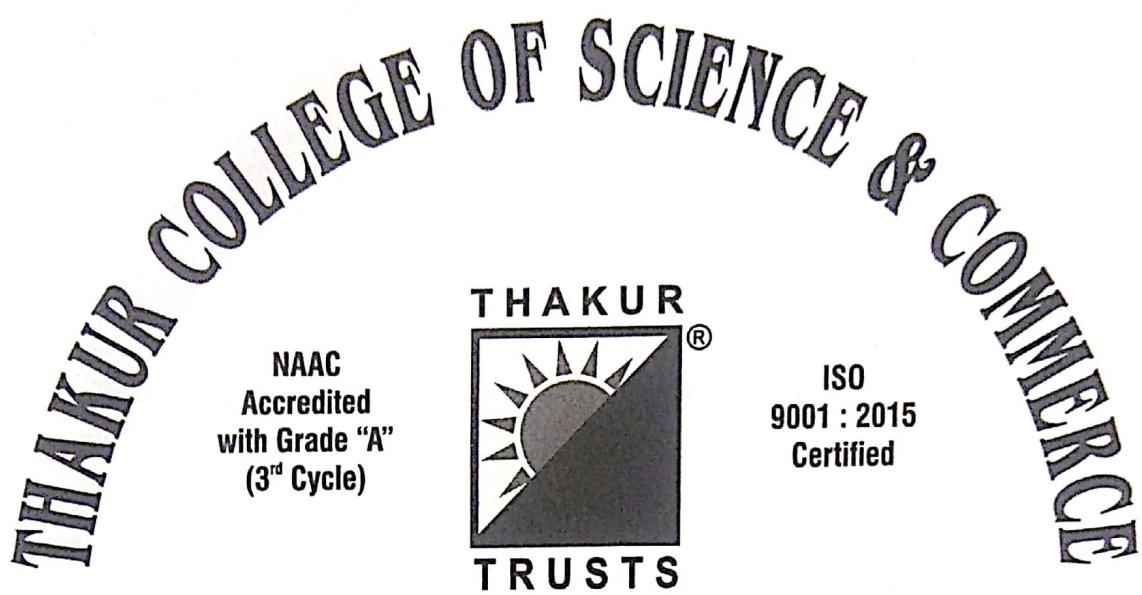


Exam Seat No. _____



Degree College

Computer Journal

CERTIFICATE

SEMESTER II UID No. _____

Class FYCS (A) Roll No. 1742 Year 2019-20

This is to certify that the work entered in this journal
is the work of Mst. / Ms. Raj waghela

who has worked for the year 2019-20 in the Computer
Laboratory.

Teacher In-Charge

Head of Department

Date : _____

Examiner

★ ★ INDEX ★ ★

| No. | Title | Page No. | Date | Staff Member's Signature |
|-----|---|----------|----------|----------------------------|
| 1. | Modes of operation | 27 | 25/11/19 | 0/0 missing Jm 27/11/19 |
| 2. | Iterators & Iterable | 28 | 2/12/19 | Jm Jan 11/19 |
| 3. | Exception Handling | 37 | 27/1/20 | Jm 1 |
| 4. | Regular expression | 41 | 27/1/20 | Jm 1 27/1/20 group 1/1 |
| 5. | GUI Application | 45 | 27/1/20 | Jm 1 |
| | 1) Basic method | 45 | 29/1/20 | |
| | 2) Radiobutton widget. | 45 | 29/1/20 | |
| | 3) Scrollbar & listbox widget | 47 | 3/2/20 | 1/1 |
| | 4) frame & button widget | 49 | 3/2/20 | 1/1 |
| | 5) messagebox widget. | 49 | 3/2/20 | 1/1 |
| | 6) Traversing various window using button widget. | 51 | 3/2/20 | |
| | 7) spinbox widget. | 53 | 17/2/20 | |
| | 8) Paned windows widget. | 53 | 17/2/20 | Jm 1/2 |
| | 9) Canvas method. | 55 | 17/2/20 | |
| | 10) Image displaying | 56 | 17/2/20 | Jm 1/2 |
| 6. | To learn the use of database in python | 59 | 17/2/20 | Jm 1/2 |

i) code

```
a = open ("myfile.txt", "w")
a.write ("Good boy", "\n")
a.write ("I am a bad boy", "\n")
a.write ("You are wicked", "\n")
a.close()
```

ii) code: using read()

```
a = open ("myfile.txt", "r")
```

```
str1 = a.read()
```

print ("The O/p and read method is", str1)

```
a.close()
```

O/p: The O/p and read method is Good boy

~~code: using read~~

~~I am a bad boy~~

~~You are wicked~~

ii) code: using readline()

```
a = open ("myfile.txt", "r")
```

```
str1 = a.readline()
```

print ("The O/p and read method is", str1)

```
a.close()
```

Q) Aim: To create a file modes of file operations

Q) Aim: To create a file

Algorithm:-

Step 1: Create a file object using the `open` method

Step 2: Using `write` access mode to the
write content in the file

Step 3: Close the file when not needed.

Q) Aim: Open the file in read mode and
use the `read`, `readline` & `readlines` method
to store the O/P in the variable

Algorithm:

Step 1: Open the file in the read mode
using a file object.

Step 2: Read the contents of the file using
~~read()~~, `readline` & `readlines`

Step 3: Close the file.

3) Aim: Use the file object for finding name of the file, mod in which file is opened, whether file is still opened.

Algorithm:

Step 1: open the file in read mode

Step 2: Using file object use a. closed,
a. mod , a.softspace

Step 3: Close the file.

Q: The output and read method is good boy
28

iii: using readlines () :-

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

str1 = a.readlines()

print("The o/p and read method is, " str1)
a.close()

O/P: The o/p and read method is ['Good boy\n',
'I am a bad boy\n', 'You are wicked']

Code:-

a

3) code:

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

b = a.closed

c = a.mode

d = a.softspace

print("a closed", b)

O/P ?

ans:-

21/2/19

print("a mode", c)

print("a softspace", d)

⁸⁸
Output: a closed - True
a mode - x
a softspace - 0

4)
myfileobj = open ('myfile.txt', 'x')
myfileobj. ~~close~~ read()
pos = myfileobj. tell()
print (pos)
str1 = myfileobj. seek(0, 1)
print(str1)
output: 80 40
None None

7)
a = open ('myfile.txt', 'r')
str1 = a.readlines()
for line in str1:
 print(unline)
output: 9
15
14

4) Aim: To implement tell and seek method.

Algorithm:

Step 1: Open the file using the object in a read mode.

Step 2: Using the variable pos implement the tell method and print it.

Step 3: Using the str use the seek method and print the value.

5) Aim: Using file operation read individual element from a file.

Step 1: Open the file using conditional statement.

Step 2: Store the read file in some variable.

Step 3: Using the write conditional statement check whether the length of variable is greater than 200.

Step 4: Print the length.

6) Aim: To find the frequency of occurrence of given word

Step 1: Open the file and read it using the `read()`.

2: Initialize a counter variable to zero

Step 3: Use the for conditional statement followed by if conditional statement to find frequency.

Step 4: Increment counter variable by 1

Step 5: Print the frequency using print statement.

7) Aim: To find frequency of occurrence of given word.

Step 1: Open the file and read it using `read()`.

Step 2: Read the line using the `readline()` and assign a variable to it

Step 3: Using the for conditional statement for the length of the line.

Step 4: Print it using print method.

