

QSO

Program:-

```
fileObject = open("abc.txt", "w") # file open(write mode)
fileObject.write("Hi, My Name is Aayushi \n I am FYBsc CS student
                    \n I love Python!") # file write.
```

fileObject.close() # file closed.

```
fileObject = open("abc.txt", "r") # file opened in read mode.
```

str1 = fileObject.read() # file is read.
print ("The output of read method is ", str1) # output is printed

fileObject.close()

Output:-

```
>>> ("The output of read method is hi, My Name is Aayushi \n I am
          I am an FYBsc CS student
          I love Python")
```

readline

```
fileObject = open("abc.txt", "r")
```

fileObject.readline()

str2 = fileObject.readline() # output of readline method is str2

print ("The output of readline method is hi, my name is

>>> The output of the readline method is hi, my name is

Aayushi

readlines()

```
fileObject = open("abc.txt", "r")
```

str3 = fileObject.readlines()

print ("The output of readlines method is ", str3)

>>> The output of the readlines method is [hi; my name is Aayushi
 \n; I am FYBsc CS student \n, 'I love Python\n']

Practical No:- I

Demonstrate the use of different file access mode, different attribute of file object and differentiate among the different read methods.

Algorithm :-

Step I :- Create the file object by using the open method and use the right access mode followed by writing some contents onto the file and then closing the file.

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

180

Step IV :- Now open the fileobject in write mode write some another content close subsequently then again open the fileobject in 'wt' mode that is the update mode and write contents.

Step V :- Open fileobject in read mode, display the update contents and close open again in 'rt' mode with parameters passed and display the output subsequently.

Step VI :- Now open file object in append mode open write method write content close the fileobject again open the fileobject in read mode and display the append output

#file attributes

```

o = fileobject.name
print ("name of file (name attribute):", o)
>>> ('name of file (name attribute)', 'abc.txt')

# fileobject.closed
print ("(close) attribute:", b)
>>> ('(close) attribute.', 'True')

c = fileobject.mode
print ("file mode", c)
>>> 'file mode', ('r')

d = fileobject.softspace
print ("softspace", d)
>>> ('softspace:', 0)

```

wt mode.

```

fileobject = open ("abc.txt", "wt")
fileobject.write ("Aayushi")
fileobject.close()

```

write mode

```

fileobject = open ("abc.txt", "w")
fileobject.write ("DBMS")
fileobject.close()

```

read mode.

```

# r+ mode.
fileobject = open ("abc.txt", "r+")
str1 = fileobject.read (6)
print ("Output of r+", str1)
fileobj.close()
>>> ('Output of r+', 'Aayushi')

```

```

# read mode.
fileobject = open ("abc.txt", "r")
str2 = fileobject.read ()
print ("Output of read mode!")
str2
>>> ('Output of read mode!', 'Aayushi')

```

SSD:

```
# append mode
fileobject = open("abc.txt", "a")
fileobject.write("data structure")
fileobject.close()
fileobject = open("abc.txt", "a")
fileobject.read()
str13 = fileobject.read()
print("Output of append mode:", str13)
fileobject.close()
>>> ('Output of append mode:', 'ayush: data structure')

# tell()
fileobject = open("abc.txt", "r")
pos = fileobject.tell()
print("tell():", pos)
fileobject.close()
>>> ('tell():', 0)

# seek()
fileobject = open("abc.txt", "r")
str14 = fileobject.seek(0, 0)
str18 = fileobject.read(10)
print("The beginning of the file:", str18)

# finding length of different lines exist within lines
fileobject = open("abc.txt", "r")
str19 = fileobject.readlines()
print("Output:", str19)
for line in str19:
    print(len(line))

    fileobj.close()
>>> ('Output:', ['college databases'])
```

Step
a v
and s

Step
fileobj

Step
the
in
use

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

Step VIII :- Use the seek method with opening the fileobject in read mode and closing subsequently.

Step IX :- Open fileobject with read mode also use the readlines method and store the output consequently in and print the same for counting the length use the for condition statement and display the length.

Jr allu9

PRACTICAL Xo. 2.

Aim:- Demonstrate the use of `iter()` and `next()` methods along with a class to display the odd numbers upto 100.

Algorithm:-

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

Step 2:- for extracting the next element from the container with an argument and compare the number of elements required in a container by using the conditional statement.

Step 3:- Now create an object from the given class and pass this object as argument to the `iter()` method.

Step 4:- Now using the conditional statement display all the values from the given container.

