

29/12/19

ECE

Practical No. 1

Aim : To Demonstrate the use of different file accessing modes, different files, different attributes and read method.

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

Step 2 : Now, To open the file use the read mode using the appropriate syntax i.e read(), readline() & readlines, then store the output in variable & Display the contents of the variable.

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

20:

```

file obj = open("demo.txt", "w")
file obj.write ("Hello World" + "\n")
file obj.write ("I am A.I\n Developed by Dhaval \n Developed in C")
file obj.close()

file obj = open ("demo.txt", "r")
# Read()
String1 = fileobj.read()
print ("The output of read method ", String1)
fileobj.close()

>>> ('The output of read method.', 'Hello World\nI am A.I\n'
     Developed by Dhaval \n Developed in C\n')

# Readlines
file obj = open ("demo.txt", "r")
String2 = fileobj.readline()
print ("The output of readline method ", String2)
fileobj.close()

>>> ('The output of readline method : 'Hello world \n')

# Readlines()
file obj = open ("demo.txt", "r")
String3 = fileobj.readlines()
print ("The output of readlines method ", String3)
fileobj.close()

>>> ('The output of readlines method ', ['Hello World\nI am A.I\n'
     Developed by Dhaval \n Developed in C\n'])

# file attributes
String4 = file obj.name
print ("name of file (name attribute)", String4)
>>> ("name of file (name attribute)"'demo.txt')

String5 = fileobj.closed
print ("(close) attribute = " String5)
>>> ((close) attribute = 'True')

```

Q15:

```
C> fileobj.mode  
Print ("file mode", c)  
>>> ("file mode", 'r')
```

```
d = fileobj.softspace  
Print ("softspace", d)  
>>> ("Softspace", 0)
```

wt mode

```
fileobj = open ("demo.txt", "w+")  
fileobj.write ("Thakur")  
fileobj.close()
```

rt mode

```
fileobj = open ("demo.txt", "r")  
e = fileobj.read (s)  
Print ("The output of r+", e)  
fileobj.close()
```

```
>>> (Output of r+, "Thakur")
```

Write mode

```
fileobj = open ("demo.txt", "w")  
fileobj.write ("college")  
fileobj.close()
```

read mode

```
fileobj = open ("demotxt", "r")  
f = fileobj.read (s)  
Print ("output of read mode", f)  
>>> (Output of read mode, "college")
```

Step 4: Now open the fileobj in write mode write some another Content close subsequently. Then again open the fileobj in 'w+' mode that is the update mode and write contents

Step 5: Open fileobj in read mode Display the update written content in, close, open again in 'r+' mode with parameter passed in. Display the output Subsequently.

~~After all the steps in fileobj write no output is there~~

Step 6 : Now open fileobj in append mode open
the write method write Content close the

fileobj again open the fileobj in read
mode & display the 'appending output'.

Step 7 : Open the fileobj in read mode declare

a variable and perform fileobj dot
tell method is store the output consequently
in variable

#append mode

```
fileobj = open("demo.txt", "a")
```

```
fileobj.write("database")
```

```
fileobj.close()
```

```
fileobj = open("demo.txt", "r")
```

```
g = fileobj.read()
```

```
print("The output of append mode", g)
```

```
fileobj.close()
```

```
>>> ("The output of append mode", 'College database')
```

College database

with open('text.txt', 'a') as f:

f.write('College database')

Output: College database

55

tell Method

```
fileobj = open("demo.txt", "r")
```

```
String 6 = (fileobj.tell())
```

```
Print (String 6)
```

```
>>> 0
```

Seek Method

```
fileobj = open("demo.txt", "r")
```

```
String 7 = file.seek(0, 0)
```

```
String 8 = file.read(10)
```

```
Print ("The beginning of the file", String 8)
```

```
>>> Hello World
```

finding length of different lines exist within lines

```
fileobj = open("demo.txt", "r")
```

```
Stg = fileobj.readlines()
```

```
Print ("Output", stg)
```

```
for line in stg:
```

```
    Print (len(line))
```

```
fileobj.close()
```

```
>>> (Output . [ College database])
```

Step 8: Use the Seek method with the offset argument with opening the fileobj in readmode & closing subsequently.

Step 9: Open file obj with readmode also use the readlines method & store the output consequently in & print the same for counting the length. Use the for Conditional Statement & Display the length.

✓
Xmail ✓

68

Step 10: 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 of print to the statement is to add special character in the content of the file.

Content with Special character

```
fileobj = open("demo.txt", "r")
```

```
Content = fileobj.read()
```

```
while len(Content) > 0:
```

```
    print(Content, end = "#")
```

```
    Content = fileobj.read()
```

```
>>> #E#L#L#O#
```

```
#I#A#M# A#I#
```

```
#D#E#V#E#L#O#P#E#D# #B#Y# # D#H#a#V#a#L
```

FS

Program:

```
mytuple = ("Dhaval", "Pinank", "Mahesh", "Pranay")
```

```
myiter = iter(mytuple)
```

```
print = (next(myiter))
```

Output:

Dhaval

Pinank

Mahesh

Pranay

Program:

```
mytuple = ("Dhaval", "Pinank", "Mahesh", "Pranay")
```

```
for a in mytuple:
```

```
    print(a)
```

Output:

Dhaval

Pinank

Mahesh

Pranay.

Dr. Anil

Practical No. 2

Aim: To display elements of a tuple using iterator method

Algorithm:

Step 1: Form a tuple with certain elements inserted in it

Step 2: Use iterator method with tuple and assign it to a variable.

Step 3: Use the next method with variable and print the elements.

Aim: To use iterator method with for loop.

Algorithm:

Step 1: Form a tuple with certain element inserted in it.

Step 2: Use the for conditional statement to access each element of tuple.

Step 3: Print the element of tuple.

Aim: To find odd number in given range using outer method

Algorithm:

Step 1: Define a class which will contain various functions.

Step 2: Define the outer method with an argument and return the value of argument.

Step 3: Define a function which increments the value of argument by two.

Step 4: Create an object which inherits the properties of class and take the user input.

Step 5: Use the for conditional statement followed by if conditional statement and print the value.

Program

26

```
class odd:  
    def __iter__(self):  
        self.num = 1  
        return self  
  
    def next(self):  
        num = self.num  
        self.num += 2  
        return num  
  
myobj = odd()  
myiter = iter(myobj)  
x = int(input("Enter range number"))  
for i in range(x):  
    if i < x:  
        print(i)
```

Output

Enter a number: 6

1
3
5

as

```
class myclass:  
    def __iter__(self):  
        self.a = 1  
        return self  
  
    def __next__(self):  
        if self.a <= 20:  
            x = self.a  
            self.a += 1  
            return x  
        else:  
            raise StopIteration
```

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

Output:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20.