6) `a = open ("myfile.txt", "r")`
`str1 = a.read()`
`count = 0`
`for word in str1:`
 `if (word == 's'):`
 `count += 1`
 `print (count)`

Output :
1
2
3
4
5
6
7
8

5) with `open ("my.txt", "r") as g:`
`s = 1`
~~`c = g.read(2)`~~
~~`while len(c) > 0:`~~
~~`print (c, end = "#")`~~
~~`c = g.read(2)`~~

Output :
Hey#, # J# am# Ra#J!# wa#gh#el#ha

1) 16

mytuple = ('Banana', 'Apple', 'orange')

myiter = iter(mytuple)

print(next(myiter))

print(next(myiter))

Output:

Banana

Apple

2) mytuple

mytuple = ('Banana', 'Apple', 'orange')

for x in mytuple:

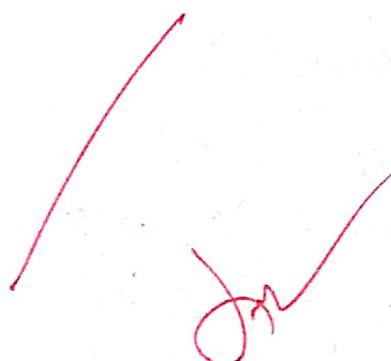
print(x)

Output:

Banana

Apple

orange



Practical 2

31

1) Aim: To display elements of tuple ^{using} among the iterator method.

Algorithm:

Step 1: form a tuple by and containing some elements in it.

Step 2: Use the iter method followed by the next method to iterate the variable.

Step 3: Print the elements.

2) Aim: To use the iterator method with for loop.

Algorithm:

Step 1: form a tuple containing elements in it.

Step 2: Use the for individual conditional statement. to iter the elements.

Step 3: Print the elements.

3) Aim : To print odd numbers using iterable method

Algorithm :

Step 1: Define a class and within that define a iter method which will initialize the first variable within the container object.

Step 2: Now use the next method and define the logic for displaying the odd values.

Step 3: Create an object to print the values.

4) Aim :- using iterable display set of first 50 numbers.

Step 1: Define a iter method with an argument and initialize it to a first value.

Step 2: For extracting the next element from the container use the next method with an argument and compare no. of elements received in a container.

Step 3: Now create an object from the given class and pass the object as an argument to the iter method.

Step 4: Now using the conditional statement display all the value.

3) 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)
```

```
a = int(input("Enter no"))
```

```
for x in myiter:
```

```
    if x < a:
        print(x)
```

output: Enter no : 20

1

3

5

7

9

11

13

15

17

19

4) ~~Q8~~

class myclass:

def __iter__(self):

self.a = 1

return self

def __next__(self):

if self.a <= 20:

n = self.a

self.a += 1

return n

else:

raise StopIteration

myobject = myclass()

myiter = iter(myobject)

for n in myiter:

print(n)

Output:

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

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

Q8/2/13

5) Aim : To find the first 10 numbers factorial using iter method.

Algorithm:

- Step 1: Define a class with name factorial. Within that define a iter method which will initialize the first variable with an object.
- Step 2: Now define the next method and assign a variable to the first initialize variable of iter method. and then increment the first variable with 1 with using appropriate logic.
- Step 3: Now create an object ob from the given class and pass the value object as an argument to the iter method. Then take and user input of an integer value and next assign a variable n to 1.
- Step 4: Use the for statement for loop variable n in range iter variable, following if statement that n is less than equal to user input value. and use an appropriate logic for factorial formula, and use the print statement to display factorial of number.

5)

34

class factorial :

def __iter__(self) :

self.a = 1

return self

def __next__(self) :

a = self.a

self.a += 1

return a

myobj = factorial()

myiter = iter(myobj)

no = int(input('Enter a number'))

r = 1

for x in myiter :

if x <= no :

r *= x

print(r)

Output: Enter number 10

1

2

6

24

120

720

5040

40320

362880

3628800

Q8:

```
1) List1 = list(input("Enter:"))
def even(n):
    if n%2 == 0:
        return ("even")
    else:
        return ("odd")
print list(map(even, list1))
```

Output:

- Enter: 0, 4, 5, 7, 9, 11, 15, 20, 25

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



To demonstrate exception handling

- 1) Aim: To find the number is even or odd using map method

Step1:

Algorithm:

Step1: Take a list datatype of some numbers by user by using list and input method subsequently.

Step2: Define a function with a single argument. And subsequently use if statement. if the parameter

Step3: return even when when the parameter parameter is modular by 2 or else. return odd.

Step3: By using map method map the input list datatype with given function and convert it into list datatype and display it using print statement.

- 2) Aim: To find square and cube of a number in a list using map method.

Algorithm:

Step1: Take input of numbers by user and convert it into list datatype.

Step 2: Define a function for square using single parameter which will return (x^{**2}) square of given data by user to user.

Simultaneously, again use the define a function for cube using single parameter which return (x^{**3}) cube of given data.

Step 3: In list take square and cube and assign it to a variable.

Step 4: Use for statement for i in list given by user. By using map method with lambda function using appropriate logic for mapping the given input with the define function.

Step 5: After mapping use convert it into list datatype and display it using print method.

3) Aim: To Display the / the list of the square of numbers using append method.

Algorithm:

Step 1: Take input of number by user and convert it into list datatype.

```
2) list1 = List(input("Enter"))  
def square(n):  
    return n**2  
def cube(n):  
    return n**3  
funct = [square, cube]  
for i in list1:  
    valueout = map(lambda x: x(i), funct)  
    print(List(valueout))
```

Output: Enter: 1, 2, 3, 4, 5, 6

[1, 1]

[4, 8]

[9, 27]

[16, 64]

[25, 125]

[36, 216]



18

3) list1 = list(input("Enter:"))

list2 = []

for i in range(list1):

list2.append(i**2)

print(list2)

Output: Enter: 1,4,2,5,8,12,9

[1, 16, 4, 25, 64, 144, 81]



Step2: Assign a variable with empty list

Step3: Use for statement for i in a loop variable i in list of given number. Subsequently use the append method using appropriate logic as to square the given user list by appending it in the empty list and display it using print statement.

Step4:

JW
16/12/19

Practical - 03

To Demonstrate Exception handling.

) Aim: To show environmental error using the exception block.

Step1: Use the try block for defining the normal course of action. For eg: define the file object and open file in the write mode. and write some content in file.

Step2: Use the except block with IOError as an environment error convey the appropriate message to user else display the message that the operation is carried out successfully.

1) try:
fileobj = open("abc.txt", 'w') 38
fileobj.write ("This program of exception will
show there is environmental error
or not")

Except IOError:

else print ("There is an environmental error")

else:

print ("The exception is successfull")

output:

The exception is successfull.

try:
file obj = open ("abc.txt", 'r')
fileobj.write ("This program of exception
will show there is environmental error
or not")

Except IOError:
print ("There is an environmental error")

else:

print ("The exception is successfull")

output:

There is an environmental error

Q) try:

```
user = int(input('Enter value: '))
a = float(user)
print('Result = ', a)
```

Except IOError:

```
    print('There is an environmental error')
```

Except ValueError:

```
    print('There is an value error')
```

else:

```
    print('Operation is successful!')
```

Output:-

i) Enter value : 5a

There is an value error

ii) Enter value : 4

Result = 1.33

Operation is successful!

2) Aim: To demonstrating use of value error in given program.

Step1: Accept the value from the user and if it is a valid value, display the entered value and terminate the condition by using break statement

Step2: Def the except block with the value error as a keyword and display the appropriate message.

Step3: We can do the multiple exception using the except statement for handling the different categories of errors, and accordingly use print statement to display type of errors.

Jas

3) Aim: To demonstrate the use of Index Error.

Step 1: Use the try block and define a list with some elements in it.

Step 2: Use the indexing of list and print the element of that list.

Step 3: Now define a list which is empty. Try to print.

Step 4: Def an Except block with any type of error and display appropriate message. We can also define multiple exception by using except statement. Else display the operation is successfull.

40

