```
    # if any of the entry field is empty
    # then show an error message and clear
    # all the entries
    if (dayField.get() == "" or monthField.get() == ""
        or yearField.get() == "" or givenDayField.get() == ""
        or givenMonthField.get() == "" or givenYearField.get() ==
"") :

        # show the error message
        messagebox.showerror("Input Error!")

        # clearAll function calling
        clearAll()
        |
        return -1

# function to calculate Age
def calculateAge() :

    # check for error
    value = checkError()
```

Scanned with CamScanner

```
# import all functions from the tkinter
from tkinter import *

# import messagebox class from tkinter
from tkinter import messagebox

# Function for clearing the
# contents of all text entry boxes
def clearAll() :

    # deleting the content from the entry box
    dayField.delete(0, END)
    monthField.delete(0, END)
    yearField.delete(0, END)
    givenDayField.delete(0, END)
    givenMonthField.delete(0, END)
    givenYearField.delete(0, END)
    rsltDayField.delete(0, END)
    rsltMonthField.delete(0, END)
    rsltYearField.delete(0, END)

# function for checking error
def checkError() :
```

Scanned with CamScanner

```
    given_month = given_month - 1
    given_day = given_day + month[birth_month-1]


# if birth month exceeds given month, then
# donot count this year and add 12 to the
# month so that we can subtract and find out
# the difference
if (birth_month > given_month):
    given_year = given_year - 1
    given_month = given_month + 12

# calculate day, month, year
calculated_day = given_day - birth_day;
calculated_month = given_month - birth_month;
calculated_year = given_year - birth_year;

# calculated day, month, year write back
# to the respective entry boxes

# insert method inserting the
# value in the text entry box.
```

Scanned with CamScanner

```
# if error is occur then return
if value == -1 :
    return

else :

    # take a value from the respective entry boxes
    # get method returns current text as string
    birth_day = int(dayField.get())
    birth_month = int(monthField.get())
    birth_year = int(yearField.get())

    given_day = int(givenDayField.get())
    given_month = int(givenMonthField.get())
    given_year = int(givenYearField.get())


    # if birth date is greater then given birth_month
    # then donot count this month and add 30 to the date so
    # as to subtract the date and get the remaining days
    month =[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

    if (birth_day > given_day):
```

Scanned with CamScanner

```
    rsltDayField.insert(10, str(calculated_day))
    rsltMonthField.insert(10, str(calculated_month))
    rsltYearField.insert(10, str(calculated_year))


# Driver Code
if __name__ == "__main__" :

  # Create a GUI window
  gui = Tk()

  # Set the background colour of GUI window
  gui.configure(background = "light green")

  # set the name of tkinter GUI window
  gui.title("Age Calculator")

   # Set the configuration of GUI window
  gui.geometry("525x260")

  # Create a Date Of Birth : label
  dob = Label(gui, text = "Date Of Birth", bg = "blue")
```

Scanned with CamScanner

```
# Create a Given Date : label
givenDate = Label(gui, text = "Today's Date", bg = "blue")

# Create a Day : label
day = Label(gui, text = "Day", bg = "light green")

# Create a Month : label
month = Label(gui, text = "Month", bg = "light green")

# Create a Year : label
year = Label(gui, text = "Year", bg = "light green")

# Create a Given Day : label
givenDay = Label(gui, text = "Given Day", bg = "light
green")

# Create a Given Month : label
givenMonth = Label(gui, text = "Given Month", bg = "light
green")

# Create a Given Year : label
givenYear = Label(gui, text = "Given Year", bg = "light
green")
```

Scanned with CamScanner

```
# Create a Years : label
rsltYear = Label(gui, text = "Years", bg = "light green")

# Create a Months : label
rsltMonth = Label(gui, text = "Months", bg = "light green")

# Create a Days : label
rsltDay = Label(gui, text = "Days", bg = "light green")

# Create a Resultant Age Button and attached to
calculateAge function
resultantAge = Button(gui, text = "Resultant Age", fg =
"Black", bg = "Red", command = calculateAge)

# Create a Clear All Button and attached to clearAll
function
clearAllEntry = Button(gui, text = "Clear All", fg = "Black",
bg = "Red", command = clearAll)

# Create a text entry box for filling or typing the
information.
dayField = Entry(gui)
monthField = Entry(gui)
yearField = Entry(gui)
```

Scanned with CamScanner

```
givenDayField = Entry(gui)
givenMonthField = Entry(gui)
givenYearField = Entry(gui)

rsltYearField = Entry(gui)
rsltMonthField = Entry(gui)
rsltDayField = Entry(gui)


# grid method is used for placing
# the widgets at respective positions
# in table like structure .
dob.grid(row = 0, column = 1)

day.grid(row = 1, column = 0)
dayField.grid(row = 1, column = 1)

month.grid(row = 2, column = 0)
monthField.grid(row = 2, column = 1)

year.grid(row = 3, column = 0)
yearField.grid(row = 3, column = 1)
```

Scanned with CamScanner

```
clearAllEntry.grid(row = 12, column = 2)

# Start the GUI
gui.mainloop()
```

Scanned with CamScanner

```
givenDate.grid(row = 0, column = 4)

givenDay.grid(row = 1, column = 3)
givenDayField.grid(row = 1, column = 4)

givenMonth.grid(row = 2, column = 3)
givenMonthField.grid(row = 2, column = 4)

givenYear.grid(row = 3, column = 3)
givenYearField.grid(row = 3, column = 4)

resultantAge.grid(row = 4, column = 2)

rsltYear.grid(row = 5, column = 2)
rsltYearField.grid(row = 6, column = 2)

rsltMonth.grid(row = 7, column = 2)
rsltMonthField.grid(row = 8, column = 2)

rsltDay.grid(row = 9, column = 2)
rsltDayField.grid(row = 10, column = 2)
```

