

NAAC
Accredited
with Grade "A"
(3rd Cycle)



ISO
9001 : 2015
Certified

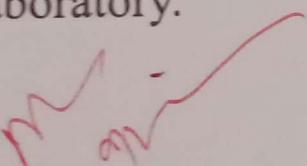
Degree College
Computer Journal
CERTIFICATE

SEMESTER II UID No. _____

Class FY BSC-CS Roll No. 1876 Year I

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

who has worked for the year 2019 - 2020 in the Computer
Laboratory.


Teacher In-Charge

Head of Department

Date : _____

Examiner



INDEX



No.	Title	Page No.	Date	Staff Member's Signature
	PYTHON - II			
1.	Demonstrate the use of different file accessing mode, different attributes & read method.	23	26/11/19	{ mn 19/12/19 }
2.	Demonstrate the use of iterables & iterators.	27	3/12/19	
3.	Demonstrate the use of exception handling.	31	19/12/19	
4.	Demonstrate the use of regular expression.	34	25/12/19	M 03/01/20
5.	Demonstrate the use of GUI components.	40	27/12/19	{ }
6.	a) Drawing of a human face with help of program. b) Kelvin scale - Fahrenheit conversion	49 52 51	21/01/20 21/11/20	{ n 03/03/20 }

INDEX

```

if choice == 4 :
    print ("# Division", div (n,y))

```

else :
choice == 5 :

```
    print ("Modull Division = ", mod (n,y))
```

else :
print ("Wrong selection by user")

SHELL ENVIRONMENT:

Enter 1st value = 5
Enter 2nd value = 2

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Module Division

Enter solution : 3

Multiplication / 10

Aim : Demonstrate the use of different file accessing mode,

different attributes and read method.

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

Step 2 - Now open the file in read mode & then use read(), readline(), readlines() & store the output in variable & 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 & then file mode in which it is open or close & finally the output of the softspace attribute.

Step 4 - Now open the file object in write mode & write some another content then again open the file object in 'w+' mode

Step 5 - Open file object in read mode, display the update written content & close, open again in '+' mode with parameter 'r+' subsequently. P.

Step 6 - Now open file object in append mode, open write method. write content close the file object again. Open the file object in read mode & display the append output.

Fileobj = open("abc.txt", "w")
Fileobj.write("Computer Science subjects" + "\n")
Fileobj.close()

Fileobj = open("abc.txt", "r")
st1 = Fileobj.read()
print("The output of read method : ", st1)
Fileobj.close()

>>> The output of read method : Computer Science
subject
DBMS
Python
DS

```
#readline()
Fileobj = open("abc.txt", "r")
st2 = Fileobj.readline()
print("The output of readline method : ", st2)
Fileobj.close()
>>> The output of readline method : Computer Science subject
```

```
#readlines()
Fileobj = open("abc.txt", "r")
st3 = Fileobj.readlines()
print("The output of readlines method : ", st3)
```

```
>>> The output of readlines method : Computer Science subject
```

```
DBMS
Python
DS
```

```
# file attributes  
a = fileobj.name  
print("Name of file (name attribute):", a)  
>>> (Name of file (name attribute, abc.txt)  
b = fileobj.close()  
print ("(fileclose) attribute:", b)  
>>> (close) attribute = "True"  
c = fileobj.mode  
print ("file mode", c)  
d = fileobj.softspace  
print ("softspace", d)  
>>> ("softspace:", o)
```

```
# wt mode  
fileobj = open("abc.txt", "wt")  
fileobj.write("Nidhi")  
fileobj.close()
```

```
# rt mode  
fileobj = open("abc.txt", "rt")  
st1 = fileobj.read()  
print("Output of rt", st1)  
fileobj.close()
```

```
>>> ("Output of rt", 'Nidhi')
```

```
# cosa mode
```

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

```
# read mode  
fileobj = open("abc.txt", "r")  
fileobj.read()  
print("Output of read mode", st1)  
>>> ('Output of read mode', 'Nidhi')
```

Step 7 - Open the file object in read mode. declare a variable q1 perform fseek & tell not in q1 store the output consequently in variable.

Step 8 - close the seek method with the arguments with opening the fileobject in read mode & closing subsequently.

Step 9 - Open fileobject with read mode also we use the readlines method q1 store the output consequently in q1 print the same for counting the length we use the for condition statement q1 display the length

append mode

```
fileobj = open ("abc.txt", "a")
fileobj.write ("Data Structure")
fileobj.close()
```

File object = open ("abc.txt", "x")

str3 = fileobj.read()

print ("output of append mode", str3)

fileobj.close()

```
>>> ("output of append mode:", "Data Structure")
```

tell()

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

pos = fileobj.tell()

print ("tell():", pos)

fileobj.close()

>>> tell(): 0

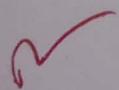
seek()

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

str4 = fileobj.seek(0, 0)

str8 = fileobj.read(10)

print ("The beginning of the line is =", str8)



```
class odd :
```

```
    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
```

Aim : Demonstrate the use of iterable & iterators.

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

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

Algorithm :

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

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

Step 3 - Now create an object of the given class E_1 pass the object in the iter method.

Write a program using an iterator 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 $\text{iter}()$ with argument E initialize value E_1 return the value

Step 2 - Now define $\text{next}()$ with an argument E compare the upper limit by using conditional statement.

Step 3 - Now create an object of the given class E_1 pass this object in the iter method.

class power:

28

def __iter__(self):
 self.sum = 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 = 2~~

>>> x.next()

~~2**1 = 2~~

>>> x.next()

~~2**2 = 4~~

>>> x.next()

~~2**3 = 8~~

class fact:

def __iter__(self):

self.f = 1

return self

def next(self):

if self.f <= 10:

num = self.f

self.f += 1

fac = 1

for i in range(1, num+1):

 fac = fac * i

print(f"!{self.f-1}!", fac)

else:

 raise StopIteration

```
>>> f = fact()
```

```
>>> x = iter(f)
```

```
>>> x.next()
```

1! = 1

```
>>> x.next()
```

2! = 2

```
>>> x.next()
```

3! = 6

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 & initialize the value & return the value

Step 2 - Define the next() with an argument & complete the upper limit by using a conditional statement.

Step 3 - Now create an object of the given class & pass this object as in the iter method.

4] write a program using iterable concept to display multiple of 2 in range 1 to 10.

Algorithm :

Step 1 - Define a iter() with argument & initialize the value & return the value.

Step 2 - Define the next() with an argument Eq
compare the upper limit by using a
conditional statement.

