

# Practical No. 1

## Modes of File Operation.

### 1. Write mode (w)

# To create a file  
a = open ("SUSHANT", "w")

# To enter values in file

a. write ("SUSHANT Parajy" + "\n")  
a. write ("Row no. 167")

# To close the file.  
a. close()

### 2. Read(r) mode:

# To open a file  
a = open ("SUSHANT", "r")

# To read the file.

b = a.read()  
print ("Output is : ", b)

Output:

Output is Sushant Poojary  
roll no. 1762

incomplete  
Output

Readline method  
Output is Sushant Podar

Readlines Method:

Output is [ 'Sushant Podar', 'vol no 146' ]

a) To read multiple lines

a reading method :

a. open ("SUSIANT", "r")

b. readline()

a. close()

print ("Output is", b)

a reading method : read()

a. open ("SUSIANT", "r")

b. readlines()

c. close()

print ("Output is", b)

a) To close the file

a. close()

or  
a. close()

#

append mode :

```
file obj = open("SUSHANT", "a")
file obj . write (" data structure ")
file obj . close()
file obj = open("SUSHANT", "a", "r")
str3 = fileobj.read()
print ("Output of append mode : ", str3)
fileobj.close()
>>> (Output of append mode : , 'SUSHANT', 'data
structure')
```

22

```
# tell()
file obj = open("SUSHANT", "r")
pos = file obj.tell()
print ("tell() : ", pos)
file obj . close()
>>> ('tell() : ', pos)
```

# tell()

```
file obj = open("SUSHANT", "r")
str4 = fileobj.seek(0,0)
str8 = fileobj.read(10)
print ("The beginning of file : ", str8)
fileobj.close()
```

↓  
Dr. (Mr)

### Code:

```
mytuple = ("Rachit", "Dhruv", "Pushman")
myiter = iter(mytuple)
print(next(myiter))
print(next(myiter))
print(next(myiter))
```

### Output :-

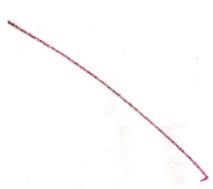
Rachit  
Dhruv  
Pushman

### Code:

```
mytuple = ("Rachit", "Dhruv", "Pushman")
for a in mytuple:
    print(a)
```

### Output:

Rachit  
Dhruv  
Pushman



Ques :-  
How to display elements of a tuple using iterator method.

Demonstration :-  
To display elements of a tuple with certain elements intact

Step 2 :- Use iter method with tuple & assign it to a variable

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

b) To use iter method with for loop :-

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

## Aim: Iterators

### Algorithm:

Step 1:

Create a tuple with elements that we need to iterate using the `iter` and `next` method the number of time we use the `iter` & `next` method we will get the next iterating element in the tuple. Display the same.

Step 2:

The similar output can be obtained by using `for` conditional statement. An iterable variable is to be declared in `for` loop which will iterate.

Step 3:

Define a function name `square` with a parameter which will obtain output of square value of the given number. In similar way declare var. to get value raised 3 & return the same.

Step 4: Call the `declare` function using function call.

```

# iter() and next():

mytuple1 = ("apple", "orange", "banana")
myiter1 = iter(mytuple1)
print(next(myiter1))
myiter2 = iter(mytuple1)
print(next(myiter2))
mytuple3 = iter(mytuple1)
print(next(mytuple3))

Output: >>> apple
banana

