

PRACTICAL-1

OBJECTIVE: Demonstrate the use of different file accessing modes different attributes and methods.

Step 1: Create a file object using open method and use the write access method 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 file object for finding the name of the file, the file mode in which it's opened whether the file is still open or close and finally the output of the softspace attribute.

```
fill obj = open ("abc.txt", "w")
fill obj . write (" computer science sub " + "\n") 024
fill obj . write (" DOMS in PYTHON AND \n")
fill obj . close ()
```

```
fill obj = open ("abc.txt", "r")
str1 = fill obj . read ()
print ("The output of read method : ", str1)
fill obj . close ()
```

```
fill obj = open ("abc.txt", "r")
str2 = fill obj . readline ()
print ("The output of readline method : ", str2)
fill obj . close ()
```

```
fill obj = open ("abc.txt", "r")
str3 = fill obj . readlines ()
print ("The output of readlines method : ", str3)
fill obj . close ()
```

Q50

a = file obj . name
print ("name of file (name attribute): ", a)

b = file obj . closed
print ("(closed) attribute: ", b)

c = file obj . mode
print ("file mode: ", c)

file obj = open ("late.txt", "w")
file obj . write ("DBMS")
file obj . close()

d = file obj . softspace
print ("soft space: ", d)

file obj = open ("abc.txt", "wt")
file obj . write ("chennai bro")
file obj . close()

file obj = open ("abc.txt", "r")
s = file obj . read()
print ("output of read: ", s)

file obj = open ("abc.txt", "rt")
s1 = file obj . read()
print ("output of r+", s1)
file obj . close()

Step 4: Now open the file obj in write mode with some another content. close subsequently then again open the file obj in 'w+' mode that is the update mode and write contents.

Step 5: Open file obj in read mode display the update written contents and close open again in 'r+' mode with parameters passed and display the output subsequently.

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

PSO

Step 7: Open the file obj in read mode declare a variable and perform file object dot tell method and store the output consequently in variable

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

Step 9: Open file obj with read mode also use the read lines method and store the output consequently in and print the same for counter the length use the for conditional statement and display the length.

```
file obj = open("abc.txt", "a")
file obj . write ("data structure")
file obj . close()
file obj = open ("abc.txt", "r")
s3 = file obj . read()
print ("output of append mode: ", s3)
file obj . close()
```

026

```
file obj = open ("abc.txt", "r")
pos = file obj . tell()
print ("tell ():", pos)
file obj . close()
```

```
file obj = open ("abc.txt", "r")
str = file obj . seek(0, 0)
print ("seek (0,0) is : ", str)
file obj . close()
```

Math
Dr
16/17

```
file obj = open ("abc.txt", "r")
str1 = file obj . seek(0, 1)
print ("seek (0,1) is : ", str1)
file obj . close()
```

ANSWER

```
file obj = open ("abc.txt", "r")
st2 = file obj . seek (0, 2)
print ("seek (0, 2) is : ", st2)
file obj . close()
```

```
file obj = open ("abc.txt", "r")
stat = file obj . readlines ()
print ("output : " stat)
for line in stat:
    print (len (line))
file obj . close()
```

ESO

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 times we use the `iter` and `next` method we will get the next iterating elements in the tuple.
Display the same.

Step 2: The similar output can be obtained by using for conventional statement. An iterator 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` to get the value raised 3 and return the same.

Step 4: Call the declared function using function call.

my tuple1 = ("banana", "orange", "apple")
my iter1 = iter(my tuple1)
print(next(my iter1))
my iter2 = iter(my tuple1)
print(next(my iter2))
my iter3 = iter(my tuple1)
print(next(my iter3))
=> banana
orange
apple
for loop:
my tuple1 = ("kevin", "stuart", "bob")
for x in my tuple1:
 print(x)
=> kevin
stuart
bob
square and cube
def square(x):
 y = x * x
 return y
cube(x):
z = x * x * x
return z

390

for i in range(5):

 value = list(map(lambda x: x**2, funct))

 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 * 1.5, listnum))

print(listnum)

def even(x):

 if (x % 2 == 0):

 return "EVEN"

else:

 return "ODD"

list (map(even, listnum))

odd numbers

class odd:

 def __init__(self):

 self.num = 1

 def self

 def __next__(self):

 num = self.num

 self.num += 2

 def __next__(self):

 num = self.num

 self.num += 2

 return num

Step 5: Using for conditional statement specifying the range use the list type casting with map method declare a 'lambda', i.e. anonymous 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 displaying the output.

Step 7: Define a function even with a parameter - then using conditional statements 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 initialize the first element within the container object.

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

~~Not
Done~~

Practical-3

Aim:- Program to demonstrate exception handling

1] Write a program using the exception method of the nature ArithmeticError.

Step 1:- Use the try block & except the input in the raw input method and convert it into the integer datatype and subsequently terminating block.

Step 2:- Use the except block with the exception name as value error & display the appropriate message if the subsequently code is part of the try block.

2] Write a program for accepting the file in a given mode and use the EnvironmentError as an exception for the given input

Step 1:- Within the try block open, the file in the write mode & write mode and write some content on the file.

Step 2:- Use the except block with the value error and display the message regarding message of the file.

Program:-

030

while True

try

x = int(input("Enter class"))

break

except ValueError:

print("Enter Numeric value")

Output:- Enter class 46

Enter class abc

ValueError

Enter Numeric value

Program:-

try:-

f = open("sample.txt", "w")

f.write("Python")

except IOError:

print("User waiting on the file")

else:

print("Operation carried out successfully")

f.close()

Output:-

operation carried out successfully -

or incompatibility of the mode use the else block to display a message that the operation is carried out successfully.

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

Step 4: Use except with value error and print the message not a valid input.

3] Write a program using the assert() to check if the list elements are empty;

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

Step 2:- Close the function & is the body of the program, if you certain elements in the list & take some appropriate action.

4] Write a program to check the range of the age of the students is given class and 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 - if so return the age else use the value error exception.