Aim: To print first 20 number using iter method

Algorithm

Step 1: Define a class which will contain various function

Step 2: Define iter method, with an argument, and return the value of argument.

Step 3: Define next method, which increments the value of argument by 1 & prints it.

Step 4: Create a object which inherits the property class and take the user input.

Step 5: Use for loop to print the value of the variance.

ES

Aim: To find square of a number without using map method.

Algorithm: ~~Algorithm will include steps for defined steps~~

Step 1: Define a list which contains certain values. Then apply the Euler's square root method.

Step 2: Define an empty list. ~~which is yet to be filled with result of the square root~~

Step 3: Use for loop followed by append method ~~to append the result into the empty list~~

Step 4: Print the value of the list.

Program:

28

```
list = [1, 2, 3, 4, 5]
empty = []
for i in list:
    empty.append(i**2)
print(empty)
```

Output:

[1, 4, 9, 16, 25]

(Given space will be provided)

Please Enter Input:

[100, 200, 300, 400, 500, 600, 700, 800, 900, 1000]

82

num = [0, 4, 5, 7, 9, 11, 13, 15, 20, 9]

num = 1st (map (lambda x: x%5, num))

print (num)

def even (x)

if x%2 == 0:

else:

return "odd"

1st (map (even, num))

Output:

[0, 4, 5, 7, 9, 11, 13, 15, 20, 9]

[even, even, odd, odd, odd, odd, odd, odd, odd, even, odd]

Aim: To find if the number is odd or even from given list using map method.

Algorithm:

Step 1: Declare a list num variable and display some elements.

Step 2: Define a function even which consists various conditional statement.

Step 3: Further use of conditional statement to check the module of numbers for each element of list and return the message accordingly.

Step 4: Use the map method to print the value in the list format.

Aim: To find square and cube of number simultaneously using map method.

Algorithm:

Step 1: Define a function which returns the square of given number.

Step 2: Define another function which returns cube of given number.

Step 3: Store the output simultaneously in a list.

Step 4: Use the for loop followed by the map function and print the value of result.

```

def square(x):
    return (x ** 2)

def cube(x):
    return (x ** 3)

func1 = [square, cube]

for i in range(4):
    values = list(map(lambda x: x(i), func1))
    print(list(values))

```

Output:

- [0, 0]
- [1, 1]
- [4, 8]
- [9, 27]
- [16, 64]

Jyoti

118 Program

try:

```
fileobj = open("abc.txt", "w")
```

```
fileobj.write("Python is an indented language.\n")
```

```
C is a Programming language!")
```

```
except IOError:
```

```
[print("There is an environment error")]
```

Output

Operation Successful.

File Created



Practical No. 3

Aim: Program to demonstrate exception handling.

Program for demonstrating the use of IOError.

Step 1: Use the try block to define the normal course of action. For e.g :- Define the file object and open the file in the write or read mode, and write some content onto the file.

Step 2: Use the except block with the IOError as an environment error and convey the appropriate message to the user, else display the message that the operation is carried out successfully.

18

Program to demonstrate the multiple exception
IOError and ValueError.

Step 1: Use the try block and define the file object

open the file in write or read mode and write

some content onto the file.

Step 2: 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

but if it is not then display an error message

Step 3: Define the except blocks for IOError and

ValueError.

While True:

try :

fileobj = open ("abc.txt", "w")

fileobj.write (" Python is an indented language \n C is a programming language")

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

print (a)

break

except IOError:

print ("There is an environment error")

except ValueError:

print ("The value is invalid")

Ans 3

SS

```
#match()
import re
pattern = r "FyCS"
sequence = "FyCS represent 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 = 'hello 123 howdy 789 45 howru'
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', ' ', ' ', '45', 'howru']
```

Practical - 4

TOPIC : Regular expression

Step 1: Import re module declare pattern and declare sequence use match method with declare arguments if arguments matched than 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 find all () with argument l and print the same.

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

88.

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

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. Use 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 number are equal to 10. If criteria matches print cell number matches otherwise print failed.

```
# 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 0x0281DF00>
    python
```

Verifying the given set of phone numbers

```
import re
list1 = ['8004567891', '9145673210', '7865432981',
         '9876543201']
for value in list1:
    if re.match(r'[8-9]\d{1}[0-9]{9}'):
        value or len(value) == 10:
            print("criteria matched for cell number!")
    else:
        print('criteria failed!')
```

```
>>> Criteria matched for cell number
```

```
Criteria matched for cell number
```

Criteria failed!

```
Criteria matched for cell number
```

```
# vowels
```

```
import re
```

```
str1 = 'plant is life overall'
```

```
Output = re.findall(r'lb [aeiouAEIOU]lw+', str1)
```

```
print(Output)
```

```
>>> ['is', 'overall']
```

```
# host & domain
```

```
import re
```

```
seq = 'abc.tcs@edu.com, xyz@gmail.com'
```

```
pattern = r'[lwl.-]+[lwl.-]'
```

```
Output = re.findall(pattern, seq)
```

```
print(Output)
```

```
>>> ['abc.tcs', 'edu.com', 'xyz', 'gmail.com']
```

```
# counting of first 2 letters.
```

```
import re
```

```
s = 'mr.a, ms.b, ms.c, mr.t'
```

```
p = r'[lms/mr.]+' +
```

```
O = re.findall(p, s)
```

```
print(O)
```

```
m = 0
```

```
f = 0
```

```
for v in O:
```

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 separating the host & domain name. Use the.findall() and print the output respectively

Step 9: Import re module enter a string use pattern to display only two element of the particular string Use.findall() declare two variable with initial value as zero use for condition and subsequently use the if condition check whether condition satisfy add up the or else increment value and display the value subsequently.

