

PRACTICAL NO-1.

```

fileobj = open("abc.txt", "w") # file open in write mode
fileobj.write("Computer science subject" + "\n")
fileobj.write("DBMS in python in DS\n")
fileobj.close() # file close

fileobj = open('abc.txt', 'r') # read mode
str1 = fileobj.read()
print("The output of read method:", str1)
fileobj.close()
>>> The output of read method: Computer science
      Subject
      DBMS
      python
      DS

# readline()
fileobj = open('abc.txt', 'r')
str2 = fileobj.readline()
print("The output of readline method:", str2)
fileobj.close()
>>> The output of readline method: Computer
      science subject

# readlines()
fileobj = open('abc.txt', 'r')
str3 = fileobj.readlines()
print("The output of readlines method:", str3)
fileobj.close()
>>> The output of readlines method: Computer science
      Subject
      DBMS
      python
      DS
  
```

Aim: Demonstrate the use of different file accessing mode, different attributes and Read method.

ALGORITHM:

Step 1: Create a file object using open method and use the write accessing mode followed up by writing some contents onto the file and then closing the file.

Step 2: Now open the file in read mode and then use read(), readline(), readlines() and store the output in variable and finally display the contents of variable.

Step 3: Now use the file object for finding the name of 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.

88

File I/O

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

Step 5: Open file object in read mode, display the updated written contents and close open again in 'r' mode with parameter passed and display the output subsequently.

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

#file attributes  
a = fileobj.name  
print("Name of file (name attribute) :", a)  
=>(Name of file (name attribute, abc.txt))  
  
b = fileobj.closed  
print("((close) attribute :", b)  
=>((close) attribute = True)  
  
c = fileobj.mode  
print("file mode :", c)  
=>("file mode", 'r')  
  
d = fileobj.softspace  
print("softspace", d)  
=>("softspace", 0)  
  
# w+ mode  
# write mode  
fileobj = open("abc.txt", "w+") fileobj = open("abc.txt", "w")  
fileobj.write("Saurabh") fileobj.write("DBMS")  
fileobj.close()  
  
# r+ mode  
# read mode  
fileobj = open("abc.txt", "r+") fileobj = open("abc.txt", "r")  
str1 = fileobj.read() str2 = fileobj.read()  
print("output of r+", str1) print("output of read mode", str2)  
fileobj.close()  
=>('output of r+', 'Saurabh')  
=>('output of read mode', 'Saurabh')

31

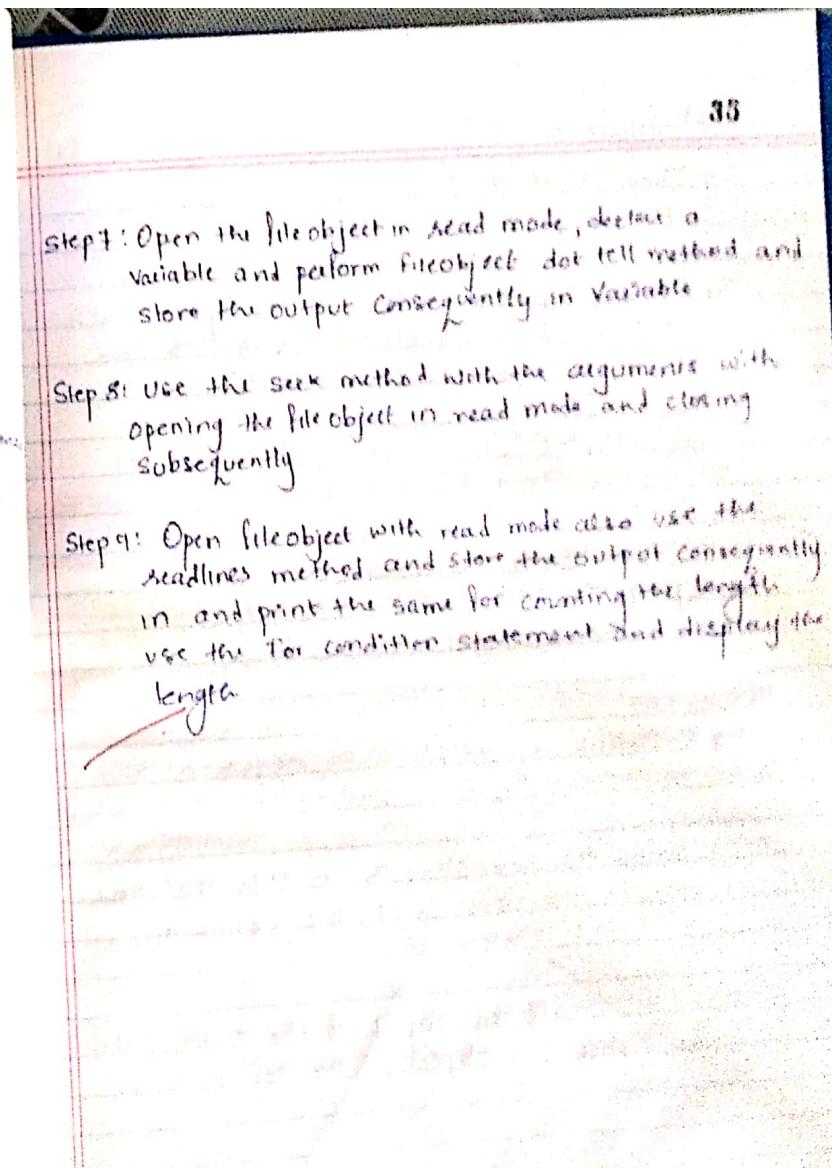
```

#append mode
18.
fileobj=open("abc.txt","a")
fileobj.write("Data structure")
fileobj.close()
fileobject=open('abc.txt',"r")
str3=fileobj.read()
print("Output of append mode",str3)
fileobj.close()
>>>('Output of append mode','saurabh','datastructure')

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

#seek()
fileobj=open("abc.txt","r")
str4=fileobj.seek(0,0)
str5=fileobj.read(10)
print("The beginning of the line is",str5)
m