Scanned with CamScanner

Age Calculator

Date Of Birth

Day

Month

Year

Today's Date

Given Day

Given Month

Given Year

Resultant Age

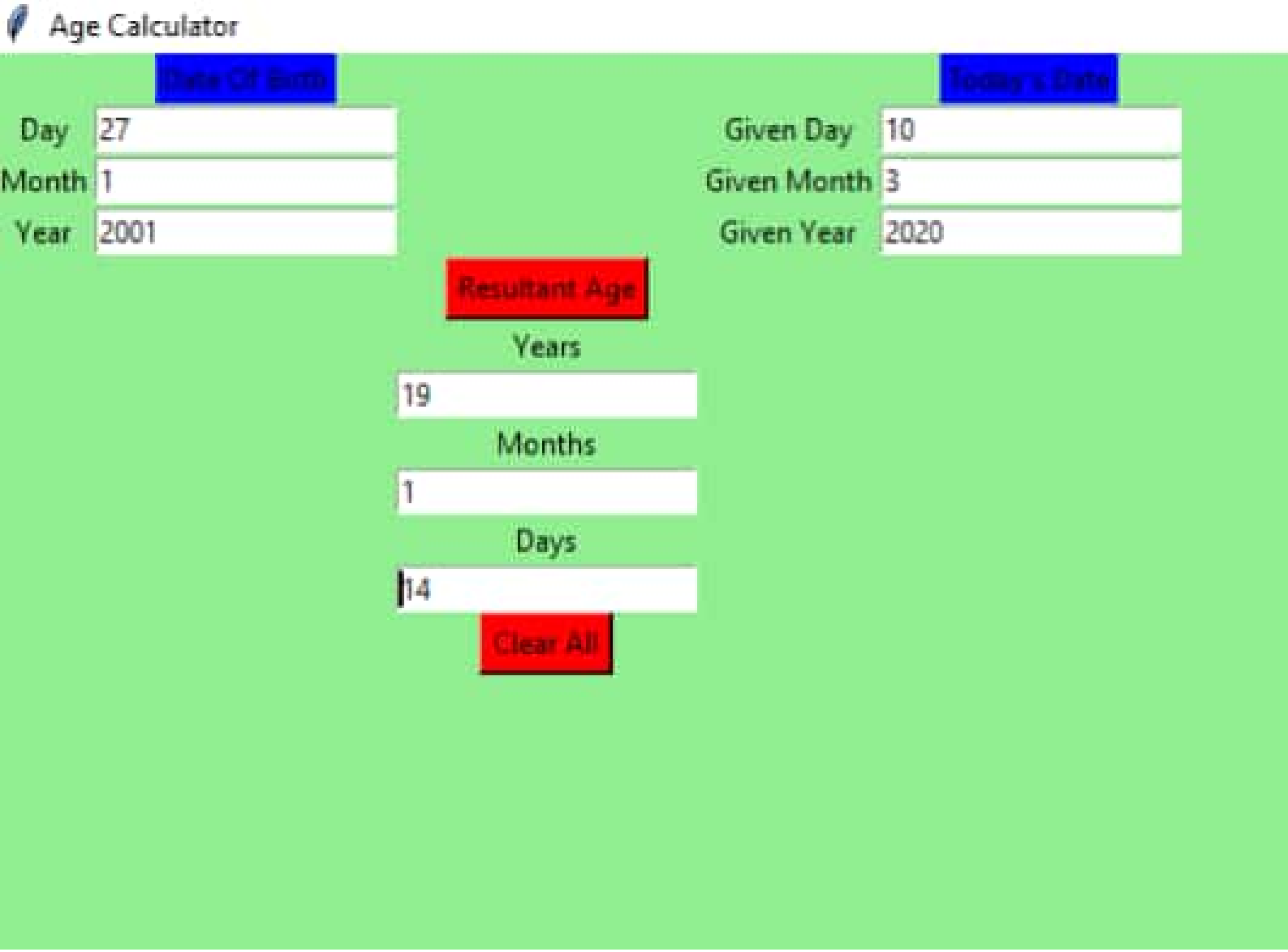
Years

Months

Days

Clear All

Scanned with CamScanner



Scanned with CamScanner

Practical- 1

Objective :- Demonstrate the use of different file accessing modes different attributes read methods

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

Step 2 :- Now open the file in read mode & then we read() readline & store the output is variable & finally display the contents of variable

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

Scanned with CamScanner

```
file obj = open("sample.txt", "w")
fileobj.write("computer science subjects" + "\n")
fileobj.write("DBMS \n PYTHON \n FOSS \n")
fileobj.close()
#read()
fileobj = open("sample.txt","r")
str1 = fileobj.read()
print("The output of read method:", str1)
fileobj.close()
>>> ('The output of read method : ', 'Computer science
    subject \n DBMS \n Python \n FOSS \n')

#readline()
fileobj = open("sample.txt","r")
str2 = fileobj.readline()
print("The output of readline method:", str2)
>>> ('The output of readline method : ', 'Comp. sci subjec
#readlines()
fileobj = open("sample.txt","r")
str3 = fileobj.readlines()
Print("The output of readlines method :", str3)
>>> ('The output of readlines method : ', ["Computer Science
    'DBMS\n', 'Python\n', 'FOSS\n']
#file attributes()
a = fileobj.name
print("Name of file (name attributes) :", a)
>>>(name of file (name attributes) ', 'sample.txt')
b = fileobj.closed
print("(close) attribute :", b)
>>> ("(close) attrib
```

Scanned with CamScanner

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

Step 5:- Open file obj in read mode display the update written contents & close open again in 'rt' mode with parameter parral & display. The output subsequently.

Step 6:- Now open file obj in append mode open write method write contents. close the file obj again open the file obj in read mode & display the 'appending' output.

Scanned with CamScanner

```
c = fileobj.mode
print ("filemode ", c)
>>> ('filemode', 'r')
d = fileobj.softspace
print ("softspace ", d)
>>> ('softspace : ', 0)
```

w+ mode

```
fileobj = open("sample.txt", "w+")
fileobj.write("loutik sir")
fileobj.close()
```

r+ mode

```
fileobj = open("sample.txt", "r+")
g1 = fileobj.read(6)
print ("output of r+", g1)
fileobj.close()
>>> ('output of r+ ', 'loutik')
```

append mode

```
fileobj = open("sampl.txt", "a")
fileobj.write("Data Analysis")
fileobj.close()
fileobj = open("sample.txt", "r")
gs = fileobj.read()
print ("output of append mode: ", gs)
fileobj.close()
>>> ('output of append mode : ', 'loutik sir
Data Analysis')
```

write mode

```
fileobj = open("sample.txt", "w")
fileobj.write("DBMS")
fileobj.close()
```

read mode

```
fileobj = open("sample.txt", "r")
g = fileobj.read()
print ("output of readmode", g)
>>> ('output of read mode', 'loutik Sir')
```