```

# for loop:

```

mytuple1 = ("Tom", "Stuart", "Terry")
for x in mytuple1:
    print(x)

Output: >>> Tom
Stuart
Terry

```

# square & cube:

```

def square(x):
    y = x * x
    return y
def cube(x):
    z = x * x * x
    return z

```

Count i = [square, cube]

x [ For i in range [5]:  
value set map (lambda x:) ]

```
for i in range [x]:  
    value = list(map(lambda x: x(i)), func)
```

```
>>> [0,0]  
[1,1]  
[4,8]  
[9,27]  
[16,64]
```

```
# map () :-  
listnum = [0,2,3,4,5,6,7,8]  
listnum = list(map(lambda x: x%5, listnum))  
print (listnum)  
  
def even(x):  
    if (x%2 == 0):  
        return "Even"  
    else:  
        return "Odd"  
>>> [0,2,4,6,8]  
  
# odd number :-  
  
class odd  
  
def __iter__(self):  
    self.num = 1  
    return self  
  
def __next__(self):  
    num.self = num  
    num.self = 2  
    def __next__(self):  
        num = self.num  
        self.num += 2  
    return num
```

Step 5: Using for conditional statement specifying the range use the list type casting with map method, declare a lambda Fun . i.e. anonymous function & print the same.

Step 6: Declare a listnum variable & declare some elements & use the map method with help of lambda function give two arguments display the output.

Step 7: Define a listnum variable and declare some elements and use the map method with help of lc

Step 8: Define a function even with a parameter then using conditional statement to check whether the number is even and odd & return the value respectively

Step 9: Define a class & within that define the iter() method which will initialize the first element within the container object.

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

Step 10: Define an object of a class

Step 11: Accept a number from the user & when we which use want to display the odd no.

Step 12: Define a function with if condition if exactly equal to one return else on parameter.

Step 13: Enter a input from the declare an empty list use append method for appending the input value. Print the output variable.

Step 14: Define a function which will accept a input to find factorial

Step 15: Define a empty set list.

Step 16: Use append method to pass the value of input

Step 17: Use map method with function name and a empty list

Step 18: Print the output.

```

myobj = add()
myiter = iter(myobj)
x = list(input("Enter a number"))
for i in myiter:
    if C(i < x):
        print(i)

```

```

>>> Enter a number: 15
1
3
5
7
9
11
13

```

```

# factorial :=
def f(n):
    if (n == 1):
        return 1
    else:
        return (n * f(n-1))

```

*Factorial*

```

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

```

```

lst = []
lst = append(n)
a = map(f, lst)
print("The factorial is : ", a)

```

```

>>> Enter a number: 5
The factorial is : 120

```

Source code:

```
try :  
    f  
    a = open("Sushant","w")  
    a.write("Hi ! How are you?") +  
        (" where do you live?")  
except IOError:  
    print("Not Possible")  
else :  
    print(" I am good")
```

Output:

I am good.

Q:- Using the exception block, related to the environment error?

### Algorithm:

Step 1:- Use the try block to define normal character of action for eg. define the file object and open the file in the write mode and write some content onto the file.

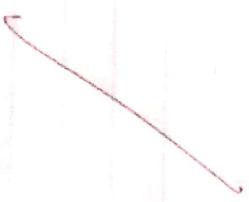
Step 2:- Use the except block with a i/o error as an environment error & convey the appropriate error to the user else display the message that the ~~operation~~ is carried out successfully.

WAP for demonstrating the use of the value error in the given program Statement

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

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

Step 3 :- we can define the multiple exception using the except statement for finding the different categories of error



Source code:

```
try :
    a = open ("sushant", "r")
    a.write ("Hi ! How are you ?")
except ValueError:
    print ("Not possible")
```

except IOError:

print ("where do you live")

else :

print ("default action")

Output:  
default action.

Q. While True,

try :

```
file a = open ("sushant.txt", "w")
a.write ("Hi how are you ?")
a = int (input ("Enter a no ."))
print (a)
```

break

except IOError:

print ("The I am good")

except ValueError:

print ("The value is invalid")

*Ans:*

Output:

Enter a number : abc.

w) The value is invalid.

Enter a number : pqr

w) The value is invalid

Enter a number : #3

73

## Topic : Regular Expression

Step 1: Import re module declare pattern and declare sequence use match method. Declare arguments if arguments matched print the same otherwise print pattern NOT FOUND!