```
try:  
    def a():  
        list1 = []  
        s = (list1)  
        return s  
    def b():  
        list2 = [1, 2, 3, 4, 5]  
        d = len(list2)  
        return d  
  
except ValueError:  
    print('There is an value error')  
  
else:  
    print('The operation is successfull!')
```

Output:

The operation is successfull!

1) import re

pattern = r"FYCS"

sequence = "FYCS represents computer science stream!"

if re.match(pattern, sequence):

 print("match pattern found")

else:

 print("NOT FOUND")

Output:

match pattern found.

2) import re

pattern = r'\d+'

sequence = 'hello123, howdy789, 45howru'

output = re.findall(pattern, sequence)

print(output)

Output: ['123', '789', '45']

3) Import re

pattern = r'\d+'

string = 'hello123, howdy789, 45howru'

result = re.split(pattern, string)

print(result)

Output:

['hello', 'howdy', 'howru']

Topic : Regular expression

1) matching the pattern

Step : 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.

2) numerical value separator

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

3) split method.

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

4) no space using sub()

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

5) To search the memory location

Step: Import re module declare a sequence use search method for finding subsequently use ch grouped() with dot operator as search() gives memory location using grouped. It will show up the match string.

6) verify the set of phone number

Step: 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 numbers are in range of 0 to 9 and check whether the entered numbers are equal to 10. if criteria matches print all numbers matches otherwise print failed.

4) Import re

pattern = r'1st'

string = 'abc def ghi'

replace = ''

v1 = re.sub(pattern, replace, string)

print(v1)

Output: 'abcdefghi'

42

5) Import re

sequence = 'Python is interesting language'

result = re.search('Python', sequence)

print(result)

Output:

a = result.group()

print(a)

Output:

<_SRE_SRE - match object at 0x0281DF0>

6) Import re

sequence = ['8004567891', '9145673210', '7865432891',
 '9876543210']

for value in list(sequence):

If re.match(r'[8-9]\d{1}\d{9}',

value or len(value) == 10):

print("Criteria match for cell number! ")

else:

print("Criteria failed!")

Output: Criteria match for cell number!

Criteria match for cell number!

Criteria failed!

Criteria match for cell number!

7) Import re

str1 = 'DS is known datstructure in python'

result = re.findall(r'\b[aeiouAEIOU]\w+', str1)

print(result)

Output: ['is', 'in']

8) Import re

seq = 'abc.tese@edu.com , xyz@gmail.com'

pattern = r'([ax-zA-Z]+@[ax-zA-Z]+\.)'

output = re.findall(pattern, seq)

print(output)

Output: ['abc', 'tese', 'edu.com', 'xyz', 'gmail.com']

9) Import re

s = 'mr.a.ms.l.mr.t'

p = r'([mr])\1'

op = re.findall(p, s)

print(op)

m = op

f = op

for i in op:

if (v == 'ms'):

f = f + 1

else:

m = m + 1

7) To find vowels using findall()

Import re module . declare a string use the module with findall() for finding the vowels in string and declare the same.

8) To print host and domain name

Step: Import re module . declare the host , and domain name declare pattern for separating the host & domain name. Use the findall() and print the output.

9) Counting of first two letter.

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

Jmjl

print("No of males is : ", m)
print("No of females is : ", f)

41

Output:

['ma', 'ms', 'ms', 'mr']

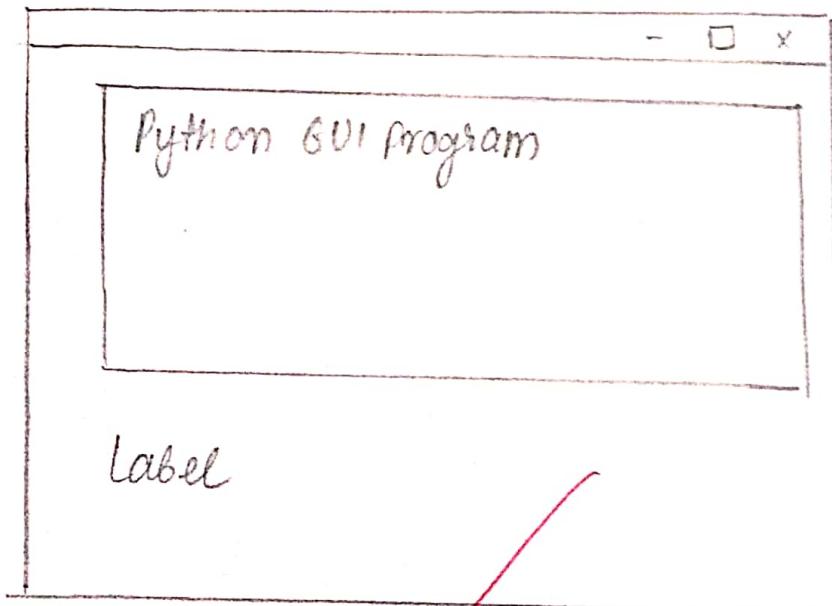
(No of males is : 2)

(No of females is : 2)

Or 22

```
from tkinter import *  
root = Tk()  
t1 = Text(root)  
t1.insert(END, "Python GUI program")  
t1.pack(side=TOP, padx=20, pady=20, ipadx=4,  
        ipady=50)  
label = Label(root, text="Label")  
label.pack(side=LEFT, padx=10, pady=20, ipadx=3,  
           ipady=40)  
root.mainloop()
```

Output :-



Practical - 5.

45

Q 1) Aim : To make use of the GUI application along with basic method.

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

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

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

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 colour (bg)

4) The foreground colour (fg) and then use the pack() with a relevant padding attributes.

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

Step 1: Use the tkinter module to include the relevant method.

Step 2: Define a function which tell the user about the given selection made from the multiple option available.

Step 3: Use the config() along with the label object and call the variable as an argument within the method.

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

Step 5: Now create an object from radiobutton() which will take following arguments.

- 1) Position on the parent window.
- 2) Defining the text variable which will takes the values option no. 1, 2, 3, 4..
- 3) Define the variable argument.
- 4) Corresponding value and trigger the given function.

Step 6: Now call the pack() for radio object so created & specify the object/argument using 'ANCHOR' attribute.

from tkinter import *

46

def sel1():

selection = "Rs 50 / per "

l1 = Label(text=selection)

l1.pack(side=TOP)

def sel2():

selection = "Rs 80 / per "

l2 = Label(text=selection)

l2.pack(side=TOP)

def sel3():

selection = "Rs 40 / per "

l3 = Label(text=selection)

l3.pack(side=TOP)

root = Tk()

root++

var = IntVar()

l1 = Label(root, text="Select any one")

l1.pack(side=TOP)

r1 = Radiobutton(root, text="Tomato", value=0,
command=sel1)

r1.pack(anchor=N)

r2 = Radiobutton(root, text="Onion", value=1,
command=sel2)

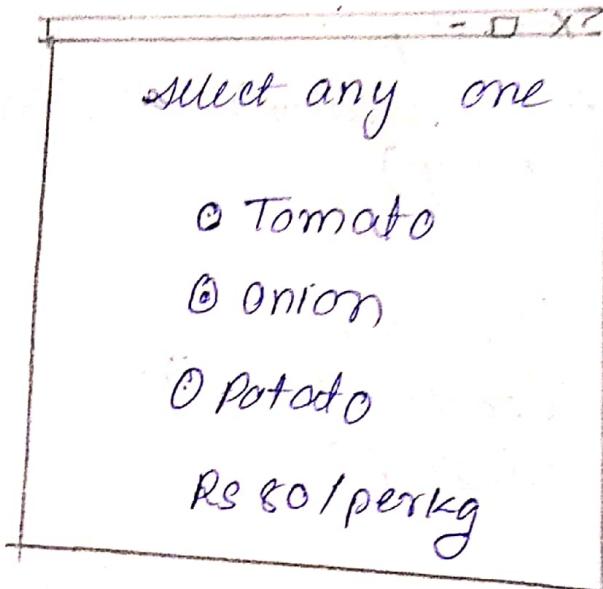
r2.pack(anchor=N)

r3 = Radiobutton(root, text="Potato", value=2,
command=sel3)

r3.pack(anchor=N)

¹¹
label = Label (root)
label.pack (side = Bottom)
root.mainloop()

output:



Step 7: Now define the 'label' from the corresponding method & place it on parent window subsequently use the pack() from the widget and finally make use of the mainloop().

~~Not
Done~~

3) Aim : To make use of scroll bar widget

Step1: Import relevant methods from the tkinter library.

Step2: Create an object corresponding to the parent window & create an object from the scrollbar () & place it on the parent window so created.

Step3: Create an object from the text () & place it on to the parent window with the height & width specified.

Step4: Use the pack() along with the object of the scrollbar method & use the argument as side & fill.

Step5: Now use the text object along with the pack () & again use the side & the fill attribute.

Step6: Now use the config() along with the pack() & again use the side and the fill attribute.

Step7: Similarly use the config along with the text object & use the scroll command argument.

Step8: Now define the textual information in terms of paragraph and use the insert() with two arguments & call the mainloop() at last.