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

```

class mult:
    def __iter__(self):
        self.m = 1
        return self
    def next(self):
        if self.m <= 10:
            num = self.num
            self.m += 1
            table = 2**num
            print("2^", num, "= ", table)
        else:
            raise StopIteration

```

>>> m = mult()

>>> n = iter(m)

>>> n.next()

$$2^1 = 2$$

>>> n.next()

$$2^2 = 4$$

~~>>> n.next()~~

~~m~~ $2^3 = 6$

>>> n.next()

$$2^4 = 8$$

```

def accept_age():
    age = int(input("Enter your age"))
    if age > 30 or age < 16:
        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

✓

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 an exception represents object which represents an error. This object is derived from given class E_1 . When the python script raises an exception it must be handled immediately, otherwise it will terminate E_1 close the program.

Q] write a program to check the range of the age of the students in given class E_1 if age does not fall in given range use value error exception otherwise return the valid no.

Algorithm :

Step 1 - Define a function which will accept the age of the student from standard input.

Step 2 - Use if conditional to check whether the input age falls in range E_1 so return the age else use ValueError exception.

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

Step 4 - Use except with value error & print the message not a valid range.

2.] Write a program to check whether the number in given class & if the number its a floating point use value error as exception for the given input.

Algorithm :

Step 1 - Use try block & accept the input using input() & convert it into integer subsequently terminate the block.

Step 2 - Use the except block with exception as ValueError & display code in appropriate message is part of try block.

while True:

try:

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

print ("Valid number")

break

except ValueError:

print ("Not a valid number ! try again ")

»»> Enter a number : 172

Not a valid number ! Try again

Enter a number : 17

Valid number

✓
m

```
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 & 1 is

```
>>> Enter first number : 1
```

```
>>> Enter second number : 0
```

Error!

3] Write a program to demonstrate use of zero division error.

Algorithm -

Step 1 - Use the try block & accept the input using `input()` & then convert it into integer datatype.

Step 2 - Define a function with 2. parameters to divide the no's given by user

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

Step 4 - Use except with zero division error & print the message.

✓
m/s
19/12/19

Ques: Demonstrate the use of regular expression.

Theory: Regular expression represents the sequence of characters which is mainly used for finding patterns in a string & for this we import re module. The 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

Q.1] Write a regular expression segregating numeric values from a given string.

Algorithm:

Step 1: Now apply string & display the pattern in.findall() 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*abc4567"
result = re.findall ("\d", string)
result1 = re.findall ("\D+", string)
print(result)
print(result1)
```

output :

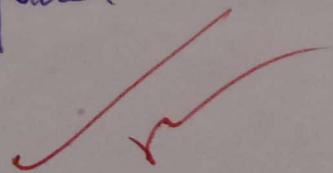
```
>>> ['1234', '4567']
>>> ['hello', 'abc']
```

✓

- 18

HT 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
```



Q2] Write a regular expression for finding the match string at the beginning of given sequence.

Algorithm :

Step 1: Import re module & apply a string

Step 2: use search() with "\APython" & string as two parameters.

Step 3: Now display the output st

Step 4: Now use the if condition of statement for we to know whether the match is found or not.

Q.3] write a regular expression to check whether the given mobile number starts with 8 or 9 & the total length of digit should be almost 10

Algorithm:

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

Step 2: Now use for conditional statement to find if the number starts with 8 or 9 & the total number length should be 10. Use match() statement to find the string.

Step 3: Use if conditional statement to know whether we have a match or not. if we have use group() to display mobile no. if output is correct don't display

```

# Code 36
import re
li = ["9876543210", "8765432109", "7654321098",
      "654321098767"]
for element in li:
    result = re.match("[8-9]{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
8765432109
incorrect mobile no
incorrect mobile no.

```

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']
```



Q4] write a regular expression for extracting a word from given string along with space character subsequently extract the word without space character.

Algorithm :

Step 1 : Import a module & apply a string .

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

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

~~Step 4 : Now display the output~~

Q5] write a regular expression for extracting first the last word from a string .

Algorithm :

Step 1 : Import re module & 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

Q6]

Write a regular expression for extracting the date in format dd-mm-yyyy by using the.findall() when the string has following format Aadhi 201 24-12-2019

Algorithm :

Step 1 : Import re module & apply string.

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

Step 3 : Now display the output.

#CODE 5 :

```
import re
string = "Python is important"
result = re.findall(r"\w+", string)
result1 = re.findall(r"\w+\d", string)
print(result)
print(result1)
```

Output

```
>>> ['Python']
>>> ['Important']
```

code 6 :

```
import re
string = "Amitti 201 24-12-2019"
result = re.findall(r"\d{2}-\d{2}-\d{4}", string)
print(result)
```

~~# output:~~

```
>>> [24-12-2019]
```

CODE 7:

```
import re
string = "abc @ tese.edu"
result1 = re.findall ("^w+", string)
result2 = re.findall ("t\w+\.\w+$", string)
result3 = re.findall ("[\w]-]", string)
print (result1)
print (result2)
print (result3)
```

Output:

```
>>> ['abc']
>>> ['tese.edu']
>>> ['abc', 'tese.edu']
```

✓ ↗

- Q7] write a re for extracting the
- ① username from email id
 - ② hostname from email id
 - ③ Both username & hostname from email id

Algorithm :

Step 1: Import re module & apply a string

Step 2: use.findall() to find username, hostname & both of email id

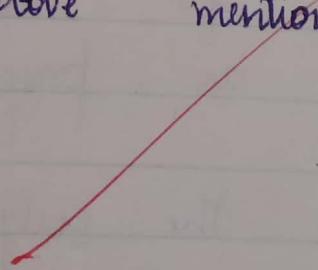
Step 3: use "^\w+" for username use "\w+\." for hostname & use "(^\w+.\w+)" for both as parameter in findall()

~~Step 4:~~ Display the output

W
orlDwOrld

Aim - To demonstrate the use of GUI components.

ALGORITHM -

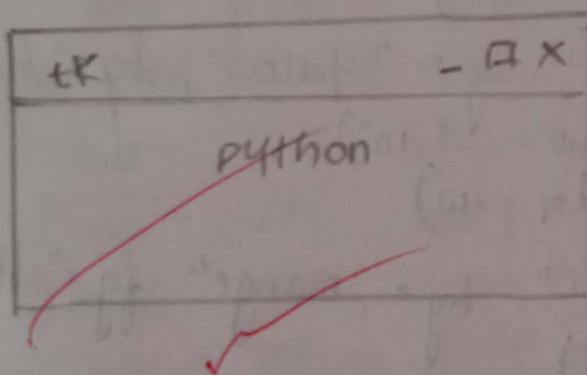
- Step 1 - Use the Tkinter library of the text widgets for importing the features.
 - Step 2 : Create an object using the TK().
 - Step 3 : Create a variable using the widget Label & use the text method.
 - Step 4 : Use the text mainloop() corresponding above mention for triggering of the events.
- 

CODE : (Creation of parent window)

40

```
from Tkinter import*
root = Tk()
l= Label (root, -text = "python")
l. pack()
root . mainloop()
```

OUTPUT :



SYNTAX (2)