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

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

```
# match()
import re
pattern = r"FYCS"
sequence = "FYCS represents 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'\dt'
string = 'hello123, howdy789, us howru'
output = re.findall(pattern, string)
print(output)

>>> ['123', '789', '45']
```

## # Split()

```
import re
pattern = r'\dt'
string = 'hello123, howdy789, us howru'
output = re.split(pattern, string)
print(output)

>>> ['hello', 'howdy', 'howru']
```

No. 7

```
# no_space:
```

```
import re
String = 'abc def ghi'
pattern = r"\S"
replace = ""
v1 = re.sub(pattern, replace, String)
print(v1)
>>> abcdefghi
```

8.

```
# group()
```

```
import re
```

```
Sequence = "Python is an interesting language"
v = re.search('IA Python', sequence)
print(v)
v1 = v.group()
print(v1)
```

9.

10

```
>>> <_SRE_Match Object at 0x0281DF00>
Python.
```

```
# Verifying the given set of phone nos.:-
```

```
import re
list1 = ['874667990', '9145673210', '7765432056',
         '9820453150']
```

```
for value in list1:
```

```
    if re.match(r'[8-9]{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.
```

Step 4: 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.

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

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

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

Q1 Vowels.

```
import re
str1 = 'Plant is life overall'
output = re.findall(r'[aeiouAEIOU](wt)', str1)
print(output)
''' ['is', 'overall']
```

# host & domain

```
import re
seq = 'abc.tsc@edu.com', xyz@gmail.com,
pattern = r'[\w+.] + [\w+.] + [\w+.] +'
output = re.findall(pattern, seq)
print(output)
''' ['abc.tsc', 'edu.com', 'xyz', 'gmail.com']
```

# Counting of first 2 letters:

```
import re
s = 'mr.a, mr.b, ms.c, mt.t'
P = r'[ms / mr /] +'
O = re.findall(P, s)
print(O)
m = 0
f = 0
for v in
```

Output  
''' ['mr', 'ms.c', 'mt.t']

tk

- □ ×

Hey There! Sushant Here

label.

## Practical - 5

33

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

Algorithm:

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

Step 2: Create a variable from a text variable & position it onto 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 mainloop method for triggering corresponding event.

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

- i) None of the parent window
- ii) Text attribute which defines the string
- iii) The background colour (bg)
- iv) The foreground colour (fg)

Now use the pack() with relevant attributes

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

Algorithm:

Step 1: Use tkinter method to import relevant methods

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 a object of Radiobutton which will take the following arguments

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

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

## Source code:

34

```
from tkinter import *
def sel1():
    Selection = "Dhruv"
    label.config(text=Selection)
def sel2():
    Selection = "Rachit"
    label.config(text=Selection)
def sel3():
    Selection = "Mahesh"
    label.config(text=Selection)
def sel4():
    Selection = "Pushpraj"
    label.config(text=Selection)
root = Tk()
var = IntVar()
L1 = Label(root, text="Select any roll number")
L1.pack(side=TOP)
R1 = Radiobutton(root, text="1746", variable=var, value=0,
                  command=sel1)
R1.pack(anchor=N)
R2 = Radiobutton(root, text="1789", variable=var, value=1,
                  command=sel2)
R2.pack(anchor=N)
R3 = Radiobutton(root, text="1766", variable=var, value=2,
                  command=sel3)
R3.pack(anchor=N)
R4 = Radiobutton(root, text="1774", variable=var, value=3,
                  command=sel4)
R4.pack()
label = Label(root)
label.pack(side=BOTTOM)
root.mainloop()
```

UR

- □ X

Select any roll number:

1746

1789

1766

1774

Dhruv

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

Aim: c) To make use of scroll bar widget in the GUI application.