```
from tkinter import *
```

```
root = Tk()
```

```
root.geometry("680x500")
```

```
l = Label(root, text= "Software team member rollno.",  
         bg= 'black', fg= 'white')
```

```
l.pack(side=TOP)
```

```
scroll = Scrollbar(root)
```

```
scroll.pack(side=RIGHT, fill=Y)
```

```
mylist = Listbox(root, yscrollcommand= scroll.set,  
                 bg= 'light blue')
```

```
for num in range(40, 70):
```

```
    mylist.insert(END, "rollnumber : " + str(num))
```

```
mylist.pack(side=LEFT, fill=BOTH)
```

```
scroll.config(command= mylist.yview)
```

```
root.mainloop()
```

output:

Software team member roll no.

- rollnumber : 1740
- rollnumber : 1741
- rollnumber : 1742
- rollnumber : 1743
- rollnumber : 1744

```
from tkinter import *
```

```
root = Tk()
```

```
root.geometry ("680x500")
```

```
frame = Frame (root)
```

```
frame.pack ( )
```

```
left = frame (root)
```

```
left . pack ( )
```

```
right = Frame (root)
```

```
right . pack ( )
```

```
b1 = Button (root, text = "Select", bg = "red", fg = "black")
```

```
b2 = Button (root, text = "Modify", bg = "yellow", fg = "black")
```

```
b3 = Button (root, text = "ADD", bg = "blue", fg = "red")
```

```
b4 = Button (root, text = "Exit", bg = "red", fg = "green")
```

```
b1 . pack (side = LEFT, padx = 20)
```

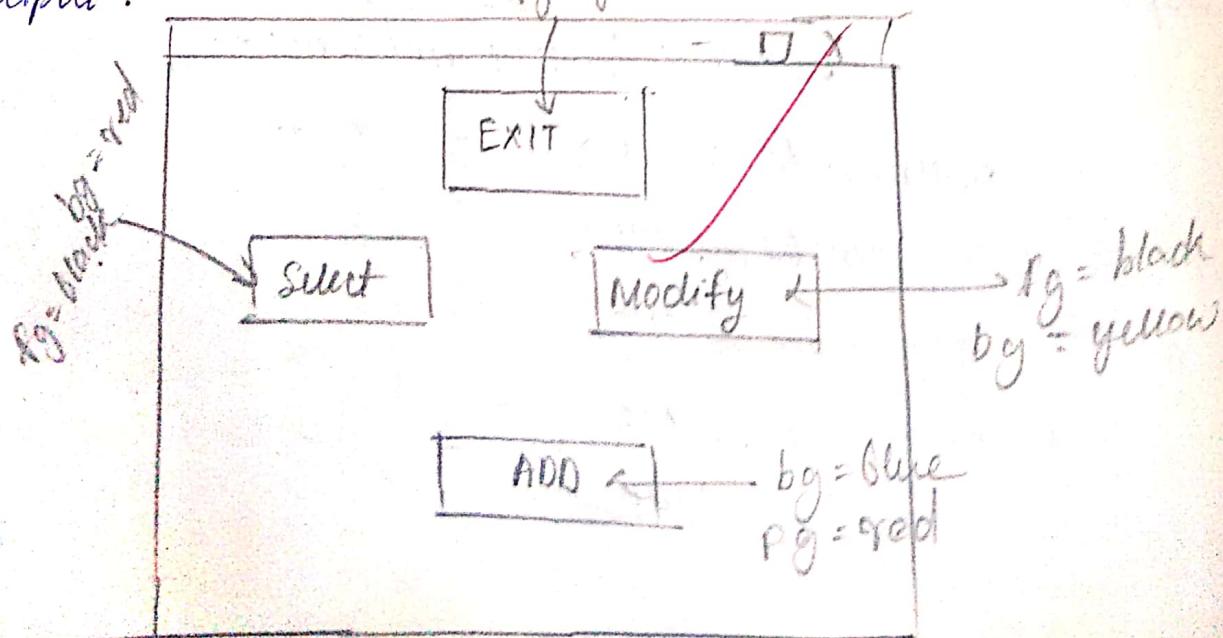
```
b2 . pack (side = RIGHT, padx = 20)
```

```
b3 . pack (side = BOTTOM, pady = 20)
```

```
b4 . pack (side = TOP)
```

```
root . mainloop ( )
```

Output :-



4) Aim : To use the frame object & combining different button objects.

Step 1: Import relevant method from tkinter library.

Step 2: Define the object corresponding to parent window & define the size of parent window in terms of pixel.

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

Step 4: Create another frame object termed as the left frame & put it on the parent window on the left side.

Step 5: Similarly define Right frame and subsequently define the button object placed onto the given frame. with the attribute as text, active background and foreground.

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

~~Step 7: Similarly create the button object corresponding to Modify operation put it onto the frame object with side = right attribute.~~

Step 8: Add another button put it on to the right frame and label as ADD

Step 9: Add another button right frame termed as exit.
and call mainloop().

5) Aim: To make use of messagebox method.

Step 1: Import relevant method from tkinter library.

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

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

Step 4: Use pack method and finally call mainloop()

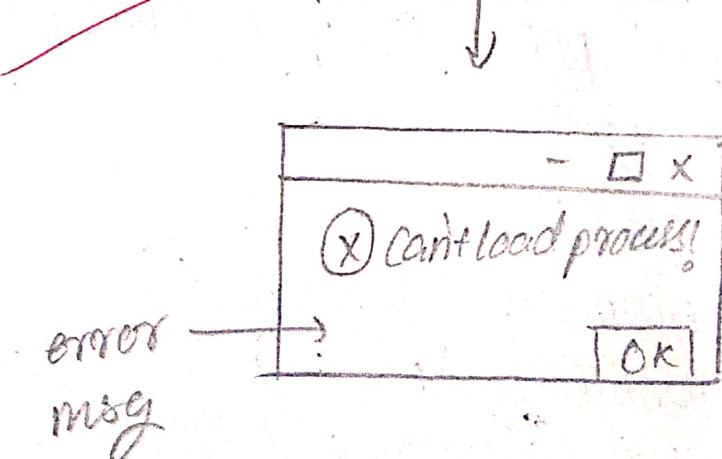
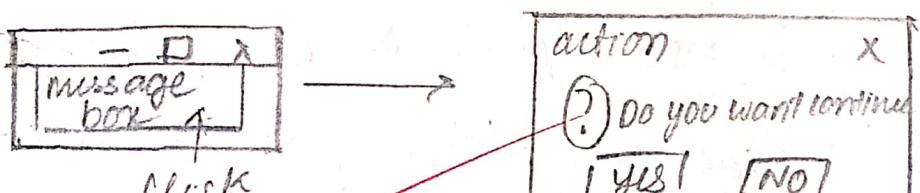
```
from tkinter import*  
from tkinter import messagebox  
def msgbox():  
    messagebox.askyesno("action", "do you want to  
    Continue?")  
    messagebox.showerror('error', 'can't load the  
    process!')
```