Scanned with CamScanner

```
#tell()
fileobj = open ("sample.txt", "r")
pos = fileobj .tell()
print ("tell() = " ,pos)
fileobj.close
>>> ('tell() = ', 0L)

#seek()
fileobj = open ("sample.txt", "r")
st = fileobj.seek(0,0)
print ("seek (0,0) is = ", st)
fileobj.close()
>>> ('seek (0,0) is = ', none)
fileobj = open ("sample.txt", "r")
st1 = fileobj.seek(0,1)
print ("seek (0,1) is : ", st1)
fileobj.close()
>>> ('seek (0,1) is , ', none)
fileobj = open ("sample.txt", "r")
st2 = fileobj.seek(0,2)
print ("seek (0,2) is = ", st2)
fileobj.close()
>>> ('seek(0,2) is: ", none)

# finding length of different lines exist within lines
fileobj = open ("sample.txt", "r")
stat = fileobj.readlines()
print ("output = ", stat)
for line in stat:
    print (len(line))
fileobj.close()
>>> ('output: ', ['Mouttin Sign, Data Analytics'])  26
```

Scanned with CamScanner

Step7:- Open the file obj in read mode, declare a variable & perform file object dot tell method & store the output consequently in variable

Step8:- Use the seek method with the arguments with opening the file obj in read mode & closing subsequently

Step9:- Open file obj with read mode also use the readlines method & store the output consequently in and print the var for counting the length use for conditional statement & display the length.

19/12

Scanned with CamScanner

Practical - 2

Aim :- Iterators

1) Iterators

Step1 :- Define a iter method with an argument & initalize it to the 1st value.

Step2 :- for extracting the next element from the container, use the next method with an argument & compare the no. of elements req. in a container by using the conditional statement.

Step3 :- Now create an object from the given class & pass this object as an argument to iter meth

Step4 :- Now using the coditional statement display all the values from the given container

(11) Step5 :- Define a function name square with a parameter which will obtain output of square values of the given number

Step6 :- Call the declared function using function call.

Scanned with CamScanner

```
class myrange:
    def __iter__(self):
        self.a = 1
        return self
    def __next__(self):
        if self.a < 30:
            x = self.a
            self += 1
            return x
        else:
            raise StopIteration
myclass = myrange()
mydata = iter(myclass)
for x in mydata:
    print(x)
```

Output

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
16
17
18
19
20
21
22
23
24
25
26
27
28
30
```

Scanned with CamScanner

```
class fact:
    def__iter__(m):
        m.value = 1
        return m
    def__next__(m):
        factor = 1
        for i in range(1, m.value+1):
            fact o = fact o * (i)
        print("the factorial", m.value, "is", fact
        m.value += 1
        if (m.value == 5):
            raise stop iteration
y = iter(fact5)
while true:
    next(y)
```

Output

The factorial 1 is : 1

The factorial 2 is : 2

The factorial 3 is : 6

The factorial 4 is : 24

The factorial 5 is : 120

The factorial 6 is : 720

Scanned with CamScanner

Step 7:- for extracting nextvalue from container use the next method with an argument & compare the container using condn statement.

Step 8:- Now create an object from the given class & pass this object as an argument to the iter method.

Step 9:- Now using the conditional method stat. display all the values from the given container

3) factorial

Step 10:- Define an object of a class

Step 11:- Accept a number from the user till which we want to display the factorial of the no.

Step 12:- Define a function with if condn if exactly input is to 1 then return one else parameter

Step 13:- Enter a input from the declare an empty list use append method, for appending the input value

Print the output value.

Scanned with CamScanner

```
class myiter:
    def __iter__(pow):
        pow.no = 1
        pow.nos = int (inp
```

Scanned with CamScanner

Practical No:-3

Aim:- Program to demonstrate exception handling

1] Write a program using the exception method of the nature Arithmatic error

step 1: Use the try block & except the input using the raw input method & connect it into the integer datatype & subsequently terminate the block

step 2: Use the except block with the exception name as value error & display the appropriate message if the suspecious code is part of the try block

2] W.A.P for accepting the file in a given mode & use the environment error as an exception for the given input

step 1: within the try block open the file using the write mode & some content on the file.

step 2: Use the except block with IO error & display the message regarding missing of the file or incompatibility of the mode use the else block to display a message that the operation is carried out successfully

Scanned with CamScanner

```
# Program.
while True
    try
        x = int(input("Enter class"))
        break
    except ValueError
        print("Enter Numeric Value")
```

Output:

```
enter class : 467
```

```
# Program -
try
    fo = open("abc.txt", "w")
            ("Shubham Jain")
except IOError
    print("Error writing on the file")
else:
    print("operation carried out successfully")
    fo close()
```

Output:

```
Operation carried out successfully
```

Scanned with CamScanner

#program

```
def assert_(n):
    assert (len(n) == 0)
    print ("list is empty")
var1 = [ ]
print (assert_(var1))
```

Output:-

list is empty

#program

```
def acceptage():
    age = int(input ("Enter age :- "))
    if age > 30 or age < 16:
        raise ValueError
    return age
Valid = false
while not valid:
  try
    age = acceptage()
    valid = true
  except ValueError:
        print ("Not a valid age")
```

Output:

Enter age : 4
Not a valid age
Enter age 18

Scanned with CamScanner

3) W.A.P using the assert() to check if the list elements are empty

Step 1: Define a function which accepts an argument & check using the assert statement whether the given list is empty list & accordingly return the message

Step 2: Close the function & in the body of program & define certain elements in list & take some appropriate action.

4) W.A.P to check the range of the age of the students in given class & if the age do not fall in given range else the value error exception otherwise return the valid no.

Step 1: Define a function which will accept the age of the student from the standard input

Step 2: Use the if condition to check whether the input age falls in the range & so return the age else use the value error exception

Scanned with CamScanner

Step 3: Define the while loop to check whether the boolean expression holds true. Use the try block to accept the age of student & terminate the looping condition.

Step 4: Use except with value Error & print the message not a valid range

Dr
18/07

Scanned with CamScanner

code

```
import re
string = "hello 1234 abc4567"
result = re.findall ("\d", string)
result = re.findall ("\!", string)
print (result)
print (result)
```

output

