero so operation cannot be performed

36

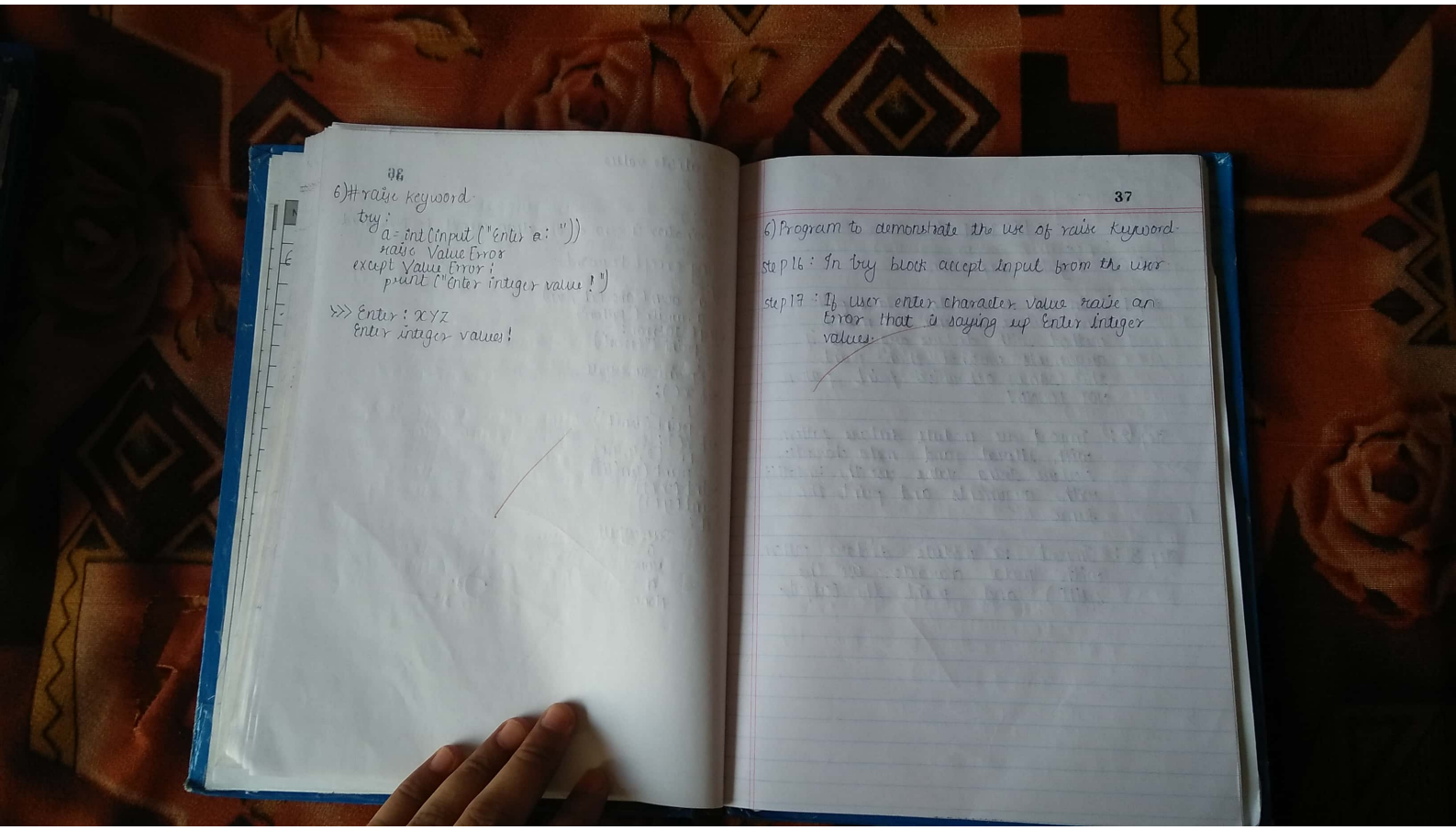
5) # using except keyword

```
try:
    a = open("abc.txt","w")
    a.write("python")
except IOError:
    print("error!")
else:
    print successfull.
def X():
    l=[]
    print(len(l))
def Y():
    li=[2,4,4,1]
    print(len(li))
print(X())
print(y())
```

output :

```
successfull
0
None
4
None
```

Scanned by CamScanner



36

6) # raise keyword

```
try:
    a = int(input("Enter a: "))
    raise ValueError
except ValueError:
    print("Enter integer value!")
```

>>> Enter : xyz
Enter integer value!

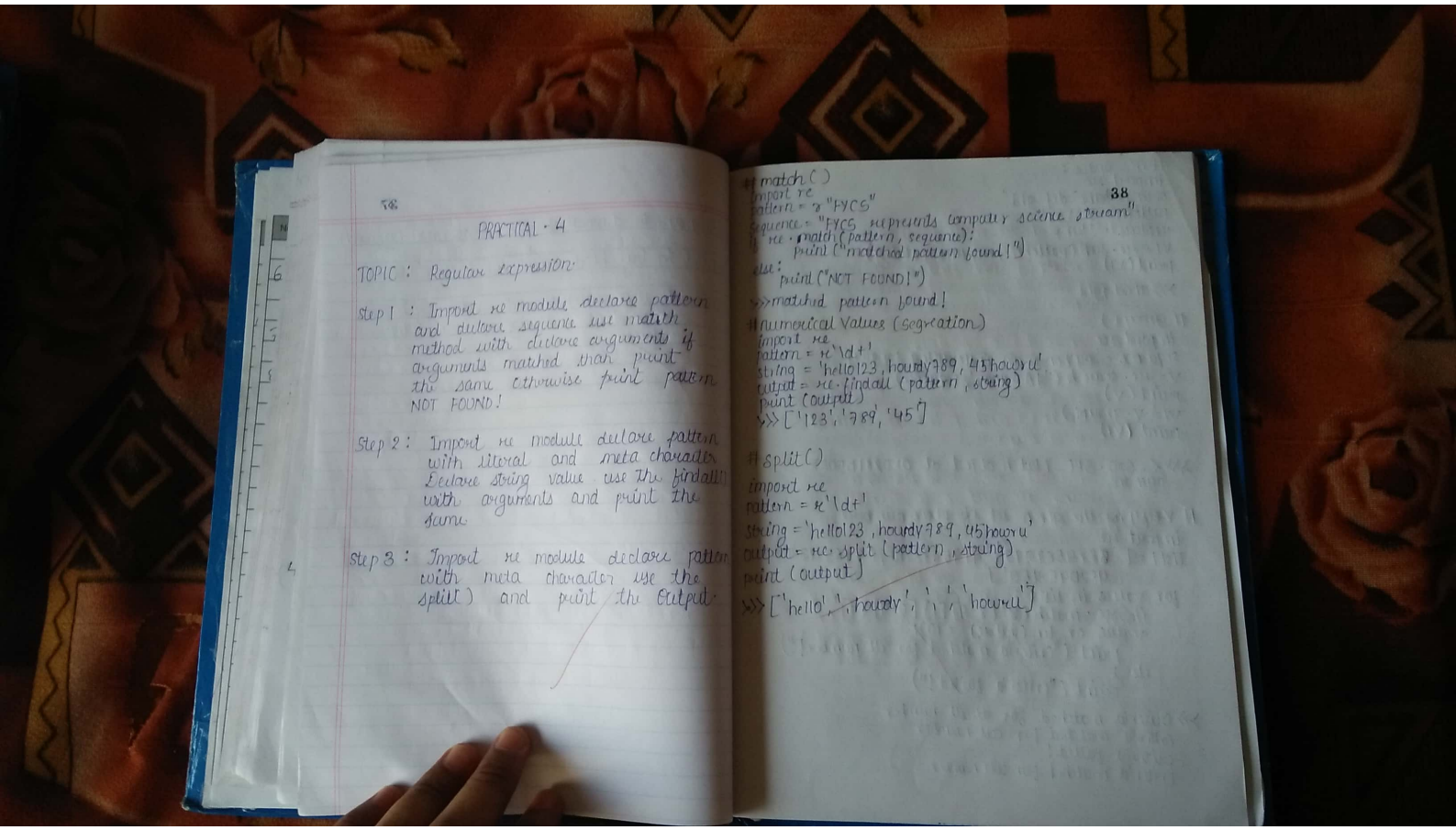
37

6) Program to demonstrate the use of raise keyword

step 16 : In try block accept input from the user

step 17 : If user enter character value raise an error that is saying up Enter Integer values.

Scanned by CamScanner



PRACTICAL - 4

TOPIC : Regular expression

Step 1 : Import re module declare pattern and declare sequence use match method with declare arguments if arguments matched then print the same otherwise print pattern NOT FOUND!