```



### PRACTICAL NO: 2

AIM: Demonstrate the use of iterables and iterators.

Theory: In python iterator is an object which implements iterator class which has 2 methods namely `--iter()--` and `next()--`.  
list, tuple, dictionary and the set all represents a iterable object.

Q1. Write a program using iterable objects for displaying the odd numbers in range 1 to 10.

Algorithm:

Step 1: Define a `iter()` with argument and initialize the value and return that var.

Step 2: Define the `next()` with an argument and compare the upper limit by using a conditional statement.

Step 3: Now create an object of the given class and pass this object in the `iter` method.

```

#CODE:
def __iter__(self):
    self.num = 1
    return self

def next(self):
    if self.num <= 10:
        num = self.num
        self.num += 2
        return num
    else:
        raise StopIteration

>>> y = count()
>>> z = iter(y)
>>> z.next()
1
>>> z.next()
3
>>> z.next()
5
>>> z.next()
7
>>> z.next()
9
>>> z.next()
11

```

```

# code:
class power:
    def __iter__(self):
        self.p = 0
        return self
    def next(self):
        if self.p <= 10:
            num = self.p
            self.p += 1
            po = 2 ** num
            print("2 ** ", self.p - 1, " = ", po)
            return po
        else:
            raise StopIteration

```

```

>>> p = power()
>>> x = iter(p)
>>> x.next()
2**0 = 1
>>> x.next()
2**1 = 2
>>> x.next()
2**2 = 4
>>> x.next()
2**3 = 8

```

37

2. write a program using an iteration for calculating the power of a given number. For instance number entered is 2 then value calculated should be  $1, 2^1, 2^2, 2^3, 2^4$ .

Algorithm:

Step 1: Define `iter()` with argument and initialize value and return the value.

Step 2: Now define `next()` with an argument and compare the upper limit by using conditional statement.

Step 3: Now create an object of the given class and pass this object in the `iter` method.

3. write a program using iterable concept to find factorial of number in range 1 to 10.

Algorithm:

Step 1: Define a `iter()` with argument and initialize the value and return the value.

Step 2: Define the `next()` with an argument and compare the upper limit by using a conditional statement.

58.

Step 3: Now create an object of the given class, pass this object in the iter method.

Q4 write a program using iterable concept to display multiple of 2 in range 1 to 10

Algorithm:

Step 1: Define a iter() with argument as, initialize the value and return value.

Step 2: Define the next () with an argument and compare the upper limit before a conditional statement

Step 3: Now create an object of the given class and pass this object in the iter method.