```
>>> ['1234', '4567']
>>> ['hello', 'abc']
```

Scanned with CamScanner

Aim:- Demonstrate the use of regular exp.

Theory:- Regular expression represents the sequence of character which is mainly used for finding & replacing the given pattern in a string and for this we impor removable and common usage of regular expression involves following functionalit
- Searching a given string
- finding a string
- Breaking a string into smaller substrin
- Replacing part of string

a) Write a regular segregating numeric & alphabetic values from a given string

Algorithm:-

Step1: Now apply string & pattern in finda and display the output.

Step: \d is used for matching all decima digit whereas is used to matc non-decimal digits

Scanned with CamScanner

Q2 Write a regular expression for finding the match string at the beginning of given sequence.

Algorithm

Step1: Import and apply a string

Step2: Use search() with "\A Python" and string as two parameters

Step3: Now display the output.

Step4: Now use if conditional for user to know whether the match is found or not

Scanned with CamScanner

```
import re
string = "Python is a programming language"
result = re.search("\Apython", string)
print(result)
if result:
    print("match found")
else:
    print("match not found")
```

output:

```
>>> rematch object.span = (0, 6),
        match = "Python"
>> match found
```

Scanned with CamScanner

```
#code
import re
li = ["9876543210", "8765432109", "7654321098",
                                "6543210987"]
for element in li:
    result = re.match ["[8-9]{1}[0-9]{9}",
                                element)
if result:
    print("correct mobile no")
    print (result.group(1))
else:
    print ("Incorrect mobile no.")
```

```
# output
>>> correct mobile no.
  9876543210
  correct mobile no.
  8765432109
  Incorrect mobile no.
```

Scanned with CamScanner

Q3 Write a regular expression to check the given mobile number starts with 8 or 9 & the total length 10

Algorithm:

Step1: Import re module & apply a string module no. 8

Step2: Now use for conditional statement to find if the number starts with 8 or 9 and the total number should be length of 10. Use match() inside for statement to find the number in given string

Step3: Use if conditional statement to know whether we have match or not if we have use groups() to display output and if we don display incorrect no.

Scanned with CamScanner

Q1 Write a regular expression for subtract a word from given string with numbers characters in between the & the word subsequently subtract. the word without space character

Algorithm

Step1. Import re.module and apply a string

Step2. Use findall() to ~~subtract~~ or extract a word from given string

Step 3. Use "\w" to extract word along with space & use "\w+" to extract without

Step4: Now display the output

Scanned with CamScanner

#code

```
import re
string= "Python is important"
result 1 = re.findall("\w.", string)
result 2 = re.findall ("\Wt", string)
print (result 1)
print (result 2)
```

#output

```
>>> ['Python', ' : ; ', ' important.']
['Python', ' ; ]', ' important']
```

16/07

Scanned with CamScanner

```
#code
import re
string = " Python is a programming language
result = re.findall ["n\w+", string]
result1 . re.findall ["\w+", string]
print (result)
print (result1)

#output
>>> ['Python']
>>>['important']
```

Scanned with CamScanner

05 Write a regular expression for extracting first and last word from a string

Algorithm

Step 1: Import re module and apply a strin

Step 2: Use findall() in which use r"\w
as one parameter to find first
word of string then use "\w
as parameter to find last wo
of string

Step 3: display the output.

Scanned with CamScanner

Write a regular expression for extracting the data in format dd.mm.yyyy by using the findall () where the string has following format AMIT 201 24-12-2019

Algorithm

Step 1: Import re module and apply string

Step 2: Use findall method and use \d 23 - \d{2} - \d{4} as an parameter

Step 3: Now display the output.

Scanned with CamScanner

```
#code
import. re
string = "Unit 201 24-12-2014"
result = re.findall ["\d {2}-\d{2}-\d{4}",
                     string)
print (result)

#output
>>> ['24-12-2014']
```

lab

27/a

Scanned with CamScanner

Code

```
import re
string = "abc@tcsc.edu"
result1 = re.findall("\w+", string)
result2 = re.findall("+\w+.\w+$", string)
result3 = re.findall("\w|...]+", string)
print(result1)
print(result2)
print(result3)
```

#output

```
>>> ['abc']
>>> ['tcsc.edu']
>>> ['abc','tcsc' edu']
```

Scanned with CamScanner

Aim: Write a re for extracting two
1) Username from email-id
2) Hostname from email-id
3) Username & hostname from email id.

Algorithm

Step1: Import re. module and apply a string

Step2: Use findall() to find username, hostname & or email id

Step3: Use "\w+" for username use "+\w.\w+$" for hostname and use "[\w\.-]+" for both as parameter in findall()

Step 4: Display the output

Scanned with CamScanner

Pratical - 5

Topic:- GUI Components

Step1: Use the tkinter library for importing the features of the text widget.

Step2: Create an object using the TK()

Step3: Create a variable using the widget label and use the method.

Step4: Use the mainloop() for triggering of the corresponding above mention events

#2

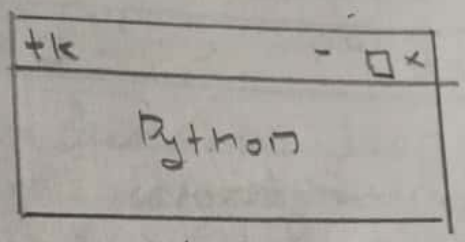
Step1: Use the tkinter library for importing the features of the text widget.

Step2: Create a variable from the text method and position it on the parent window

Scanned with CamScanner

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

output :-

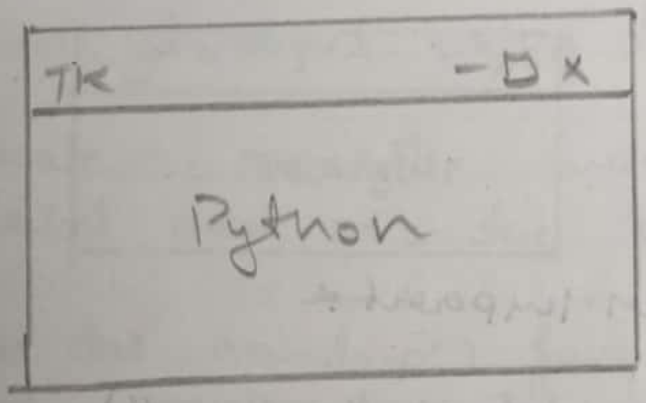


#2

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