Q50

Output :-
② my_tuple1 = ("Banana", "Orange", "Apple")
my_iter1 = iter(my_tuple1)
for i in my_iter1:
print(i)

>>> Banana
Orange
Apple

③ class MyClass:
def __iter__(self):
self.num = 1
return self
def __next__(self):
if self.num <= 10:
num = self.num
self.num += 1
return num
else:
raise StopIteration

obj = MyClass()
my_iter = iter(obj)
for i in my_iter:
print(i)

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

Simple use of iter() method.

Create a tuple object and assign it the value or elements. Then create a Iterable object to iter through the tuple with iter() method.

Now use for condition of statement to display all the elements in the iterable object.

Program using the iterable object to display set of 10 numbers.

Step 1 :- Define iter() with an argument and initialize it to the first value.

Step 2 :- for extracting the next element from the container, use the next() with an argument and compare the no. of elements required in a container by using the conditional statement.

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

Step 4 :- Now using the conditional statement display all the values from the given container.

④ Program to display the elements of list as even or odd.

Step 1:- Define a function named as evenOdd with an argument which would be integer.

Step 2:- Define the logic for determining the numbers as even or odd and return even or odd appropriately.

Step 3:- Declare a list variable and define all the numbers that you want in the list.

Step 4:- Use map() method with the above function and list variable arguments.

Step 5:- Print the result.

⑤ Program for printing the square and cube of given set of numbers using map().

Step 1 :- Define a square function with an argument and return the square of the no.

Step 2 :- Define a function cube with an argument and return the cube of the number.

Step 3 :- Declare a list variable and call function square and cube in the list.

Step 4:- Use for conditional statement and use the map() to find square and cube.

a odd
with
1 the
mont
and
ion
e

Output..

```
(4) def evenodd(n):
    if n % 2 == 0:
        return "Even"
    else:
        return "Odd"

list_num = [0, 4, 5, 7, 9, 11, 13, 5, 20, 9, 25]
list_1 = list(map(evenodd, list_num))
print(list_1)

>>> ['Even', 'Even', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd']
```

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

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

func_1 = [square, cube]

for x in range(5):
    value_out = list(map(lambda n: n(n), func_1))
    print(value_out)
```

```
>>> [1, 1]
[4, 8]
[9, 27]
[16, 64]
```

Jun 11/2019

QSO

① Program

```
try:  
    fo = open ("abc.txt", "r")  
    fo.write ("hi, this is aayushi")  
  
except IOError:  
    print ("Error occurred under Environment error")  
  
else:  
    print ("Operation successful")
```

Output

Environmental Error

②

Practical NO 1 - 3

027

exception

① Algorithm

Step 1 :- Use the try block to define the normal course of action : For eg : Define the file obj and open the file in the write mode & write some content onto the file.

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

780

Exception handling

② Program for demonstrating the use of value error in the given program statement.

Step 1 : Accept the value from the user and if it is a valid value display the entire value and terminate the condition by using the break statement

Step 2 : Define the except block with the value errors as a keyword & display the appropriate message.

Step 3 : You can define the multiple exception using the except statement for finding the different category of errors

Program :-

```
try:  
    a = int(input("Enter the number"))  
except ValueError:  
    print("Arithmatic error")  
else:  
    print("Successful")  
  
try:  
    fo = open("abct.txt", "r")  
    fo.write("Hello, how are you??")  
except IOError:  
    print("Environmental error")  
else:  
    print("Successful")
```

Output:-

Enter the number: 14
Arithmatic error

850

match()

import re

pattern = r"FYCS" represents computer science

sequence = "FYCS Sequence":

if re.match(pattern, sequence):

 print("Match found")

else:

 print("Not found!")

Output :-

Match Found.

numerical values (Segregation)

import re

pattern = r"\d+"

sequence = "Neha14, 30 days, 129 don"

Output = re.findall(pattern, sequence)

Print (Output)

Output:-

[14, 30, 129]

PRACTICAL No:- 4

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 findall() with arguments and print the same.

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

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

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

Step 6: Import re module declare list with numbers. Use the conditional statement Here we have used if the for conditional for checking first number is either 8 or 9

PRACTICAL NO:- 4

029

Regular Expression

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

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

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

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

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

Step 6:- Import re module declare list with numbers. use the conditional statement here we have used if the for conditional for checking first number is either 8 or

and next number are in range of 0 to 9 and check whether the entered number are equal to 10 if matches print no is correct otherwise Print Invalid number

Step 4: import re module declare a host and domain name declare pattern for separating the host and domain name. Use the.findall() and print the output respectively.

Step 8: import re module declare a string. Use the module with.findall() the vowels in the string and declare the same.

Step 9: Import re module .enter a string use the pattern to display only two elements 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.

#split()