33

```
: code:  
class fact:  
    def __iter__(self):  
        self.f=1  
        return self  
    def next(self):  
        if self.f <= 10:  
            num = self.f  
            self.f += 1  
            fact = 1  
            for i in range(1, num+1):  
                fact = fact * i  
            print(self.f-1, "!", "=", fact)  
        else:  
            raise StopIteration  
  
    >>> f=fact()  
    >>> x=iter(f)  
    >>> x.next()  
    1!=1  
    >>> x.next()  
    2!=2  
    >>> x.next()  
    ✓ 3!=6:  
    ↴
```

88. 例題  
# code:

No. 8  
Class mult:

def \_\_iter\_\_(self):

    self.m = 1

    return self

def next(self):

    if self.m <= 10:

        num = self.m

        self.m += 1

        table = 2 \* num

    else:  
        print("2 \* ", num, " = ", table)

        raise StopIteration

>>> m = mult()

>>> x = iter(m)

>>> x.next()

2 \* 1 = 2

>>> x.next()

2 \* 2 = 4

>>> x.next()

2 \* 3 = 6

>>> x.next()

2 \* 4 = 8

### EX-PRACTICAL No.3

Aim: Demonstrate the use of exception handling.

Theory: An exception is an event which occurs during execution of program which disrupts the normal flow of program. Thus a exception represents object which represents an error. This object is derived from given class and when the python script raises an exception it must be handled otherwise it will terminate and end the program.

Q1. write a program to check the range of age of the students in given class. If age does not fall in given range raise value error or exception otherwise return valid no.

Algorithm:

Step 1: Define a function which will accept age of the student from Standard input.

Step 2: Use if conditional to check whether input age falls in range and so raise the age else use ValueError exception.

#code:

```
def accept_age():
    age = int(input("Enter your age:"))
    if age > 30 or age < 10:
        raise ValueError
    else:
        print("your age is", age)
valid = False
while not valid:
    try:
        age = accept_age()
        valid = True
    except ValueError:
        print("your age is not in range")
```

```
>>> Enter your age: 15
Your age is not in range
Enter your age: 32
Your age is not in range
Enter your age: 17
Your age is 17.
```

Q

Write "Five!"

```
try:  
    a = input("Enter a number! ")  
    print("Valid number")  
    break  
except ValueError:  
    print("Not a valid number! Try again")
```

Enter a number: 12.2

Not a valid number! Try again

Enter a number: 17

Valid number!

✓

12

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

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

Q) write a program to check whether the number in given class and if the number is a floating point use value error as exception for the given input.

Algorithm:

Step 1: Use try block and accept the inputs using input() and convert it into integer datatype and subsequently terminate the block.

Step 2: Use the except block with exception as ValueError and display appropriate message if suspicious code is part of try block.

Q3 write a program to demonstrate use of zero division error.

Algorithm:

Step 1: use the try block and accept the input using input() and then convert it into integer datatype

Step 2: Define a function with 2 parameters to divide the numbers given by user

Step 3: Define while loop to check whether the boolean expression holds true

Step 4: Use except with zerodivision error and print the message.

# Code:

```
def divide(a, b):
    ans = a/b
    return ans
while True:
    try:
        a = int(input("Enter first number: "))
        b = int(input("Enter second number: "))
        ans = divide(a, b)
        print("division of", a, "and", b, "is", ans)
        break
    except ZeroDivisionError:
        print("Error!")
>>> Enter first number: 1
>>> Enter second number: 1
Division of 1 and 1 is 1
>>> Enter first number: 1
>>> Enter second number: 0
Error!
```

MM  
19/12/19

## PRACTICAL NO. 4

AIM: Demonstrate the use of regular expression.

THEORY: Regular expression represents the sequence of characters which is mainly used for finding and replacing the given pattern in a string and for this we import re module and common usage of regular expression involves following functionalities:

- Searching a given string
- Finding a string
- Breaking a string into smaller substring
- Replacing part of string

Q1] Write a regular expression segregating numeric and alphabetic values from a given string

Algorithm:

Step 1: Now apply string and pattern in.findall() and display the output

Step 2: \d is used for matching all decimal digits whereas D is used to match non decimal digits.

```
# CODE 1:
import re
string = "hello1234 abcD4567"
result = re.findall("\d+", string)
result1 = re.findall("\D+", string)
print(result)
print(result1)
```

# OUTPUT:

1234  
4567

~~✓~~

Ex:-

Q.2. write a regular expression for finding the match string at the beginning of given sequence

Algorithm!

Step 1: Import re module and apply a string

Step 2: Use search() with "\A Python" and string as two parameters

Step 3: Now display the output

Step 4: Now use if conditional statement for used to know whether the match is found or not

# CODE 2:

```
import re
string = "python is an important language"
result = re.search("\A python", string)
print(result)
if result:
    print("Match found")
else:
    print("Match not found")
```

# OUTPUT:

```
>><re.Match object: span=(0, 6);  
match='python'>
```

>> match found.