```
l4 = Label (root, text = "CS1", bg = "orange",
            fg = "black", font = "10")
l4.pack (side = TOP, ipady = 50)
root.mainloop()
```

Output:



Scanned with CamScanner

Step 3: Use the pack() along with the object created from the text() and use the parameter
1) side = LEFT , padx = 20
2) side = LEFT , pady = 30
3) side = TOP , ipadx = 40
4) side = TOP , ipady = 50

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

Step 5: Now repeat above steps with the label() which takes the following arguments
1) Name of the parent window
2) Text attribute which defines the string
3) The background color (bg)
4) The background fg and then use the pack() with a relevant padding attributes

Nat
Jm 27/12

Scanned with CamScanner

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 () defining specific fixed size of the parent window

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

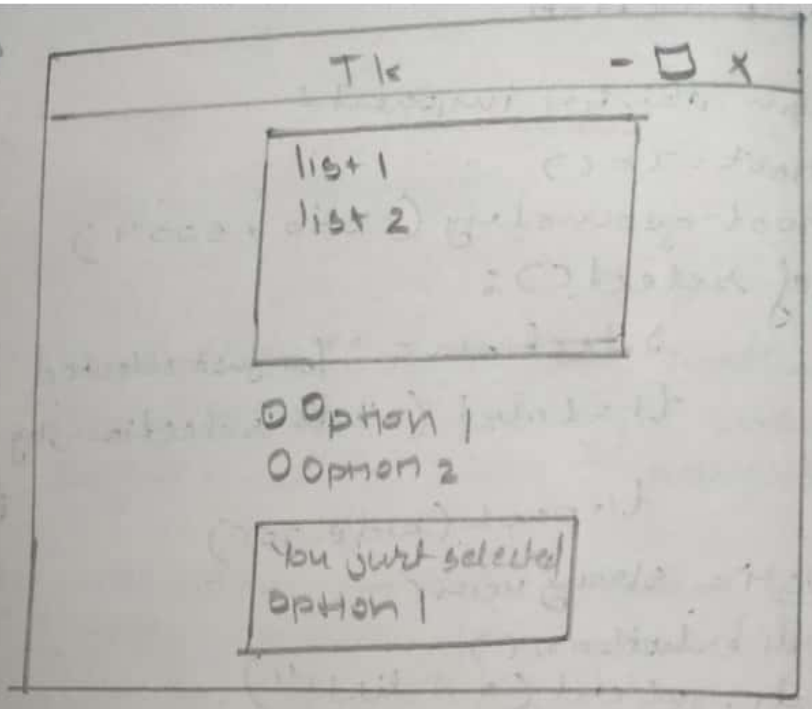
Step 4: Now define the parent window and define the option with control variable

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

Scanned with CamScanner

```
#Radio button
from tkinter import *
root = Tk()
root.geometry("500x500")
def select():
        selection = "You just selected" + str(var.get())
        t1 = Label(text=selection, bg="White",
                                        fg="green")
        t1.pack(side=TOP)
var = StringVar()
l1 = listbox()
l1.insert(2,"list1")
l2.insert(2,"list2")
l1.pack(anchor=N)
r1 = Radiobutton(root, text="option1", variable=
                var.value="option1", command=
                                select)
r1.pack(anchor=N)
r2 = RadioButton(root, text="option2", variable=
                var.value="option2", command=
                                select)
r2.pack(anchor=N)
root.mainloop()
```

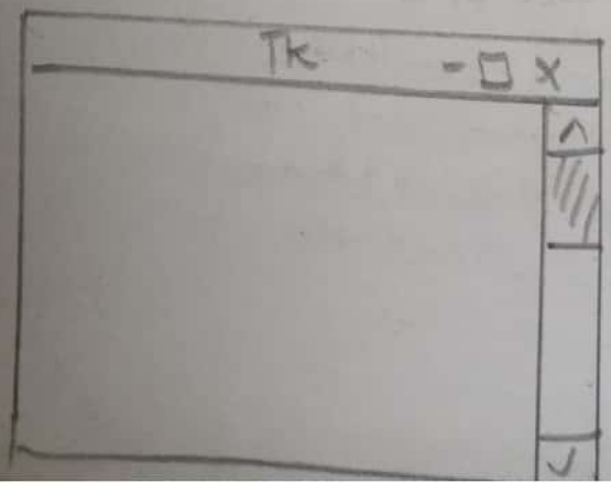
Scanned with CamScanner



#2

```
scrollbar()
from tkinter import*
root=Tk()
root.geometry("300x300")
s=scrollbar()
s.pack(side="right", fill='y')
root.mainloop()
```

Output:-



Scanned with CamScanner

step 6:- Create an object from radiobutton which will take following arguments (parent window object, text variable which will take the values option no. 1, 2, 3 ... variable argument, corresponding values & trigger the function declared

step 7:- Now call the pack() for radio object so created & specify the argument using anchor attribute

step 8:- finally make use of the mainloop() above 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 ~~pack() along~~ geometry() with the resolution for laying the window

step 4 Create an object & use the scrollbar()

step 5 Use the pack() along with the scrollbar object with side & fill attribute

Scanned with CamScanner

Step6:- Use the mainloop with the parent object

#3

Step1:- Import the relevant libraries from the tkinter method

Step2:- Create an corresponding object of the parent window

Step3:- Use the label widget along with the parent object created & subsequently use the pack method

Step4:- Use the geometry manager with pixel size (680×500) suitable pixel value

Step5:- Use the frame widget along with the ~~parent~~ object attributes created & use the pack method.

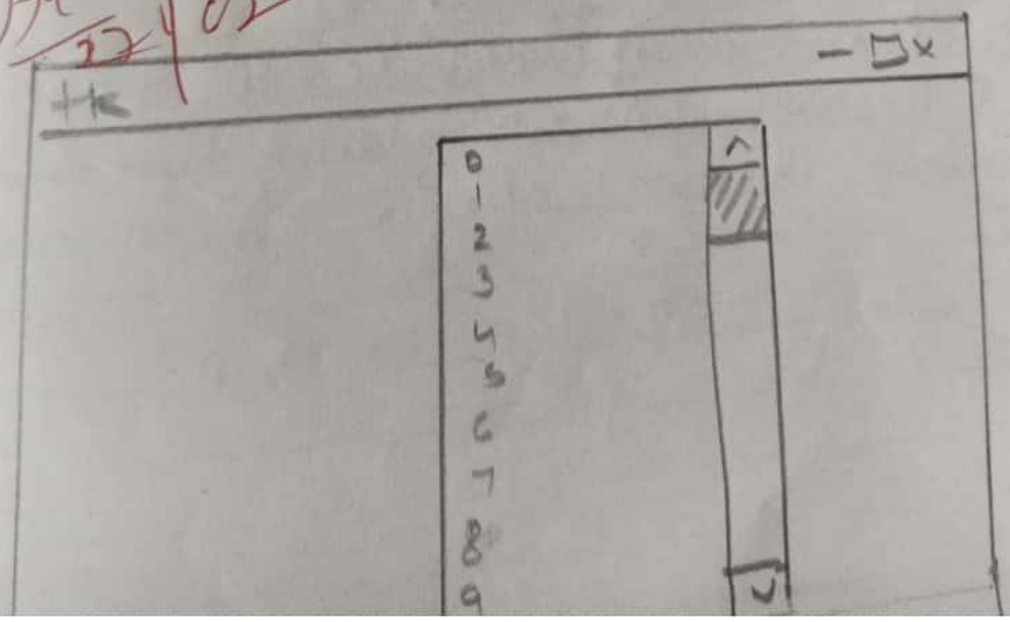
Step6:- Use the listbox method along with the attributes like width, height, font, DO create a listbox method object use pack() for the same

Step7:- Use the scrollbar() with an object w the attributes of vertical.

Step8:- Trigger the event with