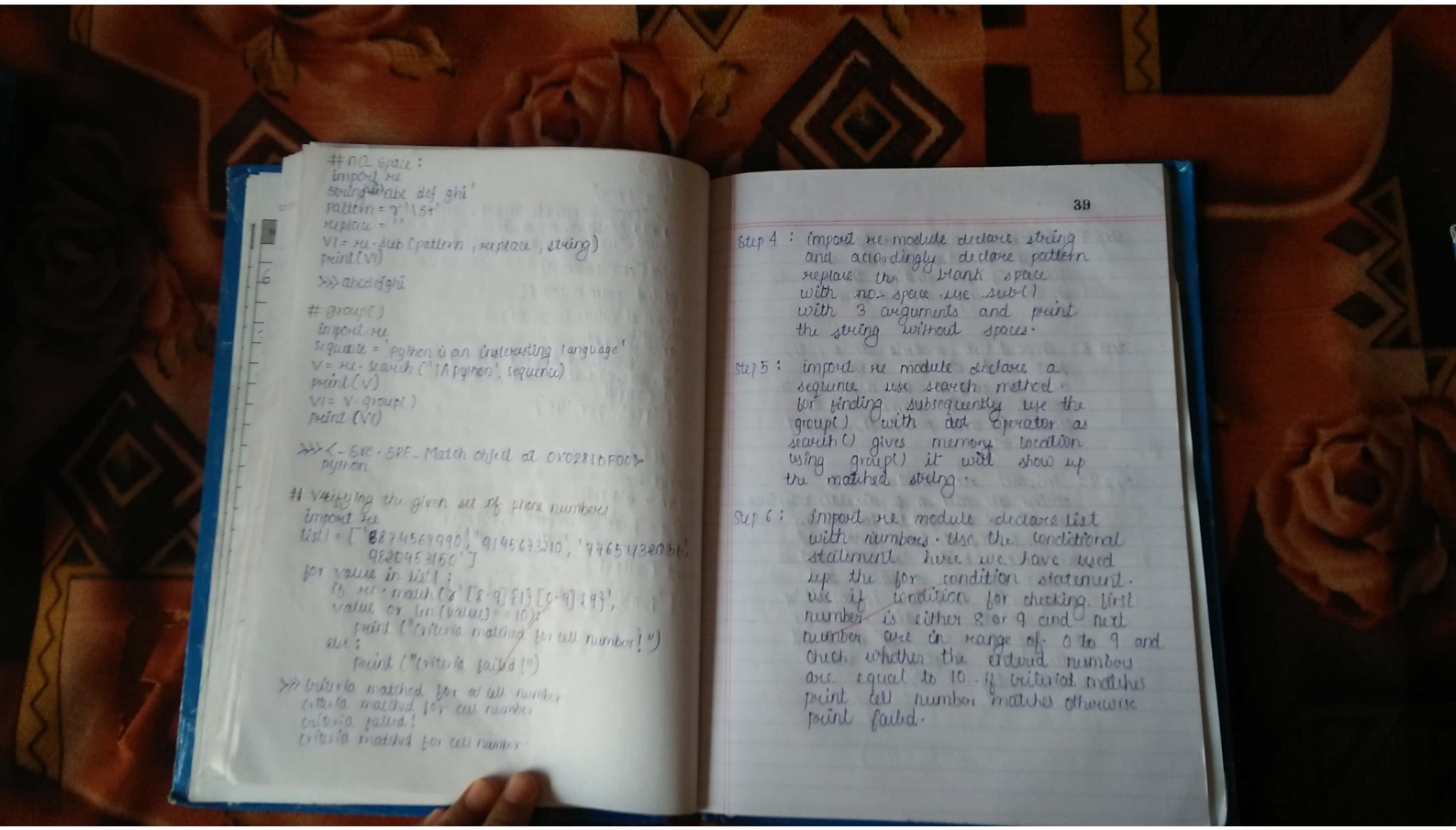
Step 2 : Import re module declare pattern with literal and meta character Declare string value use the findall() with arguments and print the same.

Step 3 : Import re module declare pattern with meta character use the split() and print the output.

38

```
#match()
import re
pattern = r"FYCS"
sequence = "FYCS represents computer science stream"
if re.match(pattern, sequence):
    print("matched pattern found!")
else:
    print("NOT FOUND!")
>>>matched pattern found!
#numerical values (segregation)
import re
pattern = r'\d+'
string = 'hello123, howdy789, 45howru'
output = re.findall(pattern, string)
print(output)
>>> ['123', '789', '45']

#split()
import re
pattern = r'\d+'
string = 'hello123, howdy789, 45howru'
output = re.split(pattern, string)
print(output)
>>> ['hello', ', howdy', ', ', 'howru']
```

Scanned by CamScanner



```
# NO space :
import re
string = 'abc def ghi'
pattern = r'\s+'
replace = ''
V1 = re.sub(pattern, replace, string)
print(V1)
>>> abcdefghi

# group()
import re
sequence = 'python is an interesting language'
V = re.search('^Python', sequence)
print(V)
V1 = V.group()
print(V1)

>>> <_sre.SRE_Match object at 0x02810F0C>
  python

# Verifying the given set of phone numbers
import re
list1 = ['8874567990', '9145673210', '7765432056',
         '9820453150']
for value in list1:
    ch = re.match(r'[8-9][0-9]{9}', value or len(value) = 10):
        print("Criteria matching for all numbers!")
    else:
        print("Criteria failed!")

>>> Criteria matched for all number
Criteria matched for all number
Criteria failed!
Criteria matched for all number
```

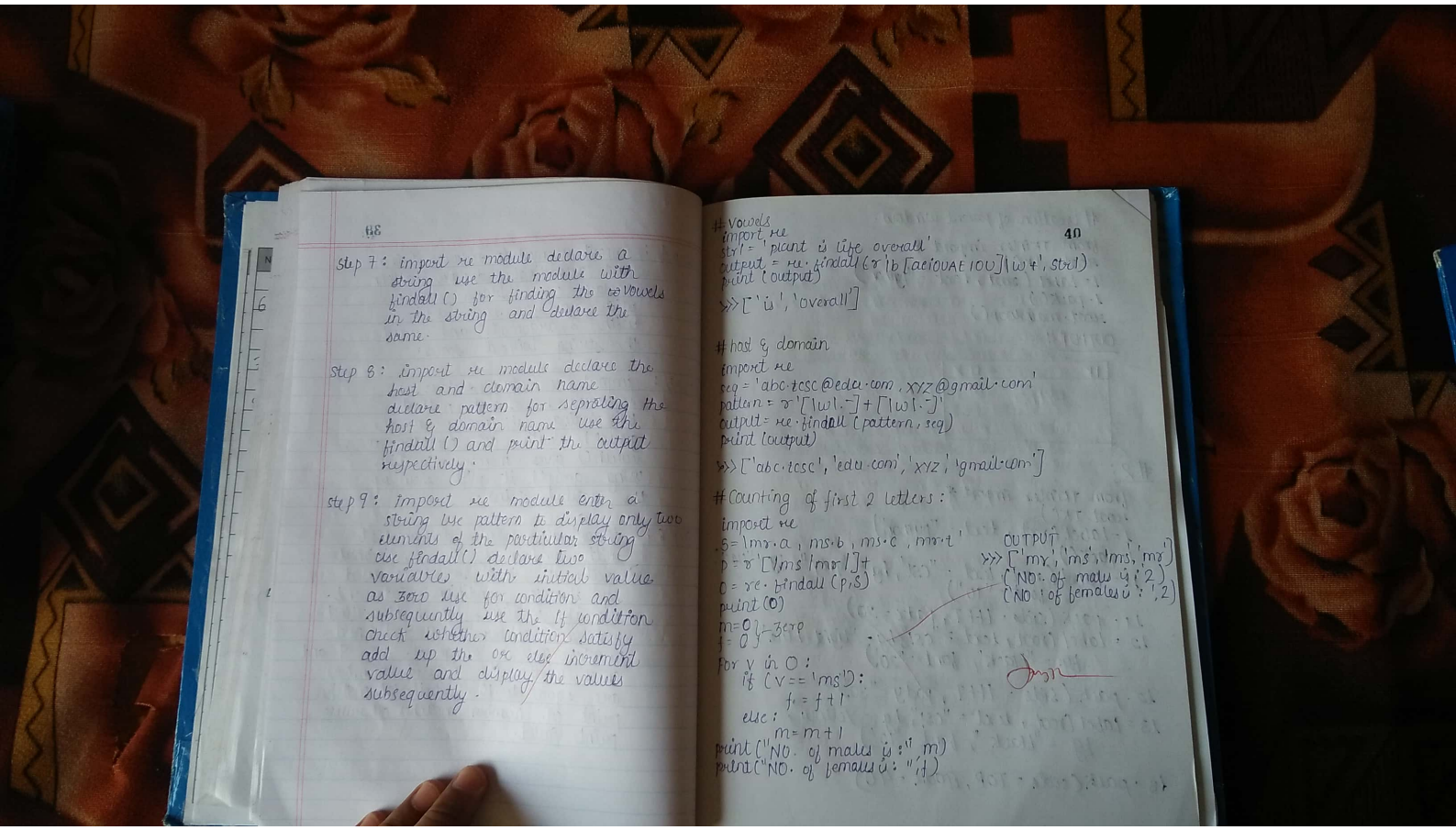
39

Step 4 : import re module declare string and accordingly declare pattern replace the blank space with no space use sub() with 3 arguments and print the string without spaces.

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

Step 6 : import re module declare list with numbers. Use the conditional statement here we have used up the for condition statement. We if condition for checking first number is either 8 or 9 and next number are in range of 0 to 9 and check whether the entered numbers are equal to 10. If criteria matches print all number matches otherwise print failed.

Scanned by CamScanner



Step 7: import re module declare a string use the module with findall() for finding the vowels in the string and declare the same.

Step 8: import re module declare the host and domain name declare pattern for seprating the host & domain name use the findall() and print the output respectively.

step 9: imposed re module enter a string use 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 check whether condition satisfy add up the one else increment value and display the values subsequently.