```
import re
pattern = r"\D+"
string = "Neha14, 90days, 129don"
output = re.split(pattern, string)
print(output)
```

Output:

['Neha', '14', '90days', '129don']

no space:

```
import re
pattern = "\S+"
sequence = "abc def ghi"
replace = ''
op = re.sub(pattern, replace, sequence)
print(op)
```

Output:

abcdefghijklm

Q80

Group()

```
import re
sequence = "Python is an interesting lang"
j = re.search(" \A Python", sequence)
Print(j)
OP = j.group(1)
Print(OP)
```

Output:-

<_sre.SRE_Match object at 0x02E1D960>
Python.

Verifying the Given set of Phone. No. :-

```
import re
list1 = ['9819492767', '7021181118']
for values in list1:
    if re.match(r'[8-9]\d{9}|[0-9]\d{9};  
            len(values) == 10):
        Print("No. is correct")
    else:
        Print("Invalid Number")
```

Output

No is correct
No is correct

vowels:

```
import re
str = "My name is Payushi"
Pattern = r"\b[aeiouAEIOU]\w+"
op = re.findall(Pattern, str)
print(op)
```

Output:

['is']

host & domain:

```
import re
seq = "ayushis3011@gmail.com"
Pattern = r"\b[\w]+\.\w+\b"
op = re.findall(Pattern, seq)
print(op)
```

Output:

['ayushis', 'gmail.com']

Q80

Counting of first 2 letters:

import re

S = "mr.a, ms.b, ms.c, mr.z"

pattern = r'(\ms{1})\ms{1}(\.)'

OP = re.findall (Pattern, S)

Print (OP)

m = 0

f = 0

for v in D:

if (v == 'ms'):

f = f + 1

else :

m = m + 1

Print ("No. of Males:", m)

Print ("No. of Females:", f)

Output:

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

('No. of Males : 2')

('No. of Females : 2')

✓ JMK

PRACTICAL No - 51

GUI

① Algorithm: (parent window) (Label widget)

1. Import the tkinter module for using the various features of the widget.

2. Generate a parent window using TK() method.

3. Add a label widget onto the parent window as much as you want with the help of the class Label.

4. Use the mainloop method to launch the window and start the event loop.

② Algorithm (Radio Button)

1. Import the tkinter module for using the various features of the widget.

2. Generate a parent window using TK() method.

3. Add a ~~Label~~^{radiobutton} widget onto the parent window as much as the user want. This will create a radiobutton along with the text written in the radiobutton widget.

4. Use the mainloop method to launch the window and to start the event loop.

Label widget.

```
from tkinter import *
root = Tk()
w = Label(root, text="Hello world!!", bg="yellow", fg="red")
w.pack(pady=30)
w1 = Label(root, text="welcome", bg="red", fg="yellow")
w1.pack(padx=20)
root.mainloop()
```

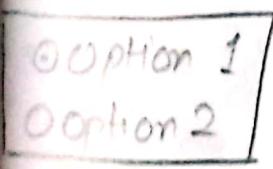
Output:



Radio Button

```
from tkinter import *
root = Tk()
r = Radiobutton(root, text="Option 1", value=1)
r.pack()
r1 = Radiobutton(root, text="Option 2", value=2)
r1.pack()
root.mainloop()
```

Output:

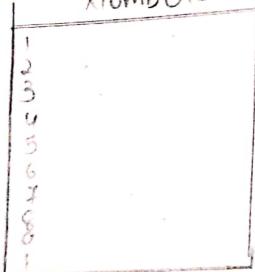


180

Listbox

```
from tkinter import *
root = Tk()
root.geometry("680x500")
l = Label(root, text="Numbers:")
l.pack()
frame = Frame(root)
frame.pack()
listbox = Listbox(frame, font=("Times New Roman", 20))
listbox.pack(side=LEFT, fill=Y)
for n in range(100):
    listbox.insert(END, str(n))
root.mainloop()
```

Output:

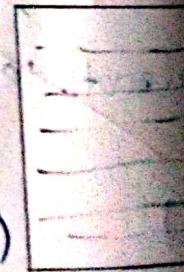


(2)

Scrollbar