✓ MZ

```

#P:
# CODE 3:
import re
li = ["9876543210", "8675324109", "6542109871"]
for element in li:
    result = re.match("[8-9][1]{1}[0-9]{9}", element)
    if result:
        print("Correct mobile no")
        print(result.group(1))
    else:
        print("Incorrect mobile no")
#OUTPUT:
>>> Correct mobile no
9876543210
Correct mobile no
8675324109
Incorrect mobile no

```

Q5 write a regular expression to check whether the given mobile number starts with 8 or 9 and the total length of digit should be atmost 10.

Algorithm:

Step 1: Import re module and apply a string of mobile no.s

Step 2: Now use for Conditional Statement to find if the number starts with 8 or 9 and the total number should length of 10. use match() inside for Statement to find the match in given string.

Step 3: Use if Conditional statement to know whether we have a match or not. If we have use group() display the output and if we don't display Incorrect mobile no.

Q4. Write a regular expression for extracting a word from given string along with space character in between the word and subsequently extract the word without space character

Algorithm:

Step 1: Import re module and apply a string

Step 2: Use.findall() to extract a word from given string

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

Step 4: Now display the output

Q5. Write a regular expression for extracting first and last word from a string

Step 1: Import re module and apply a string -

Step 2: Use.findall() in which use "`^|\w+`" as one parameter to find first word of string then use "`|\w+\$`" as parameter to find last word of string.

Step 3: Now display the result

# CODE 4:

```
import re
string = "python is Important"
result1 = re.findall("\w*", string)
result2 = re.findall("\w+", string)
print(result1)
print(result2)
```

# OUTPUT:

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

# CODE 5:

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

# OUTPUT:

```
>>> ['python']
>>> ['Important']
```

m

m

```

# Q6
import re
string = "Amit 201 24-12-2019"
result = re.findall("\d{2} - \d{2} - \d{4}", string)
print(result)
# OUTPUT:
>>> ['24-12-2019']

```

```

# CODE 7:
import re
string = "abc@tcsc.edu"
result1 = re.findall("@", string)
result2 = re.findall("t\w+", string)
result3 = re.findall("[\w\.-]+", string)
print(result1)
print(result2)
print(result3)
# OUTPUT:
>>> ['@']
>>> ['tcsc']
>>> ['abc', 'tcsc.edu']

```

Q6  
47  
write a regular expression for extracting the date in format dd-mm-yyyy by using the `.findall()` where the string has following format Amit 201 24-12-2019.

Step 1: Import `re` module and apply `string`

Step 2: Use `.findall` method and use '`\d{2} - \d{2} - \d{4}`' as an parameter.

Step 3: Now display the output.

Q7  
write a `re` for extracting the (1)username from email id @hostname from email id (2) Both username and host name from email id:

Step 1: Import `re` module and apply a `string`

Step 2: Use `.findall()` to find username, hostname and both of email id.

Step 3: Use "`\w+`" for username Use "`@\w+\.\w+$`" for hostname and both use "`[\w\.-]+`" for both as parameter in `.findall()`.

Step 4: Display the output.

Q7  
Ans

### PRACTICAL NO: 5.

#### GRAPHICAL USER INTERFACE (GUI)

- (a) write a program to implement Label and Text on the parent window. It has sub menu frame.

Algorithm:

1. Import relevant method from Tkinter library

2. Create an object corresponding to the parent window using Tk()

3. Create an object for label method and place it onto parent window. Now use pack method for positioning of label

4. Similarly Create an another label object now in pack() use attributes of internal padding.

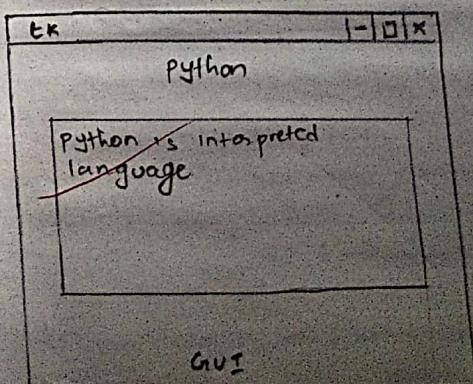
5. Create an object for text method and place it onto the parent window. Use the insert method for applying string in the text.