```
#3 # using from widget
from tkinter import *
window=Tk()
window.geometry("680x500")
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()
```

lab
Dr 22/03



Scanned with CamScanner

```
#message Box
from tkinter import *
import tkmessagebox
root=Tk()
def func:
    tkmessageBox.showinfo ("info window", "Pyth
    b1=Button (root, text = "Python version info",
                command = func)

    b1.pack()
    root.mainloop()
```

Scanned with CamScanner

Practical - 5(c)

Aim:- GUI components.

Step1:- Import the relevant methods from the tkinter library

Step2:- Import tkmessagebox

Step3:- Define a parent window object along with the parent window.

Step4:- Define a function which will use tk messagebox with showinfo method along with info window attribute.

Step5:- Declare a button with parent window object along with the command attribute.

Step6:- Place the button widget onto the parent window & finally call mainloop() for the triggering of events called above

Scanned with CamScanner

Step1: Import the relevant methods fr
the tkinter library along with pa
window object declared

Step2: Use parent window object cla
with minsize function for wind
size

Step3:- Define a function main() declare
parent window object & use config(
title(), minsize(), title() as well a
Button() & use pack() & main
simultaneously

Step4:- Similarly define the function rec
& use the attribute accordingly

Step5:- Declare customer function button
along with parent object & declare
button with attribute like FLAT, RI
GROOVE, RAISED, along with relief
widget.

Step6:- Finally call the mainloop() for ev
programming

Scanned with CamScanner

```
#multipale 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 In Places of
            Interest In Golden Gate Bridge In lom
            Street In Coit Tower)
    l.pack()
    b1 = Button(top, text = " next", command : recom
    b1.pack (side = RIGHT)
    b2 : Button (top, text = " exit, command : termine
    b2.pack (side = LEFT)
    top.mainloop()
```

Scanned with CamScanner

```
b5 = Button (top3, text "Bridge Cutter", delay=
top3.mainloop
def terminate():
    quit()
b5 = Button (root, text="TOUR DETAILS", command=
b5.pack()
b6 = Button (root, text="Button Details", command=
                                        Button)
b6.pack()
root.mainloop
```

Scanned with CamScanner

```
def second():
    top2 = Tk()
    top2.config(bg="orange")
    top2.title("About us!")
    top2.minsize(300,300)
    L = label(top2, text="Created by: Rachit
        for more details contact to our
        official account")
    L.pack()
    b3 = Button(top2, text="prev", command=main
    b3.pack(side=left)
    b2 = Button(top2, text="exit", command=
                                terminate)
    b2.pack(side=right)
    top2.mainloop()
```

lat

```
def button():
    top3 = Tk()
    top3.geometry("300x300")
    b1 = Button(top3, text="flat button", relief=FLAT
    b1.pack()
    b2 = Button(top3, text "group button", relief=
                                GROOVE)
    b2.pack
    b3 = Button(top3, text="sunken button,
                                relief=SUNKEN)
    b3.pack()
```

Scanned with CamScanner

Practical - 5(D)

* Displaying the image

Algorithm

Step1: Create an object corresponding to parent window & use the following 3 methods, Title, mesrize, config

Step2: Create a left frame object the frame method & place it on patient window with the height & bg specified

Step3: Now - create a right frame of frame object from the frame method with width, height, specified

Step4: Create a label object from the label method & place it onto the left with text attribute denoting the ... ing with relief attribute used as RAISED value & subsequently ... geud method

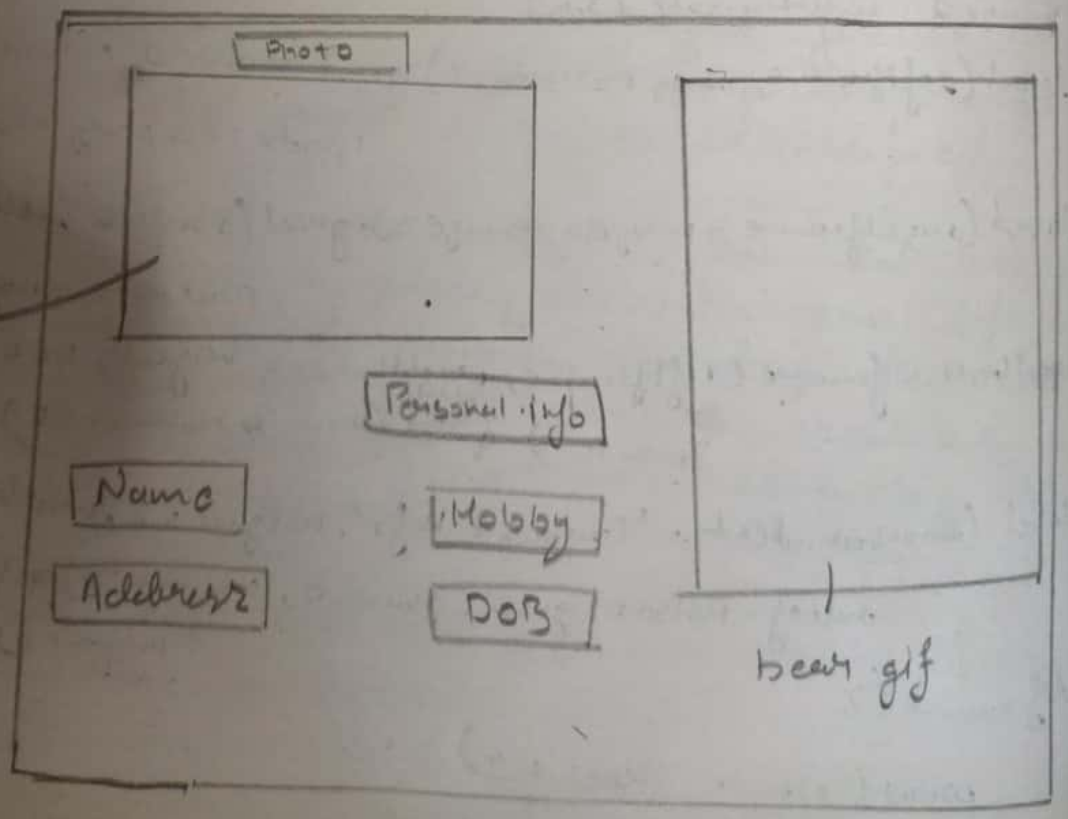
Step5:- Now use the photo image method the file attribute ...

Scanned with CamScanner