Program:-

032

```
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 >= 40 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 = 5
Not a valid age
... age : 18

Q80

```
import re
string = "hello 1234 abc 4579"
result = re.findall("Id", string)
result1 = re.findall("V", string)

print(result)
print(result1)
```

Output

```
[[1234, '4579']]
['hello', 'abc']
```

Practical-4

Aim :- Demonstrate the use of regular expression

Theory :- Regular expression represents the sequence of characters which is mainly used for finding & replacing the given pattern in a string & for this very important, reproduce & common usage of regular expression.

Q.) Write a regular expression segregating numeric & alpha-numeric values from a given string
Algorithm

Step 1:- Now apply step 2-pattern in findall()
display the output

2:- ~~I'd is used for matching all decimal digits whereas is used to match non-decimal digits~~

(Q) write a regular expression for finding the match string at beginning of given sequence.

Algorithm:

Step 1: Import re module & apply a try

2: use search() with "ip* python" & string as two parameters

3: Now display the output

4: Now use if conditional statement to check whether the known whether the match is found or not

3) write a regular expression to check whether the given mobile no starts with 8 or 9 & total length 10

Step 1: Import re module & apply a try of mobile no 8

2: Now use for conditional statement to find if the no starts with 8 or 9 & total length of 10 use match()

code 2 :

```
import re
```

```
string = "Python is a important language  
result = re.match('A Python', string)  
print(result)
```

```
if result:
```

```
    print("Match found")
```

```
else:
```

```
    print("No match not found")
```

output :

```
>>> re.match object span(0, 6)
```

~~match = "Python"~~

~~>>> match found~~

Jm 1110

Q.80

```
# code 4
import re
string = "Python is important"
result1 = re.findall("\w+", string)
result2 = re.findall("\w+", string)
print(result1)
print(result2)
```

output

```
>>> ['Python', 'is', 'important']
[ 'Python', 'is', 'important' ]
```

code 5

```
import re
string = "Python is important"
result = re.findall("\w+", string)
result1 = re.findall("\w+", string)
print(result)
print(result1)
```

output

```
>>> ['Python']
[ 'Important' ]
```

Step 3:- Use if conditional statement to know whether we have a match or not if we have use group(1) to display the output & if we don't display nothing.

- Q) Write a program expression for extracting a word from given string along with space characters in between the word and subsequently extract the word without space character.

Algorithm

Step 1: Import re module and apply a string

Step 2: Use find all() to extract a word from given string

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

Step 4: Now display the output

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

Algorithm

Step 1: Import re module & apply a string

Step 2: Use find all() in which use ' \w+' as one parameter to find first word of string this will
" \w+g' as parameter to find last word of string

Step 3: Display the output

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.

Step 3: Use the pack() along with the object created from the text(). and use the parameters

- i) side = LEFT, padx = 20
- ii) side = LEFT, pady = 20
- iii) side = TOP, padx = 30