Now call the mainloop()

#### # CODE:

```
from tkinter import*
root = Tk()
l1 = Label(root, text = "Python")
l1.pack(ipadx = 20, ipady = 50, side = TOP)
l2 = Label(root, text = "GUI")
l2.pack(ipadx = 50, ipady = 80, side = BOTTOM)
t1 = Text(root)
quote = "Python is interpreted language"
t1.insert(END, quote)
t1.pack(ipadx = 70, ipady = 90, side = RIGHT)
root.mainloop()
```

#### # OUTPUT:

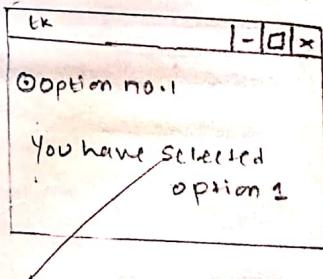


```

#CODE
from tkinter import*
def set():
    selection = "you selected the option "+str(v)
    label.config(text=selection, justify=LEFT)
root = Tk()
var = IntVar()
r1 = Radiobutton(root, text="Option no:1", variable=var, value=1, command=set)
r1.pack(anchor=W)
label = Label(root)
label.pack()
root.mainloop()

#OUTPUT:

```



49

write a program making use of the radiobutton and the control variable for selection of the given choice

Algorithm:

Step 1: Import the relevant method from tkinter library

Step 2: Define a function and define a variable which will keep track of option selected

Step 3: Use the config() along with the label object and the attribute for displaying the selection name

Step 4: Now define the parent window object outside the function definition and define the control variables.

Step 5: Now define an object corresponding to the radiobutton with the following attributes

- (i) Parent window
- (ii) Text attribute
- (iii) Variable
- (iv) Value
- (v) Command

Step 6: Likewise define the five different radiobutton and use the pack method simultaneously by defining the anchor attribute

Step 7: Now define the label object and put it onto the parent window

Q. Write a program implementing the scrolling function using the scroll method.

Algorithm:  
step 1: Import the scroll method from tkinter library

step 2: Create an object corresponding to the parent window using Tk()

Step 3: Create an object from scrollbar and place it onto the parent window

Step 4: Create an object from text method placing it onto the same parent window with the height and width attribute specified

Step 5: Use the pack() with the argument side and fill

Step 6: Create an object from the scrollbar method and use its pack with the side and the fill attribute. Similarly with the text object use the pack()

Step 7: Now use the config() along with the scrollbar object and use the command argument

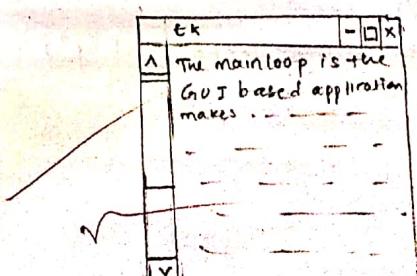
ps: Similarly use the config() with text object and use yscroll command

Q: Now define string variable and use the insert method along with the object and use the mainloop()

# CODE:

```
from tkinter import *
root = Tk()
para = "the mainloop in the GUI based application makes the given widget avail"
s = Scrollbar(root)
t = Text(root, height=1, width=20)
t.pack(side=RIGHT, fill=Y)
s.pack(side=RIGHT, fill=Y)
s.config(command=t.yview)
t.config(yscrollcommand=s.set)
t.insert(END, para)
root.mainloop()
```

# OUTPUT:

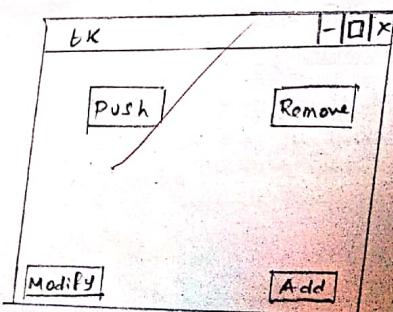


```

#CODE:
from tkinter import *
root = Tk()
frame = Frame(root)
frame.pack(padx=20, pady=50, side=TOP)
leftframe = Frame(root)
leftframe.pack(side=LEFT)
rightframe = Frame(root)
rightframe.pack(side=RIGHT)
buttonPush = Button(frame, text="push", activebackground="red",
                    fg="blue", bg="yellow")
buttonPush.pack(side=LEFT)
buttonRemove = Button(frame, text="Remove", fg="green", bg="yellow")
buttonRemove.pack(side=BOTTOM)
buttonAdd = Button(rightframe, text="Add")
buttonAdd.pack(side=LEFT)
buttonModify = Button(leftframe, text="Modify")
buttonModify.pack(side=RIGHT)
root.mainloop()

```

# OUTPUT:



- 51
- (4) write a program for implementing a frame widget and explain its relevance.
- Algorithm:
- 1 Import relevant method from `tkinter` library.
  - 2 Create an object corresponding to the parent window using `Tk()`.
  - 3 Create an object from `frame` method and place such an object onto the parent window so created.
  - 4 Use the `pack()` for positioning the widget onto the parent window.
  - 5 Create an object termed as `leftframe` and position it onto the left side of window. Similarly create the `rightframe` object and position on the right side of the parent window.
  - 6 Now Create an `button` object and place it onto the frame widget with the `text` attribute `fg` and `bg` and position this on the left side.
  - 7 Similarly create other `button` object named `remove` position on frame and position on `right` side. Further create `add` button. put it onto `rightframe` and on `left` side and create `modify` button on `leftframe` and position on `right` side and call the `mainloop()`.