```
from tkinter import *
root=tk()
root.title("Python GUI")
root.maxsize(900,800)
root.config (bg="black")
leftframe = frame(root,bg="yellow", height="300")
leftframe.grid (row=0, column=0)
rightframe = frame (root, bg="blue", height="300", width="150")
rightframe.grid (row=0, column=0)
label (leftframe, text="Photo, height=2, width=20)
                grid (row=0, column=0)
image1= PhotoImage (file="bear.gif")
image1.subsample (1,2)
image2 = PhotoImage (file="bear.gif")
image2.subsample (3,4)
label (leftframe, image=image1).grid (row=0, column=0,
                padx=20, pady=10)
label (rightframe, image=image2).grid (row=0 column=1,
                padx=10, pady=10)
toolbar = frame (leftframe, width=200, height=400, bg=
                "white") grid (row=2, column=0)
label (toolbar, text="Personal Info", height=2, width=2,
                relief=RAISED) grid (row=0, column=0, padx=20,
                pady=10)
def name():
    print("Name: Rachit")
```

Scanned with CamScanner

```
def hob():
    print ("Hobby: Coding")
def add():
    print ("Address: Mumbai")
def dob():
    print ("DOB= 21/10/2001")
Button(toolbar, text= "Name", height=1,
        width=16, cmd=Name).grid(row=1, col=2
Button(toolbar, text= "Hobby", hegt=1, w=16,
        command= hob).grid(row=1, col=1)
Button(toolbar, text="Address", height=1, width
        cmd=add).grid(row=2, col=0
Button(toolbar, text = "DOB", height=1, width=16
        command: dob).grid(row=2, col=2
root.mainloop()
```



Scanned with CamScanner

Step 6:- Use the sub sample method with the object of the img. & give the x, y coordinate values.

Step 7:- Use the label method & position it onto the left frame and place the img. after the sampling.

Step 8 :- Create another label object positioning it onto the right frame & specifying the image & background attribute with row & column attribute specify it as (0,0)

Step 9:- Now create a toolbar object from the frame method & position it onto the left frame with height & width specified at second row.

Step 10 - Now define the various function for different tool bar options provided in the left frame

Step 11:- From the label method position the text on the toolbar use the relief attribute and corresponding grid value.

Step 12:- Create the label method position it on the toolbar with next title at 10 second row but

Scanned with CamScanner

Step13:- Now make use of mainloop method

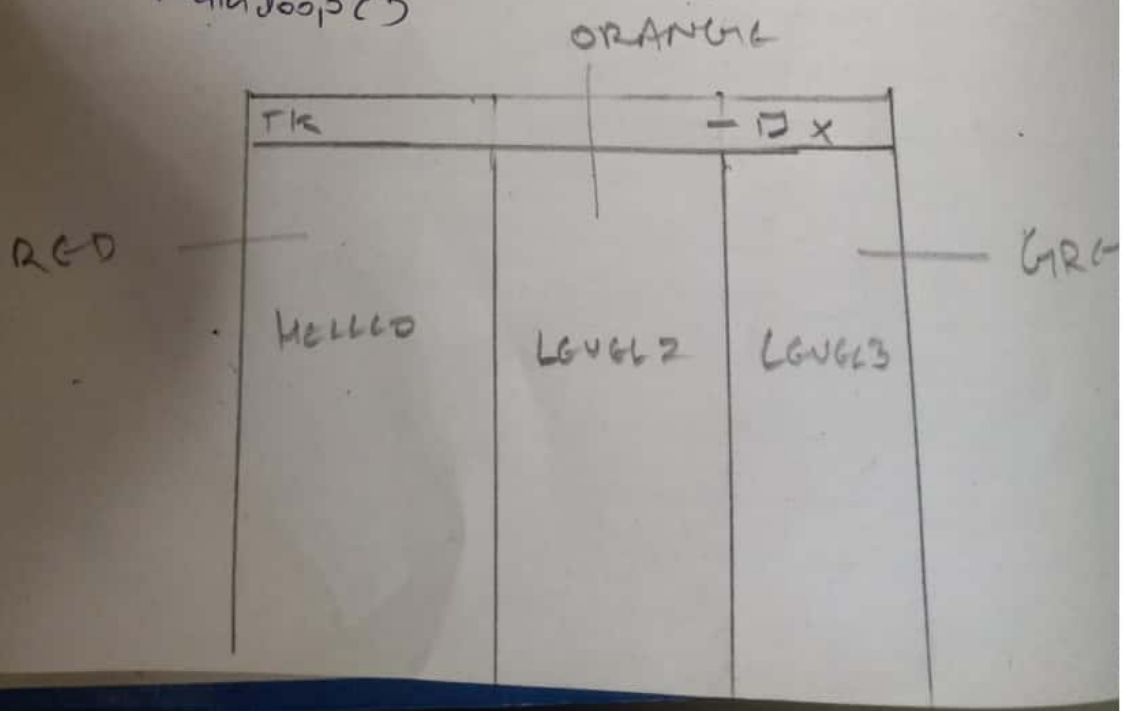
Lab

27/11

Scanned with CamScanner

Code:-