```
from tkinter import *
root = Tk()
para = " — — — "
s = Scrollbar(root)
t1 = Text(root, height=20, width=20)
s.pack(side=RIGHT, fill=Y)
t1.pack(side=LEFT, fill=Y)
s.config(command=t1.yview)
t1.config(yscrollcommand=s.set)
t1.insert(END, para)
root.mainloop()
```

Output:



③ Algorithm (List Box).

1. Import the tkinter module, create a parent window.
2. Use the geometry method to arrange the size of the parent window.
3. Use the Label widget according to your choice.
4. Use the frame widget to create a frame in the parent window.
5. Use the Listbox widget with its different attributes such as font, fontsize.
6. Use the for loop to print the number upto 99.
7. Use the mainloop method to launch the window and to start the event loop.

④ Algorithm (Scrollbar).

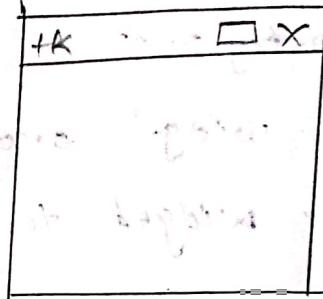
1. Import the relevant method from tkinter library.
2. Create a parent window.
3. Use the pack method to write the content.
4. Create an object corresponding to scrollbar widget and place it on the parent window.
5. Along with the pack method use the argument such as side & fill.
6. Now use the text object along with pack method. Use the side & fill argument.
7. Now use the config method along with the object of the scrollbar and use the command attribute.
8. Similarly use the config method along with text object & use the yscrollcommand argument.
9. Use the insert method with two arguments.
10. Use the mainloop method to launch the window and to start the event loop.

280

frame

```
from tkinter import *
root = Tk()
root.geometry("650x500")
frame = Frame(root)
frame.pack()
leftframe = Frame(root)
frame.pack(side=LEFT)
rightframe = Frame(root)
frame.pack(side=RIGHT)
root.mainloop()
```

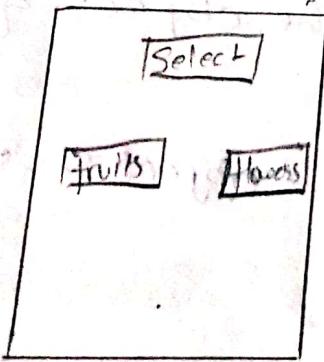
Output



Button

```
from tkinter import *
root = Tk()
root.geometry("650x500")
b1 = Button(root, text="Select")
b1.pack(side=TOP, padx=50, pady=10)
b2 = Button(root, text="flowers")
b2.pack(side=RIGHT, padx=60, pady=0)
b3 = Button(root, text="fruits")
b3.pack(side=LEFT, padx=50, pady=10)
root.mainloop()
```

Output



⑤ Frame: Algorithm

- 1) Import the relevant method from tkinter library.
2. Create an object corresponding to the parent window.
Define the size of the parent window in terms of pixels with the help of geometry method.
3. Use the frame method and create the leftframe rightframe object with text attribute.
4. Pack the frame on the Parent window.
5. Finally use the mainloop method to launch the window and to start the event loop.

⑥ Button: Algorithm

- 1) Import the relevant methods from tkinter library
2. Create the Parent window, Adjust the size of the parent window with the help of geometry method.
3. Use the button widget along with the different attributes such as ~~size~~ font , font size.
4. Pack the buttons on the parent window along with the pack method ~~&~~ use different widget such as side , padm , pady .
5. Use the mainloop method ~~&~~ to launch the window and ~~to~~ start the event loops.

Drill

280

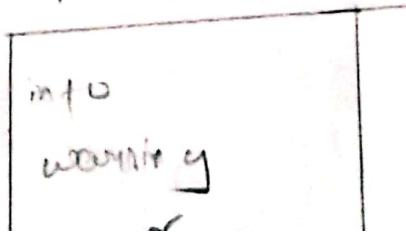
Message Box (Algorithm)

1. Import relevant method from tkinter library
2. Create a parent window
3. Define a functions according to your need.
4. Use the message box widget along with it
5. Create a object corresponding to button widget as much as you have defined function
6. Pack the button widget
7. If the user want to wide the parent window only the information window should be visible corresponding to the options given above the withdraw method is used.
8. Use the mainloop method to launch the window and to trigger the event.