### PRACTICAL NO. 6

(a) Aim: Write a program to convert temperature in celsius to fahrenheit.

Algorithm:

- 1 Import relevant method from tkinter library.
- 2 Create an object corresponding to the parent window using Tk().
- 3 Now initialize fahrenheit as DoubleVar() and set it to 32.0
- 4 Define a function convert with argument celsius and to convert celsius into fahrenheit using .set()
- 5 Now create an object lc using Label() and place it onto parent window and use text attribute as enter a no:
- 6 Now use grid() for position the object onto the parent window and initialize celsius as integer using IntVar()
- 7 Create another object and use entry widget to enter the input and place it onto the parent window
- 8 Now use grid() for positioning the object onto parent window with textvariable attribute.
- 9 Now again use label() along with textvariable attribute to display output and use grid() for positioning
- 10 Finally use mainloop()

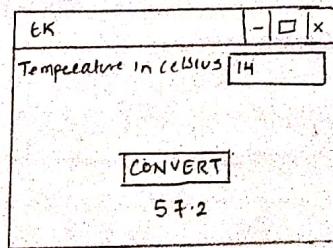
### CODE

```

from tkinter import *
window = Tk()
fahrenheit = DoubleVar()
fahrenheit.set(32.0)
def convert(celsius):
    fahrenheit.set((9.0/5)*celsius+32)
lc = Label(window, text="Temperature in celsius:")
lc.grid(row=0, column=0)
celsius = IntVar()
e = Entry(window, textvariable=celsius)
e.grid(row=0, column=1)
b = Button(window, text="Convert", command=lambda:
            convert(celsius.get()))
b.grid(row=1, column=0, columnspan=2)
l1 = Label(window, textvariable=fahrenheit)
l1.grid(row=2, column=0, columnspan=2)
window.mainloop()

```

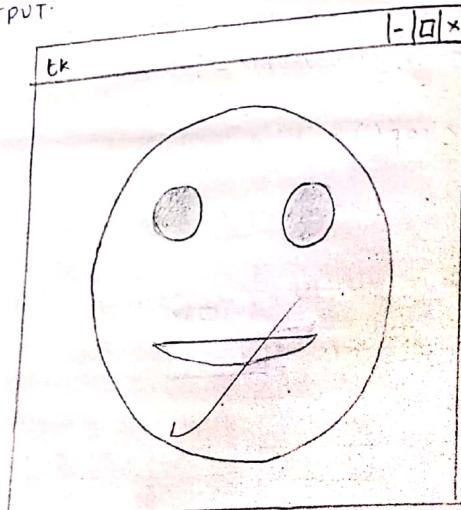
### OUTPUT



#CODE:

```
from tkinter import*
root = Tk()
c = Canvas(root, width=500, height=500)
c.pack()
face = c.create_oval(50, 50, 350, 350, outline="black", fill="yellow")
eye1 = c.create_oval(125, 125, 175, 175, fill="black")
eye2 = c.create_oval(225, 225, 275, 275, fill="black")
mouth = c.create_arc(125, 225, 275, 275, start=0, extent=-180, width=5, fill="red")
root.mainloop()
root.mainloop()

#OUTPUT:
```



(b) Aim: write a program for implementing a canvas widget to draw human face using GUI.

Algorithm:

Step 1: Import relevant methods from tkinter library.

Step 2: Create an object corresponding to the parent window from Tk()

Step 3: Create an object from Canvas () and place it onto parent window along with height and width.

Step 4: Now use pack() for positioning of widget onto the parent window

Step 5: Now create an object face and use object create\_oval() with coordinates 50, 50, 350, 350 and outline = 'black', fill = "yellow" as attribute to create face.

Step 6: Now create eye1 object and again use object.create\_oval() with appropriate coordinates along with fill as attribute to create left eye.

Step 7: Now repeat the same step 6 to create right eye

Step 8: Create an object mouth and use object.create\_oval() with appropriate coordinates, start=0, extent = -180 and fill = "red", width = as attribute to create mouth

Step 9: Finally use the mainloop().