Drsl ✓

if ($\text{v} == \text{ms}$) :

36

$f = f + 1$

else:

$m = m + 1$

print ("No. of males is: ", m)

print ("No. of females is: ", f)

>> ['mr', 'ms', 'ms', 'mr']

('No. of males is: ', 2)

('No. of females is: ', 2)

Program:

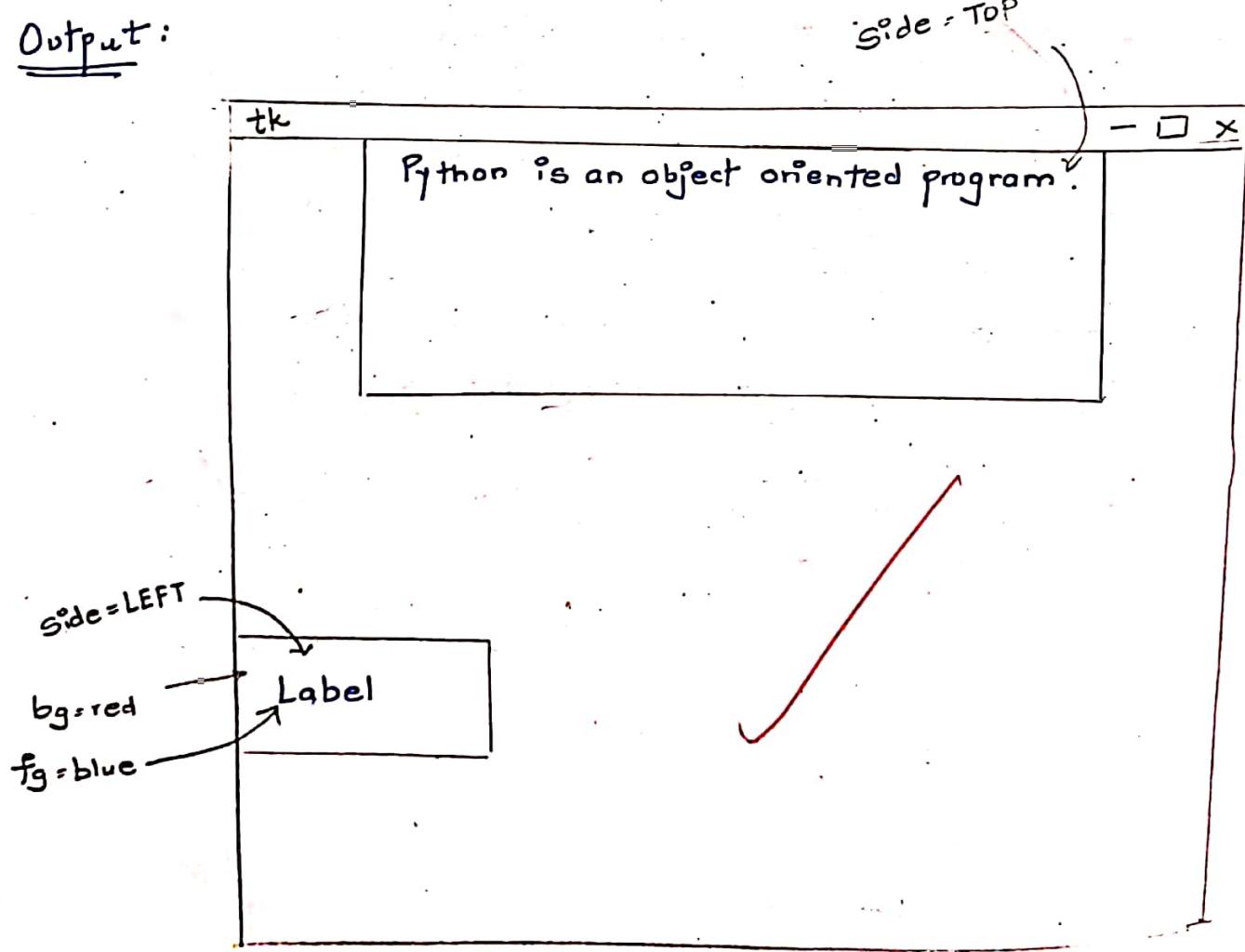
Q8

```

from tkinter import *
root = Tk()
T1 = Text(root)
T1.insert(END, "Python is an object oriented program")
T1.pack(side=TOP, padx=20, pady=30, ipadx=40, ipady=50)
L1 = Label(root, text="Label", bg="red", fg="blue")
L1.pack(side=LEFT, padx=10, ipadx=20, ipady=30)
root.mainloop()

```

Output:



Practical 5.

Aim: To make use of GUI application along with the basic method

Algorithm :

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

Step 2: Create a window from a text variable & position it into the parent window.

Step 3: Use the pack() along with the object created from text method & use the parameter.

i) side = TOP, padx = 20, ipadx = 40, ipady = 50.

Step 4: Use the main loop method for triggering corresponding event.

Step 5: Now repeat above step with a label method which takes the following argument

i) Name of parent window

ii) Text attribute which defines the string

iii) The background colour (bg)

iv) The foreground colour (fg)

Now use pack() with relevant attributes.

Aim: To make use of Radiobutton widget for selection of one of the option.

Algorithm:

Step 1: Use the tkinter method to import the relevant method.

Step 2: Define a function which tells user about given selection mode from multiple option available.

Step 3: Use the config method along with label method & call the variable as an argument within method

Step 4: Now define the parent window & define option using control variable.

Step 5: Now create object of Radio button which will take following argument:

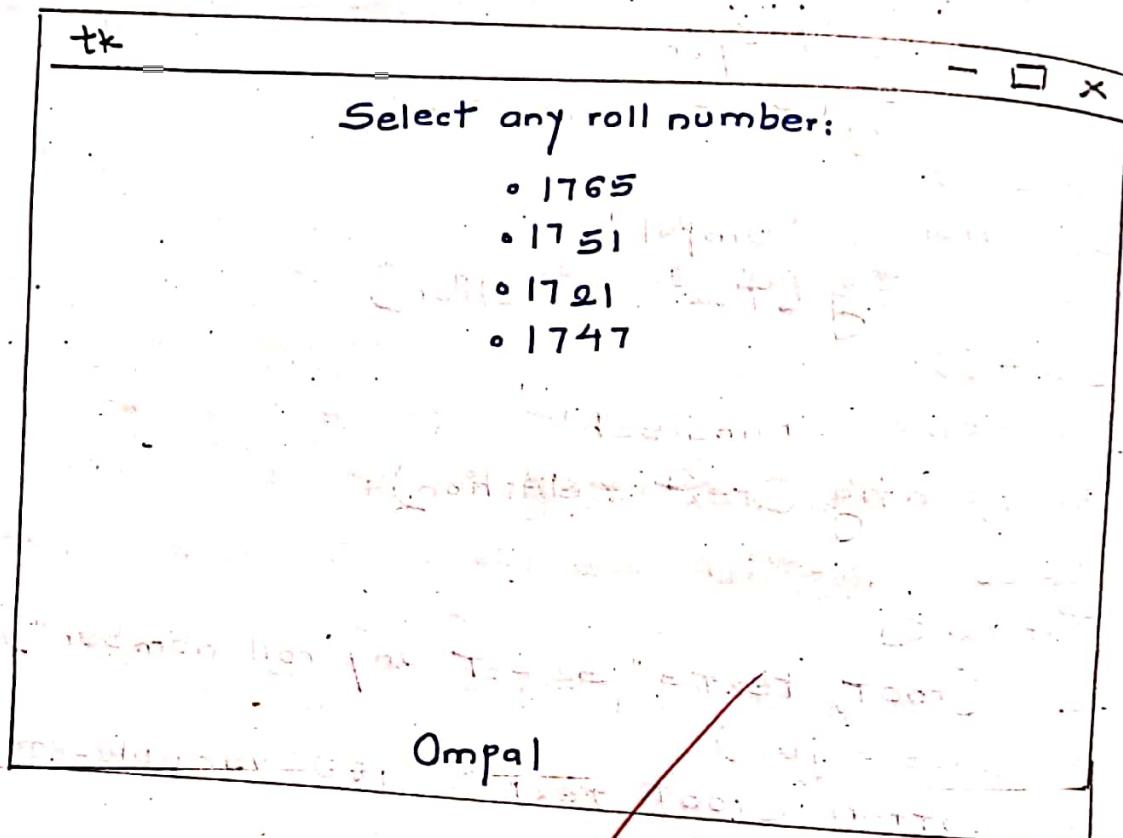
- i) Positioning on Parent Window
- ii) Text Variable
- iii) Define variable argument
- iv) Corresponding value and trigger the given function.