```
from Tkinter import *
root = Tk()
l = Label(root, text = "Python")
l.pack()
l1 = Label(root, text = "CS", bg = "grey", fg = "black",
           font = "10")
l1.pack(side = LEFT, padx = 20)
l2 = Label(root, text = "CS", bg = "light blue",
           fg = "black", font = "20")
l2.pack(side = LEFT, pady = 30)
l3 = Label(root, text = "CS!", bg = "yellow", fg = "black",
           font = "10")
l3.pack(side = TOP, ipadx = 40)
l4 = Label(root, text = "CS!", bg = "orange", fg = "black",
           font = "10")
l4.pack(side = TOP, ipady = 50)
root.mainloop()
```

#2

ALGORITHM -

Step 1 - Use the Tkinter library for importing the features of the text widgets.

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

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

1) side = LEFT , padx = 20

2) side = LEFT , pady = 30

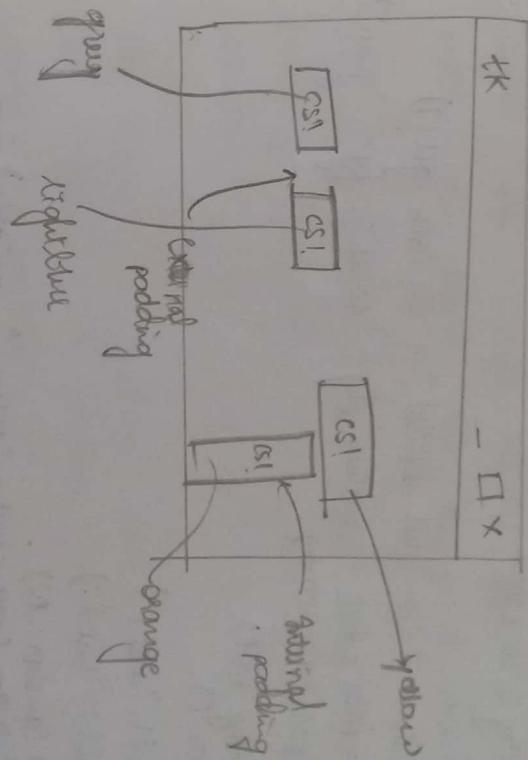
3) side = TOP , ipadx = 40

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

OUTPUT :



- 1) Name of the parent window
- 2) Tk attribute which defines the styling
- 3) The background color (bg)
- 4) The background of a Tk window has the pack() with a relevant padding attribute.

```

from Tkinter import*
root = Tk()
root.geometry("500x500")
var = StringVar()

def sel():
    selection = " You just selected " + str(var.get())
    t1 = Label(text=selection, bg="white", fg="green")
    t1.pack(side=TOP)

var = StringVar()
l1 = Listbox()
l1.insert(1, "List 1")
l1.insert(2, "List 2")
l1.pack(anchor=N)

r1 = Radiobutton(root, text="option1", variable=var, value="option1", command=sel)
r2 = Radiobutton(root, text="option2", variable=var, value="option2", command=sel)

root.mainloop()

```

#1 Algorithm -

Step 1: Import the relevant methods from the tkinter library class an object with the parent window.

Step 2: Use the parent window object along with the geometry() method for creating specific pixel size of the parent window.

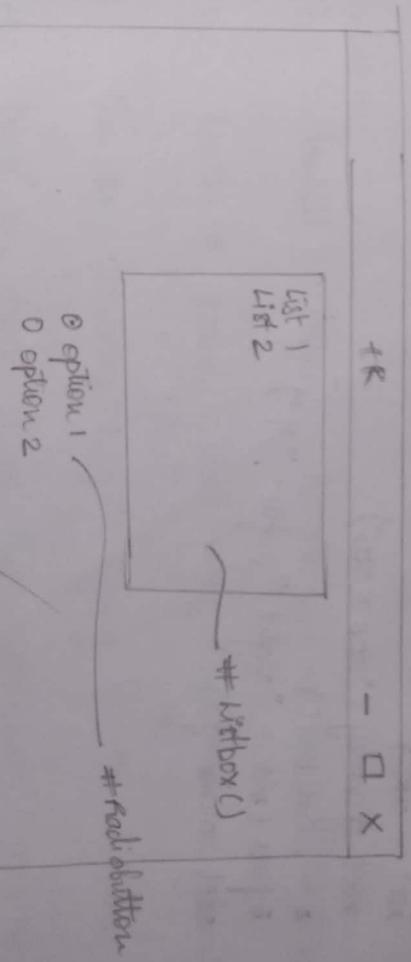
Step 3: Now define a function which tells the user about the given selection made from multiple option available.

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

Step 5: Use the listbox() and insert option on the parent window along with the pack() with specifying anchor attribute.

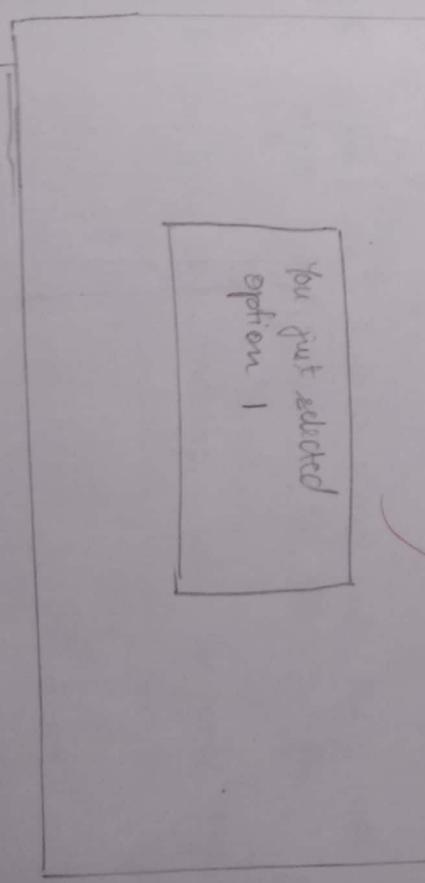
Step 6: Create an object from radiobutton which will take following arguments parent window object, text variable which take the values option no 1, 2, 3, variable argument, corresponding value & trigger the function declared.

#OUTPUT:



Step 7: Now with the `path()` for radio object
we created to specify the argument
using anchor attribute.

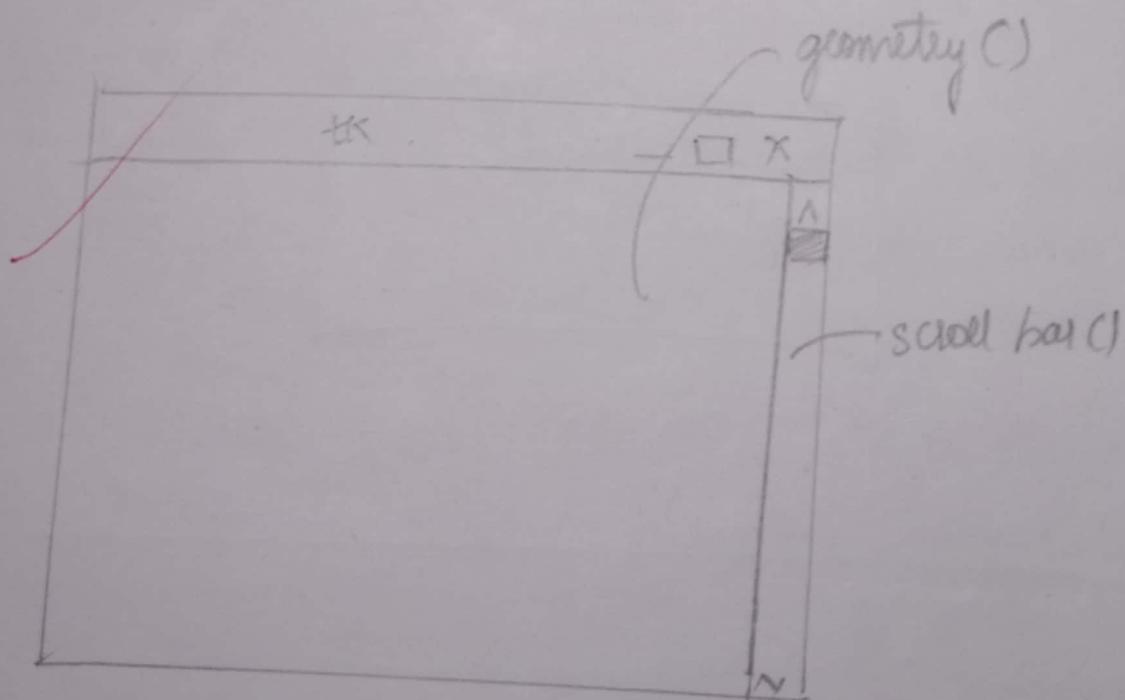
Step 8: Finally make use of the `mainloop()`
along with parent object



~~# OUTPUT~~ SCROLLBAR