```
root = Tk()  
root.config(bg='grey')  
b1 = Button(root, text='message box', bg='blue',  
            command=msgbox)
```

```
b1.pack()
```

```
root.mainloop()
```

output:



from tkinter import *

as

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=10, padx=20, pady=10)

def term():

ans1()

TOS = Tk()

TOS.geometry('450x800')

TOS.config(bg='red')

TOS.title('main window')

B3

b3 = Button(TOS, text='Continue', command=main)

b3.grid(ipadx=50, ipady=10, padx=20, pady=30)

b2 = Button(TOS, text='Exit', command=term)

b2.grid(ipadx=50, ipady=10, padx=20, pady=30)

def main1():

top = Tk()

top.geometry('450x500')

top.config(bg='pink')

top.title('window 2')

b4 = Button(top, text='Mainpage', command=main)

b4.grid(ipadx=50, ipady=10, padx=20, pady=30)

6) pir

A

S

di

X

X

6) Aim : Program to traverse various windows using the button widget.

Algo :

Step 1 : Import the relevant method from tkinter library.

Step 2 : Def a function and create a object of given window by using the tk method namely config title nsize.

Step 3 : Def 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 on to the 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.

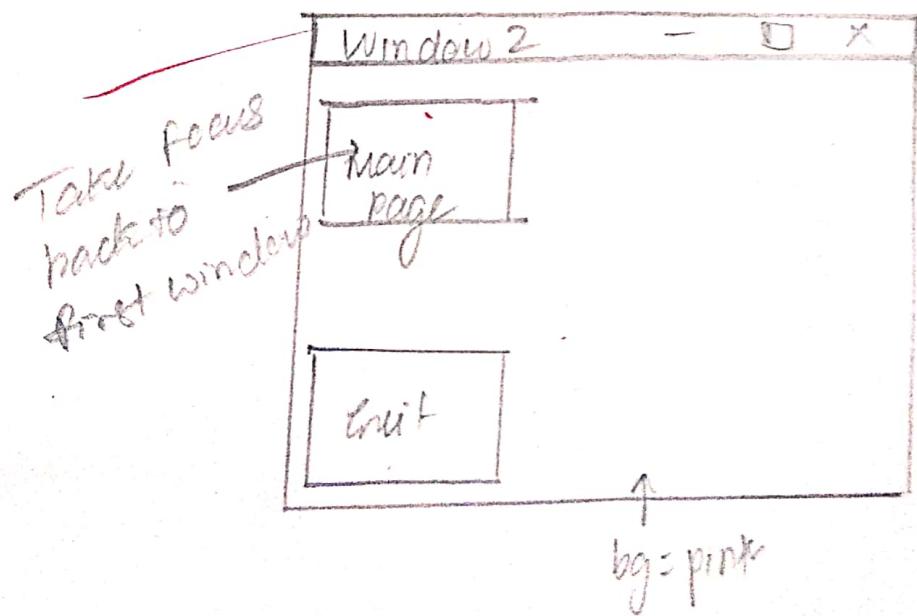
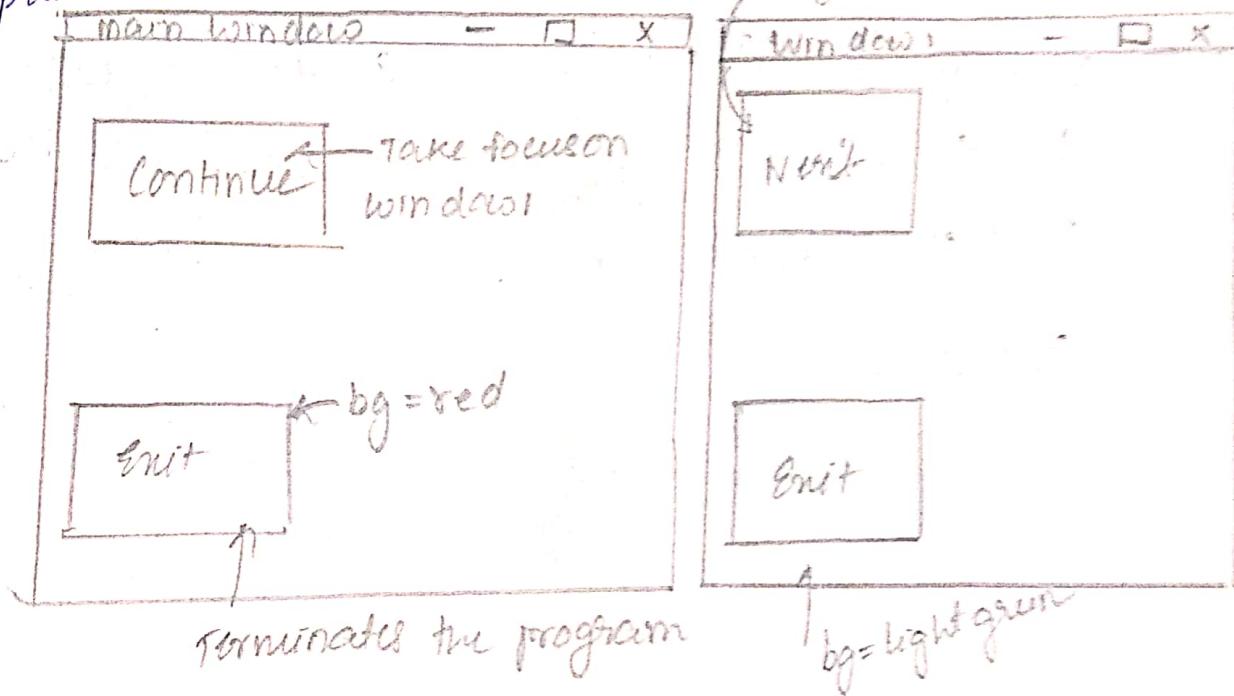
13

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

Step 8: Call mainloop().

Jan 17/2017

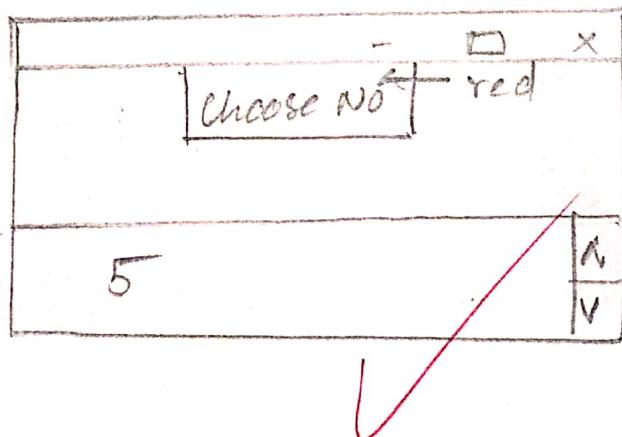
02 = Button (top, text = 'Exit', command=52 term)
 na. grid (ipadx=50, ipady=40, padx=20, pady=30)
 root.mainloop()
 output.



Q8