Algorithm:

Step 1: Import tkinter library to use Scrollbar wi

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

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

Step 4: Use pack method along with object of scrollbar method & use arguments side & fil

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

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

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

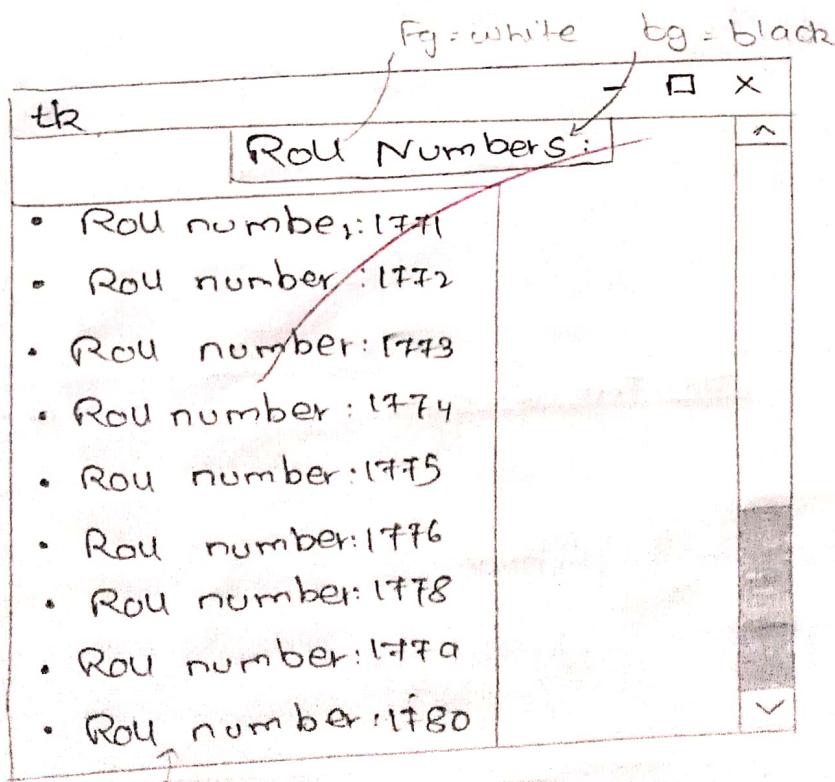
Step 8: Finally call the mainloop method.

Program:

```

from tkinter import *
root = Tk()
root.geometry('450x400')
l = Label(root, text="Roll numbers:", bg="Black", fg="white")
l.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:

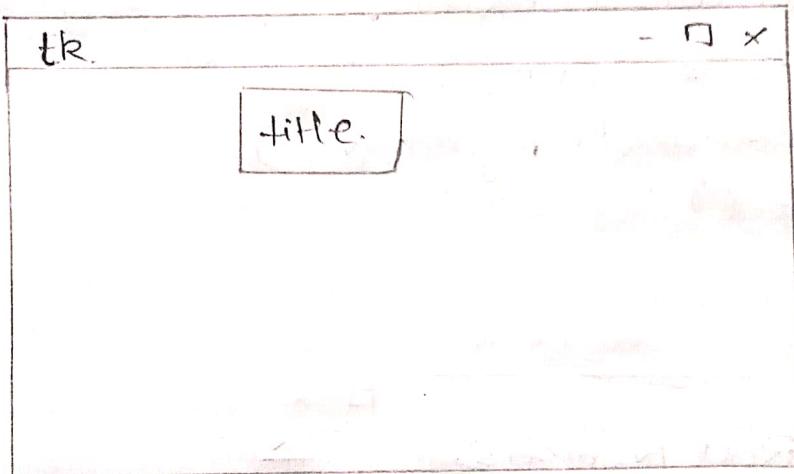
bg = light blue

Jan 1 L

## Message Box