```
from tkinter import *
root = Tk()
root.geometry ("500 x 500")
s = scrollbar()
s.pack (side = "right", file = "Y")
root.mainloop()
```

OUTPUT:



Aim: Demonstrate the use of scrollbar using GUI.

#Algorithm -

Step 1: Import relevant methods from the tkinter library.

Step 2: Create a parent object corresponding to the parent window.

Step 3: Use the geometry() for laying of the window.

Step 4: Create an object & use the scrollbar()

Step 5: Use the pack() along with the scroll bar object with side & fill attributes.

Step 6: Use the mainloop with the parent object

#3:

Step 1: Import the relevant libraries from the Tkinter method

Step 2: Create an corresponding object of the parent window

Step 3: Use the geometry manager with pixel size (680 x 500) or any other suitable pixel value.

Step 4: Use the label widget along with the parent object created & subsequently use the pack method.

Step 5: Use the frame widget along with the parent object created & use the pack method.

Step 6: Use the listbox method along with the attribute like width, height, font. Do create a listbox method object use pack() for the same.

Step 7: Use the scrollbar() with an attribute of vertical then same with object created scrollbar() use object use the configure the from the pack()

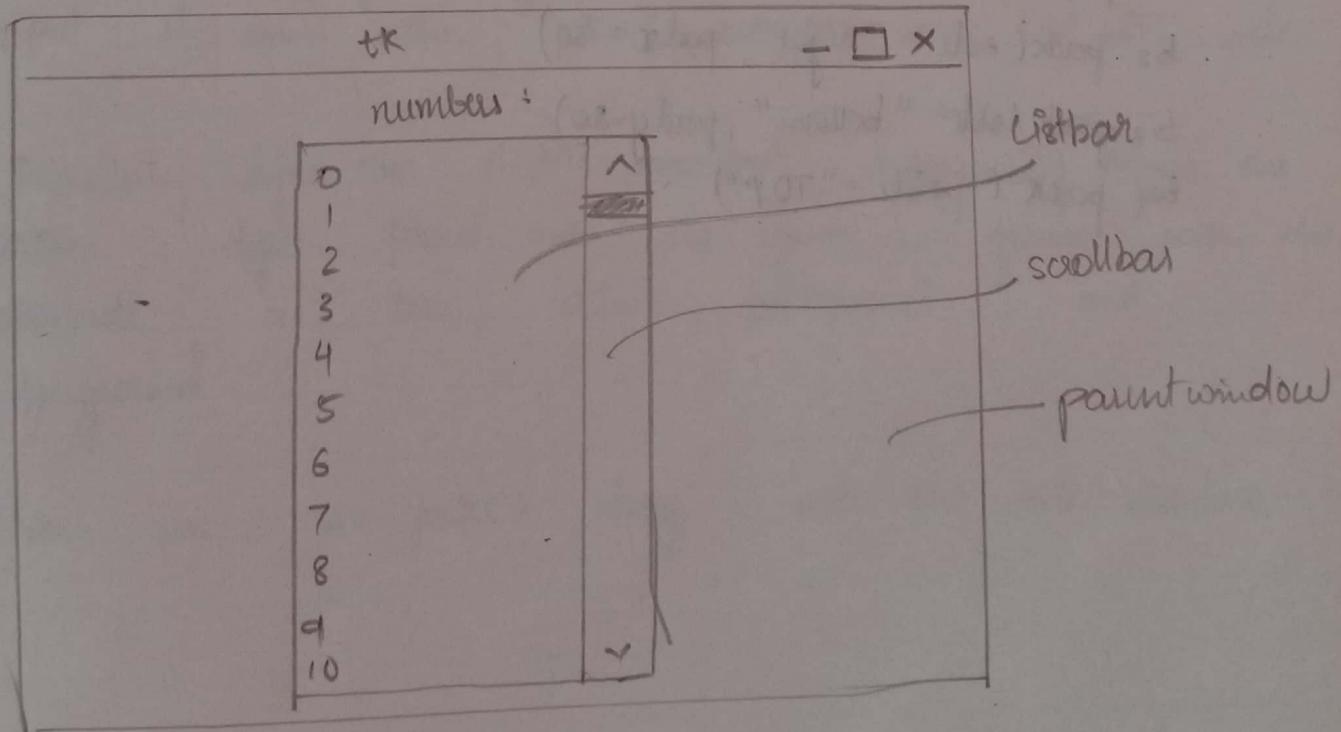
Step 8: Trigger the events using mainloop.