- iv) side = TOP, pady = 40
- v) side = TOP, pady = 50

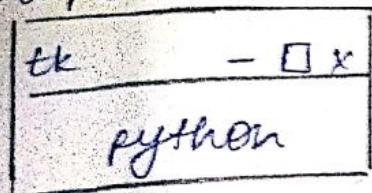
Step 4: Use the mainloop()

creation of parent window
from tkinter import *

036

```
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", fg="grey",  
          fg="black", font="10")  
l1.pack(side=LEFT, padx=20)  
l2 = Label(root, text="CS", fg="light blue",  
          fg="black", font="20")  
l2.pack(side=LEFT, padx=30)  
l3 = Label(root, text="CS", fg="yellow",  
          fg="black", font="10")  
l3.pack(side=TOP, ipadx=40)
```

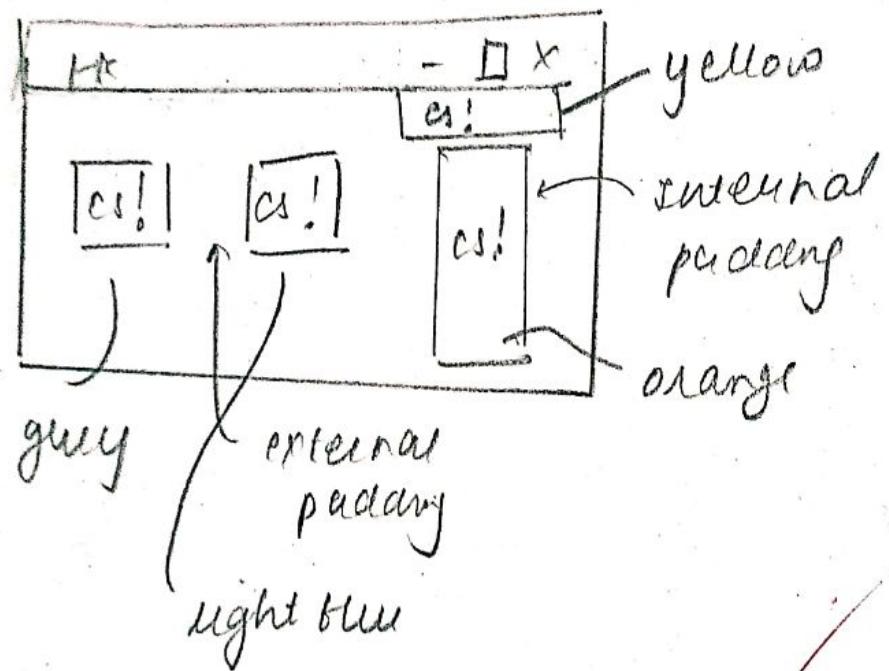
380

2a = label (root, text= "cs!", fg= "orange")

fg= "black", font= "o.")

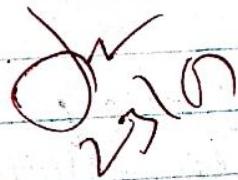
lg.pack (side= top, ipady= 50)

root.mainloop()



Step 5 Now repeat above steps with the Label() which takes the following arguments.

- i) Name of the parent window.
- ii) Text attribute which defines the string.
- iii) The background color (bg).
- iv) The foreground fg and then use the pack() with a relevant padding attributes.



780

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 size of the parent window.

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

Step 4 : Now define the parentwindow 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 entry object which will take following arguments (parentwindow, .option, textvariable) which will take the values option no 1, 2, 3 -- variables arguments corresponding value & trigger the function declared

place button

038

from tkinter import *

root = Tk()

root.geometry("500x500")

def select():

selection = "You just selected " + str(var.get())

l1 = Label(text=selection, fg="white",

fg="green")

l1.pack(side="top")

var = StringVar()

l1 = Label(text=)

l1.insert(1, "cut")

l2 = Label(text="cut")

l2.pack(anchor="n")

rr1 = Radiobutton(root, text="option 1", variable=

=var, value="option 1", command=select)

rr1.pack(anchor="n")

rr2 = Radiobutton(root, text="option 2", variable=

=var, value="option 2", command=select)

rr2.pack(anchor="n")

root.mainloop()

800

ff

111

nnn

nnn

nnnnnn

all padding turned

① Option 1

② Option 2

| You just selected
| option 1

112

scrolltage)

from tkinter import *

root = Tk()

root.geometry ("800x800")

sc = Scrollbar()

sc.pack(side="right", fill="y")

root.mainloop()

geometry()

—□—

X

—□—

—□—

—□—

—□—

—□—

—□—

—□—

—□—

—□—

—□—

surface)

Step 7: Now call the pack() for shadow object to create 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 windows

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

Step 5: Use the pack() along with the scrollbar object with scroll and fill attribute.

Step 6: Use the mainloop with the parent object

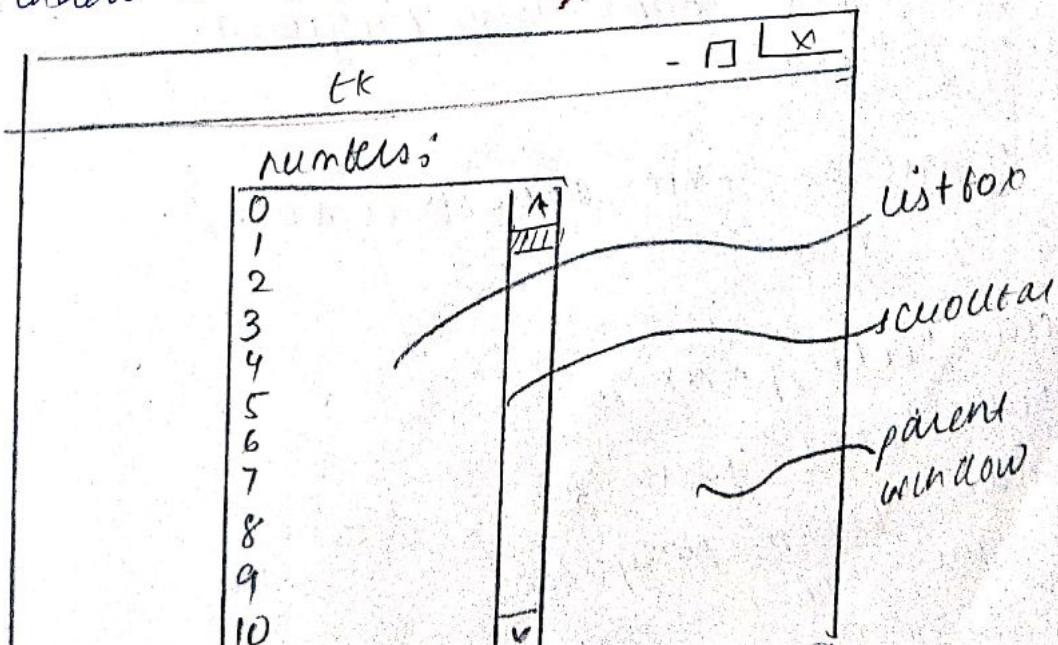
- Step 1: Import the relevant libraries from the tkinter method
- Step 2: Create an corresponding object of the parent window.
- Step 3: Use the geometry manager with pixel size (680x500) or any other suitable pixel value
- Step 4: Use the label widget along with the parent object created & subsequently use the pack method
- Step 5: Use the frame widget along with the parent object created & use the pack method
- Step 6: Use the listbox method along with the attributes like width, height font. To create 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() & use the pack()
- Step 8: trigger , the oven opens using manloop .

using frame widget
from tkinter import *