Step 6: Now call the pack method for corresponding Radio object so created and specify argument as an anchor attribute.

Program:

```
from tkinter import *
def sel 1():
    selection = "Dhaval"
    label.config(text=selection)
def sel 2():
    selection = "Mayur"
    label.config(text=selection)
def sel 3():
    selection = "Ompal"
    label.config(text=selection)
def sel 4():
    selection = "Chandresh"
    label.config(text=selection)
root = Tk()
var = IntVar()
L1 = Label(root, text="Select any roll number")
L1.pack(side=TOP)
R1 = Radiobutton(root, text="1765", variable=var, value=0, command=sel 1)
R1.pack(anchor=N)
R2 = Radiobutton(root, text="1751", variable=var, value=1, command=sel 2)
R2.pack(anchor=N)
R3 = Radiobutton(root, text="1721", variable=var, value=2, command=sel 3)
R3.pack(anchor=N)
R4 = Radiobutton(root, text="1747", variable=var, value=3, command=sel 4)
R4.pack(anchor=N)
label = Label(root)
label.pack(side=BOTTOM)
root.mainloop()
```

Output:



Step 7: Now define a label object and place it onto parent window using pack method & finally use main loop method.

EE

Aim: To make use of Scroll Bar widget of the GUI application.

Algorithm:

Step 1: Import tkinter library to use Scroll Bar widget

Step 2: Create an object corresponding to scroll parent window & create an object from Scroll bar & place it on the parent window so created.

Step 3: Create an object of Label method to provide a heading & place it on parent window.

Step 4: Use pack method along with object of scroll bar method & use argument side & fill.

Step 5: Create an object of List Box method & place it onto parent window with attribute yscroll command

Step 6: Use for loop to insert values in the object of List Box by using insert method.

Step 7: Use config method along with scroll bar object & use command attribute.

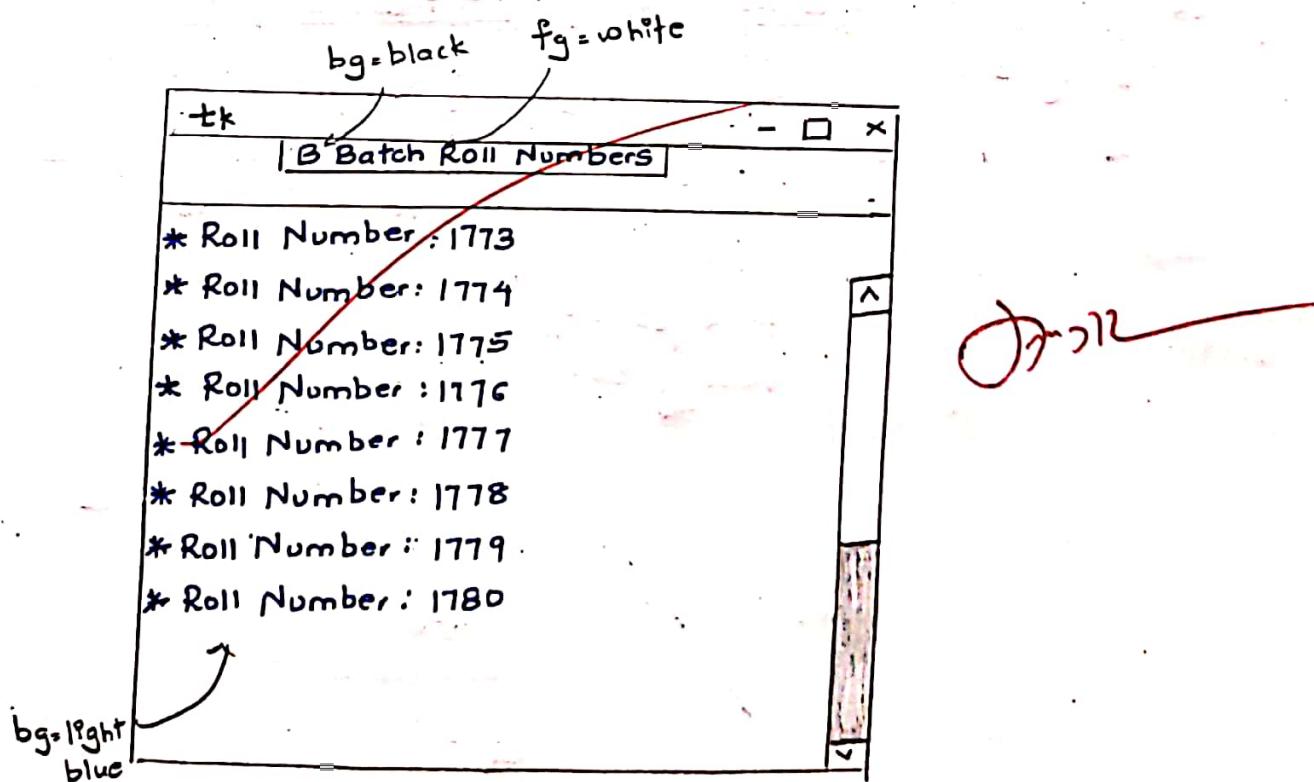
Step 8: Finally call the mainloop method.