```
from tkinter import *
root = Tk()
l = Label(root, text = 'choose NO', bg = 'red')
l.pack()
b1 = Spinbox(root, from_ = 0, to = 10, font = 'x')
b1.pack(side = TOP)
root.mainloop()
```

Output:-



7) AIM: Spinbox To use the SpinBox.

Step 1: Create an object from the method & subsequently create an object from the Spinbox().

Step 2: Make the obj. be created onto the parent window & triggers the corresponding event.

8) Aim : To use Paned window.

Step1: Create an obj from the paned window & use the pack() with the attribute fill & expand.

Step2: Create an obj from the label method & put it onto the paned window with the font attribute & use the add method to insert new obj.

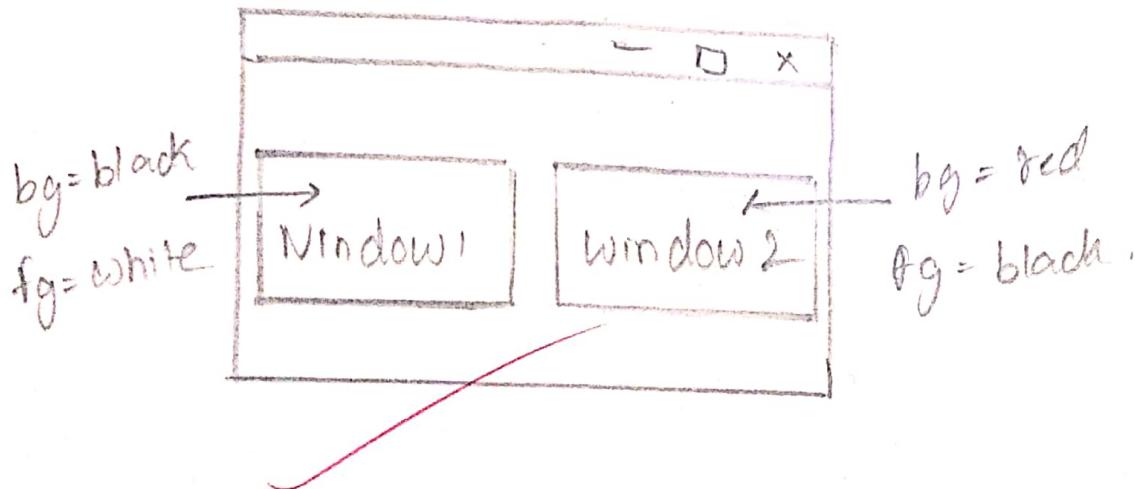
Step3: Similarly create a 2nd paned window obj & add it onto the 1st paned window with orientation specified.

Step4: Now create another level obj & place it onto the 2nd paned window obj & add it onto the 2nd paned.

Step5: Trigger the mainloop()

from tkinter import*
 root = Tk()
 54
 p = PanedWindow()
 p.pack(fill=BOTH, expand=1)
 l = Label(p, text='Window1', bg='black', fg='white')
 p.add(l)
 p1 = PanedWindow(p, orient=VERTICAL)
 p.add(p1)
 l1 = Label(p1, text='Window 2', bg='red', fg='black')
 p2 = PanedWindow(p, orient=HORIZONTAL)
 p.add(p2)
 root.mainloop()

Output :



from Tkinter import *

root = Tk()

cl = Canvas(root, height = '400', width = '400', bg = 'black')

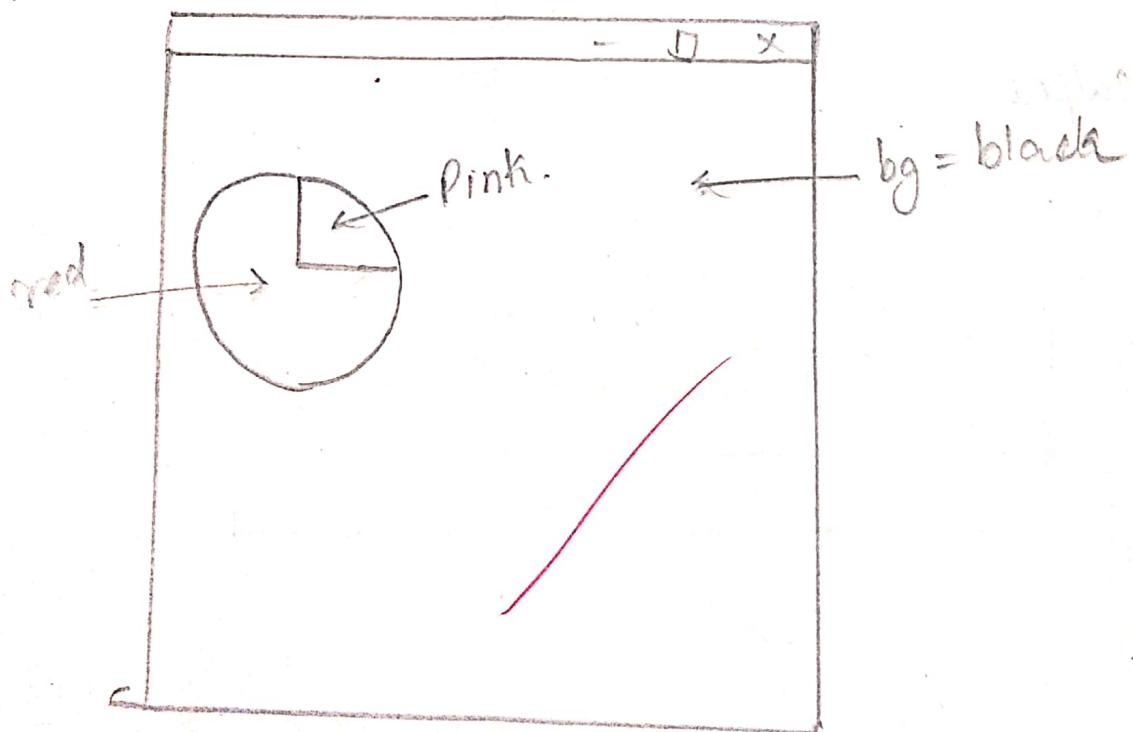
line = cl.create_line(10, 90, 70, 170, 205, fill = 'black')

oval = cl.create_oval(10, 90, 70, 170, 205, fill = 'red')

arc = cl.create_arc(10, 90, 170, 205, fill = 'pink')

cl.pack()

Output.



g) Aim: To use Canvas method.

Step 1: Create an obj from the canvas method & use the attribute height, width, bgcolor & the parent window of object.

Step 2: Use the method create line, creat oval etc Create one along with the canvas object so create & use the co-ordinate values.

Step 3: Similarly use the other methods & code pack() & mainloop().

Jan 12/02

Q3

Aim: To display Image using frame, grid, button.

Step 1: Create an obj corresponding to the parent window & use the following 3 methods.

1 TITLE

2 MINSIZE

3 Config

Step 2: Create a left frame obj 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, attributes specified.

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

Step 4: Create a label obj 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 as (0,0) some external padding values.

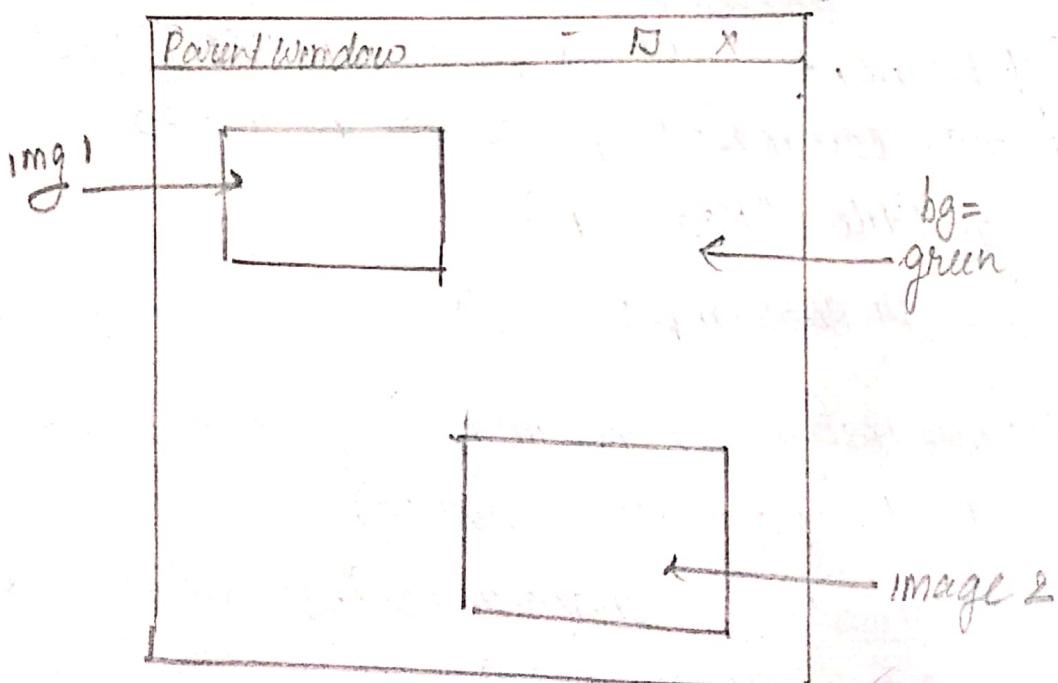
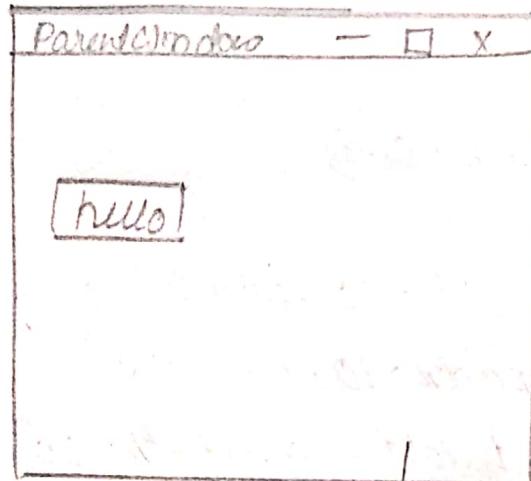
Step 5: Now use the photo image method with the file attribute specified.