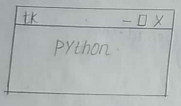
```
# Vowels
import re
str1 = 'plant is life overall'
output = re.findall(r'\b[aeiouAEIOU]\w+', str1)
print(output)
>>> ['a', 'overall']

# host & domain
import re
seq = 'abc.tcsc@edu.com, xyz@gmail.com'
pattern = r'[\w\.-]+[\w\.-]'
output = re.findall(pattern, seq)
print(output)
>>> ['abc.tcsc', 'edu.com', 'xyz', 'gmail.com']

# Counting of first 2 letters:
import re
S = 'mr.a, ms.b, ms.c, mr.t'
p = r'[\ms|mr]+'
O = re.findall(p,S)
print(O)
m=0; f=0
for v in O:
    if (v=='ms'):
        f = f+1
    else:
        m = m+1
print("NO. of males is:", m)
print("NO. of females is:", f)
```

OUTPUT
>>> ['mr', 'ms', 'ms', 'mr']
('NO. of males is', 2)
('NO. of females is', 2)

Scanned by CamScanner

```
# creation of parent window.
from Tkinter import *
root = Tk()
l = Label (root, text = "python")
l.pack()
root.mainloop()
```

OUTPUT :



#2:

```
from Tkinter import *
root Tk()
l = Label (root, text = "python")
l.pack()
l1 = Label (root, text = "CS!", bg = "grey", bg = "black"
            font = "10")
l1.pack (side = LEFT, padx = 20)
l2 = Label (root, text = "CS!" bg = "light blue",
     bg = "black", font = "20")
l2.pack (side = LEFT, pady = "30")
l3 = Label (root, text = "CS!", bg = yellow",
     bg = "black", font = "10")
l3.pack(side = TOP, ipadx = 40)
```


PRACTICAL - 5

TOPIC : GUI Components

Step 1 : Use the tkinter library for importing the features of the text widget

Step 2 : Create an object using the tk()

Step 3 : Create a variable using the widget label and use the text method

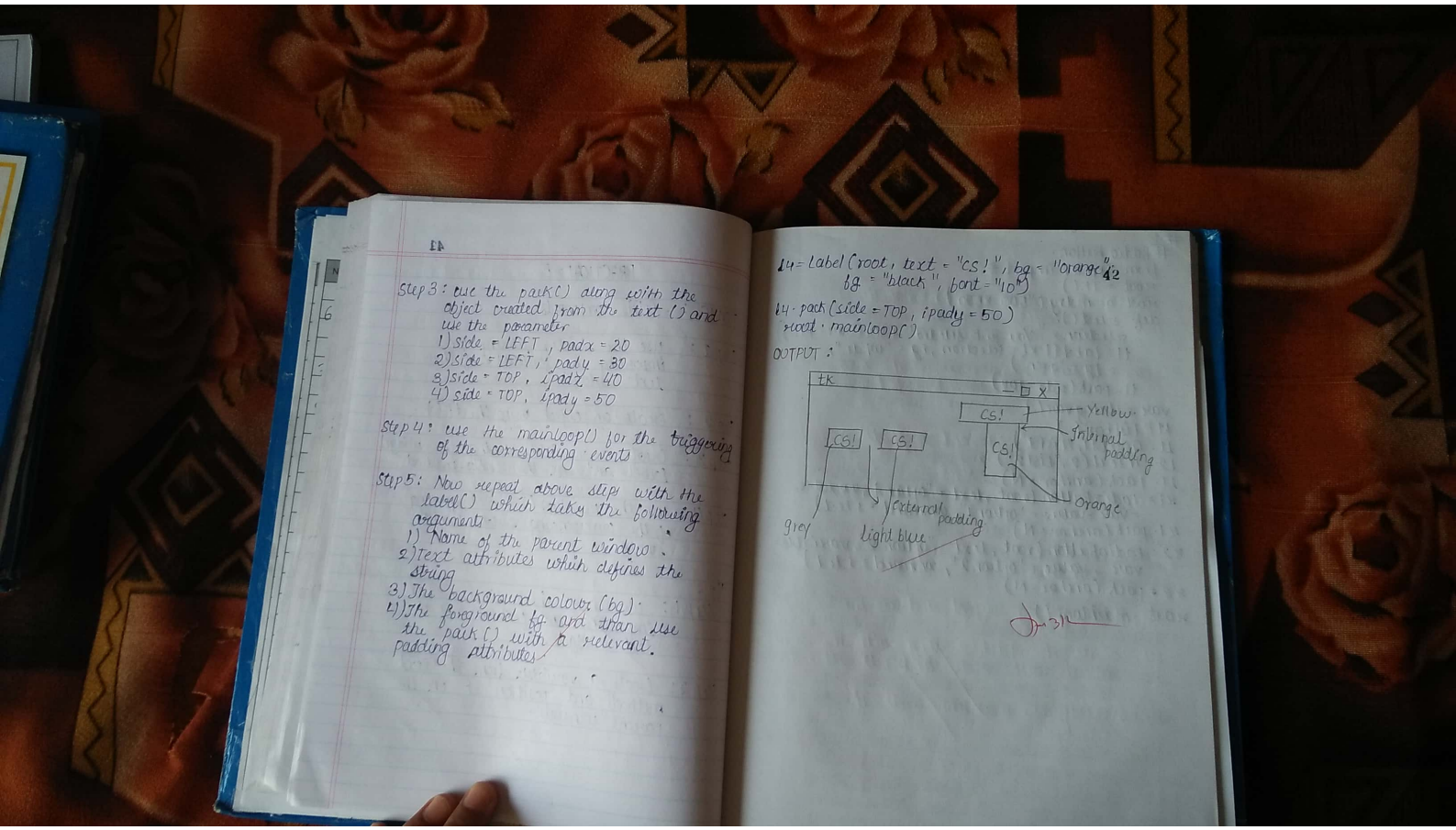
Step 4 : Use the mainloop() for triggering of the corresponding above mentioned events

#2 :

Step 1 : Use the tkinter library for importing the features of the text widget

Step 2 : Create a variable from the text method and position it on the parent window.

Scanned by CamScanner



Scanned by CamScanner

#1
```
#Radio button.
from tkinter import *
root = TK()
root geometry ("500 x500")
def select():
    selection = " You just selected" + str(var.get())
    tl = Label (text = selection, bg = "white",
        bg = "green")
    tl. pack (side = TOP)
var = stringvar()
l1 = Listbox()
l1. insert (1, "List 1")
l2. insert (2, "List 2")
l1. pack (anchor = N)
r1 = RadioButton (root, text = "option 1", variable
    = var, value = "option1", command = select)
r1. pack (anchor = N)
r2 = RadioButton (root, text = "option 2", variable =
    var, value = "option 2", command = select)
r2. pack (anchor = N)
root. mainloop()
```


PRACTICAL - 5 (B)

AIM : GUI Components

#1 :

Step 1: Import the relevant methods from the tkinter library Create an object with the parent window.

Step 2: use the parent window object along with the geometry() declaring specific pixel size of the parent window.

Step 3: Now define a function which tells the user about the given selection made from multiple option available

Step 4: Now define the parent window and define the option with control variable

Step 5: Use the listbox() and insert options on the parent window along with the pack() with specifying anchor attribute

Step 6: Create an object from radio button which will take following arguments parent window object, text variable which will take the values option no 1,2,3... variable arguments, corresponding value & trigger the function declared.

Scanned by CamScanner

Step 7: Now call the pack() for radio object so created and specify the argument using anchor attribute.

Step 8: Finally make use of the mainloop() along with parent object.

#2:

Step 1: Import relevant methods from the tkinter library.

Step 2: Create a parent object corresponding to the parent window.

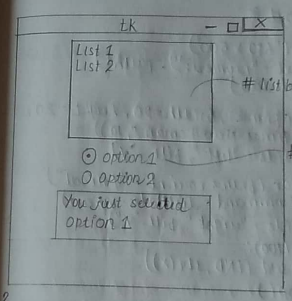
Step 3: Use the geometry() for laying of the window.

Step 4: Create an object and use the scrollbar()

Step 5: Use the pack() along with the scrollbar object with side and fill attributes.

Step 6: Use the mainloop with the parent object

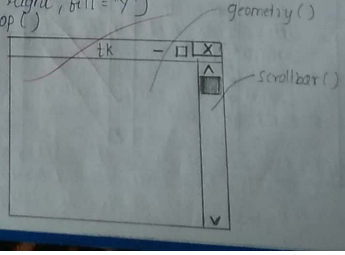
OUTPUT:



#2

```
Scrollbar()
from tkinter import*
root = TK()
root.geometry("500x500")
S=Scrollbar()
S.pack(side="right", fill="y")
root.mainloop()
```

OUTPUT:

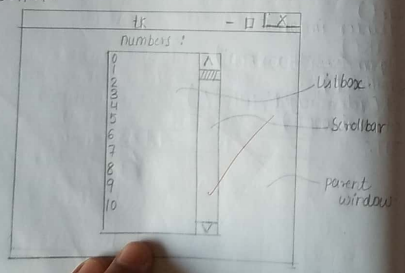


Scanned by CamScanner

#3
#using frame widget

```
from tkinter import *
window = TK()
window.geometry("680 x 500")
Label(window, text = "numbers:").pack()
frame = Frame(window)
frame.pack()
listNodes = Listbox(frame, width=20, height=20,
                    font=("Times New Roman", 10))
listNodes.pack(side = "left", fill = 'y')
scrollbar = Scrollbar(frame, orient = "vertical")
scrollbar.config(command = listNodes.yview)
scrollbar.pack(side = "right", fill = "y")
for x in range(100):
    listNodes.insert(END, str(x))
window.mainloop()
```

OUTPUT:



#3 :

Step 1 : Import the relevant libraries from the tkinter method.

Step 2 : Create an corresponding object of the parent window.

Step 3 : use the geometry manager with pixel size (680 x 500) or any other suitable pixel value.

Step 4 : use the label widget along with the parent object created and subsequently use the pack method.