Program

40

```
from tkinter import *
root = Tk()
root.geometry ('450x400')
l1 = Label (root, text = "B Batch Roll Number : ", bg = "black", fg = "white")
l1.pack ()
scroll = Scrollbar (root)
scroll.pack (side = RIGHT, fill = Y)
mylist = Listbox (root, yscrollcommand = scroll.set, bg = "light blue")
for num in range (41, 81):
    mylist.insert (END, "* Roll Number: " + str (num))
mylist.pack (side = LEFT, fill = BOTH)
scroll.config (command = mylist.yview)
root.mainloop ()
```

Output:



Program :

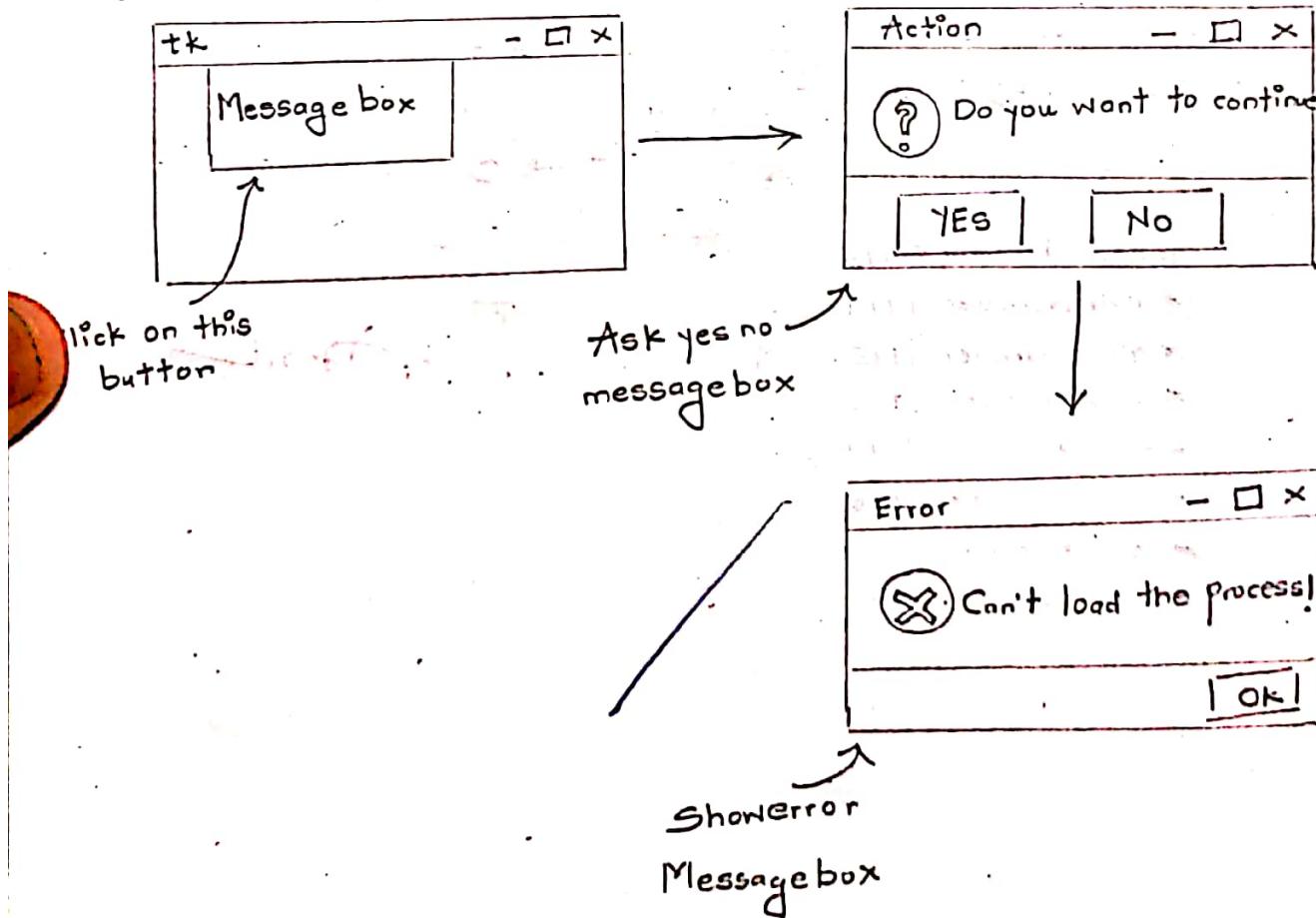
Q1

```

from tkinter import *
from tkinter import messagebox
def msgbox():
    messagebox.askyesno("Action", "Do you want to continue?")
    messagebox.showerror("Error", "Can't load the process!")
root = Tk()
root.config(bg="grey")
B1 = Button(root, text="Message Box", bg="blue", command=msgbox)
B1.pack()
root.mainloop()

```

Output:



Aim: To make use of messagebox method of the GUI application.

Algorithm:

Step 1: Import relevant method from tkinter library.

Step 2: Define a function and use messagebox along with different methods available which contain one or more arguments.

Step 3: Create an object from button method and place it onto the parent window with text and command attribute specified.

Step 4: Use pack method and finally use the mainloop method.

Aim: Program to traverse various window using the button widget.

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 method 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 method like config, title, geometry etc.