```
window = Tk()
window.geometry("1168x800d")
label(window, text="numbers").pack()
frame = Frame(window)
frame.pack()
```

listNodes = Listbox(frame, width=20, height=10,
font="Times New Roman", 0)

```
listNodes.pack(side="left", fill="y")
scrollbar = Scrollbar(frame, orient="vertical")
scrollbar.config(command=listNodes.yview)
scrollbar.pack(side="right", fill="y")
for i in range(100):
    listNodes.insert(END, str(i))
window.mainloop()
```



H 4. 040

from tkinter import *

window = Tk()

window.geometry("680x500")

frame = Frame(window)

frame.pack()

listframe = Frame(frame)

leftframe = frame.pack(side="left")

rightframe = frame.pack(side="right")

rightframe.pack(side="right")

b1 = Button(leftframe, text="select", activebackground="red", fg="blue")

b2 = Button(leftframe, text="modify", activebackground="yellow", fg="black")

b3 = Button(leftframe, text="ADD", activebackground="blue", fg="red")

b4 = Button(leftframe, text="EXIT", activebackground="red", fg="green")

b1.pack(side="left", padx=20)

b2.pack(side="right", padx=30)

b3.pack(side="bottom", padx=20)

b4.pack(side="top",

4:

Step 1: Import relevant methods from tkinter library

Step 2: Define the objects corresponding to parent window ?
define the size of parent window in terms of no of pixels

Step 3: Now define the frame objects from the method 2
place it on 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, activebackground
and foreground

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

Step 7: Similarly create the button object corresponding
to the MODIFY operations put it into frame object
on side = "RIGHT"

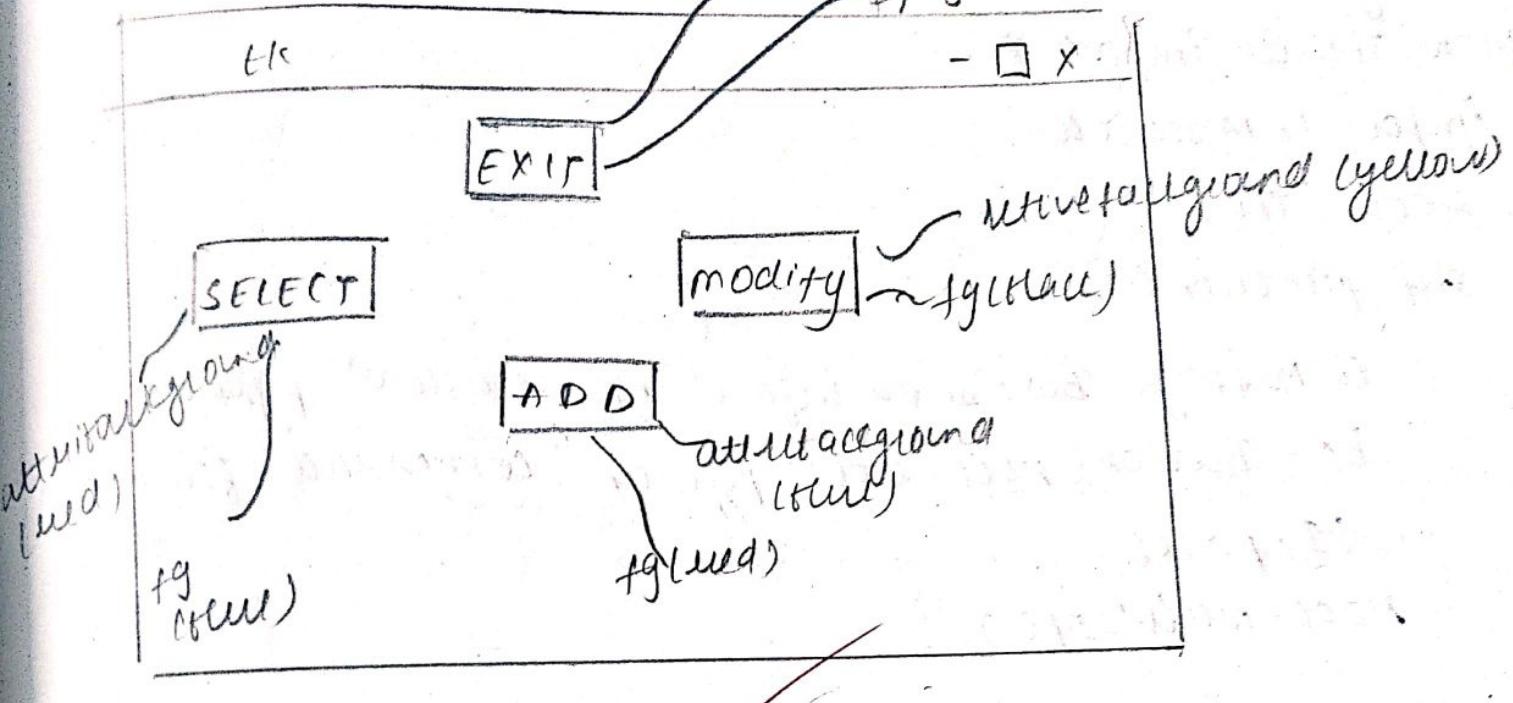
Step 8: Create another button object & place it on to the
right frame and the button as ADD.

Step 9: Add another button & text is on the top of frame
and label it as EXIT.

180

step 10: Use the `put()` simultaneously for all the
objects & finally use the `mainloop()`

042

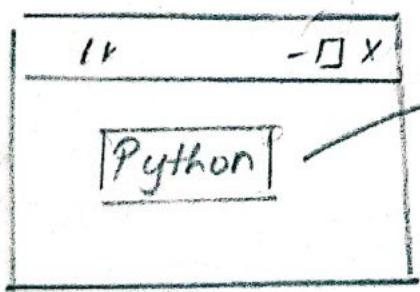


On 27/7/22

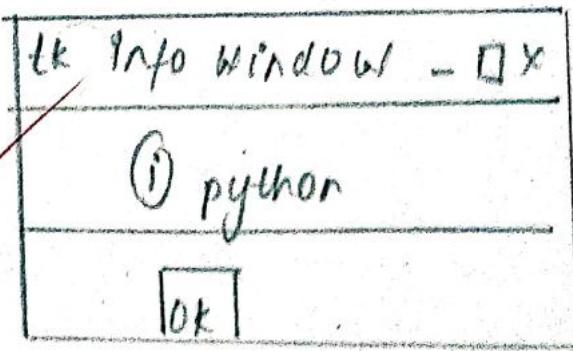
840

```
# Message Box
from tkinter import *
import tkinter.messagebox
root = Tk()
def function():
    tk.messagebox.showinfo("Info Window", "python")
b1 = Button(root, text="python", command=function)
b1.pack()
root.mainloop()
```

Output:



click through this window
to pop up



PRACTICAL - 5(C)

File & GUI components

- Step 1: Import the relevant methods from tkinter library
- Step 2: Import tkMessageBox
- Step 3: Define a parent window object along with the parent window
- Step 4: Define a function which will use tk.messagebox 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

- Step 1: Import the relevant methods from the tkinter library along with parent window object declared.
- Step 2: use parentwindow object along with message function for window size.
- Step 3: Define a function main, declare parent window object & use config(), title(), minsize(), label() as well as button() and use pack() & mainloop() simultaneously.
- Step 4: Similarly define the function second & use the attributes accordingly.
- Step 5: Declare another function fiction along with parent object & declare fiction with attributes like FLAT, RIDGE, GROOVE, RAISED, SUNKEN along with relief/relief
- Step 6: finally call a the mainloop() for event driven programming.

multiple windows
different function (relief I))

from Tkinter import *

root = Tk()

root.minsize(300, 300)

def main():

top = Tk()

top.config(bg="black")

top.title("HOME")

top.minsize(300, 300)

label1 = Label(top, text="SAN FRANCISCO in place of selected! In Golden
Gate Bridge in comand street in chinatown in control")

b1.pack()

b1 = Button(top, text="NET", command=second)

f1.pack(side=RIGHT)

f1 = Button(top, text="cd", command=terminal)

f2.pack(side=LEFT)

top.mainloop()

```
def second():
    top2 = Tk()
    top2.config(bg="orange")
    top2.title("About Us!")
    top2.minsize(300, 300)
    l=Label(top2, text="Created by : Chinmay Naik for my  
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()

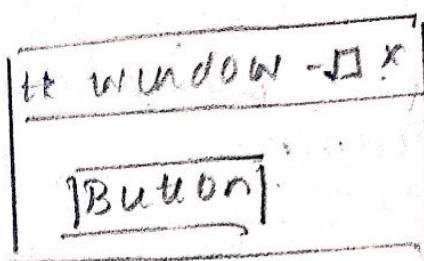
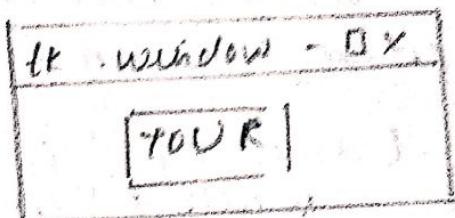
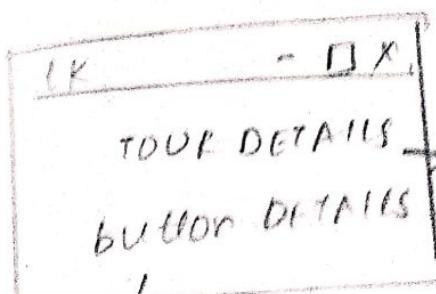
def button():
    top3 = Tk()
    top3.geometry("300x300")
    b1=Button(top3, text="flat button", relief=FLAT)
    b1.pack()
    b2=Button(top3, text="groove button", relief=GROOVE)
    b2.pack()
    b3=Button(top3, text="raised button", relief=Raised)
    b3.pack()
    b4=Button(top3, text="sunken button", relief=SUNKEN)
    b4.pack()
    b5=Button(top3, text="ridge button", relief=RIDGE)
    top3.mainloop()

def terminate():
    b5.quit()

b5=Button(root, text="TOUR DETAILS", command=main)
b5.pack()
b6=Button(root, text="BOTTON DETAILS", command=b6)
b6.pack()
```

C46

Output:



Dx(1)