```

Message box:
from tkinter import *
import messagebox
root=TK()
def info():
    messagebox.showinfo ("Aayushi", "FYBSc (S)")
def warning():
    messagebox.showwarning ("Danger")
def error():
    messagebox.showerror ("wrong")
def askyesno():
    messagebox.askyesno("Do you study Python")
b1=Button(root, text="info", command=info)
b1.pack()
b2=Button(root, text="warning", command=warning)
b2.pack()
b3=Button(root, text="error", command=error)
b3.pack()
b4=Button(root, text="Yes or No", command=askyesno)
b4.pack()
root.mainloop()

```



Traversing.

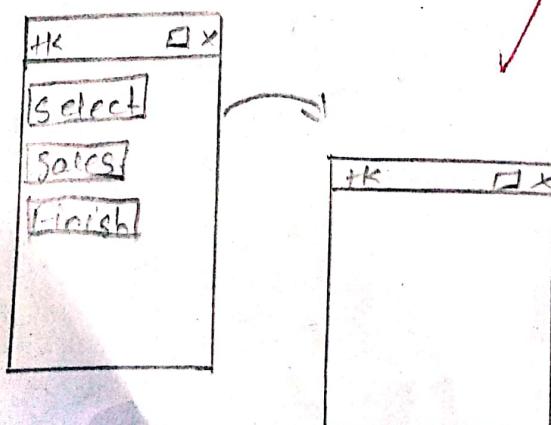
```
from tkinter import *
root = Tk()
def main():
    root = Tk()
    root.title()
    root.config()
def sales():
    root = Tk()
    root.config()
    root.title()
def finish():
    quit()
```

B1 = Button(root, text="Select", command=main)
B1.pack()

B2 = Button(root, text="Sales", command=sales)
B2.pack()

B3 = Button(root, text="Finish")

B3.pack()
root.mainloop()



⑧ Algorithms (Fraversing)

1. Import the relevant method from the Tkinter library
2. Create a parent window
3. Define the different functions by using three Methods namely config, title & minsize.
4. Now create the button widget along with the command attribute that is the event should be triggered.
5. Create as much button widget as there are number of functions defined.
6. Use the mainloop method to launch the window and to start the event loop.

~~Final~~

PRACTICAL NO-8

Algorithm (Image)

(practical 8)

1. Use the Tkinter library.
2. Create a parent window & use the Geometry widget.
3. Create a frame object by using the frame method and place it onto the parent window.
4. Now create the leftframe & rightframe object by using the frame method and use the side attribute within the pack method and subsequently use the padx, pady, ipadx & ipady attributes.
5. Create a object by using the label method and place it onto the parent window.
6. Use the photoimage method to place the photo & subsequently use the subsample method give the coordinates. And use the label method to place the original in the leftframe.
- Finally use the mainloop method for triggering the events.

image.

```

from tkinter import *
root = Tk()
root.geometry("380x480")
frame = Frame(root)
frame.pack()
leftframe = Frame(frame, bg = "blue", height = "20",
                  width = "20")
leftframe.pack(side = LEFT, padx = 40, pady = 40)
rightframe = Frame(frame, bg = "red", height = "40",
                    width = "20")
rightframe.pack(side = LEFT, padx = 20, pady = 20)
l1 = Label(leftframe, text = "Image", relief = RAISED)
l1.pack()
image = PhotoImage(file = "Penguin.gif")
Originalimage = image.subsample(3, 5)
label(leftframe, image = Originalimage).pack()
root.mainloop()

```

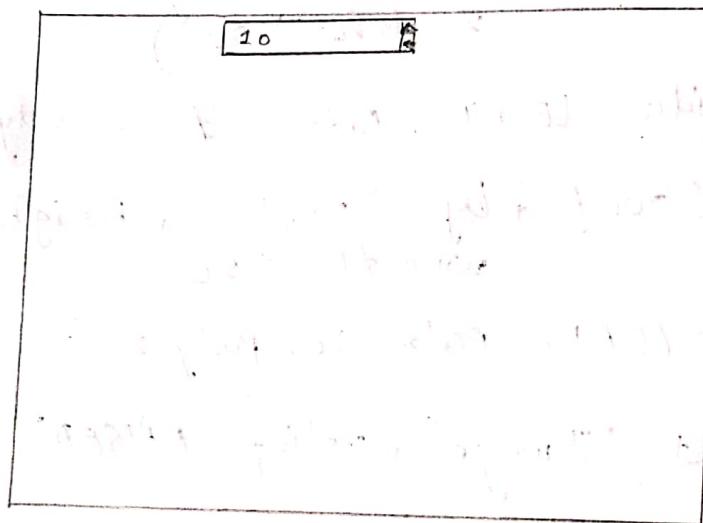
Dr 1012
Z

Q8

Spinbox

```
from tkinter import*
master=Tk()
master.geometry("350x350")
S1=Spinbox(master, from_=1,to=10)
S1.pack()
mainloop()
```

Output :-



PanedWindow