```
from tkinter import *  
import tkMessageBox  
root = Tk()  
def fun():  
    tkMessageBox.showinfo ("info window", "Python")  
    b1 = Button(root, text = "title", command = fun)  
    b1.pack()  
root.mainloop()
```

Output:



the Info Window

i Python

OK

d) Message Box Method :-

Step 1 :- Import the relevant method from tkinter library.

Step 2 :- Define a function by use of the message box along with different methods available which contains one or more arguments.

Step 3 :- Thus different options, which are available are showinfo(), showwarning(), askyesno(), askquestion()

Step 4: Create object from button method & place it onto parent window with title of button specified.

Step 5:- Use the pack method to display the button widget of & finally use mainloop method.

Step 6:- If this user wants to hide the parent window and only the info window should be visible corresponding to the six option given below

## Robot Style

Step 1: Use the button with the following  
standards:

1. The Robot window
2. Text attached
3. Robot

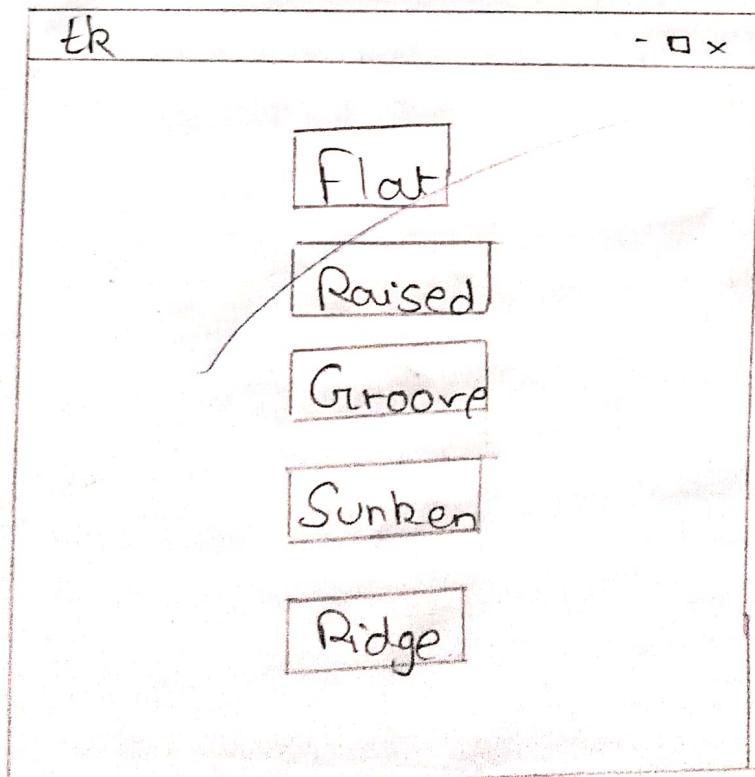
Step 2: Use the corresponding pack method  
for the respective button object and  
trigger the corresponding event.

Step 3: Finally use the onwindow method.

## Source code:

```
from tkinter import *
root = Tk()
b1 = Button(root, text = "Flat", relief = FLAT)
b1.pack()
b2 = Button(root, text = "Raised", relief = RAISED)
b2.pack()
b3 = Button(root, text = "Groove", relief = GROOVE)
b3.pack()
b4 = Button(root, text = "Sunken", relief = SUNKEN)
b4.pack()
b5 = Button(root, text = "Ridge", relief = RIDGE)
b5.pack()
root.mainloop()
```

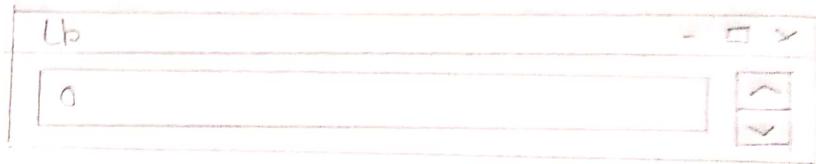
## Output:-



No. 68  
Source Code:

```
from tkinter import *  
root = Tk()  
sl = Spinbox(root, from_=0, to_=10)  
sl.pack(anchor=s)  
root.mainloop()
```

No. 69  
Output:



## Spinbox:

Step 1:

Create an object from `tkinter` method and subsequently create an object from the `Spinbox` 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.

## \* Paned Window.

Step1: Create an object from paned window and use the pack method with the attribute fill and expand.

Step2: Create an object from the label method and put it onto the paned window with the text attribute and use the add method to embed the new object.

Step3: Similarly create a second paned window object and add it onto the 1<sup>st</sup> paned window with orientation specified.

Step4: Now create another label object and place onto the 2<sup>nd</sup> paned window object and add onto the 2<sup>nd</sup> pane.

Step5:- Now use the mainloop method to terminate.

## Source Code :