```

# code (3)
[using frame widget]
from tkinter import *
window = Tk()
window.geometry ("680x500")
label (window, text = "numbers :").pack()
frame = Frame (window)
frame.pack()
listNodes = Listbox (frame, width = 20, height = 20, font = "Times New Roman",
                     listNodes.pack (side = "left", fill = "y"))
                     (10))
scrollbar = Scrollbar (frame, orient = "vertical")
scrollbar.config (command = listNode.yview)
scrollbar.pack (side = "right", fill = "y")
for x in range (100):
    listNodes.insert (END, str(x))
window.mainloop()

```

#output-



#4 :

```
from tkinter import *
window = Tk()
window.geometry ("680x500")
frame = Frame (window)
leftframe.pack (side = "Left")
rightframe = Frame (window)
rightframe.pack (side = "right")
b1 = Button (frame, text = "select", active background = "red",
             fg = "blue")
b2 = Button (frame, text = "modify", active background = "yellow",
             fg = "black")
b3 = Button (frame, text = "ADD", active background = "red", fg = "green")
b4 = Button (frame, text = "EXIT", active background = "blue",
             fg = "red")
b1.pack (side = "LEFT", padx = 20)
b2.pack (side = "right", padx = 30)
b3.pack (side = "bottom", pady = 20)
b4.pack (side = "TOP")
```

#4

Algorithm -

- Step 1: Import relevant methods from tkinter library.
- Step 2: Define the object corresponding to parent window and define the size of parent window in terms of no of pixels.
- Step 3: Now define the frame object from the method and place it on to the parent window.
- Step 4: Create another frame object termed as the left frame and put it on the parent window on its LEFT side.
- Step 5: Similarly define the RIGHT frame and subsequently define the bottom 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.

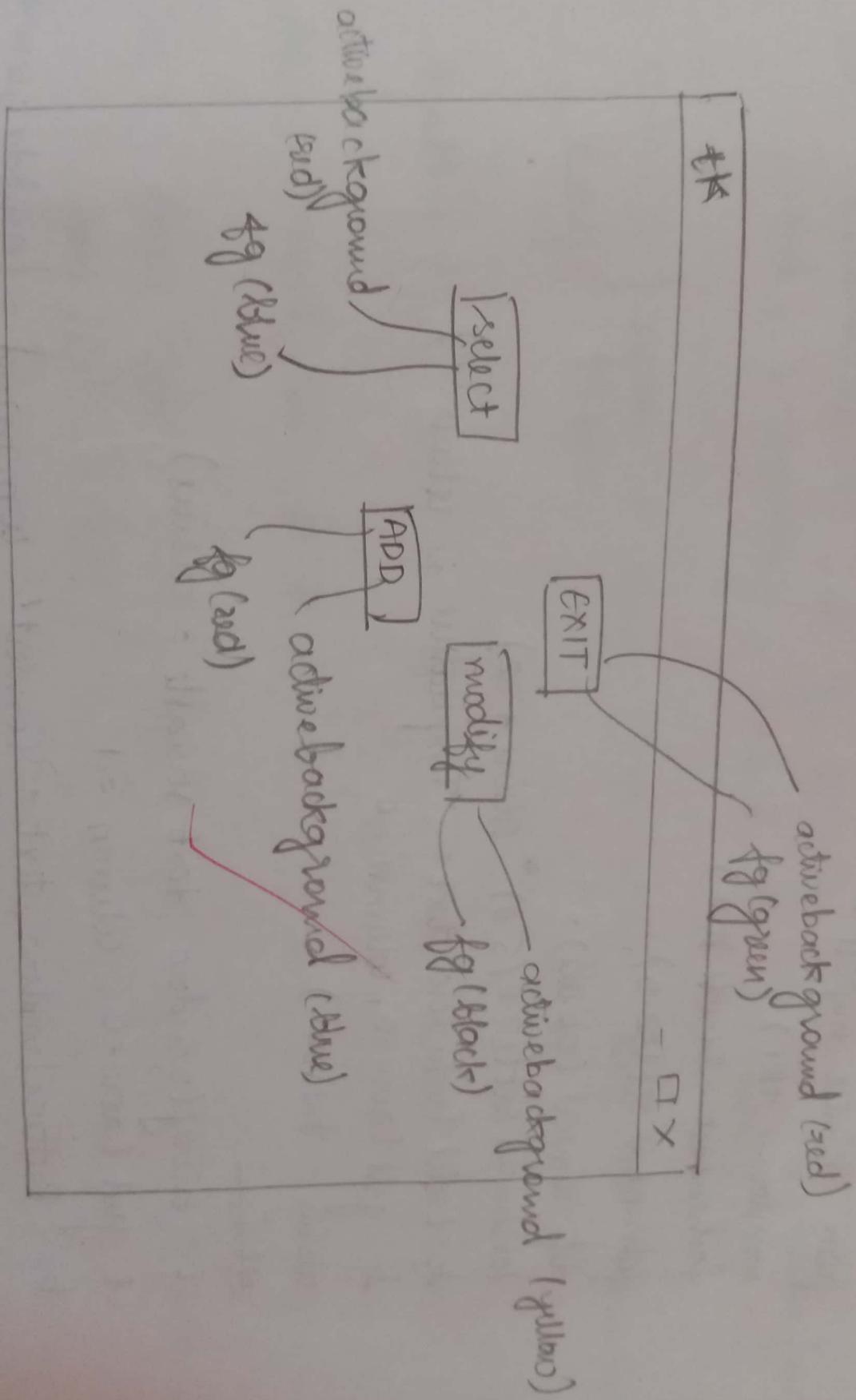
Step 7 : Similarly create the button object corresponding to the MODIFY operation put it into frame object on side = "right"

Step 8 : Create another button object & place it on the RIGHT frame & label the button as ADD.

Step 9 : Add another button & put it on the frame & label it as EXIT

Step 10 : Use the pack() simultaneously for all the objects & finally use the mainloop()

OUTPUT



SYNTAX (1) :

```

from tkinter import *
window = Tk()
fahrenheit = DoubleVar()
fahrenheit.set(32.0)
def convert(celsius):
    fahrenheit.set((9.0/5.0)*celsius + 32)
lc = Label(window, text = "Temperature in Celsius :")
lc.grid(row=0, column=0)
celsius = IntVar()

```

~~celsius =~~