### PRACTICAL NO: 7

Aim: Write a program to find factorial of number using GUI.  
Algorithm:

Step 1: Import relevant methods from tkinter library

Step 2: Now define a function factorial to calculate factorial using recursive function

Step 3: Define another function calculate to call factorial function

Step 4: Now create an object with entry() and use pack() for positioning on parent window

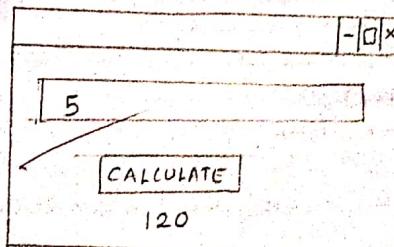
Step 5: Now create an object with button() along with command attribute to calculate factorial.

Step 6: Now again create an object with label() to show output

Step 7: Finally use the mainloop()

```
#code:  
from tkinter import*  
def factorial(n):  
    if n==0 or n==1:  
        return 1  
    else:  
        return n*factorial(n-1)  
def calculate():  
    result=factorial(int(entrytext.get()))  
    info.config(text=result)  
root=Tk()  
entrytext=Entry(root)  
entrytext.pack()  
button=Button(root, text="calculate", command=calculate)  
button.pack()  
info=Label(root, text="Factorial")  
info.pack()  
root.mainloop()
```

### OUTPUT



```

#CODE
from tkinter import*
def calculate():
    if int(v.get()) == 1:
        res = int(e1.get()) + int(e2.get())
        l3.config(text = res)
    elif int(v.get()) == 2:
        res = int(e1.get()) - int(e2.get())
        l3.config(text = res)
    elif int(v.get()) == 3:
        res = int(e1.get()) * int(e2.get())
        l3.config(text = res)
    else:
        res = int(e1.get()) / int(e2.get())
        l3.config(text = res)
root = Tk()
l1 = Label(root, text = "Enter no. 1:")
l1.grid(row=0, column=0)
e1 = Entry(root)
e1.grid(row=0, column=1)
l2 = Label(root, text = "Enter no. 2:")
l2.grid(row=1, column=0)
e2 = Entry(root)
e2.grid(row=1, column=1)
r = IntVar()
r1 = Radiobutton(root, text = "Add", variable = v, value = 1)
r1.grid(row=2, column=0)
r2 = Radiobutton(root, text = "Sub", variable = v, value = 2)
r2.grid(row=2, column=1)
r3 = Radiobutton(root, text = "Mult", variable = v, value = 3)
r3.grid(row=2, column=2)
r4 = Radiobutton(root, text = "Div", variable = v, value = 4)
r4.grid(row=2, column=3)

```

55

(b) Write a program to perform arithmetic operation on two numbers using GUI.

Algorithm:

Step 1: Import relevant methods from tkinter library.

Step 2: Now create an object corresponding to parent window

Step 3: Now define a function calculate to carry out arithmetic operations on 2 numbers.

Step 4: Now create object with label() as num 1 and num 2 and use grid() to place it onto parent window

Step 5: Create objects with entry() to take input from user()

Step 6: Now initialize v as integers using intvar()

Step 7: Now create 4 Objects with Radiobutton() to choose any one of arithmetic operators and use grid() for positioning onto parent window.

Step 8: Now create a object with button() along with command and attribute to carry out the arithmetic operation of user's choice

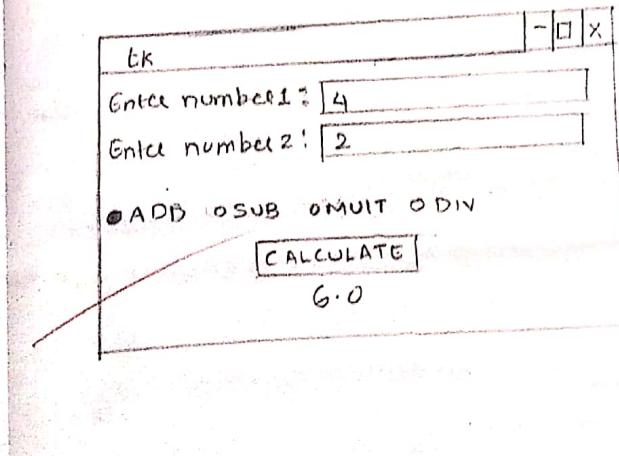
Step 9: Now create a object with label() to show output

18:

step 10: Finally use the mainloop()

```
B=Button(root, text="Calculate", command=calculate)
B.grid(row=3, column=1, columnspan=2)
l3=Label(root)
l3.grid(row=4, column=1)
root.mainloop()
```