```
(a)
from tkinter import *
root = Tk()
root.title("Python GUI")
root.minsize(900, 800)
root.config(bg="black")
leftframe = Frame(root, bg="yellow", height="30",
                  width="150")
leftframe.grid(row=0, column=0)
rightframe = Frame(root, bg="blue", height="300",
                    width="150")
rightframe.grid(row=0, column=0)
label1(leftframe, text="Photo", height=2, width=20)
image1 = PhotoImage(file="bear.gif")
label1.grid(row=0, column=0)
Image1 = PhotoImage(file="bear.gif")
Image1.subsample(1, 2)
Image2 = PhotoImage(file="bear.gif")
Image2.subsample(3, 4)
label2(leftframe, image=Image1).grid(row=0, column=0)
label3(rightframe, image=Image2).grid(row=0, column=1)
label4 = Label(leftframe, width=200, height=400,
               bg="white").grid(row=2, column=0)
label5(toolbar, text="Personal Info", height=2, width=20,
       relief=RIDGE).grid(row=0, column=0, padx=20,
                           pady=20)
label5.entry_name1 = Entry(toolbar, width=20).grid(row=1, column=0)
print("Name: Chinmay")
```

Displaying the Image

Algorithm:

Step 1: Create an object corresponding to the parent window & use the following 3 methods with Master config.

Step 2: Create a left frame object from the frame method & place it onto the parent window with the height width and the bg specified. Subsequently use the grid method with the row, column, padx, pady specified.

Step 3: Now, create a right frame object from the frame method with the width, height specified and the row & the column value should be specified.

Step 4: Create a label object from the label method & place it onto the left frame with text attribute denoted as the original image with relief attribute used as RAISED value & subsequently use grid method with row, column value specified as (0, 0) with some column padding value.

Step 5: Now use the photo image method with the file attribute specified.

7AO

Step 6: Use the sub sample method with the org of the image & give the x, y co-ordinate values.

Step 7: Use the label methods & position it onto the left frame and placing the image after the sampling and use the grid method for the positioning in the first row.

Step 8: Create another label object positioning onto the right frame and specifying the image & background attribute with row 2 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 & position it onto the second row.

Step 10: Now define the various function from diff tool bar options provided in the left frame.

Step 11: From the label method position the text to the toolbar use the relief attribute & corresponding grid value & incorporate the internal padding as well.

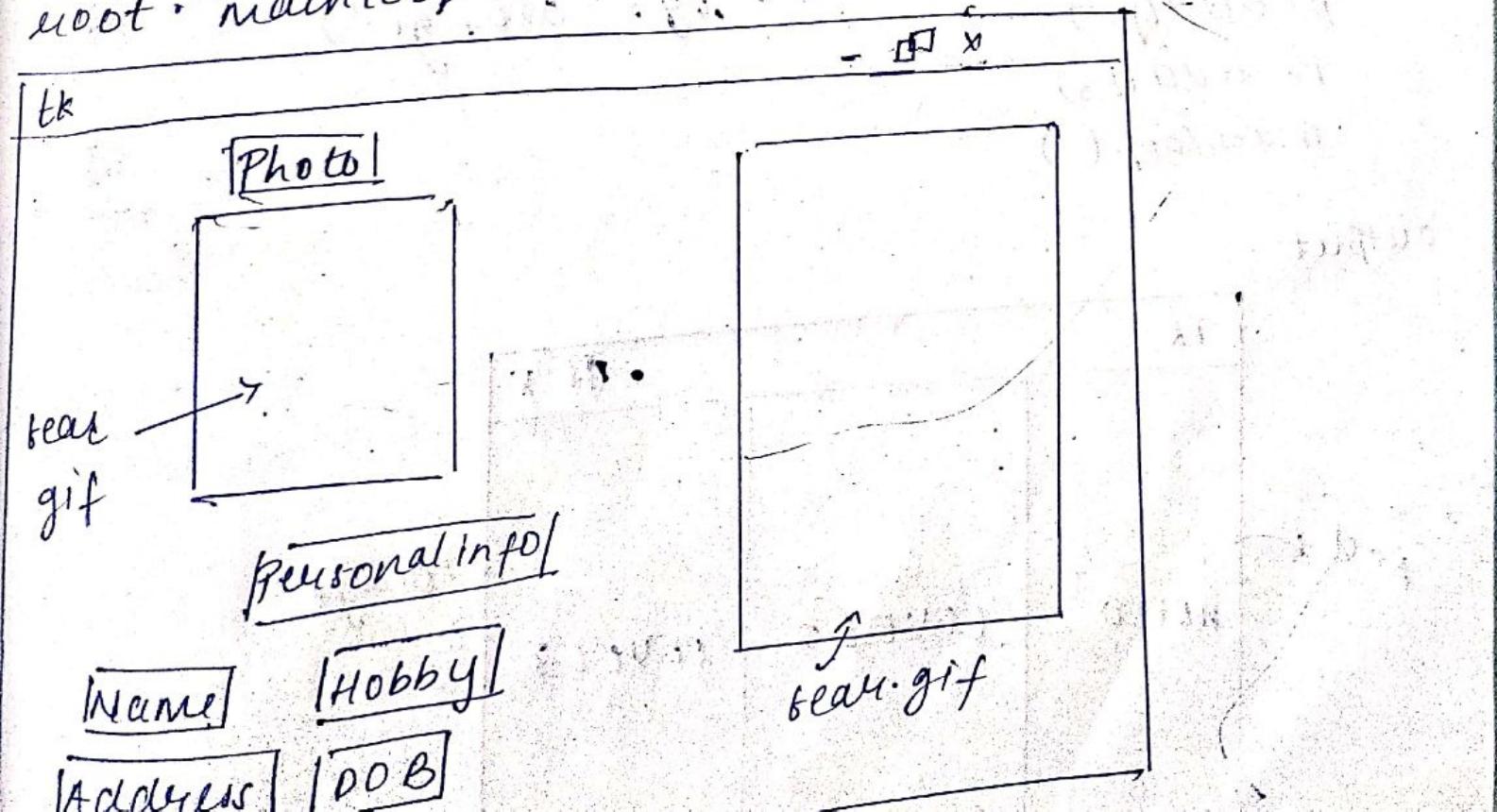
Step 12: Create the label method position it on the toolbar with the next title as personal information & position on some random but next column.

Step 13: Now we make the remaining methods