```

from tkinter import *
root = Tk()
root.geometry("450x500")
root.title("Parent Window")
root.minsize(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=0, (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="koala.gif")
originalimage = p1.subsample(4, 5)
label(leftframe, image=originalimage).grid(row=0,
                                             column=1, padx=10, pady=30)
label(rightframe, image=originalimage).grid(row=2,
                                             column=2, padx=20, pady=30)
root.mainloop()

```

Output:



Step 6: Use the sub sample method with the object of the image & if the x,y coordinate values

Step 7: Use the 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 level abj positioning it onto right frame & specifying the image & the background attribute with the row & column attribute specified as (0,0).

Step 9: Now create a toolbar abj 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 Jntn for the different toolbar options provided in the left frame.

~~Step 11: From the label (l) position text onto the toolbar use the 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

In
a

88

```
import dbm
db = dbm.open ("database", flag = "c")
if db ["WWW"] != None:
    print ("found")
else:
    print ("not found")
```

Output :

Found.



Aim: To learn the use of database.

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

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

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

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

Step 2: How Create connection objects using `Sqlite` library and connecting method for Create the new database?

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

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

Step 5: Now with the Cursor objects use insert statements for entering the values co-ordinating into the different field considering the datatypes.

Step 6: Use the commit method to complete the operation use the connect object

import os, sqlite3

67

connection = sqlite3.connect('softwareteam
student.db')

c = connection.cursor()

c.execute('create table student (Name char,
RollNo int, Dept char)')

c.execute('insert into student values ('Sachin',
1456, 'CS')') values

c.execute('insert into student values ('Hareesh', 1478,
, 'CS')')

c.execute('insert into student values ('Sanjana',
1879, 'IT')')

c.execute('insert into student values ('Dilip',
1506, 'DataScience')')

6.

connection.commit()

c.execute('select * from student')

c.fetchall()

c.execute('drop table student')

13
Output:

[('Sachin', 1456, 'CS'), ('Haresh', 1478, 'CS'),
('Sanjana', 1879, 'IT'), ('Dilip', 1506, 'DataScience')]

Step 8: use the execute statement along with the cursor objects for accessing the value in the data base using selecting from when clause.

Step 9: finally use the fetchall method for displaying the value for the table using the cursor objects.

Step 10: use the execute method & the drop table syntax for terminating the database
finally use close method.

18

GUI-project

Temperature conversion

Step 1: Import the relevant method from tkinter library.

Step 2: Define a function which will take a float input from user using entry widget and use expression which will give result in celsius degree. And use label() to display result in label form.

Step 3: Similarly again define a function which will convert in fahrenheit. And use label() to display result in label form.

Step 4: Define a parent window frame object on parent window

Step 5: Use label method to display the title 'Temperature conversion' on framee.

Step 6: Use entry widget. to enter the number degree to convert into fahrenheit or in celsius.

Step 7: Use a Radio button 'celsius' to with command which will convert fahrenheit to 'celsius'

```
from tkinter import*
def celsius():
    f=float(num1.get())
    c=(f-32)*5/9
    label=Label(root,text=".2f celsius"%c,fg="dark blue").pack(side=BOTTOM,ipadx=15,ipady=20)
    return
def fahrenheit():
    c=float(num1.get())
    f=(c*9/5)+32
    label=Label(root,text=".2f fahrenheit"%f,fg="Brown").pack(side=BOTTOM,ipadx=15,ipady=20)
    return
def kelvin():
    c=float(num1.get())
    k=(c+273.15)
    label=Label(root,text=".2f kelvin"%k,fg="Purple").pack(side=BOTTOM,ipadx=15,ipady=20)

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

root=Tk()
root.geometry('450x450+400+200')
frame=Frame(root).pack()
num1=StringVar()
radbtn=StringVar()
radbtn.set(None)
frame1=Frame(root,bg="light blue")
```