```

e = Entry(window, textvariable = celsius)
e.grid(row=0, column=1)
b = Button(window, text = 'Convert', command = lambda: convert(e.get()))
b.grid(row=1, column=0, columnspan=2)
lf = Label(window, textvariable = fahrenheit)
lf.grid(row=2, column=0, columnspan=2)
window.mainloop()

```

PRACTICAL - G

Aim: Demonstrate the use of GUI by creating a human face and converting celsius into fahrenheit.

1] Write a program to draw human convert celsius into fahrenheit using GUI.

Algorithm:

Step 1 - Import ~~the~~ all the relevant methods in tkinter library.

Step 2 - Create object corresponding to the parent window from Tk().

Step 3 - Now initialize fahrenheit as DoubleVar() & set it to 32.0.

Step 4 - Now define a function convert with argument celsius & to convert celsius into fahrenheit using set()

Step 5 - Now create an object l2 using Label() & place it onto parent window & use text attribute as enter a number.

Step 6 - Now use grid() for position the object onto the parent window.

Step 7 - Initialize celsius as integer using IntVar().

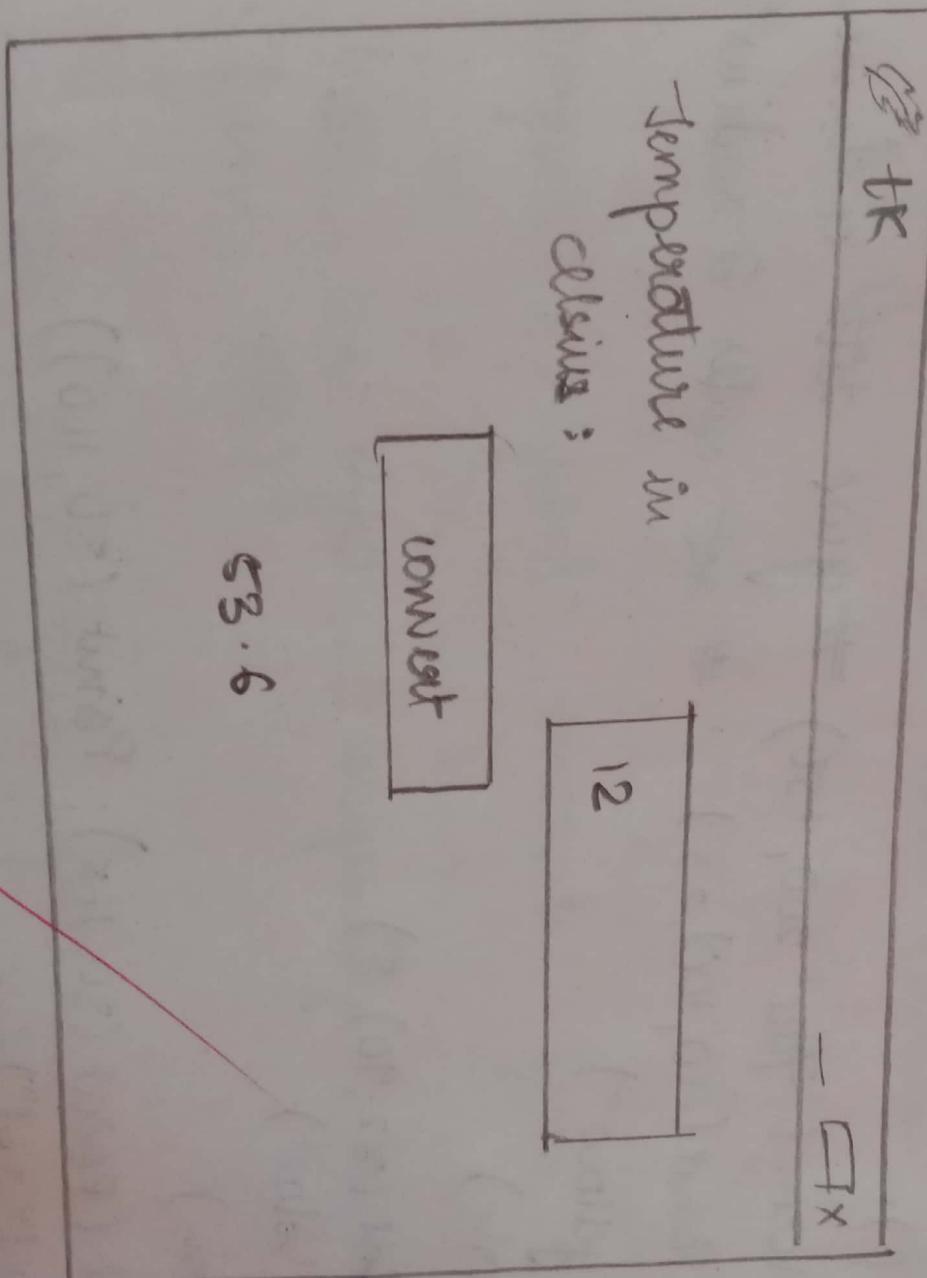
Step 8 - Create another object ϵ_1 use entry widget to enter the input and place it onto the parent window.

Step 9 - Now use grid() for positioning the object onto parent window with text variable attribute.

Step 10 - Now again use label() along with textvariable attribute to display output ϵ_1 use grid() for positioning.

Step 11 - Finally use mainloop()

output :



50

#SYNTAX (2):

```
from graphics import*
def mainpg():
    win = GraphWin('Face', 200, 150) # give title & dimension
    head = Circle(Point(100, 100), 25) # set center & radius - for face
    head.setFill("yellow")
    head.draw(win)
    eye1 = Circle(Point(32, 90), 5)
    eye1.setFill('blue')
    eye1.draw(win)
    eye2 = Circle(Point(68, 90), 5)
    eye2.setFill('blue')
    eye2.draw(win)
    mouth = Oval(Point(30, 115), Point(50, 110))
    mouth.setFill("red")
    mouth.draw(win)
    label = Text(Point(100, 120), 'A face')
    label.draw(win)
    message = Text(Point(win.getWidth() / 2, 20), 'click anywhere to quit')
    message.draw(win)
    win.getMouse()
    win.close()
mainpg()
```

2] write a program 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 C1 & place it onto parent window along with height & width.

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

Step 5 - Now create an object face & use object . Create - oval() with coordinates 50,50,350,350 , & outline = 'black' , "fill" = "yellow": as attribute to create face.

Step 6 - Now create eye1 object & 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.

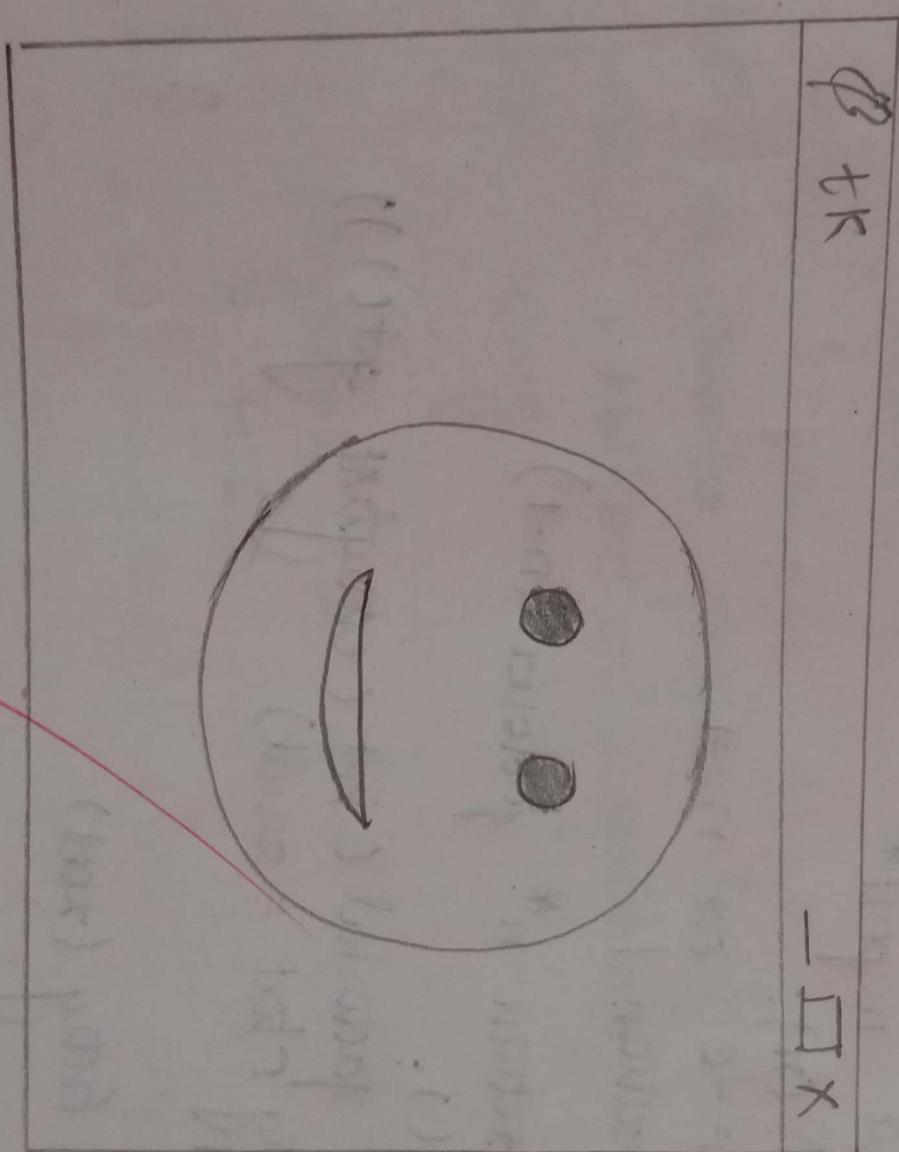
Step 8: Create an object mouth & we object.
Create - arc(), with appropriate co ordinates,
start = 0, extent = -180 & fill = "red",
width = 5 as attribute to create
mouth.

Step 9: Finally use the main loop()

#outfit:

φ tk

-□x

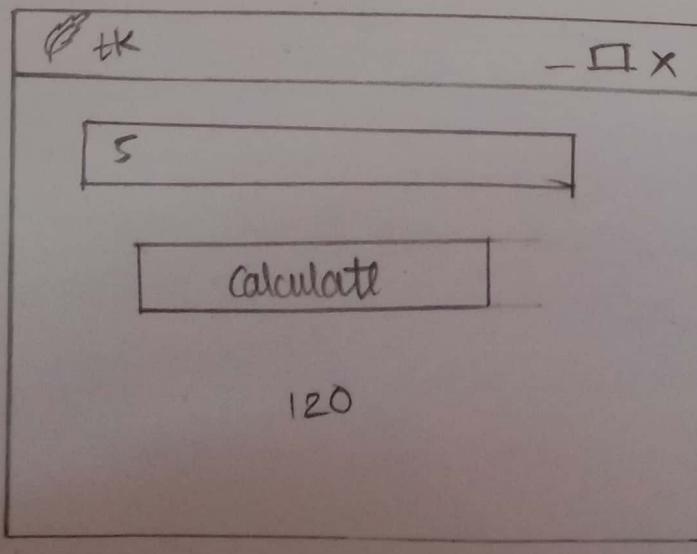


52

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()
btn = Button(root, text="calculate", command=calculate)
btn.pack()
info = Label(root, text="factorial")
info.pack()
root.mainloop()
```