```

def lab():
    print("Hobby : Coding")
048
def add():
    print("Address Number")
def dob():
    print("DOB = 15/07/2001")
button(toolbar, text="Name", height=1, width=16,
       command=name).grid(row=1, column=0)
button(toolbar, text="Hobby ", height=1, width=16,
       command=hob).grid(row=1, column=1)
button(toolbar, text="Address ", height=1, width=16,
       command=add).grid(row=2, column=0)
button(toolbar, text="DOB", height=1, width=16,
       command=dob).grid(row=2, column=1)
root.mainloop()

```



code

from tkinter import *

root = Tk()

p = Panedwindow(bg = "pink")

p.pack(fill = BOTH, expand = 1)

l1 = Label(p, text = "HELLO", bg = "red")

p.add(l1)

p1 = Panedwindow(p, orient = VERTICAL, bg = "yellow")

p.add(p1)

l2 = Label(p1, text = "LEVEL 2", fg = "green")

p1.add(l2)

p2 = Panedwindow(p1, orient = HORIZONTAL, fg = "blue")

l3 = Label(p2, text = "LEVEL 3", fg = "orange")

p2.add(l3)

mainloop()

output

TK		
red ↪ HELLO	LEVEL 2	LEVEL 3 → orange

paned window

step 1: create an object from the paned window method and use the pack method to make this object visible.

step 2: Now create an object from the entry widget & place it onto the paned windows and use the add method. Similarly, create an object of a paned window.

step 3: Create an object from the scale widget & place it onto the preceding paned window & use the add method accordingly.

step 4: Create a button widget and place it onto the paned window define a functionality along with the button widget

step 5: - use the pack method & main loop method for the corresponding event to trigger.

* Canvas Widget

Step 1: Create an object from the canvas widget by using the attribute height width by color window object

Step 2: Use the corresponding method for drawing simple geometrical shape like circle oval & rectangle specific the co-ordinate values.

Step 3: Similarly use the create line & create oval method along with the co-ordinate values & the fill attribute for specifying the color.

Step 4: Finally use the pack & mainloop method

from tkinter import *

root = Tk()

u1 = canvas(root, height = 500, width = 500, bg = "white")

oval = cl.create_oval(50, 12, 12, 400, fill = "purple")

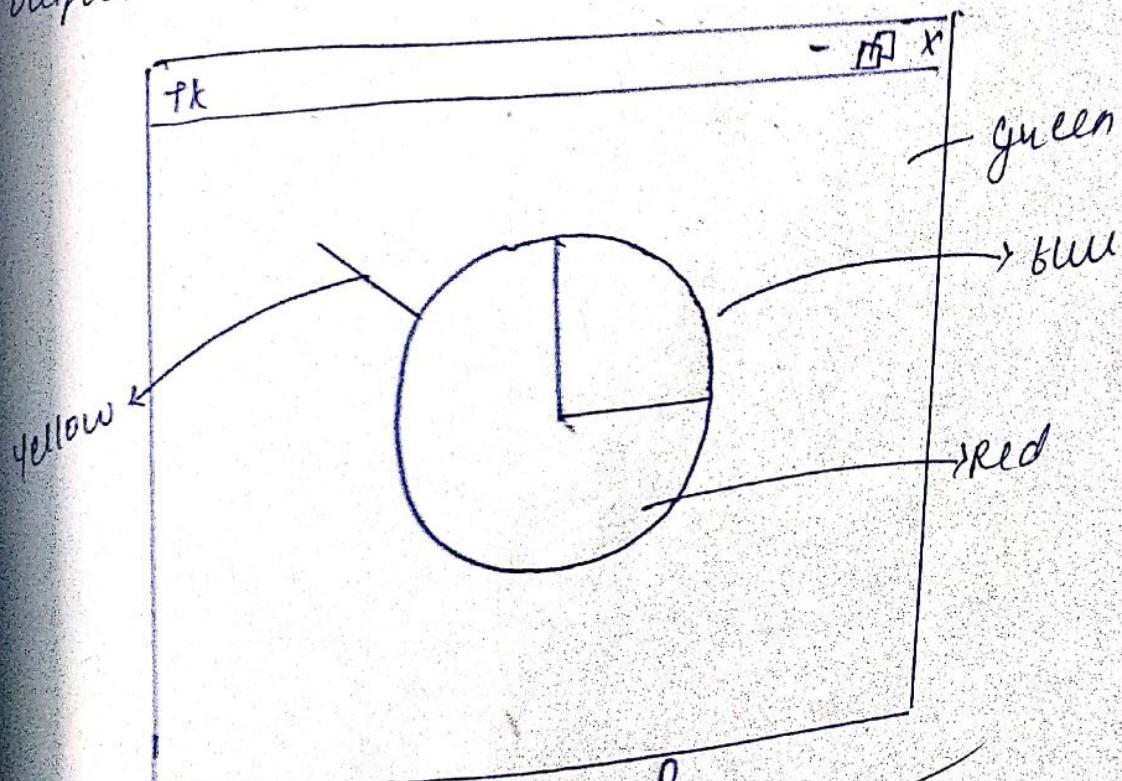
arc = cl.create_line(30, 20, 70, 60, fill = "yellow")

arc = cl.create_oval(300, 18, 18, 400, fill = "blue")

cl.pack(side = TOP)

root.mainloop()

output



Kash

DR SB

#1: 060

```
>>> import dbm  
>>> db = dbm.open("data base", flag='c', mode=0600)  
>>> db["name"] = "name"  
>>> if db["name"] != None:  
    print ("database not empty || match!")  
else:  
    print ("database empty! || NOT match")  
>>> match  
>>> db.close()
```

ATM's Database connectivity

10

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

Step 2: Use the object so created for accessing the given website & correspondingly regular name for the website

Step 3: Check whether the given url address matches with the regular name of the page is not equal to none than display the message that particular found a match or else not found / unmatched

Step 4: Use the close() to terminate database if used

11.2: Step 1: Import corresponding library to make database connection i.e. `java.sql.*`

Step 2: Now create the connection object using `sql` library & the `connect()` for creating new database.

Step 3: Now create cursor object using the `cursor()` from the connection object created -

Step 4: Now use the `executel()` for entering the data with the column name & respective datatype.

Step 5: Now with cursor-object use the `insert` statement for entering the value corresponding to different fields, corresponding the datatype.

Step 6: Use the `commit()` to complete the transaction using the connection object.

Step 7: Use the `execute` statement along with cursor-object for accessing the values from the database using the `select from where` clause.

Step 8: Finally use the `fetch()` or `fetchall()` for displaying the values from the table using the cursor-object.