```

from tkinter import *
root = Tk()

P = PanedWindow(bg = "red")
P.pack(fill = BOTH, expand = 1)
L1 = Label(P, text = "Python Gui", bg = "green")
P.add(L1)

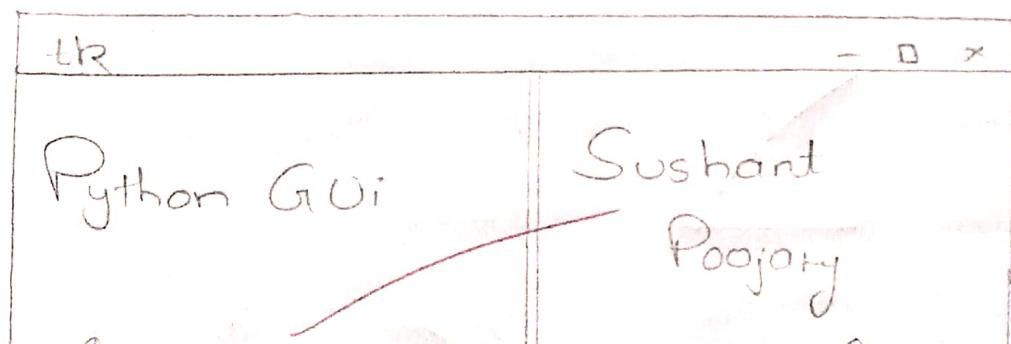
P1 = PanedWindow(P, orient = VERTICAL, bg = "blue")
P.add(P1)

L2 = Label(P1, text = "Sushant Poojary", bg = "gray")
P1.add(L2)

root.mainloop()

```

## Output:



## Source code:

```
from tkinter import *
```

```
root = Tk()
```

```
c1 = Canvas(root, height = 400, width = 400, bg = "blue")
```

```
oval = c1.create_oval(20, 140, 150, 250, fill = "white")
```

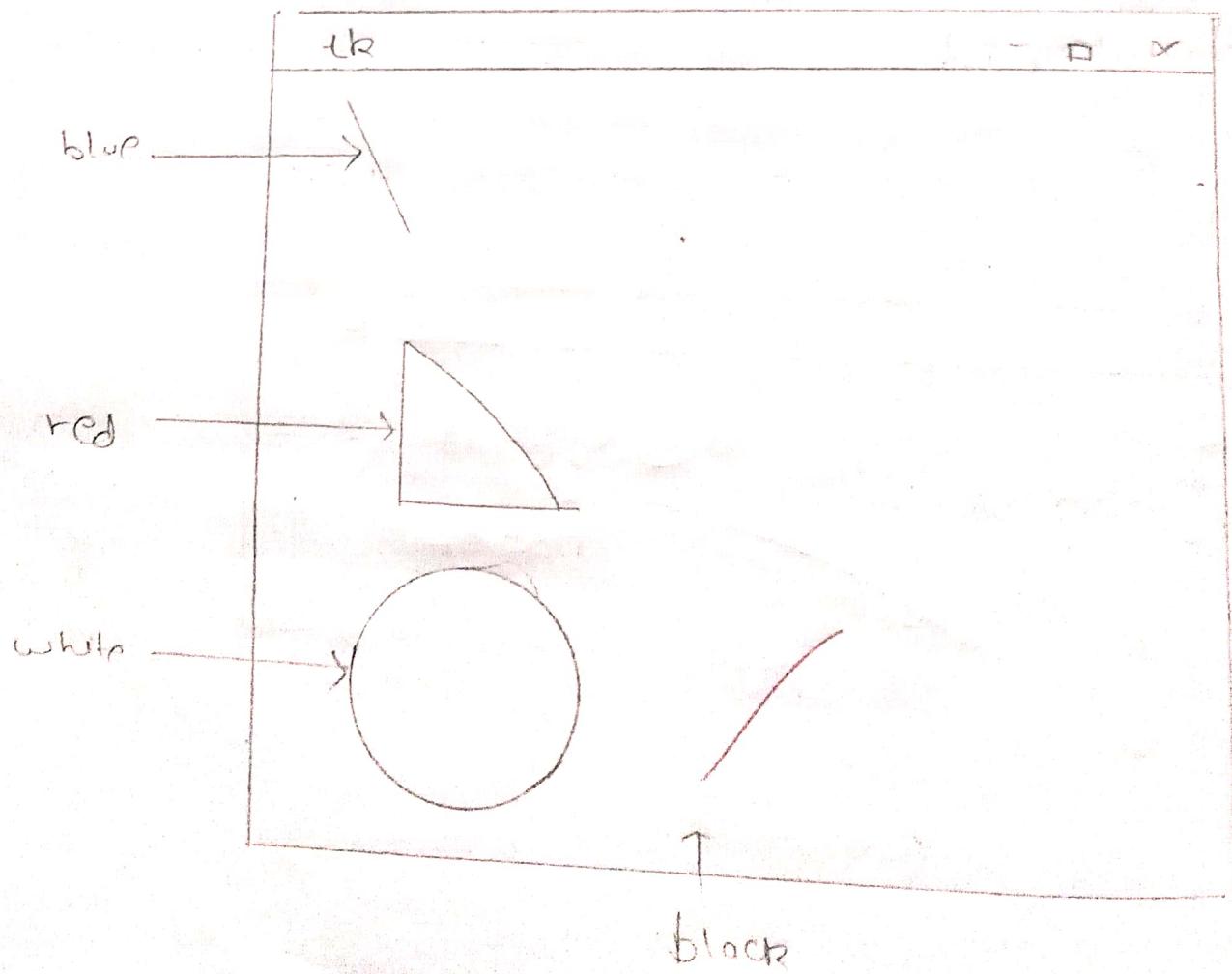
```
line = c1.create_line(30, 40, 60, 60, fill = "blue")
```

```
arc = c1.create_arc(20, 140, 150, 60, fill = "red")
```

```
c1.pack()
```

```
root.mainloop()
```

## Output:



## Canvas Widget

- Step 1:- Use the tkinter method and create an object from the canvas method and use the attribute height , weight , bg colour and the parent window obj.
- 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 colours.
- Step 3:- Now call the pack method and mainloop method.

Cat

Dog

Scanned with CamScanner

# 1  
Aim: To make use of database library

Algorithm:

Step 1: Import the dbm library to implement various features of sqlite and use the open method form creating the data base along with the corresponding flag

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

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

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

```
>>> import dbm  
>>> db = dbm.open("database", flag='c', mode=0666)  
>>> db["name"] = "name"  
>>> if db["name"] != None:  
    print("database not empty // match!")  
else:  
    print("database empty! // Not match")  
>>> db.close()
```

Output:

```
>>>match
```

# 2: Step 1: Import corresponding library to make database connection, os & SQLite-3

Step 2: Now create the connection object using SQLite-3 library & the connect() for creating new database.

Step 3: Now create cursor object using the cursor() & from the connection object created.

Step 4: Now use the execute() for creating the table with the column name & respective datatype.

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

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

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

```
import os, sqlite3  
conn = sqlite3.connect("employee.db")  
cur = conn.cursor()  
cur.execute('create table dos(Name char,  
Roll no. int')  
cur.execute("insert into dos values ('Sushant', 1767),  
('Dhruv', 1746)");  
conn.commit()  
cur.execute('Select * from dos')  
print(cur.fetchall())  
conn.close()
```

Output:

[('Sushant', 1767), ('Dhruv', 1746)]

7/11/23