OUTPUT:



PRACTICAL →

Aim : Write a number program to find factorial of number & use arithmetic operations on two numbers using GUI.

a) 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 using recursive function.

Step 3 : Define another function calculate to call factorial function.

Step 4 : Now create an object with entry() & 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()

2] WAP to perform arithmetic operation on 2 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 num1 & num2 and use grid() to place it onto parent windows.

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

#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)
    l3.grid(row = 4, column = 1)
root.mainloop()
root = Tk()

```

#OUTPUT:

```

l1 = Label(root, text = "Enter a number :")
l1.grid(row = 0, column = 0)
e1 = Entry(root)
e1.grid(row = 0, column = 1)
l2 = Label(root, text = "Enter 2nd numbers")
l2.grid(row = 1, column = 0)
e2 = Entry(root)
e2.grid(row = 1, column = 1)
v = IntVar()
r1 = Radiobutton(root, text = "odd", variable = v, value = 1)
r1.grid(row = 2, column = 0)

```

`a2 = Radiobutton (root, text = "sub", variable = v, value = 2)`

`a2.grid (row=2, column = 1)`

`a3 = Radiobutton (root, text = "Mult", variable = v, value = 3)`

`a3.grid (row=2, column = 2)`

`a4 = Radiobutton (root, text = "Div", variable = v, value = 4)`

`a4.grid (row=2, column = 3)`

`B = Button (root, text = "calculate", command = calculate)`

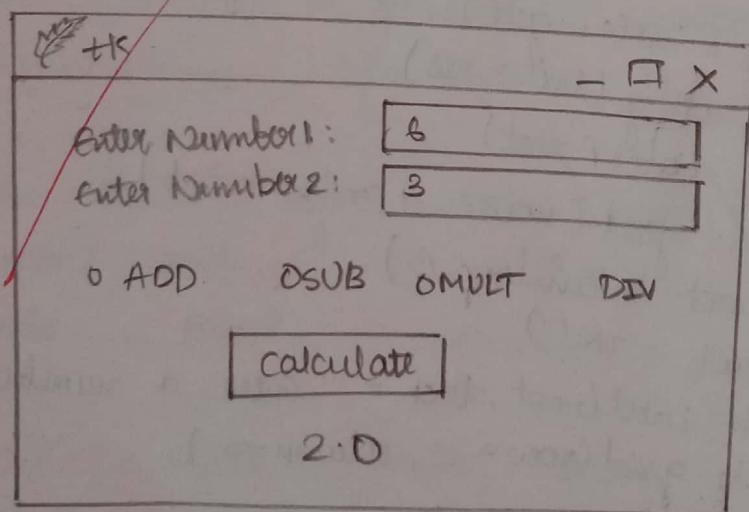
`B.grid (row=3, column = 1, columnspan = 2)`

`l3 = Label (root)`

`l3.grid (row=4, column = 1)`

`root.mainloop()`

#OUTPUT:



- Step 6 : Now initialize v as integer using AntVar()
- Step 7 : Now create & 4 objects with Radiobutton() to choose any one of arithmetic operations & use grid() for positioning onto parent window.
- Step 8 : Now create a object with button() along with command attribute to carry out the arithmetic operation of user's choice.
- Step 9 : Now create a object with label() to show output.
- Step 10 : Finally use the mainloop().

PRACTICAL - 8

AIM : Demonstrate the use of socket module and server client programs

Write a program to demonstrate use of socket module and server client programs.

Algorithm :

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

Step 2 - Define a function as server - program to get host name.

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 pair port together to many client simultaneously.

Step 6 - Now use accept() to accept new connection.

#CODE:

56

```
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()
    print("now right side client")
```

(Now run the program & now right side client program)

code

```
import socket  
def client_program():  
    host = socket.gethostname()  
    port = 5600  
    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()
```

OUTPUT for socket_program