```
#2: import os, sqlite3  
conn = sqlite3.connect("employee.db")052  
cur = conn.cursor()  
cur.execute('create table dos (name char,  
ROLLNO int')  
cur.execute("insert into dos values ('chinmay', 1788),  
( 'Raj', 1230)")  
conn.commit(),  
cur.execute('select * from dos')  
print(cur.fetchall())  
conn.close()  
  
output:-  
[( 'chinmay', 1788), ( 'Raj', 1230)]
```

Step 9: Execute() 2 drop table syntax for terminating
the database & finally use the close()

620

PYTHON PROJECT.

```
File Edit View Insert Run Options Window Help
random
random
def submit():
    def database():
        name=str(name.get())
        contactl=int(contact.get())
        emailstr=email.get()
        genderl=ct.get()
        beachl=['uhu Beach','Aksa Beach','Pader Chowpatty Beach','Madh Island Beach','Girgaum Chowpatty Beach','Versova Beach','Manori Beach','Marve Beach','Gorai Beach']
        beach=random.choice(beachl)
        addressl=c1.get()
        dayl=ct2.get()

        conn = sqlite3.connect('beachclean.db')
        with conn:
            cursor=conn.cursor()
            cursor.execute('CREATE TABLE IF NOT EXISTS volunteer (Name TEXT, Contact INTEGER, Email TEXT, Gender TEXT, Address TEXT, Beach TEXT, Day_OF_Volunteering TEXT)')
            cursor.execute('INSERT INTO volunteer (Name,Contact,Email,Gender,Address,Beach,Day_OF_Volunteering) VALUES(?,?,?,?,?,?,?)',(name,contactl,emailstr,genderl,addressl,beach,dayl))
            conn.commit()

        rec=cursor.execute('SELECT *FROM volunteer')
        for row in rec:
            print(row)

        name.delete(0, END)
        con.delete(0, END)
        email.delete(0, END)
        ct.set('Select Your Gender')
        c1.set('Select Your Location')
        ct2.set('Select Day of Volunteering')

    def exitl():
        end.destroy()
        rot.destroy()

    end=Tk()
    end.config(bg='light blue')
    end.geometry('+150+200')
    end.title("Registration & Confirmation !")
    end.maxsize(height=90, width=500)
    end.minsize(height=90, width=500)
    Label(end, text="Your Entry is Successfully Registered !\nYou will get a Confirmation Message & Date and Time !", font='Verdana 12 bold', bg='light blue').pack()

    Button(end, text="OK", width=10, command=exitl, bg='dark blue', fg='white', relief=FLAT).pack()
```

```

beachcleaner -> Beach Cleaning GUI (S&I)
File Edit Normal Run Options Window Help
name=stc(name.get())
contactl=stc(contact.get())
email=stc(email.get())
gender=stc(gender.get())
beachl=['Juhu Beach','Alax Beach','Dadar Chowpatty Beach','Madd Island Beach','Girgaum Chowpatty Beach','Vernarosa Beach','Masori Beach','Marve Beach','Gorai Beach']
beachr=random.choice(beachl)
add=stc(c1.get())
day=stc(c2.get())

rot=TK()
rot.config(bg="#eaffed")
rot.geometry('1100x500')
rot.title('Western Mumbai Beach Cleaning')
Label(rot, text="WESTERN MUMBAI BEACH CLEANING\nVOLUNTEER FORM PREVIEW\n",font="Times 19 bold",width=50,bg='#000075',fg='white').grid(row=0, columnspan=3)

Label(rot, text="Name",width=20,bg="#eaffed",font="Verdana 13 bold").grid(row=1, column=0, padx=20, pady=20, sticky=W)
Label(rot, text=name, width=25,bg="#eaffed",font="Verdana 13 bold").grid(row=1, column=1, columnspan=2, sticky=W)

Label(rot, text="Contact",width=20,bg="#eaffed",font="Verdana 13 bold").grid(row=2, column=0, padx=20, pady=20, sticky=W)
Label(rot, text=contactl, width=25,bg="#eaffed",font="Verdana 13 bold").grid(row=2, column=1, columnspan=2, sticky=W)

Label(rot, text="Email",width=20,bg="#eaffed",font="Verdana 13 bold").grid(row=3, column=0, padx=20, pady=20, sticky=W)
Label(rot, text=email, width=25,bg="#eaffed",font="Verdana 13 bold").grid(row=3, column=1, columnspan=2, sticky=W)

Label(rot, text="Gender",width=20,bg="#eaffed",font="Verdana 13 bold").grid(row=4, column=0, padx=20, pady=20, sticky=W)
Label(rot, text=genderl, width=25,bg="#eaffed",font="Verdana 13 bold").grid(row=4, column=1, columnspan=2, sticky=W)

Label(rot, text="Address",width=20,bg="#eaffed",font="Verdana 13 bold").grid(row=5, column=0, padx=20, pady=20, sticky=W)
Label(rot, text=add, width=25,bg="#eaffed",font="Verdana 13 bold").grid(row=5, column=1, columnspan=2, sticky=W)

Label(rot, text="Beach",width=20,bg="#eaffed",font="Verdana 13 bold").grid(row=5, column=0, padx=20, pady=20, sticky=W)
Label(rot, text=beachr, width=25,bg="#eaffed",font="Verdana 13 bold").grid(row=5, column=1, columnspan=2, sticky=W)

Label(rot, text="Day of Volunteering",width=20,bg="#eaffed",font="Verdana 13 bold").grid(row=6, column=0, padx=20, pady=20, sticky=W)
Label(rot, text=day, width=25,bg="#eaffed",font="Verdana 13 bold").grid(row=6, column=1, columnspan=2, sticky=W)