```
frame1.pack(side=TOP)
label2=Label(frame1,text="Temperature for Conversion",bg="grey",fg="white",font=22,relief=RIDGE)
label2.pack(side=TOP)
label2=Label(frame1,text="\n\n")
label2.pack(side=TOP)
msg=Message(root,text="Note : This application will show the conversion of temperature from Fahrenheit to Celsius and from Celsius to Fahrenheit.",fg='black')
msg.pack(side=BOTTOM)
txtdisplay=Entry(frame1,textvariable=num1,bd=30,insertwidth=1,font=14,justify='center')

txtdisplay.pack(side=TOP)
radio1=Radiobutton(frame1,text="Celsius",variable=radbtn,value="celsius",command=celsius).pack(side=BOTTOM)
radio2=Radiobutton(frame1,text="fahrenheit",variable=radbtn,value="fahrenheit",command=fahrenheit).pack(side=BOTTOM)
radio3=Radiobutton(frame1,text="Kelvin",variable=radbtn,value="kelvin",command=kelvin).pack(side=BOTTOM)
button1=Button(frame1,padx=16,pady=8,bd=8,text="Clear",bg="grey",fg='white',font=48,command=btnClearDisplay).pack(side=BOTTOM)
root.mainloop()
```

Step 8: Use another RadioButton 'fahrenheit' which will convert celsius to fahrenheit.

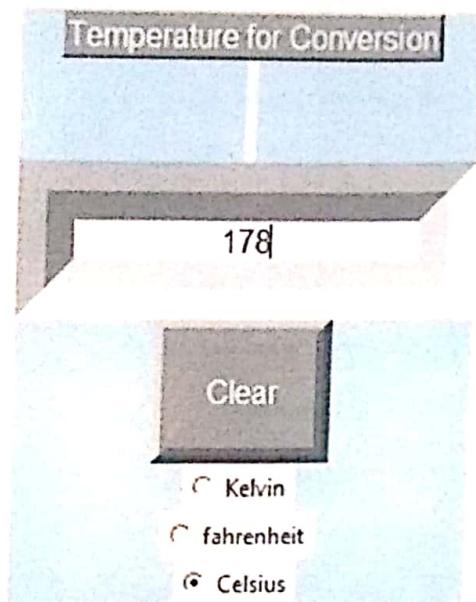
Step 9: Define a block similarly as step 2 for Kelvin degree conversion.

Step 10: Use a RadioButton 'kelvin' which will convert celsius to kelvin.

Step

Step 11: user Define a function for clear the user input. Use button method for declaring a clear.

Step 12: Define a messagebox, it will display the message.



81.11 celsius

352.40 fahrenheit

451.15 kelvin

Note : This application
will show the conversion
of temperature from
Fahrenheit to celsius
and from celsius to
fahrenheit.

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

```
>>> import os,sqlite3
>>> connection=sqlite3.connect("population.db")
>>> c=connection.cursor()
>>> c.execute("create table Andhrapradesh(District char,Population int,Area_kmsq int)")
>>> c.execute("insert into Andhrapradesh values('Anantapur',4083315,19130)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Andhrapradesh values('Chittoor',4170468,15152)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Andhrapradesh values('EastGodavari',5151549,10807)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Andhrapradesh values('Guntur',4889230,11391)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Andhrapradesh values('YSR Kadapa',2884524,15351)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Andhrapradesh values('Krishna',4529009,8727)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Andhrapradesh values('Kurnool',4046601,17658)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Andhrapradesh values('Nellore',296608,13076)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Andhrapradesh values('Prakasam',3392764,17626)")
```

Database - project

65

Database of population on each district of two state.

Step1: Import os, sqlite3

Step2: Create connection object using sqlite3 library and connecting method for create population database.

Step3: Create Cursor objects using cursor method from connection object

Step4: Use the execute method for creating the Andhraapradesh table and similarly create table for kerala state.

Step5: Now with Cursor object use insert statement for enter values of like District, population and area(km²). for Andhraapradesh

Step6: Similarly insert values for kerala

Step7: Use the Commit method to complete the operation use the connect object.

Step8: Use create and fetchall method for selecting and displaying the created database.

20

Step 10: Use Drop statement to terminating
the database.

```
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Srikakulam',2699471,5837)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Visakhapatnam',4288113,11161)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Vizianagaram',2342868,6539)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('WestGodavari',3934782,7742)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("create table Kerala(District char,Population int,Area_kmsq int)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Malappuram',4110956,3550)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Thiruvananthapuram',3307284,2192)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Ernakulam',3279860,3086)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Thrissur',3110327,3032)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Kozhikode',3089543,2344)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Palakkad',2810892,4480)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Kollam',2629702,2491)")
```

```
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Kannur',2525637,2966)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Alappuzha',2121943,1414)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Kottayam',1979384,2208)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Kasaragod',1302600,1992)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Pathanamithitta',1195537,2637)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Idukki',1107453,4358)")

<sqlite3.Cursor object at 0x000001985DF0A340. -
>>> c.execute("insert into Kerala values('Wayanad',816558,2131)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> connection.commit()

>>> c.execute('select * from AndhraPradesh')

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()
[('Anantapur', 4083315, 19130), ('Chittoor', 4170468, 15152), ('EastGodavari', 5151549, 10807), ('Guntur', 4889230, 11391), ('YSR Kadapa', 2884524, 15351), ('Krishna', 4529009, 8727), ('Kurnool', 4046601, 17658), ('Nellore', 296608, 13076), ('Prakasam', 3392764, 17626), ('Srikakulam', 2699471, 5837), ('Visakhapatnam', 4288113, 11161), ('Vizianagaram', 2342868, 6539), ('WestGodavari', 3934782, 7742)]

>>> c.execute('select * from Kerala')

<sqlite3.Cursor object at 0x000001985DF0A340>
```

```
>>> c.fetchall()
[('Malappuram', 4110956, 3550), ('Thiruvananthapuram', 3307284, 2192), ('Ernakulam', 3279860, 3086),
('Thrissur', 3110327, 3032), ('Kozhikode', 3089543, 2344), ('Palakkad', 2810892, 4480), ('Kollam', 2629702,
2491), ('Kannur', 2525637, 2966), ('Alappuzha', 2121943, 1414), ('Kottayam', 1979384, 2208), ('Kasaragod',
1302600, 1992), ('Pathanamithitta', 1195537, 2637), ('Idukki', 1107453, 4358), ('Wayanad', 816558, 2131)]
>>> c.execute('select District,Population from AndhraPradesh union select District,Population from Kerala')
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()
[('Alappuzha', 2121943), ('Anantapur', 4083315), ('Chittoor', 4170468), ('EastGodavari', 5151549), ('Ernakula
3279860), ('Guntur', 4889230), ('Idukki', 1107453), ('Kannur', 2525637), ('Kasaragod', 1302600), ('Kollam',
2629702), ('Kottayam', 1979384), ('Kozhikode', 3089543), ('Krishna', 4529009), ('Kurnool', 4046601),
('Malappuram', 4110956), ('Nellore', 296608), ('Palakkad', 2810892), ('Pathanamithitta', 1195537), ('Prakasai
3392764), ('Srikakulam', 2699471), ('Thiruvananthapuram', 3307284), ('Thrissur', 3110327), ('Visakhapatnam
4288113), ('Vizinagaram', 2342868), ('Wayanad', 816558), ('WestGodavari', 3934782), ('YSR Kadapa', 288452
>>> c.execute('select count(District) from AndhraPradesh')
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()
[(13,)]
>>> c.execute('select count(District) from Kerala')
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()
[(14,)]
>>> c.execute('select sum(Population) from AndhraPradesh')
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()
[(46709302,)]
>>> c.execute('select sum(Population) from Kerala')
```

```
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()
[(33387676,)]
>>> c.execute("drop table Andhrapradesh")
<sqlite3.Cursor object at 0x00000269D76DB5E0>
>>> c.execute("drop table Kerala")
<sqlite3.Cursor object at 0x00000269D76DB5E0>
>>> c.fetchall()
[]
```

8/V