Step 6: Define Two button which will be placed on the main window to traverse and another window and the other to terminate the program.

```

from tkinter import *
def main():
    root = Tk()
    root.geometry("450x800")
    root.config(bg="light green")
    root.title("Window 1")
    B1 = Button(root, text="Next", command=main)
    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="red")
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)

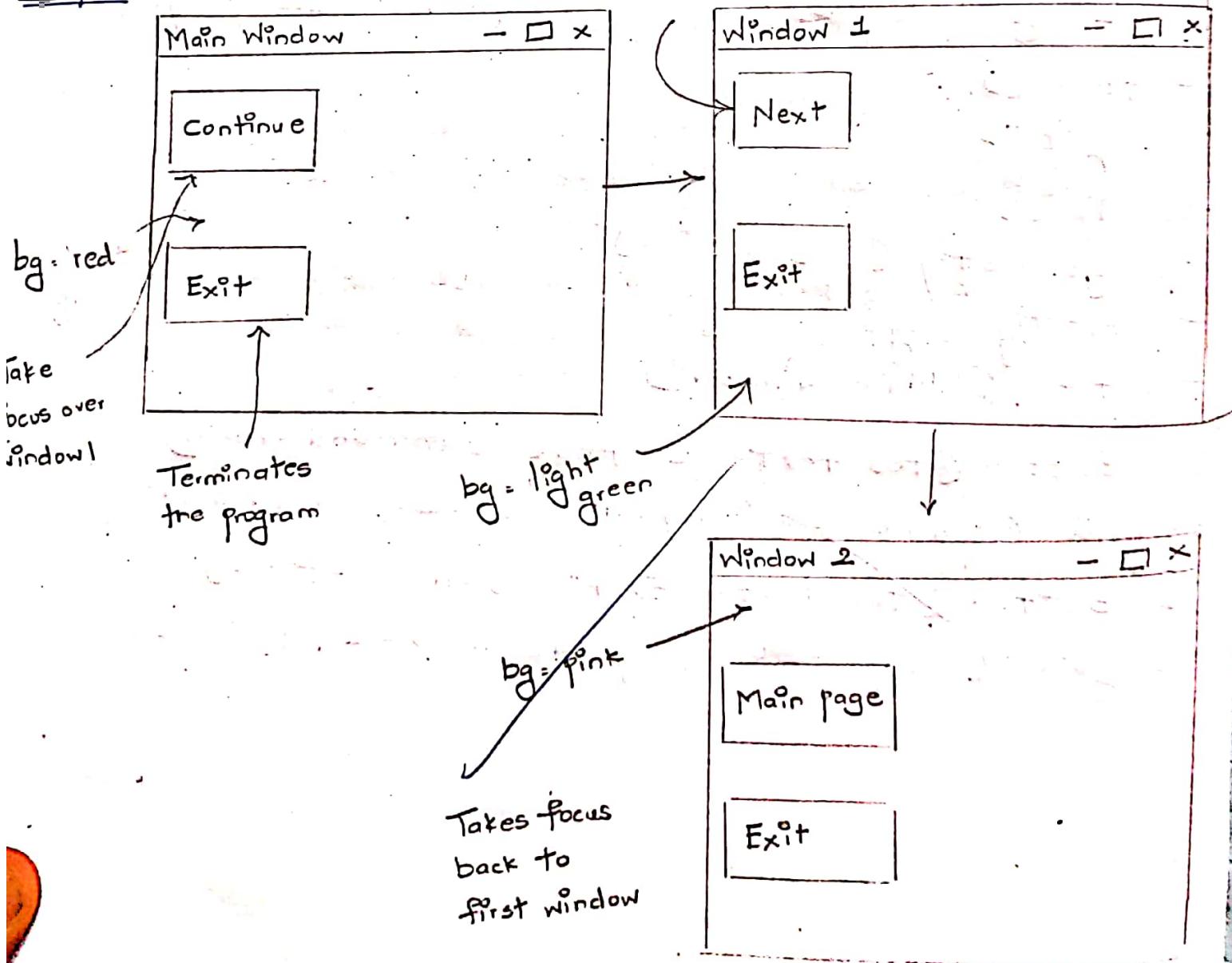
```

```

def main():
    top = Tk()
    top.geometry("450x500")
    top.config(bg = "pink")
    top.title("Window 2")
    B4 = Button(top, text = "Main page", 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()

```

Output:



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

Step 8: Finally call the mainloop method.

The buttons are placed with help of grid

method and naming "Third window"

DO

fp

- Spinbox

Algorithm

Step 1: Create an object from the tk method and subsequently create an object from the spin box method.

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

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

Step 4: Use the mainloop method to terminate.

Program:

```
from tkinter import *
root = Tk()

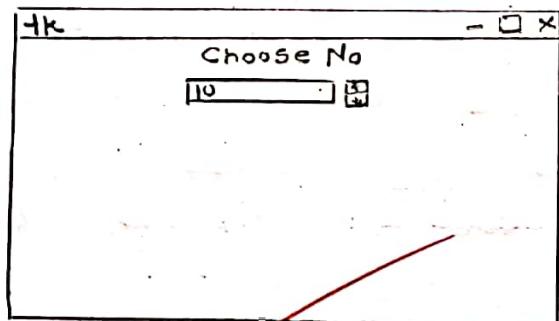
L = Label(root, text= "Choose No.", bg = "red")
L.pack()

bl = Spinbox(root, from_=0, to = 10, font = 22)
bl.pack(side = TOP)

root.mainloop()
```

44

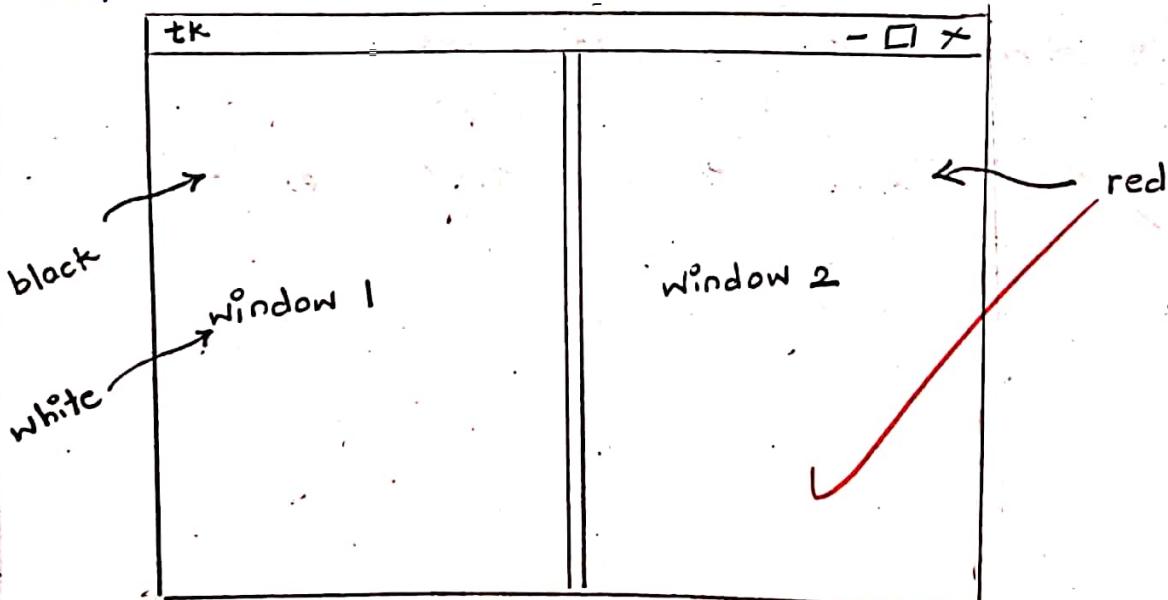
Output:



Program :

```
from tkinter import *
root = Tk()
p = PanedWindow()
p.pack (fill=BOTH, expand=10)
L = Label (p, text="window 1", bg="black", fg="white")
p.add (L)
p1 = PanedWindow (p, orient=VERTICAL)
p.add (p1)
L1 = Label (p1, text="window 2", bg="red")
p1.add (L1)
p2 = PanedWindow (p, orient=HORIZONTAL)
p.add (p2)
mainloop()
```

Output:



Paned Window.