Button(rot, text='Register',bg='#0022ff',fg='white',command=database, width=10, font='Verdana 11 bold').grid(row=7, column=2, padx=20, pady=20, sticky=W)
Button(rot, text='Edit',bg='#0022ff',fg='white',command=rot.destroy, width=10, font='Verdana 11 bold').grid(row=7, column=1, padx=20, pady=20, sticky=E)

rot.mainloop()

rot = Tk()
rot.config(bg="#eaffed")
rot.geometry('1100x500')
rot.title('Western Mumbai Beach Cleaning')

Label(root, text="WESTERN MUMBAI BEACH CLEANING\nVOLUNTEER FORM\n",font="Times 19 bold",width=50,bg='#000075',fg='white').grid(row=0, columnspan=3)

```

```

beachapp - F:\Beach Clean beachapp (35)
File Edit Format Run Options Window Help
ct.config(bg="#eaffed")
ct.geometry("1100x200")
ct.title("Western Mumbai Beach Cleaning")
el(root, text="WESTERN MUMBAI BEACH CLEANING VOLUNTEER FORM", font="Times 19 bold", width=50, bg="#0000675", fg="white").grid(row=0, columnspan=3)
el(root, text="Full Name", width=20, bg="#eaffed", font="Verdana 11 bold").grid(row=1, column=0, padx=20, pady=20)
e = Entry(root, width=35, font="Verdana 11 bold")
e.grid(row=1, column=1, columnspan=2, sticky=W, padx=40, pady=20)
el(root, text="Contact", width=20, bg="#eaffed", font="Verdana 11 bold").grid(row=2, column=0, padx=20, pady=20)
e = Entry(root, width=35, font="Verdana 11 bold")
e.grid(row=2, column=1, columnspan=2, sticky=W, padx=40, pady=20)
el(root, text="Email", width=20, bg="#eaffed", font="Verdana 11 bold").grid(row=3, column=0, padx=20, pady=20)
i1 = Entry(root, width=35, font="Verdana 11 bold")
i1.grid(row=3, column=1, columnspan=2, sticky=W, padx=40, pady=20)
el(root, text="Gender", width=20, bg="#eaffed", font="Verdana 11 bold").grid(row=4, column=0, padx=20, pady=20)
l = ['Male', 'Female', 'Other']
ringVar()
list=OptionMenu(root,c, *list1)
list.config(width=30, bg="#00c0ff", activebackground="#0022ff", font="Verdana 11 bold")
t('Select Your Gender')
list.grid(row=4, column=1, columnspan=2, sticky=W, padx=40, pady=20)

el(root, text="Address", width=20, bg="#eaffed", font="Verdana 11 bold").grid(row=5, column=0, padx=20, pady=20)
l = ['Chorlpata', 'Marine Lines', 'Charni Road', 'Grant Road', 'Mumbai Central', 'Mahalaxmi', 'Lower Parel', 'Elphinstone Road', 'Pader', 'Matunga Road', 'Mahim Junction', 'Bandra', 'Ehat Road', 'Santa Cruz', 'Ville Parle', 'Andheri', 'Jogeshwari', 'Goregaon', 'Ram Mandir', 'Malad', 'Kandivali', 'Borivali', 'Dahisar', 'Mira Road', 'Bhayandar', 'Kurla', 'Vasai Road', 'Maliwasopata', 'Virar']
ringVar()
ist=OptionMenu(root,c1, *list1)
ist.config(width=30, bg="#00c0ff", activebackground="#0022ff", font="Verdana 11 bold")
t('Select Your Location')
ist.grid(row=5, column=1, columnspan=2, sticky=W, padx=40, pady=20)

el(root, text="Day Of Volunteering", width=20, bg="#eaffed", font="Verdana 11 bold").grid(row=6, column=0, padx=20, pady=20)
l = ['Friday', 'Saturday', 'Sunday', 'Friday & Saturday', 'Saturday & Sunday', 'Friday & Sunday', 'All Three Days']
ringVar()
list=OptionMenu(root,c2, *list1)
list.config(width=30, bg="#00c0ff", activebackground="#0022ff", font="Verdana 11 bold")
t('Select Day of Volunteering')
list.grid(row=6, column=1, columnspan=2, sticky=W, padx=40, pady=20)

el(root, text="Submit", fg="white", command=submit, width=10, bg="#0022ff", font="Verdana 11 bold").grid(row=7, column=2, padx=20, pady=20)
mainloop()

```

- □ ×

WESTERN MUMBAI BEACH CLEANING VOLUNTEER FORM

Full Name

Aayushi

Contact

8855221144

Email

aayushi001@gmail.com

Gender

Female

Address

Borivali

Day Of Volunteering

Sunday

Submit

WESTERN MUMBAI BEACH CLEANING VOLUNTEER FORM PREVIEW

Name

Aayushi

Contact

8855221144

Email

aayushi001@gmail.com

Gender

Female

Beach

Manori Beach

Day of Volunteering

Sunday

Edit

Register

Western Mumbai Beach Cleaning

WESTERN MUMBAI BEACH CLEANING VOLUNTEER FORM PREVIEW

Name

Aayushi

Registration & Confirmation

Your Entry is Successfully Registered !
You will get a Confirmation Message & Date and Time !

OK

ll.com

Gender

Female

Beach

Manori Beach

Day of Volunteering

Sunday

Edit

Register

C:\Windows\py.exe

```
('abc', 123, 'abc', 'Male', 'Borivali', 'Aksa Beach', 'All Three Days')
('Chinmay', 9999557788, 'chinmay0076@gmail.com', 'Male', 'Borivali', 'Juhu Beach', 'All Three Days')
('Dinesh', 7788995511, 'dineshk007@gmail.com', 'Male', 'Kandivali', 'Juhu Beach', 'Saturday & Sunday')
('Dharan', 6688995544, 'dharan01@gmail.com', 'Male', 'Kandivali', 'Dadar Chowpatty Beach', 'Saturday & Sunday')
('Raj', 6655448899, 'raj00067@gmail.com', 'Male', 'Borivali', 'Juhu Beach', 'Sunday')
('Raj Konkar', 9988552211, 'rajkonkar@gmail.com', 'Male', 'Borivali', 'Girgaum Chowpatty Beach', 'Friday & Sunday')
('Aayushi', 8855221144, 'aayushi001@gmail.com', 'Female', 'Borivali', 'Girgaum Chowpatty Beach', 'Sunday')
```