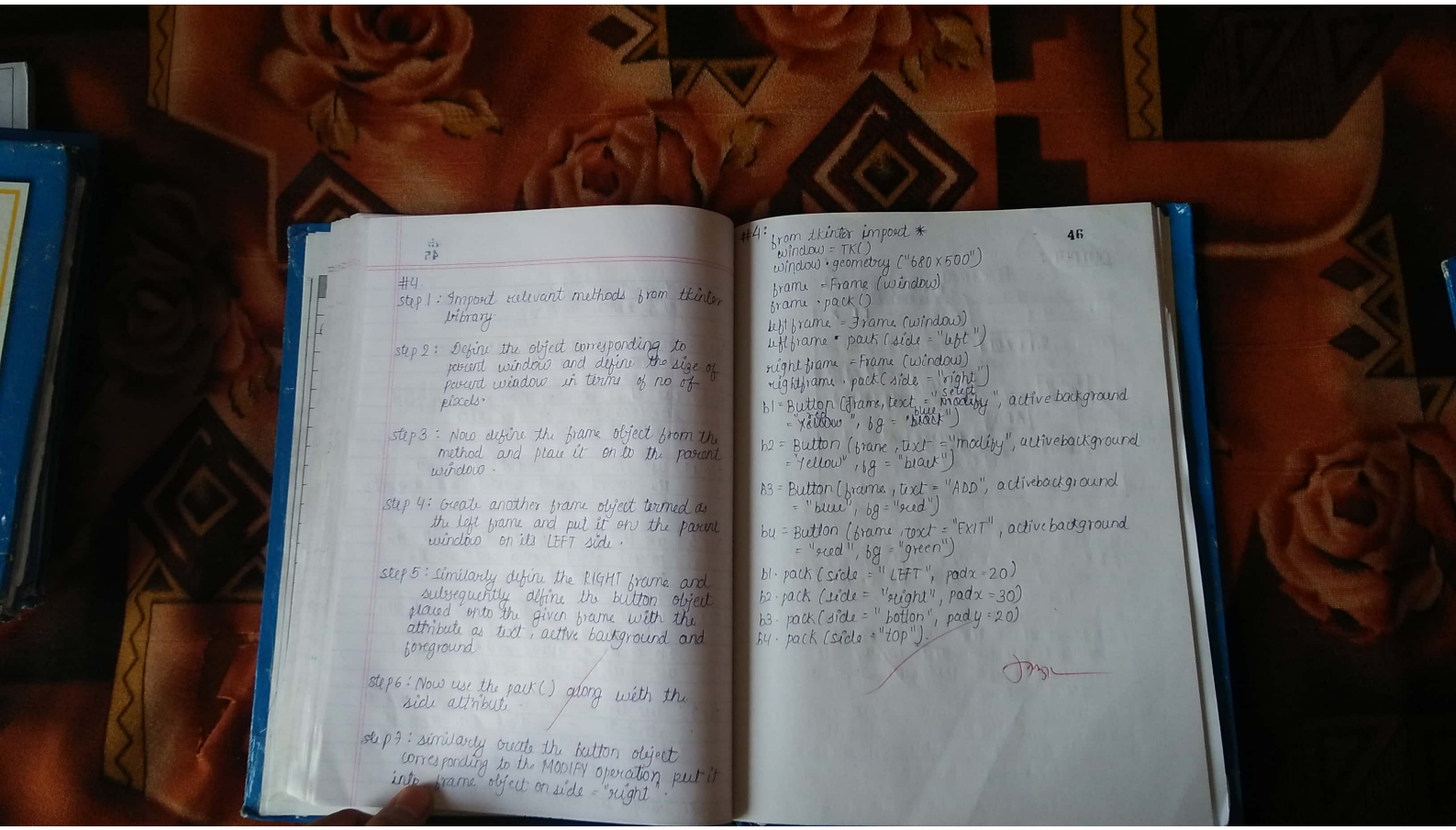
Step 5 : use the frame widget along with the parent object created and use the pack method.

Step 6 : use the listbox method along with the attributes like width, height font. Do create a listbox method's object use pack() for the same

Step 7 : use the scrollbar with an object use the attribute of vertical then configure the same with object created from the Scrollbar() and use pack()

Step 8: Trigger the events - using mainloo

Scanned by CamScanner



#4
step 1 : Import relevant methods from tkinter library

step 2 : Define the object corresponding to parent window and define the size of parent window in terms of no of pixels.

step 3 : Now define the frame object from the method and place it on to the parent window.

step 4 : Create another frame object termed as the left frame and put it on the parent window on its LEFT side.

step 5 : Similarly define the RIGHT frame and subsequently define the button object placed onto the given frame with the attributes as text, active background and foreground

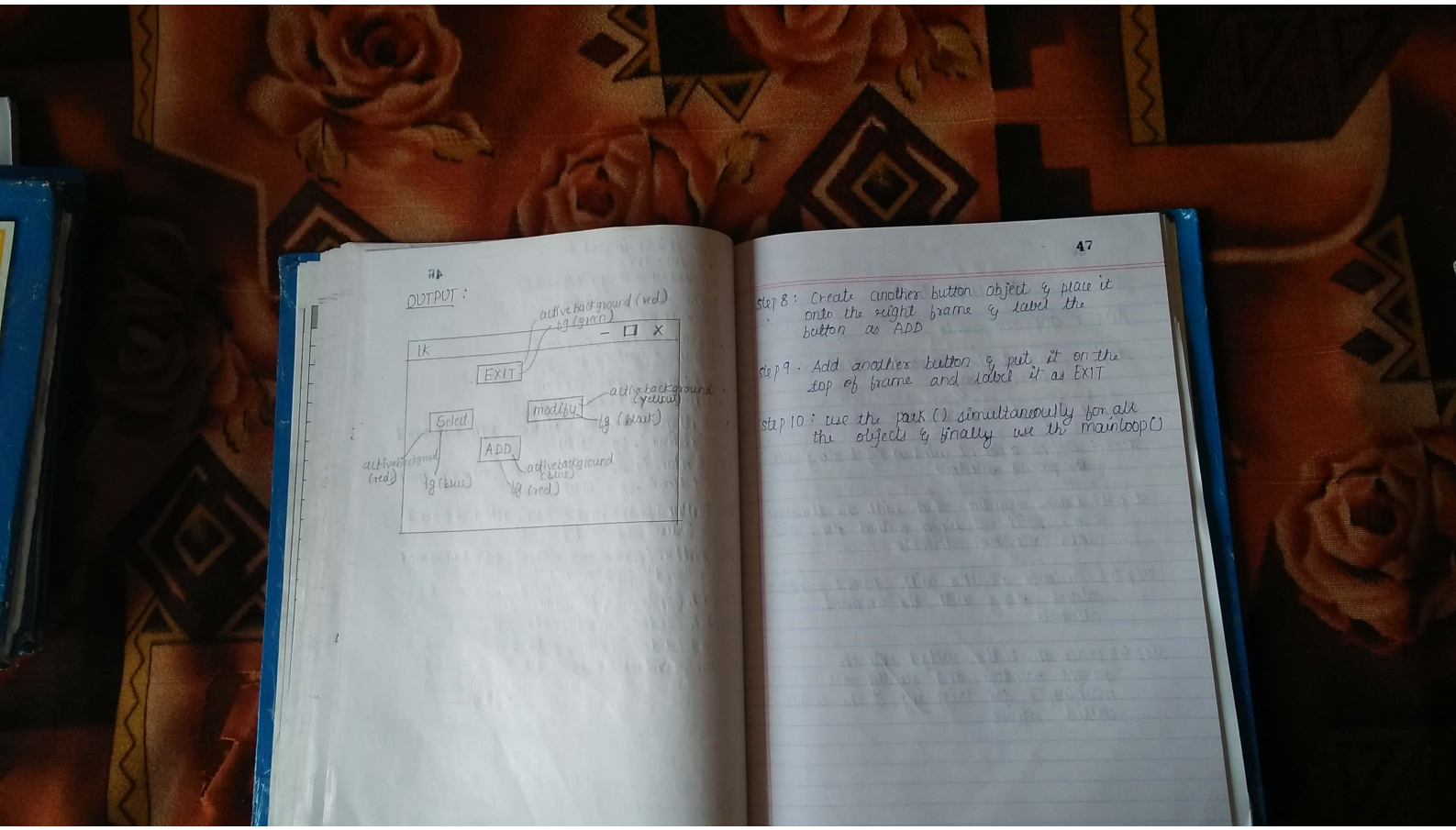
step 6 : Now use the pack() along with the side attribute.

step 7 : similarly create the button object corresponding to the MODIFY operation put it into frame object on side = "right".

#4:

```
from tkinter import *
window = TK()
window.geometry("680x500")
frame = Frame(window)
frame.pack()
leftframe = Frame(window)
leftframe.pack(side = "left")
rightframe = Frame(window)
rightframe.pack(side = "right")
b1 = Button(frame, text = "modify", activebackground
    = "yellow", bg = "blue")
b2 = Button(frame, text = "modify", activebackground
    = "yellow", bg = "blue")
b3 = Button(frame, text = "ADD", activebackground
    = "blue", bg = "red")
b4 = Button(frame, text = "EXIT", activebackground
    = "red", bg = "green")
b1.pack(side = "LEFT", padx = 20)
b2.pack(side = "right", padx = 30)
b3.pack(side = "bottom", pady = 20)
b4.pack(side = "top").
```


Scanned by CamScanner



OUTPUT :

step 8 : Create another button object & place it onto the right frame & label the button as ADD

step 9 : Add another button & put it on the top of frame and label it as EXIT

step 10 : use the pack() simultaneously for all the objects & finally use the mainloop()

Scanned by CamScanner

PRACTICAL - 5(C)

AIM : GUI components

step 1 : Import the relivant methods from tkinter library.

step 2 : Import tk massagebox.

step 3 : Define a parent window object along with the parent window.