```
from tkinter import *
root=Tk()
P=PanedWindow(bg="Pink")
P.pack(fill=Both, expand=10)
L1=Label(P.text="HELLO", bg="RED")
P.add[L1]
P1=PanedWindow(P.orient=VERTICAL, bg="Yel
P.add(P1)
L2=Label(P1.text="LEVEL2", bg="green")
P1.add(L2)
P2=PanedWindow(P.orient=HORIZONTAL, bg
L3=Label(P.text="LEVEL3", bg="orange")
P.add(P2)
P2.add(L3)
mainloop()
```



Scanned with CamScanner

Practical- 5(c)

Paned Windows:-

Step 1: Create an Object from the paned window method & use the pack method to make this object visible

Step 2: Now create an object from the entry widget & placed it onto the paned window & use the add meth similarly, create an object of a paned window.

Step 3: Create an object from the scale widget & place it onto the preceeding paned window and the add method accordingly

Step 4: Create a button widget & place onto the paned window define functionality along with the widget

Step 5:- Use the pack method & mainlo method for the corresponding ev trigger

* Canvas widget

Step1: Create an object from the canvas widget by using the attribute height width bg colour & the parent window object.

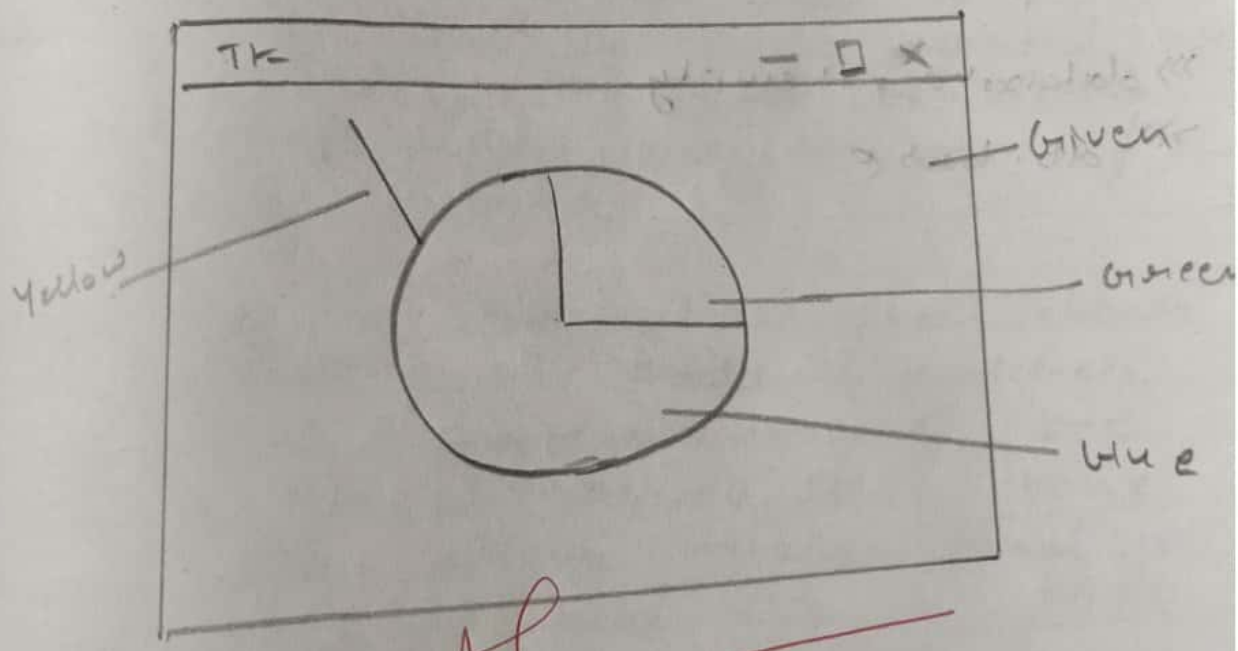
Step2: Use the corresponding method for drawing the simple geometric shape like oval & line & specify co-ordinate values

Step3: Similarly use the create line & create oval method along with the co-ordinate values & fill attribute for specifying the colour

Step4:- Finally use the pack & mainloop method

Scanned with CamScanner

```
from tkinter import *
root = Tk()
cl = Canvas(root.height = 5000, width=5000,
                                bg="GREEN")
oval = cl.Create_oval(300,12,12,400,fill="RED")
line = cl.Create_line(30,20,70,60,fill="yellow")
arc = cl.Create_arc(300,18,18,400,fill="blue")
cl.pack(side=TOP)
root.mainloop()
```



last
27/12

```
import dbm
db = dbm.open("database", flag='c', mode=438)
~~db = dbm.o~~
db["http://www.google.com"] = "" ("student.db")
if db("http://www.google.com") != None:
        print ("database not empty
else:

    print ("database empty")
>>> database not empty
>>> db.keys()
```

Scanned with CamScanner

Aim:- Database Connectivity

ALGORITHM:

Step 1:- Import the DBM library & use the open() for creating the database by supplying the name of the database along with corresponding flag.

Step 2:- Use the object so created for accessing the given website & corresponding regular name for the website.

Step 3:- Check whether the given ~~user name~~ URL address matches with regular name of the page is equal to NULL none then display message found or else not found

Step 4:- Use the close() to terminate database library

Scanned with CamScanner

II step1: Import OS & Sqlite-3 libraries make database connection

step2: Now create the connection obj using SQL Lite-3 library and connect() for creating new data-base.

step3: Now create cursor object using SQL the cursor() from the connection object created.

step4: Now use the execute() for creating the table with column name & representative data-type

step5: Now with cursor object use the insert statement for entering the values corresponding to different fields corresponding the data type

step6: Use the commit() to complete the transaction using the connection object

step7: Use the execute statement along with cursor object for accessing the values from data base

Scanned with CamScanner

```
import os, sqlite3
con = sqlite3.connect ("Student.db")
cur = con.cursor()
cur.execute ('create table tos (Name char,
                Rollno int')
cur.execute ('insert into dos value ('abc',
                '1789'), ('xyz'), '1786')
con.commit()
cur.execute ('select * from dos')
print (cur.fetchall())
con.close.
```

output:

```
['('Abc', '1789'), ('Shubham', '1786')]
```

Scanned with CamScanner

step 8: finally use the fetch() or fetchall() for displaying the value from the table using cursor object

Scanned with CamScanner