```
from tkinter import*
root=Tk()
P=PanedWindow(root)
P.pack(fill=BOTH, expand=1)
L=Label(P, text="Hello World")
P.add(L)
P1=PanedWindow(root, orient=VERTICAL)
P.add(P1)
L1=Label(P1, text="Aayushi")
P1.add(L1)
B1=Button(P1, text="Female")
P1.add(B1)
root.mainloop()
```

Spinbox

Algorithm

- 1) Import the tkinter module
- 2) Create an object from the tk method and subsequently create an object from spinbox()
- 3) Make the object of created on to the parent window and trigger the corresponding event

PanedWindow

Algorithm

- 1) Import the tkinter module , create a parent window
- 2) Create an object from panedwindow() and use the pack method with attribute fill and expand .
- 3) Create an object method from the label method and put it on to the panedwindow with the text attribute and use the add method to imbed the new object .
- 4) Similarly create the second paned window object and add it on the paned window with the orientation

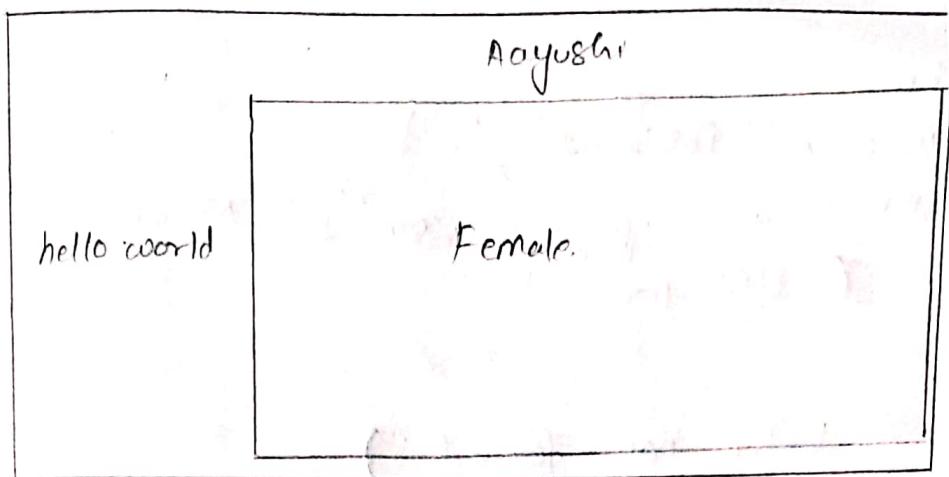
Q8

- 5) Now create another label object and place it onto the second panedwindow and
- 6) Use the mainloop method to trigger the event

Canvas:

Algorithm:

- 1) Import the tkinter module.
- 2) Create a parent window.
- 3) Create an object from the canvas method and use the attributes height, width, bg colour and parent window object).
- 4) Use the method to create line, create oval or create arc along with the canvas object so created and use the co-ordinate values.
- 5) Similarly use the other method to create other shapes and call the pack method.
- 6) Use the mainloop method to trigger the event.



canvas