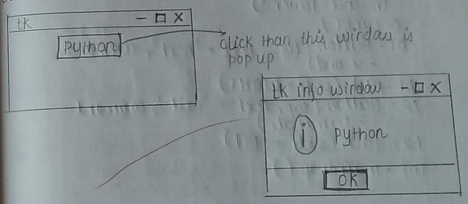
step 4 : Define a fundion which will use tkmessage box with showinfo method along with info window attribute.

step 5 : Declare a button with parent window object along with the command attribute.

step 6 : place the button widget onto the parent window and finally call mainloop() for triggering of the events called above.


```
#message box
from Tkinter import *
import tkMessageBox
root = Tk()
def function():
    tkMessageBox.showinfo("info window",
    "python")
    b1 = Button(root, text = "python", command
        = function)
    b1.pack()
root.mainloop()
```

OUTPUT :



Scanned by CamScanner

```
#Multiple window
#Different button (Relief())
from Tkinter import *
root = TK()
root.minsize (300,300)
def main():
    top = TK()
    top.config(bg = "black")
    top.title("HOME")
    top.minsize (300,300)
    L = Label (top, text = " SAN FRANCISCO
    \n place of interest : \n Golden gate
    Bridge \n lombard street \n chinatown
    \n coit Tower ")
    L.pack()
    b1 = Button (top, text = "next", command
        = second)
    b1.pack (side = RIGHT)
    b2 = Button (top, text = "exit", command
        = terminate)
    b2.pack (side = LEFT)
    top.mainloop ()
```


step 1: Import the relevant methods from the Tkinter library along with parent window object declared

step 2: use parentwindow object along with minsize function for window size

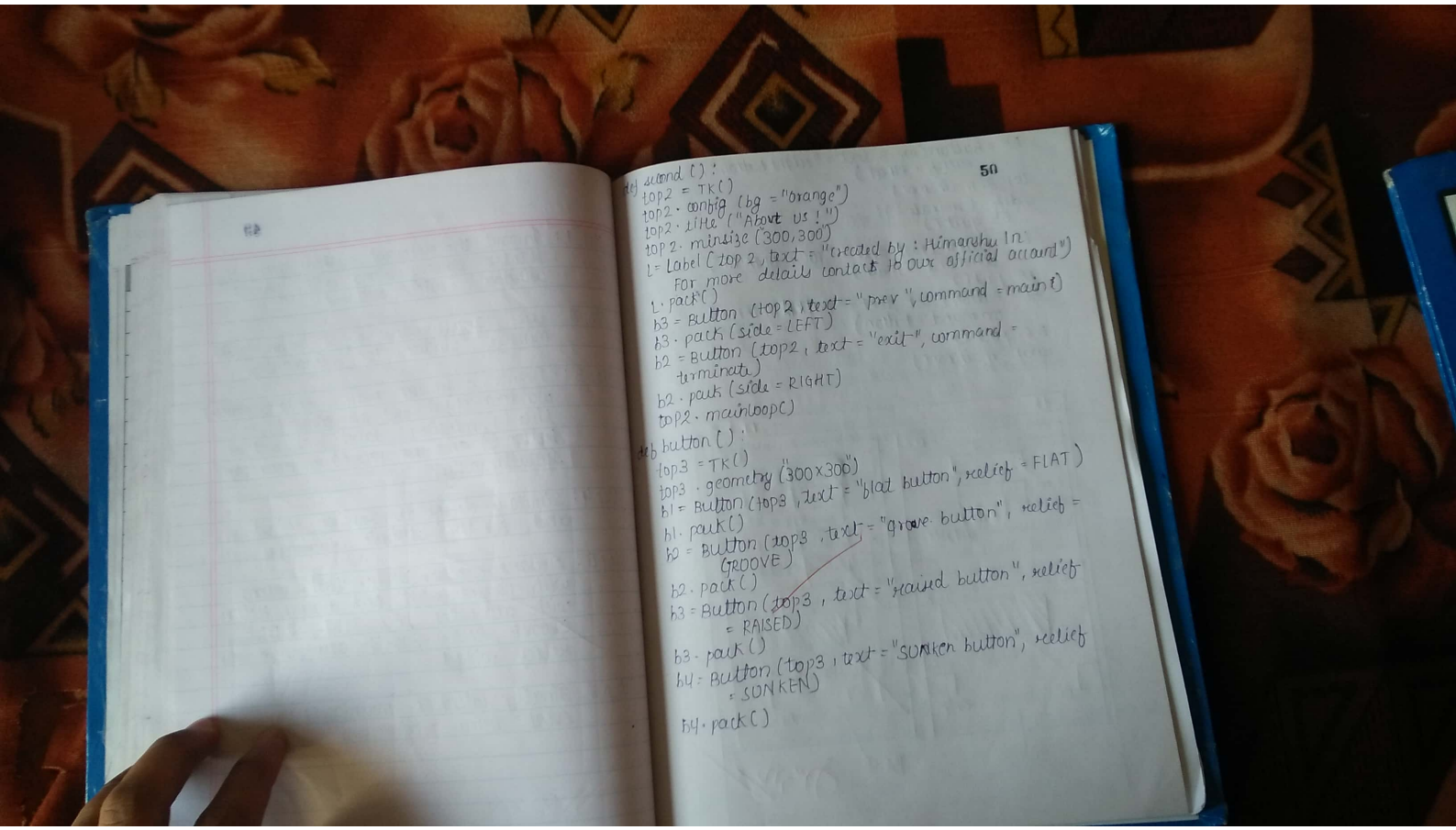
step 3: Define a function main, declare parent window object and use config(), title(), minsize(), label() as well as button() and use pack() & mainloop() simultaneously

step 4: similary define the function second and use the attribute accordingly

step 5: Declare another function button along with parent object and declare button with attributes like FLAT, RIDGE, GROOVE, RAISED, SUNKEN along with the relief widget

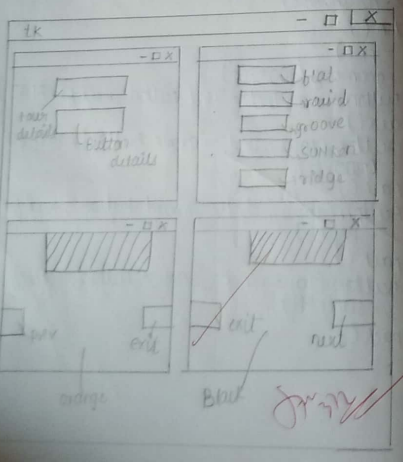
step 6: Finally called the mainloop() for event driven programming.

Scanned by CamScanner



Scanned by CamScanner

```
b5 = Button(top 3, text = "right button",
           relief = RIDGE)
top 3. mainloop()
def terminate():
  b5 quit()
b5 = Button(root, text = "TOUR DETAILS",
           command = main)
b5.pack()
b6 = Button(root, text = "BUTTON DETAILS",
           command = button)
b6. pack()
root.mainloop()
```



PRACTICAL - 5(D)

AIM : GUI Components
#TRAVERSING
(A) Program to traverse various windows using the button widgets

ALGORITHM:

Step 1 : Import the relevant method from tkinter library.

Step 2 : Define a function and create a object of given window by using the three methods namely config, title, minsize.

Step 3 : Define a button object which will be placed on the current window to traverse and define another button which will be used to exit from the window and place it onto current window.

Step 4 : Define another function which will use the quit method to terminate the program.

Step 5 : Now create an object of main window and use various methods like config, title, geometry etc.

Scanned by CamScanner

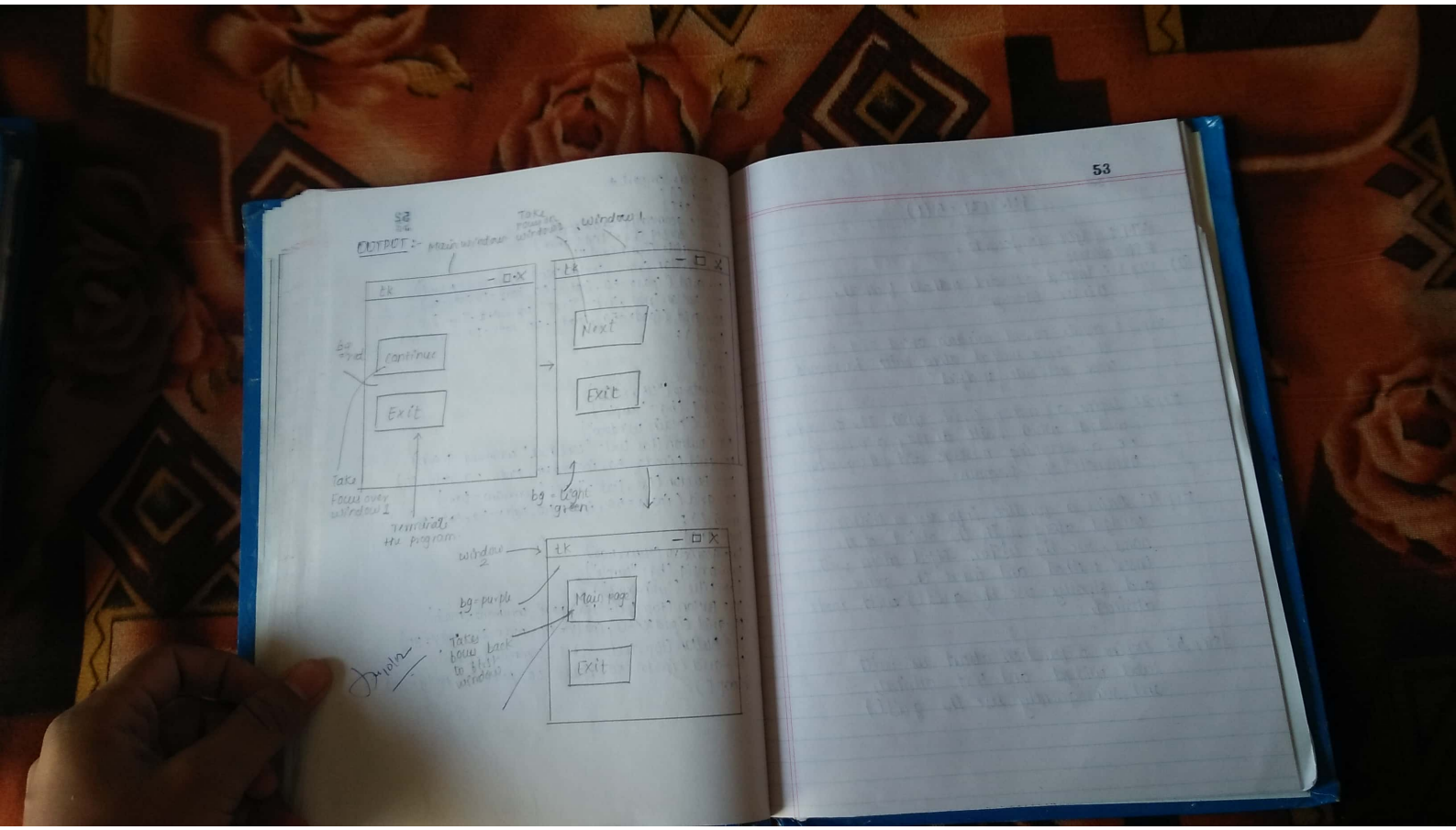
Step 6 : Define two buttons which will be placed on the main window ; one to traverse another window and the other to terminate the program

Step 7 : Define another function which will carry various button placed on third window . Define two buttons respectively and use the grid method along with the two buttons .