Algorithm:

Step 1: Create an object from paned 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 paned window with the text attribute and use add method to embed the new object.

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

Algorithm

Step 1: Use the tkinter method and create an object from the canvas method and use the attribute height, width, bg colour and the parent window object.

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

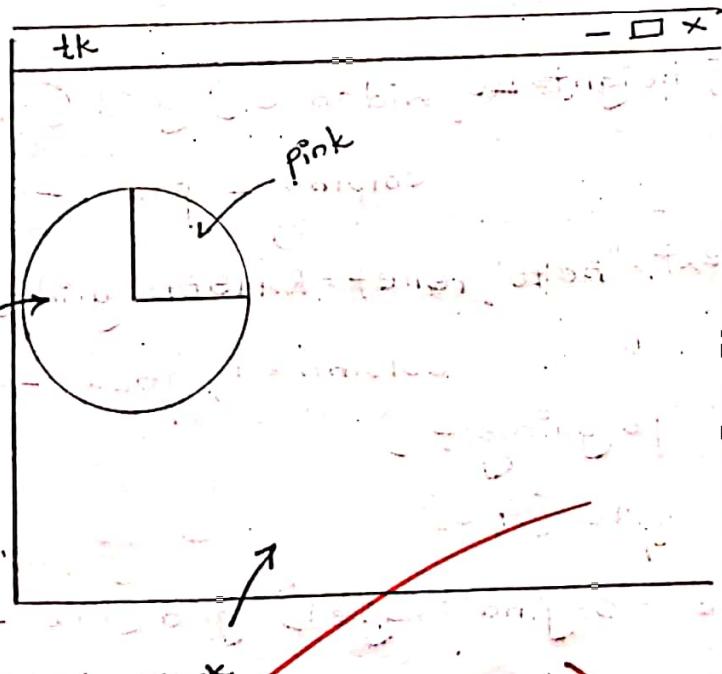
Step 3: Now call the pack method and mainloop method.

Program:

```
from tkinter import *
root = Tk()
cl = Canvas(root, height = "400", width = "400", bg = "black")
line = cl.create_line(10, 90, 170, 250, fill = "black")
oval = cl.create_oval(10, 90, 170, 250, fill = "red")
arc = cl.create_arc(10, 90, 170, 250, fill = "pink")
```

46

Output:



Program:

```
from tkinter import *
root = Tk()
root.geometry ("450x500")
root.title ("Parent Window")
root.maxsize (height=450, width=600)
root.config (bg = "green")
leftframe = Frame (root, height=20, width=30).grid (row=0,
                                                 column=1, padx=10, pady=30)
rightframe = Frame (root, height=20, width=30).grid (row=2,
                                                    column=2, padx=20, pady=30)
l1 = Label (leftframe, text = "hello", relief=RAISED).grid (row=0,
                                                            column=1, padx=20, pady=20)
p1 = PhotoImage (file = "penguin.gif")
originalimage = p1.subsample (4,5)
Label (leftframe, image = originalimage).grid (row=2, column=2,
                                               padx=20, pady=30)
mainloop ()
```



Image

Algorithm:

Step 1: Create an object corresponding to the parent window and use title, maxsize & config method.

Step 2: Create a leftframe object from the frame method & place it on to the parent window with the height, width & the background attribute specified subsequently use the grid method with the row, column padx & pady attribute specified.

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

Step 4: Create a label object from the label method & place it onto the left frame with the text attribute denoting original image with relief attribute used as a raised value & subsequently use the grid method with row, column value specified.

5A

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

Step 6: Use the subsample method with the object of the image & if the x & y co-ordinate values.

Step 7: Use label method & position it onto the left frame & placing the image after the Sampling & use the grid method for positioning in the 1st row.

Step 8: Create Another label object, positioning it onto rightframe & specifying the image & the background attribute with the row & column attribute specified.

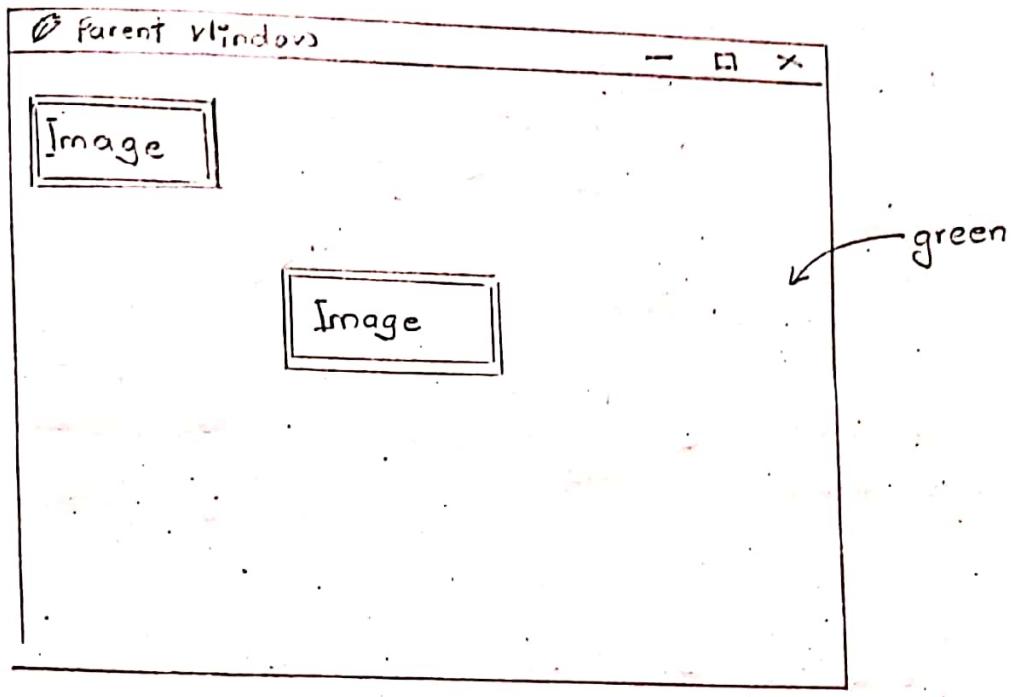
Step 9: Now create a toolbar object from the frame method & position it onto the left frame with the height & width specified & position it onto the 2nd row.