```
from tkinter import *
```

```
root = Tk()
```

```
c = Canvas(root, bg="blue", height=500, width=500)
```

```
cord = 60, 50, 240, 210
```

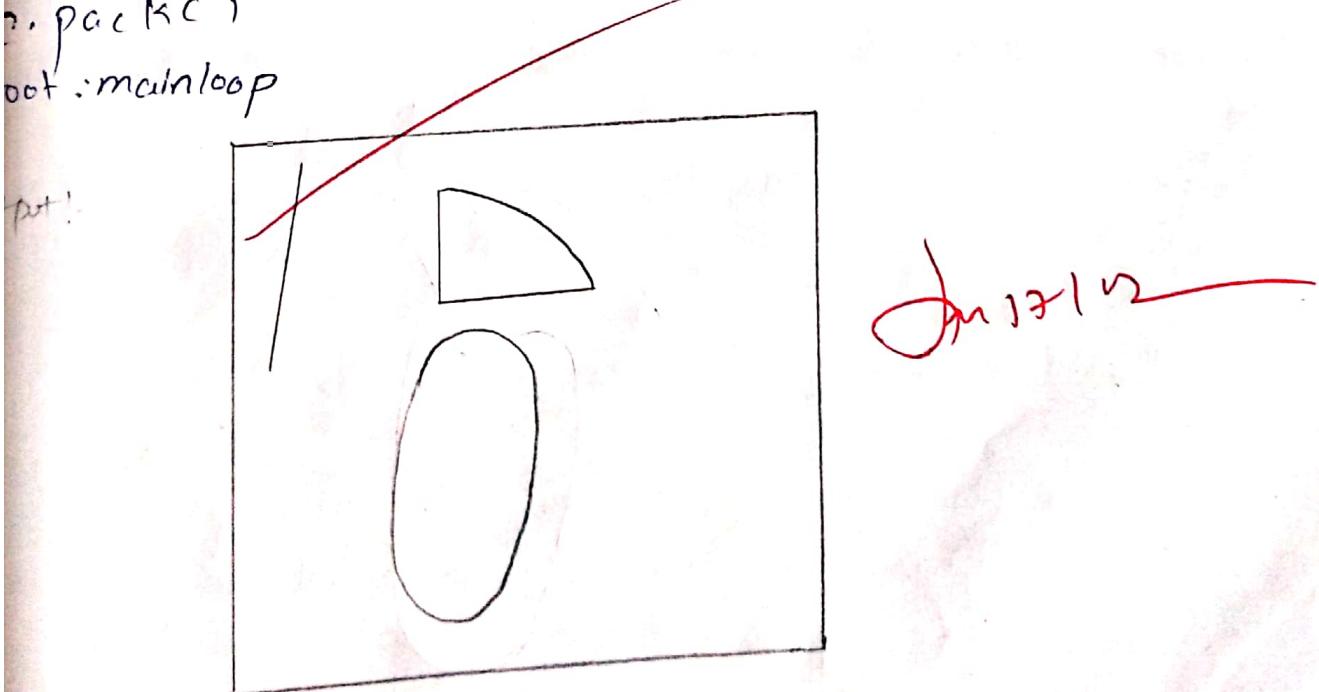
```
arc = c.create_arc(cord, start=0, extent=150, fill="red")
```

```
ne = c.create_line(20, 10, 10, 90, fill="red")
```

```
val = c.create_oval(60, 90, 210, 410, fill="red")
```

```
? .pack()
```

```
root.mainloop()
```



180
Program:-

```
import dbm
db = dbm.open("Database1", "c")
if db["https://www.googledrive.com"] != None:
    print ("URL found")
else:
    print ("URL not found")
db.close()
```

Output :-

URL found

PRACTICAL No - 6

042

Aim:- Demonstrate the use of database connectivity.

Algorithm.

1. Import the dbm library and use the open() method for creating the database by specifying the name of the database along with the corresponding flag.
2. The flag attribute such as c/in and w. Use the object so created for accessing the given website and the corresponding regular name for the website.
3. Check whether the given url address with the regular name of the page is not equal to the null then display the message that the particular url address is found else not found.
4. Finally use the close method to terminate the database library

SD

b) Creating the table and inserting value.

Algorithm:

1. Import the corresponding libraries for making the database which are OS, sqlite 3

2. Now create the connection object by sqlite 3 library & the connect method for creating the new database.

3. Now create the cursor object by using the method from the connection object created in the earlier step.

4. Now use the execute method for creating table with the column name and the respective datatype.

5. Now with the cursor object use the insert for entering the values corresponding to different fields, considering the Datatype.

6. Use the commit method to complete the transaction using the connection object.

7. Use the execute along with the cursor object for accessing the values from the database using the select from where clause.

code:-

```
import Connection  
cursor1 =  
cursor1 =
```

```
cursor1 =
```

```
cursor1 =
```

```
connection
```

```
cursor1 =
```

```
cursor1 =
```

```
b. close
```

```
output
```

```
'Ayu
```

```
ite 3.
```

```
ite 3
```

```
ite 3
```

code:-

```

import os, sqlite3
connection = sqlite3.connect("employee.db")
cursor1 = connection.cursor()
cursor1.execute('create table dcs(NAME char,
CLASS char, RNO INTEGER, PERCENT FLOAT)')
cursor1.execute('insert into dcs values("Aayushi Singh",
"FYBScs", 1759, 80.7)')
cursor1.execute('insert into dcs values("Neha Singh", 1780,
"FYBScs", 1780, 87.9)')
connection.commit()
cursor1.execute('Select Name from dcs')
cursor1.fetchall()
db.close()

```

Output :-

[('Aayushi Singh') ('Neha Singh')]

sqlite3.cursor object at 0x02560520
 sqlite3.cursor object at 0x02E60520
 sqlite3.cursor object at 0x02E60520 }

Jun 12 -

8. Finally use the fetch() for displaying the values from the table using the cursor object
9. Use the execute method and the Drop table syntax for terminating the database & finally use the close method.

100

GUI Project

Database code:

```

def Reg():
    import os
    connection = sqlite3.connect ("People.db")
    cursor1 = connection.cursor()
    cursor1.execute ('create table dcs (First Name CHAR,
    Last Name CHAR, GENDER varchar, Date of Birth varchar,
    Subjects you want to study varchar)')
    cursor1.execute ('insert into dcs values')

```

045

Code! —

```

from tkinter import *
root = Tk()
root.geometry("850x750")
l1 = Label (root, text = "Hello welcome !!", bg = "Yellow", fg = "Red")
l2 = Label (root, text = "First name").grid (row = 0, sticky = W)
e1 = Entry (root)
e1.grid (row = 0, sticky = E)
l3 = Label (root)
l3.grid (row = 1, column = 1)
l4 = Label (root, text = "Gender", bg = "Yellow", fg = "Red").grid (row = 2,
    sticky = W)
radio1 = Radiobutton (root, text = "Male", value = 1).grid (row = 3, sticky = W)
radio2 = Radiobutton (root, text = "Female", value = 2).grid (row = 3, sticky = W)
l5 = Label (root, text = "Date of birth", bg = "Yellow", fg = "Red").grid (row = 4, sticky = W)
l6 = Label (root).grid (row = 5)
spin1 = Spinbox (root, from_ = 1, to = 31).grid (row = 6, column = 0)
label1 = Label (root, text = "Date", bg = "Yellow", fg = "Black").grid (row = 6, column = 0)
spin2 = Spinbox (root, from_ = 1900, to = 2020).grid (row = 6, column = 1)
label2 = Label (root, text = "Year", bg = "Yellow", fg = "Black").grid (row = 6, column = 1)
l7 = Label (root).grid (row = 7)
l8 = Label (root, text = "Subjects you want to study", bg = "Yellow", fg =
    "Red").grid (row = 8)
l9 = Label (root).grid (row = 9)
checkbox1 = Checkbutton (root, text = "Physics", fg = "Black").grid (row = 10)
checkbox2 = Checkbutton (root, text = "Chemistry", fg = "Black").grid (row = 11)
checkbox3 = Checkbutton (root, text = "Maths", fg = "Black").grid (row = 12)
checkbox4 = Checkbutton (root, text = "Biology", fg = "Black").grid (row = 13)
label3 = Label (root).grid (row = 14)
label4 = Label (root, text = "Thankyou").grid (row = 15)

```

140

```
bl=Button(root, text="Submit").grid(row=18, column=2)
w3=Label(root).grid(row=19)
w1=Label(root, text="Thank you", bg="yellow", fg="red").grid(row=20, column=2, sticky=W)
root.mainloop()
```

Output:-

The diagram illustrates a graphical user interface (GUI) window. At the top, a title bar contains the text "Hello welcome!!". Below the title bar, there are two text input fields labeled "First name" and "Last name", both containing the placeholder text "Krenden". Underneath these fields are two radio button options: "Male" and "Female". Further down, there is a text input field labeled "Date of Birth" containing the value "1/1/1961". Below this date field, the labels "Date", "Month", and "Year" are positioned above three separate input boxes. To the left of these input boxes, there is a label "Subjects you want to study". Below this label, four checkbox options are listed: "Physics", "Chemistry", "Maths", and "Bio". A red arrow points from the "Bio" checkbox towards a "Submit" button located at the bottom right of the window. The word "Thank you" is written in cursive at the bottom center of the window.

import random
print ("winning rules of the game are as follows:

from sqflite3 import Error

upper flintier as the

um glob import glob

ef create(obj);

`obj.e.get()`

if $db[C-3] == "db"$,

pass

else :

11

try:
conn = lite.connect(db)

return conn

cept Error as e.

print(e)

114:

Con. close C)

obj::(b::insert (HK::END, db))

`obj[i].lb.insert(th, END, db)`

~~obj' = db.set("n")~~

get

`def __init__(self):`

```
self.win=tk.Tk()
```

Self-label (1)

self. entoyc)

```
self.button()
```

R&D

```
def label(self):
    self.l = tk.Label(self.win, text="Create a db")
    self.l.pack()

def entry(self):
    self.db = tk.StringVar()
    self.e = tk.Entry(self.win, textvariable=self.db)
    self.e.pack()

def button(self):
    self.b = tk.Button(self.win, text="Create DB", command=create_db)
    self.b.pack()

def listbox(self):
    self.lb = tk.Listbox(self.win)
    self.lb.pack()
    self.show_db()

def show_db(self):
    for file in glob("*db"):
        self.lb.insert(tk.END, file)

self.win.mainloop()
```

win=Tk()

050

Output

Create a db [insert the name]