```
$ python3.4 socket-server.py  
connection from: ('127.0.0.1', 57822)  
from connected user: flis  
→ Hello  
from connected user: Awesome!  
→ OK then, bye!
```

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()

Step 8 : Now print the address

Step 9 : Use while loop as True to receive data stream

Step 10 : Now close the program

(for socket client program)

Algorithm:

Step 1 - Import socket module to import methods that are relevant.

Step 2 - Define a function client_program get the host name & give port a value 5000.

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

Step 4 - Use connect() to connect the server.

Step 5 - Now take the input ("→")

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 now, bye !

→ Bye

```
# code
>>> import sqlite3
>>> conn = sqlite3.connect("student.db")
>>> cur = conn.cursor()
>>> cur.execute('Create table student (roll_no int(5) primary key, name varchar(50) not null, class varchar(50), dob date)')
<sqlite3.cursor object at 0x0322FBED>
>>> cur.execute('insert into student values (101, "Esha", "Kandivali", "FyCS", "13-08-2001")')
<sqlite3.cursor object at 0x0322FBED>
>>> cur.execute('insert into student values (102, "Nidhi", "Borivali", "FyCS", "11-04-2001")')
<sqlite3.cursor object at 0x0322FBED>
>>> cur.execute('select * from student')
<sqlite3.cursor object at 0x0322FBED>
>>> cur.fetchall()
[(101, 'Esha', 'Kandivali', 'FyCS', '13-08-2001'), (102, 'Nidhi', 'Borivali', 'FyCS', '11-04-2001')]
>>> cur.execute('update student set dob = "13/09/2002" where roll_no = 101')
<sqlite3.cursor object at 0x0322FBED>
>>> cur.execute('select * from student where address = "Kandivali"')
<sqlite3.cursor object at 0x0322FBED>
>>> cur.fetchall()
[(101, 'Esha', 'Kandivali', 'FyCS', '13-08-2001')]
>>> cur.execute('commit')
<sqlite3.cursor object at 0x0322FBED>
>>> cur.close()
```

PRACTICAL - 9

Aim : Demonstrate the use of database connectivity.

Algorithm :

Step 1 : Import sqlite3 module to import relevant methods.

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

Step 3 : Now initialize a variable to connect to cursor().

Step 4 : Now use cur.execute() to create a table, insert values into table & 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 changes.

~~Step 7 : use close() to terminate the program.~~

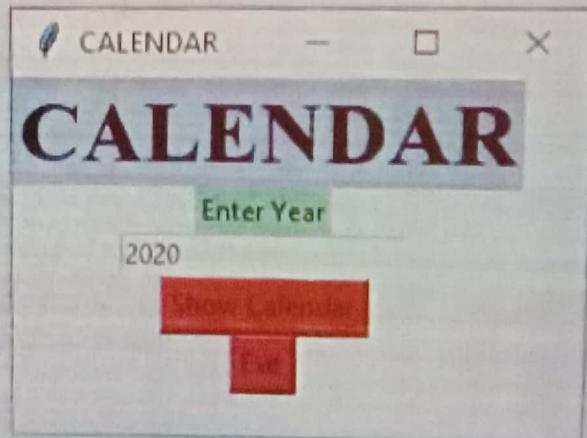
```

from tkinter import*
import calendar

new_gui=Tk()
new_gui.config(background='white')
new_gui.title("CALENDAR")
new_gui.geometry("550x600")
fetch_year=int(year_field.get())
cal_content=calendar.calendar(fetch_year)
cal_year=Label(new_gui,text=cal_content,font="Consolas 10 bold")
cal_year.grid(row=5,column=1,padx=20)
new_gui.mainloop()

if __name__ == "__main__":
    gui=Tk()
    gui.config(background="white")
    gui.title("CALENDAR")
    gui.geometry("250x140")
    cal=Label(gui,text="CALENDAR",bg="dark gray",font=("times",28,'bold'))
    year=Label(gui,text="Enter Year",bg="light green")
    year_field=Entry(gui)
    Show=Button(gui,text="Show Calendar",fg="Black",bg="Red",command=showCal)
    Exit=Button(gui,text="Exit",fg="Black",bg="Red",command=exit)
    cal.grid(row=1,column=1)
    year.grid(row=2,column=1)
    year_field.grid(row=3,column=1)
    Show.grid(row=4,column=1)
    Exit.grid(row=6,column=1)
    gui.mainloop()

```



calendar.py - C:/Users/Nidhi Waghela/calendar.py (3.7A)

File Edit Format Run Options Window Help

```
from tkinter import*
```

```
import calendar
```

```
def showCal():
```

```
    new_gui=Tk()
```

```
    new_gui.config(background='white')
```

```
    new_gui.title("CALENDAR")
```

```
    new_gui.geometry("550x600")
```

```
    fetch_year=int(year_field.get())
```

```
    cal_content=calendar.calendar(fetch_year)
```

```
    cal_year=Label(new_gui,text=cal_content,font="Consolas 10 bold")
```

```
    cal_year.grid(row=5,column=1,padx=20)
```

```
    new_gui.mainloop()
```

```
if __name__ == "__main__":
```

```
    gui=Tk()
```

```
    gui.config(background="white")
```

```
    gui.title("CALENDAR")
```

```
    gui.geometry("250x140")
```

```
    cal=Label(gui,text="CALENDAR",bg="dark gray",font=("times",28,'bold'))
```

```
    year=Label(gui,text="Enter Year",bg="light green")
```

```
    year_field=Entry(gui)
```

```
    Show=Button(gui,text="Show Calendar",fg="Black",bg="Red",command=showCal)
```

```
    Exit=Button(gui,text="Exit",fg="Black",bg="Red",command=exit)
```

```
    cal.grid(row=1,column=1)
```

```
    year.grid(row=2,column=1)
```

```
    year_field.grid(row=3,column=1)
```

```
    Show.grid(row=4,column=1)
```

```
    Exit.grid(row=6,column=1)
```

```
    gui.mainloop()
```

July
1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30

31

August

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

September

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

October

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

November

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

December

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

January

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

February

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

March

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

April

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

May

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

June

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

July

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

August

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

September

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

October

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

November

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

December

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

January

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

February

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

March

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

April

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

May

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

June

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

July
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

August
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

September
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

October
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

November
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

December
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

January
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

February
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

March
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

April
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

May
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

June
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

July
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

August
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

September
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

October
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

November
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

December
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

January
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

February
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

March
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

April
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

May
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

June
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23

3.7.1 (tags/v3.7.1:6093511, 2019-06-04) on win32
Help, "copyright", "credits" or "license()" for more information.

DESKTOP C:\Users\Nishi Hashmi\Downloads\calendar.exe

CALENDAR

2020

January

Mo	Tu	We	Th	Fr	Sa	Su
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

February

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	

March

Mo	Tu	We	Th	Fr	Sa	Su
					1	
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

April

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

May

Mo	Tu	We	Th	Fr	Sa	Su
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	
25	26	27	28	29	30	31

June

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

July

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

August

Mo	Tu	We	Th	Fr	Sa	Su
		1	2			
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

September

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

October

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4			
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

November

Mo	Tu	We	Th	Fr	Sa	Su
		1	2	3	4	5
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
					30	

December

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Type here to search