Step 10: Now define the various function for different toolbar options provided in the left frame.

Step 11: From the label method position the text onto the toolbar use the

Output:

48



relief attribute & corresponding the grid & incorporate the internal padding as well.

Step 12: Create the label method position it onto the toolbar with the next title as personal information & position it on the same row but new column.

✓ Jinal

PP

Practical - 6.

Aim: To make use of various statements of database library.

Step 1: Import db library and use the open method for creating the database by specifying name of the database along with the corresponding flag.

Step 2: Use the objects for accessing to given web size and the corresponding regular for the web size.

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

Program

50

```
import dbm  
db = dbm.open ("data base", flag="c")  
if db["www"] != None:  
    print ("Good")  
else:  
    print ("Bad")
```

Output:

Good.

Jinal

12

Program

```
import os, sqlite3  
Connection = sqlite3.connect("student.db")  
c1 = Connection.cursor()  
c1.execute('Create Table student (Name, RNO, DOB)')  
c1.execute('insert into student values ("Dhaval",  
1765, 24/06/2001)  
              "Chinmay",  
1760, 25/03/2001)  
              "Ritik"  
1773, 21/04/2001)
```

```
c1.execute('insert into student values ("Ritik"  
1773, 21/04/2001)
```

Connection.commit()

```
c1.execute('Select * from student')  
c1.fetchall()  
c1.execute('Drop Table student')
```

b)

Step 1: Import the Corresponding library taking of database connection.

Step 2: Now Create Connection objects using Sqlite library and Connecting method for creating the new database.

Step 3: Now Create the Cursor object using Cursor method from the connection object created in step 2.

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

Step 5: Now with the Cursor object use insert statements for entering the value co-ordinating into the different field field considering the data types.

Step 6: Use the Commit method to complete the transaction use the connection object.

Step 7: Use the execute statement along with the Cursor object for executing the value the data base 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.

Output:

52

[('Dhaval', 1765, 24/05/2001), ('Chinmoy', 1760, 25/03/2001),
('Ritik', 1773, 21/04/2001)]

Project 1 GUI

25

```
from tkinter import*

def btnClick(numbers):
    global operator
    operator+=str(numbers)
    text_Input.set(operator)

def btnClearDisplay():
    global operator
    operator=""
    text_Input.set("")

def btnEqualsInput():
    global operator
    sumup=str(eval(operator))
    text_Input.set(sumup)
    operator=""

cal=Tk()
cal.title("Calculator")
pic=PhotoImage(file='calc.png')
pic.subsample(3,3)
cal.iconphoto(False, pic)
operator=""
text_Input=StringVar()

frame=Frame(cal, width=4, height=25).grid(row=0, columnspan=4)
```

```

Label(frame, text='CALCULATOR', font=('AvenirNext LT Pro Bold', 20, 'bold'), relief=SOLID,
      bg='powder blue', width=20, height=2).grid(row=0, columnspan=4)

Label(cal, image=pic, bg='powder blue').place(x=5, y=2)

txtDisplay=Entry(cal, font=('arial', 22, 'bold'), textvariable=text_Input, bd=10, insertwidth=4,
                  bg="white", justify='right').grid(row=1, columnspan=4, ipady=20)

btn7=Button(cal, padx=16, pady=16, bd=8, fg="black", font=('arial', 20, 'bold'), width=2, height=1,
            text="7", bg="powder blue", command=lambda:btnClick(7)).grid(row=2, column=0)

btn8=Button(cal, padx=16, pady=16, bd=8, fg="black", font=('arial', 20, 'bold'), width=2, height=1,
            text="8", bg="powder blue", command=lambda:btnClick(8)).grid(row=2, column=1)

btn9=Button(cal, padx=16, pady=16, bd=8, fg="black", font=('arial', 20, 'bold'), width=2, height=1,
            text="9", bg="powder blue", command=lambda:btnClick(9)).grid(row=2, column=2)

Addition=Button(cal, padx=16, pady=16, bd=8, fg="black", font=('arial', 20, 'bold'), width=2, height=1,
                 text="+", bg="#26c2ff", command=lambda:btnClick("+")).grid(row=2, column=3)

btn4=Button(cal, padx=16, pady=16, bd=8, fg="black", font=('arial', 20, 'bold'), width=2, height=1,
            text="4", bg="powder blue", command=lambda:btnClick(4)).grid(row=3, column=0)

btn5=Button(cal, padx=16, pady=16, bd=8, fg="black", font=('arial', 20, 'bold'), width=2, height=1,
            text="5", bg="powder blue", command=lambda:btnClick(5)).grid(row=3, column=1)

btn6=Button(cal, padx=16, pady=16, bd=8, fg="black", font=('arial', 20, 'bold'), width=2, height=1,
            text="6", bg="powder blue", command=lambda:btnClick(6)).grid(row=3, column=2)

Subtraction=Button(cal, padx=16, pady=16, bd=8, fg="black", font=('arial', 20, 'bold'), width=2, height=1,
                     text="-", bg="#26c2ff", command=lambda:btnClick("-")).grid(row=3, column=3)

btn1=Button(cal, padx=16, pady=16, bd=8, fg="black", font=('arial', 20, 'bold'), width=2, height=1,
            text="1", bg="powder blue", command=lambda:btnClick(1)).grid(row=4, column=0)

btn2=Button(cal, padx=16, pady=16, bd=8, fg="black", font=('arial', 20, 'bold'), width=2, height=1,
            text="2", bg="powder blue", command=lambda:btnClick(2)).grid(row=4, column=1)

btn3=Button(cal, padx=16, pady=16, bd=8, fg="black", font=('arial', 20, 'bold'), width=2, height=1,
            text="3", bg="powder blue", command=lambda:btnClick(3)).grid(row=4, column=2)

```

```
Multiply=Button(cal,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'), width=2, height=1,  
    text="*",bg="#26c2ff",command=lambda:btnClick("*")).grid(row=4,column=3)  
  
btn0=Button(cal,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'), width=2, height=1,  
    text="0",bg="powder blue",command=lambda:btnClick(0)).grid(row=5,column=1)  
  
btnClear=Button(cal,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'), width=2, height=1,  
    text="C",bg="#fc392b",command=btnClearDisplay).grid(row=5,column=0)  
  
btnEquals=Button(cal,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'), width=2, height=1,  
    text="=",bg="#8ce300",command=btnEqualsInput).grid(row=5,column=2)  
  
Division=Button(cal,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'), width=2, height=1,  
    text="/",bg="#26c2ff",command=lambda:btnClick("/")).grid(row=5,column=3)  
  
cal.mainloop()
```