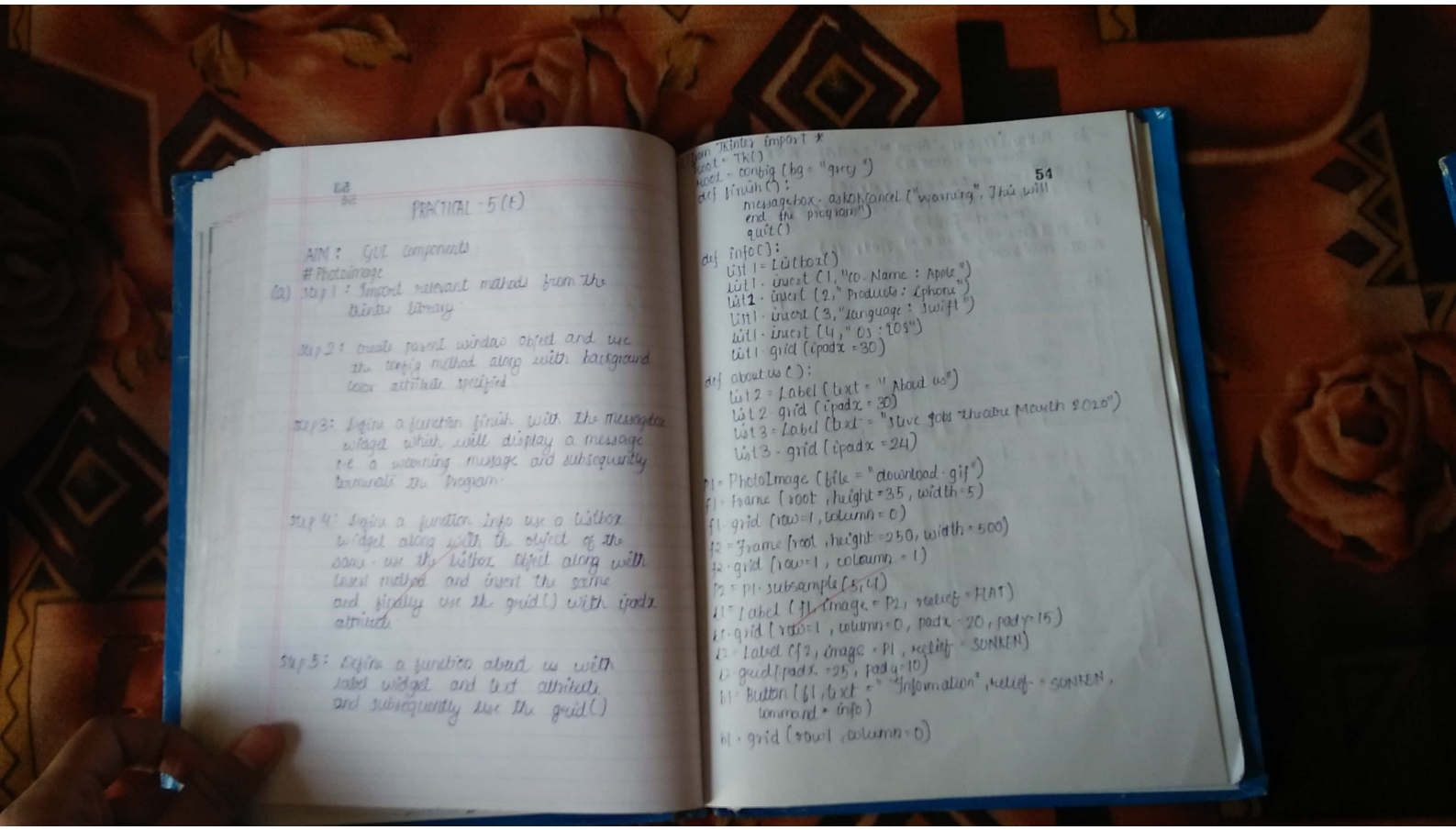
Step 8 : Finally call the mainloop method .


```
#Traversing
from tkinter import *
def main():
    root = TK()
    root.geometry("450x500")
    root.config(bg="Light green")
    root.title("Window1")
    B1 = Button(root, text="Next", command=main1)
    B1.grid(ipadx=50, ipady=40, padx=20, pady=30)
    B2 = Button(root, text="EXIT", command=term)
    B2.grid(ipadx=50, ipady=40, padx=20, pady=30)
def term():
    quit()
tos = TK()
tos.geometry("450x500")
tos.config(bg="Purple")
tos.title("Main window")
    B3 = Button(tos, text="continue", command=main)
    B3.grid(ipadx=50, ipady=40, padx=20, pady=30)
    B2 = Button(tos, text="EXIT", command=term)
    B2.grid(ipadx=50, ipady=40, padx=20, pady=30)
def main1():
    top = TK()
    top.geometry("450x500")
    top.config(bg="purple")
    top.title("Window 2")
    B4 = Button(top, text="Mainpage", command=main)
    B4.grid(ipadx=50, ipady=40, padx=20, pady=30)
    B2 = Button(top, text="EXIT", command=term)
    B2.grid(ipadx=50, ipady=40, padx=20, pady=30)
mainloop().
```

Scanned by CamScanner



Scanned by CamScanner



PRACTICAL - 5 (1)

AIM : GUI Components

#PhotoImage

(a) Step 1 : Import relevant methods from the tkinter library

Step 2 : Create parent window object and use the config method along with background color attribute specified

Step 3 : Define a function finish with the messagebox widget which will display a message i.e a warning message and subsequently terminate the program.

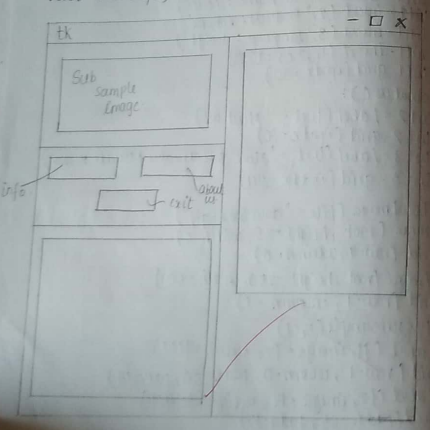
Step 4 : Define a function info use a listbox widget along with the object of the same. Use the listbox object along with insert method and insert the same and finally use the grid() with ipadx attribute.

Step 5 : Define a function about us with label widget and text attribute and subsequently use the grid()

```
from tkinter import *
root = Tk()
root.config(bg = "grey")
def finish():
    messagebox.askokcancel("warning", This will
    end the program")
    quit()
def info():
    list1 = Listbox()
    list1.insert(1, "ID Name : Apple")
    list1.insert(2, "Products : Iphone")
    list1.insert(3, "Language : Swift")
    list1.insert(4, "OS : IOS")
    list1.grid(ipadx = 30)
def aboutus():
    list2 = Label(text = "About us")
    list2.grid(ipadx = 30)
    list3 = Label(text = "Steve Jobs Theatre Mouth 2020")
    list3.grid(ipadx = 24)
p1 = PhotoImage(file = "download.gif")
f1 = Frame(root, height = 35, width = 5)
f1.grid(row = 1, column = 0)
f2 = Frame(root, height = 250, width = 500)
f2.grid(row = 1, column = 1)
p2 = p1.subsample(5,4)
l1 = Label(f1, image = p2, relief = FLAT)
l1.grid(row = 1, column = 0, padx = 20, pady = 15)
l2 = Label(f2, image = p1, relief = SUNKEN)
l2.grid(padx = 25, pady = 10)
b1 = Button(f1, text = "Information", relief = SUNKEN,
    command = info)
b1.grid(row = 1, column = 0)
```


Scanned by CamScanner

b2 = Button(f1, text = "About us", relief = SUNKEN,
command = about us)
b2.grid(row = 1, column = 2, padx = 5)
b3 = Button(f1, text = "EXIT", relief = RAISED,
command = finish)
b3.grid(row = 2, column = 1, padx = 15)
root.mainloop().



step 6: use photoimage widget with file and filename with gif attribute.

step 7: create a frame object along with the Frame() along with parent window object height & width specified and subsequently use the grid() with row & column attribute specified.

step 8: similarly create another frame object as declared by step 7.

step 9: create another object & use the subsample (5,4)

step 10: use label widget along with the frame object, relief attribute and subsequently use the grid()

step 11: Now create button object dealing with different section of frame.

Scanned by CamScanner

b. #Spinbox:

Step 1: Create an object from tk method and subsequently create an object from the str Spinbox() method.

Step 2: Make the object so created onto the parent window should trigger the corresponding events

Step 3: use the pack method to provide the direction using anchor method.

Step 4: use the mainloop() to terminate.

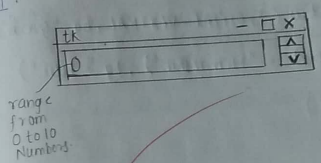
c. #Panel window.

Step 1: Create an object from panel window and use the pack method with the attribute fill and expand.