# OUTPUT



```

# core
import socket
def server_program():
    host = socket.gethostname()
    port = 5000
    server_socket = socket.socket()
    server_socket.bind((host, port))
    server_socket.listen(2)
    conn, address = server_socket.accept()
    print("Connection from: " + str(address))
    while True:
        data = conn.recv(1024).decode()
        if not data:
            break
        print("from connected user: " + str(data))
        data = input("→")
        conn.send(data.encode())
    conn.close()

```

(Now run the program and now write client program)

```

# core
import socket
def client_program():
    host = socket.gethostname()
    port = 5000
    client_socket = socket.socket()
    client_socket.connect((host, port))
    message = input("→")
    while message.lower().strip() != 'bye':
        client_socket.send(message.encode())
        data = client_socket.recv(1024).decode()
        print("Received from server: " + data)
        message = input("→")
    client_socket.close()

```

### PRACTICAL NO: 8

57

AIM: Demonstrate the use of socket module and server client program.

Algorithm:

Step 1: Import the socket module to import relevant methods.

Step 2: Define a function as server\_program to get hostname

Step 3: Now get value for port variable to initialize port no above 1024

Step 4: Use .socket() to get instance:

Step 5: Now use bind() function to bind host address and port together to configure how many client the server can list simultaneously

Step 6: Now use accept() to accept new connection

Step 7: Now print the address

Step 8: use while loop as True to receive data stream

Step 9: Now close the program

Algorithm:

Step 1: Import socket module to import methods that are relevant

Step 2: Define a function Client program get the host name and give port a value 5000.

Step 3: Now again initiate by using socket.socket()

Step 4: Use connect() to connect the servers.

Step 5: Now take the input ("→")

Step 6: Use while conditional loop to send a message

Step 7: Now use decode to receive response

Step 8: Now show the data.

Step 9: Again take input

Step 10: Close the program by using close()

#OUTPUT for socket - program  
\$ python 3.6 socket-server.py  
connect from: ('127.0.0.1', 57822)  
from connected user: +Hi  
→ Hello  
from connected user: How are you  
→ Good  
from connected user: Awesome!  
→ Ok then, bye!

#output for client - program  
\$ python 3.6 socket-client.py  
→ Hi  
Received from server: +Hello  
→ How are you?  
Received from server: Good  
→ Awesome!  
Received from server: Ok then, bye!  
→ Bye

# CODE IN SHELL ENVIRONMENT :-

```
88.2  
'''import sqlite3  
conn = sqlite3.connect("student1.db")  
cur = conn.cursor()  
cur.execute('create table student(roll_no int(5) primary key,  
name varchar(50) not null, address varchar(50) not null,  
class varchar(50), dob date)')  
<sqlite3 cursor object at 0x02E10520>  
cur.execute('insert into student values (101, "Manisha",  
"FYCS", "12-07-2001")')  
<sqlite3 cursor object at 0x02E10520>  
cur.execute('insert into student values (102, "Isha",  
"FYCS", "13-08-2001")')  
<sqlite3 cursor object at 0x02E10520>  
cur.execute('insert into student values (103, "Deeksha",  
"FYCS", "24-04-2002")')  
<sqlite3 cursor object at 0x02E10520>  
cur.fetchall()  
[(101, 'Manisha', 'Barivali', 'FYCS', '12-07-2001'),  
(102, 'Isha', 'Malad', 'FYCS', '13-08-2001'),  
(103, 'Deeksha', 'Vira', 'FYCS', '24-04-2002')]  
cur.execute('update student set  
dob = "13-09-2002" where roll_no = 101')  
<sqlite3 cursor object at 0x02E10520>  
cur.execute('select * from student where address = "Malad"  
<sqlite3 cursor object at 0x02E10520>  
cur.fetchall()  
[(102, 'Isha', 'Malad', 'FYCS', '13-08-2001')]  
cur.execute('commit')  
<sqlite3 cursor object at 0x02E10520>  
cur.close()
```

### PRACTICAL NO: 7

53

Aim: Demonstrate the use of database connectivity.

Algorithm:

Step 1: Import SQLite3 module to import relevant methods

Step 2: Now initialise a variable Conn to connect by using connect() to a new database using extension .db.

Step 3: Now initialise a variable to connect to cursor()

Step 4: Now use cur.execute() to create a table, insert value into table and use DML, DDL statements to manipulate the data in this database

Step 5: Use fetchall() to show the output

Step 6: Use commit to save all the changes

Step 7: Use close() to terminate the program