Step 2: Create an object from the Label method and put it onto the panel window with the text attribute and use the add method to embed the new object.

```
from tkinter import *
root = Tk()
S1 = Spinbox (root, from_=0 to_=10)
S1.pack (anchor = S)
root.mainloop()
```

OUTPUT:

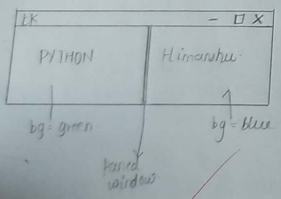


range from 0 to 10 Numbers

Scanned by CamScanner

(c.) from tkinter import *
root = Tk()
p = panedWindow(bg = "red")
p.pack(fill = BOTH, expand = 10)
l1 = Label(p, text = "PYTHON"), bg = "green")
p.add(l1)
p1 = panedWindow(p, orient = VERTICAL, bg = "yellow")
p.add(p1)
l2 = Label(p1, text = "Himanshu"), bg = "blue")
p1.add(l2)
root.mainloop()

OUTPUT



Step 3: Similarly create a second paned window object and add it onto the 1st paned window with orientation specified.

Step 4: Now create another label object and place it onto the 2nd paned window object and add it onto the 2nd pane.

Step 5: Now use the mainloop method to terminate

) # Canvas Widget

Step 1: use the Tkinter method and create an object from the canvas method and use the attribute height, width, background color and the parent window object

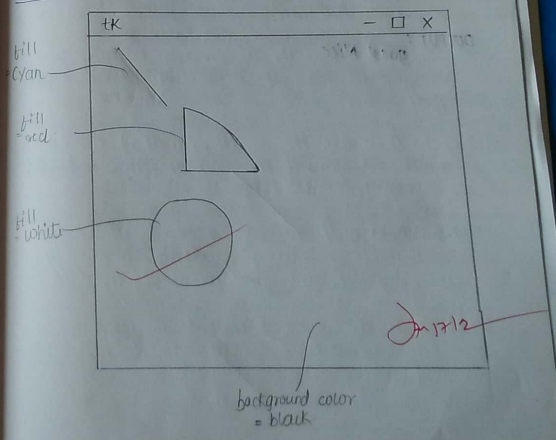
Step 2: use the method of create oval, create line and create arc along with the canvas object to so created and use the co-ordinate value. Also use the fill attribute to assign various colours.

Scanned by CamScanner

Step 3 : Now call the pack method and mainloop() method.

```
from tkinter import *
root = Tk()
C1 = Canvas(root, height=400, width=400, bg="black")
oval = C1.create_oval(20,140,150,250, fill="white")
line = C1.create_line(30,40,50,60, fill="cyan")
arc = C1.create_arc(20,140,150,60, fill="red")
C1.pack()
root.mainloop()
```

OUTPUT :



Scanned by CamScanner

a) import dbm
db = dbm.open("database", flag, "c")
if db["www"] != None :
    print ("Nice")
else:
    print ("Not Nice")

OUTPUT:
good Nice

PRACTICAL - 6

AIM : To make use of Database library.

1) Step 1 : import DB library and use the open method for creating the data base by specify name of the data base along with the corresponding flag.

Step-2 : use the objects for accessing to given web site and the corresponding regular for the web site

Step 3 :- Check whether the given URL address with the regular of the page is not equal to None than display the message from URL address else not found.

Scanned by CamScanner

b.

Step 1: Import the corresponding library taking of data base connection

Step 2: Now create connection objects using sqlite library and connecting method for create the new database.

step 3: Now create the cursor objects using cursor method from the connection objects create in steps

step 4: Now use the executing method for creating the tables with the column name and respective data type.

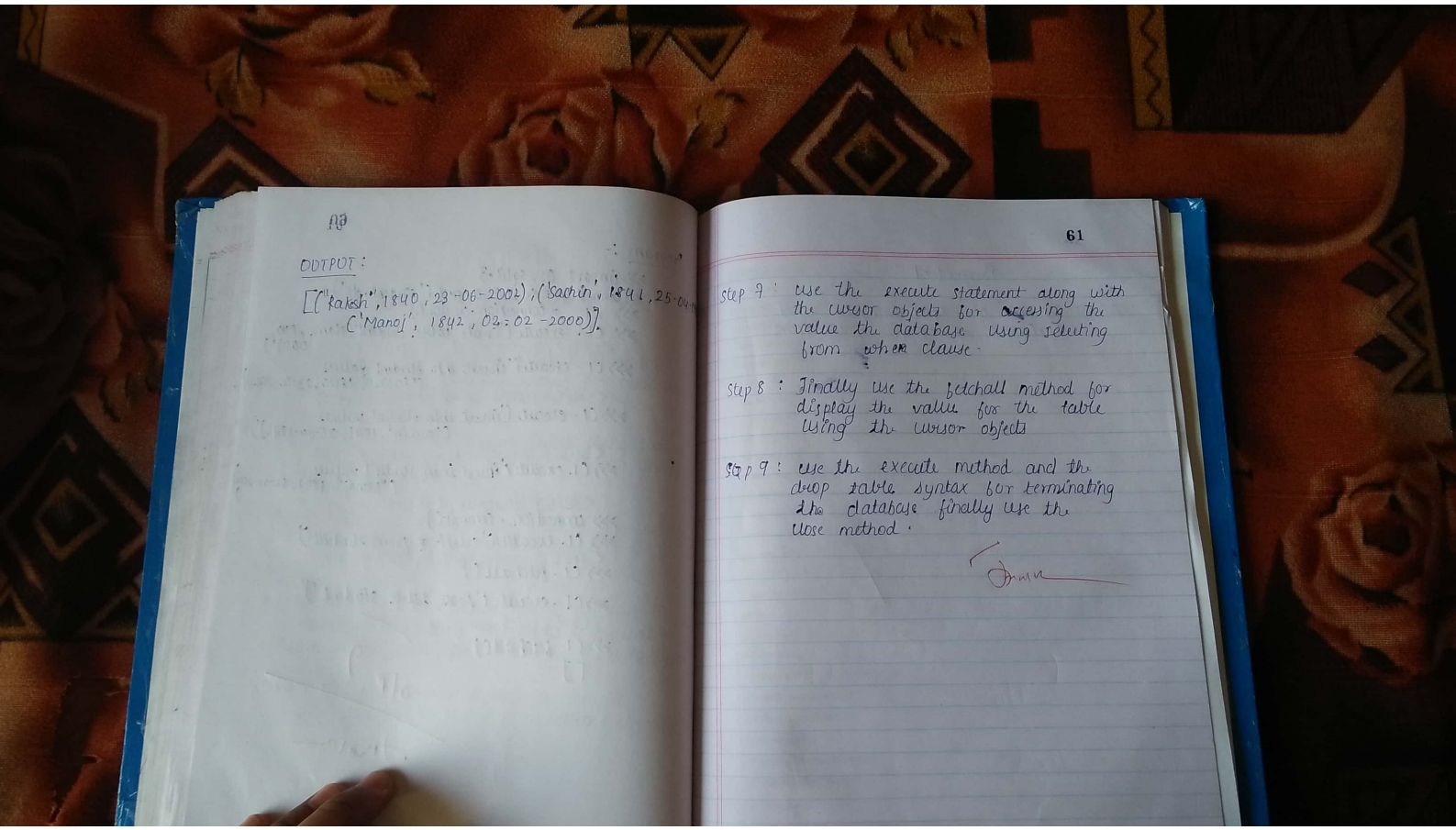
step 5: Now with the cursor objects use insert statements for entering the values co-ordinating into the different field considering the data types

step 6: Use the commit method to complete the transaction use the connection objects.

Program :-

```
>>>import os, sqlite3
>>>connection = sqlite3.connect("student.db")
>>> c1 = connection.cursor()
>>>c1.execute("create table student(Name, RNO,
<sqlite3.Cursor object at 0x02E34520> DOB)')
>>>c1.execute('insert into student values
                ("Rakesh", 1840, 23-06-2002))
  <sqlite3.cursor object at 0x02E34520>
>>>c1.execute('insert into student values
                ("sachin", 1841, 25-04-1996)')
  <sqlite3.cursor object at 0x02E34520>
>>>c1.execute('insert into student values
                ("Manoj", 1842, 02-02-2000)')
  <sqlite3.cursor object at 0x02E34520>
>>> connection.commit()
>>>c1.execute('select * from student')
  <sqlite3.cursor object at 0x02E34520>
>>>c1.fetchall()
>>>c1.execute('Drop table student')
  <sqlite3.Cursor object at 0x02E34520>
>>>c1.fetchall()
  []
```

Scanned by CamScanner



AIM

OUTPUT:

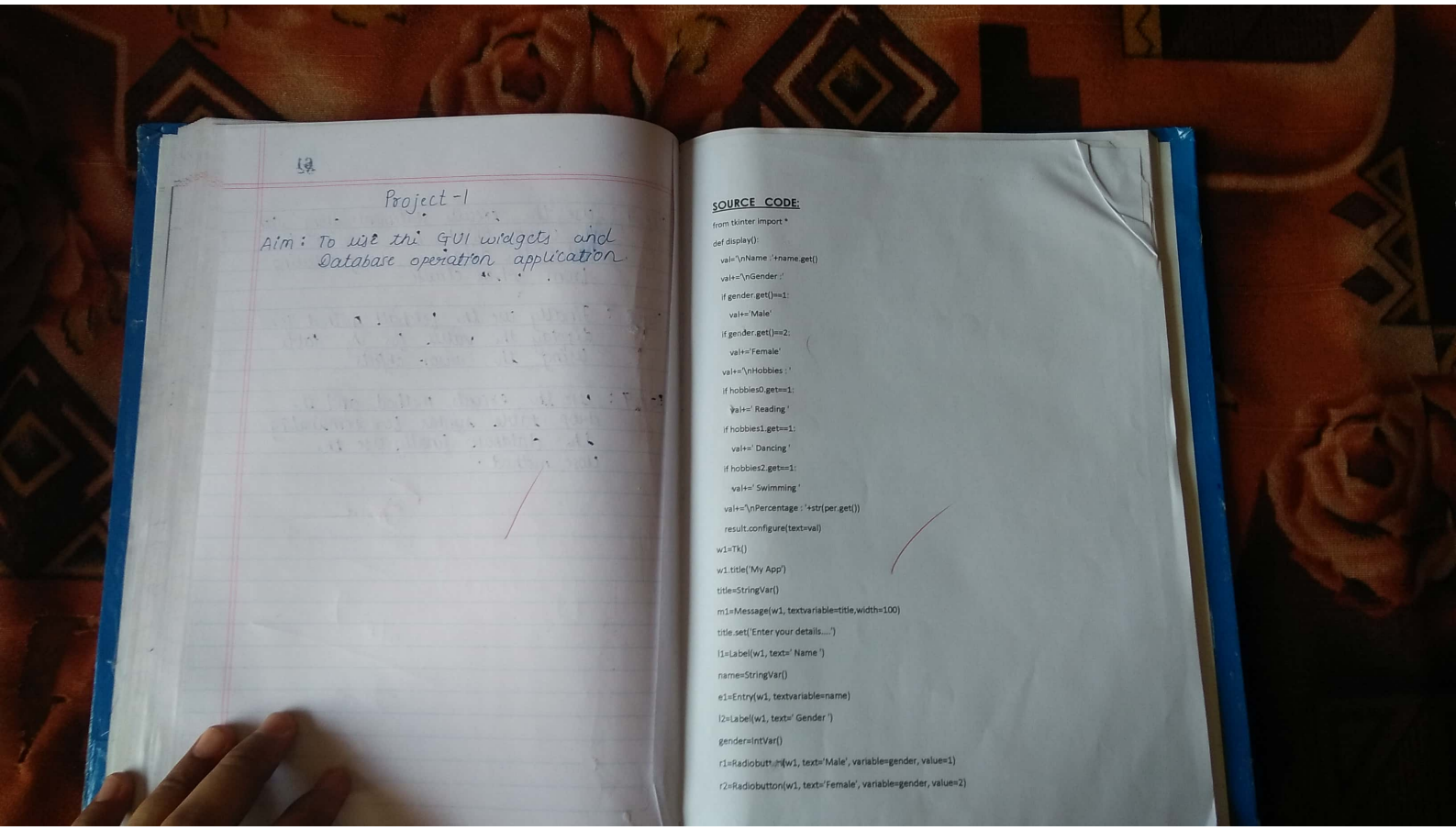
[('Rakesh', 1840, 23-06-2002), ('Sachin', 1841, 25-04-
('Manoj', 1842, 02-02-2000)].

Step 7: use the execute statement along with the cursor objects for accessing the value the database using selecting from where clause.

Step 8: Finally use the fetchall method for display the value for the table using the cursor objects

Step 9: use the execute method and the drop table syntax for terminating the database finally use the close method.

Scanned by CamScanner



Project - 1

Aim : To use the GUI widgets and Database operation application

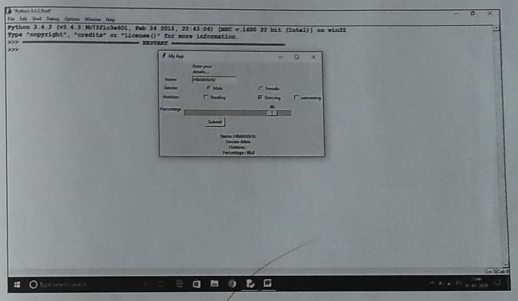
SOURCE CODE:

```
from tkinter import *
def display():
  val='\nName :'+name.get()
  val+='\nGender :'
  if gender.get()==1:
    val+='Male'
  if gender.get()==2:
    val+='Female'
  val+='\nHobbies : '
  if hobbies0.get==1:
    val+=' Reading '
  if hobbies1.get==1:
    val+=' Dancing '
  if hobbies2.get==1:
    val+=' Swimming '
  val+='\nPercentage : '+str(per.get())
  result.configure(text=val)
w1=Tk()
w1.title('My App')
title=StringVar()
m1=Message(w1, textvariable=title,width=100)
title.set('Enter your details....')
l1=Label(w1, text=' Name ')
name=StringVar()
e1=Entry(w1, textvariable=name)
l2=Label(w1, text=' Gender ')
gender=IntVar()
r1=Radiobutton(w1, text='Male', variable=gender, value=1)
r2=Radiobutton(w1, text='Female', variable=gender, value=2)
```

Scanned by CamScanner

```
l3=Label(w1, text=' Hobbies ')
hobbies0=IntVar()
hobbies1=IntVar()
hobbies2=IntVar()
ch1=Checkbutton(w1, text='Reading',variable=hobbies0)
ch2=Checkbutton(w1, text='Dancing',variable=hobbies1)
ch3=Checkbutton(w1, text='swimming',variable=hobbies2)
l4=Label(w1, text='Percentage ')
per=DoubleVar()
s1=Scale(w1, variable=per, orient=HORIZONTAL,length=300)
b1=Button(w1, text=' Submit ', command=display)
result=Label(w1)
m1.grid(row=0, column=0,columnspan=2)
l1.grid(row=1, column=0)
e1.grid(row=1, column=1)
l2.grid(row=2, column=0)
r1.grid(row=2, column=1)
r2.grid(row=2, column=2)
l3.grid(row=3, column=0)
ch1.grid(row=3, column=1)
ch2.grid(row=3, column=2)
ch3.grid(row=3, column=3)
l4.grid(row=4, column=0)
s1.grid(row=4, column=1, columnspan=2)
b1.grid(row=5, column=1)
result.grid(row=6, column=1, columnspan=2 )
w1.mainloop()
```


OUTPUT:



Scanned by CamScanner

PROJECT - 2

AIM : TO make student database using sqlite3 and tkinter

SOURCE CODE:

```
from tkinter import *
import sqlite3

root=Tk()
root.geometry('410x450')
root.title("Student DataBase using Sqlite3 and Tkinter")
root.configure(background="green")

textin=StringVar()
textinn=StringVar()

menu=Menu(root)
root.config(menu=menu)

menu = Menu(root)
root.config(menu=menu)

def helpp():
  help(sqlite3)

subm = Menu(menu)
menu.add_cascade(label="Help",menu=subm)
subm.add_command(label="Sqlite3 Docs",command=helpp)


db = sqlite3.connect('mysq.db')
cursor = db.cursor()
cursor.execute("CREATE TABLE IF NOT EXISTS people(name TEXT, phone TEXT)")
db.commit()

lab=Label(root,text='Name:',font=('none 13 bold'))
lab.place(x=0,y=0)

entname=Entry(root,width=20,font=('none 13 bold'),textvar=textin)
entname.place(x=80,y=0)

entphone=Entry(root,width=20,font=('none 13 bold'),textvar=textinn)
entphone.place(x=80,y=40)
```

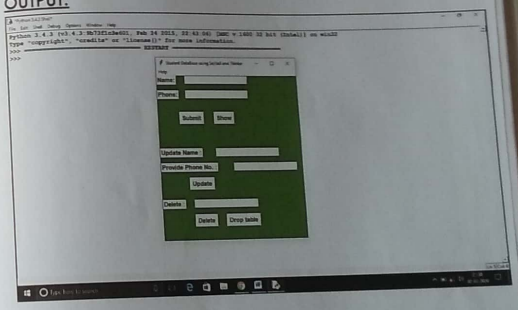


```
lab1=Label(root,text='Phone:',font=('none 13 bold'))
lab1.place(x=0,y=40)

def insert():
  name1 = textin.get()
  phone1 = textinn.get()
  conn = sqlite3.connect('mysq.db')
  with conn:
    cursor = conn.cursor()
    cursor.execute('INSERT INTO people(name, phone) VALUES(?,?)',(name1, phone1,))
    db.close()

def show():
  connt = sqlite3.connect('mysq.db')
  cursor = connt.cursor()
  cursor.execute('SELECT * FROM people')
  for row in cursor.fetchall():
    print(row)

name=StringVar()
phone=StringVar()
def updateContact():
  nam=name.get()
  ph=phone.get()

  connnt=sqlite3.connect('mysq.db')
  cursor = connnt.cursor()
  cursor.execute("UPDATE people SET name = ? WHERE phone = ?",(nam,ph))
  connnt.commit()

dell=StringVar()
def det():
  dee=dell.get()
  connnt=sqlite3.connect('mysq.db')
  cursor = connnt.cursor()
  cursor.execute("DELETE FROM people WHERE name = ?",(dee,))
  connnt.commit()


def drop():
  connnt=sqlite3.connect('mysq.db')
  cursor = connnt.cursor()
```

Scanned by CamScanner

```
    cursor.execute("DROP table people")
    connnt.commit()

  buttdrop=Button(root,padx=2,pady=2,text='Drop table',command=drop,font=('none 13
  bold'),relief='raise')
  buttdrop.place(x=180,y=380)

  buttupdate=Button(root,padx=2,pady=2,text='Update',command=updateContact,font=('none 13
  bold'),relief='raise')
  buttupdate.place(x=80,y=280)

  labuname=Label(root,text='Update Name :',font=('none 13 bold'))
  labuname.place(x=0,y=200)

  enttupadtename=Entry(root,width=20,font=('none 13 bold'),textvar=name)
  enttupadtename.place(x=160,y=200)

  labuphone=Label(root,text='Provide Phone No. :',font=('none 13 bold'))
  labuphone.place(x=0,y=240)

  entupdatephone=Entry(root,width=20,font=('none 13 bold'),textvar=phone)
  entupdatephone.place(x=210,y=240)

  labdelete=Label(root,text='Delete :',font=('none 13 bold'))
  labdelete.place(x=0,y=340)

  endelete=Entry(root,width=20,textvar=dell,font=('none 13 bold'))
  endelete.place(x=90,y=340)


  butdel=Button(root,padx=2,pady=2,text='Delete',command=det,font=('none 13 bold'))
  butdel.place(x=90,y=380)


  but=Button(root,padx=2,pady=2,text='Submit',command=insert,font=('none 13 bold'))
  but.place(x=60,y=100)

 res=Button(root,padx=2,pady=2,text='Show',command=show,font=('none 13 bold'))
 res.place(x=160,y=100)
```


OUTPUT:



Scanned by CamScanner