

```

fileobj = open("abc.txt", "w")
fileobj.write("Dance tunes" + "\n")
fileobj.write("bangarang in revolution in getlow")
fileobj.close()

file = open("abc.txt", "r")
# read
str1 = fileobj.read()
print("The output of read method:", str1)
fileobj.close()

>>> ("The output of read method : ", 'Dance tunes\nbangarang\nrevolution in getlow\n')

# readline()
fileobj = open("abc.txt", "r")
str2 = fileobj.readline()
print("Output:", str2)
fileobj.close()

>>> ('Output:', 'Dance tunes\n')

# readlines()
fileobj = open("abc.txt", "r")
str3 = fileobj.readlines()
print("Output:", str3)
fileobj.close()

>>> ('Output:', ['Dance tunes\nbangarang\nrevolution in getlow'])

# 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')

```

119

### Practical No.1

21

Objective: Demonstrate the use of different file accessing modes different attributes read methods

Step 1: Create a file object using open method & use the write access mode followed 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() and store the output in variable and finally display the contents of variable.

Step 3: Now use the fileobject for finding the name of the file mode in which its opened whether the file is still open or close & finally the output of the softspace attribute.



15

Step 4: Now open the fileobj in write mode write some another content close subsequently. Then again open the fileobj in 'w+' mode that is the update mode & write contents.

Step 5: Open fileobj in read mode display the update written contents & close. Open again in 'r+' mode with parameter passed & display the output subsequently.

Step 6: Now open fileobj in append mode open, write method write contents close the fileobj again open the fileobj in read mode the display the appending output.

c = fileobj.mode  
print ("filemode ", c)

>>> ('file mode', 'r')

d = fileobj.softspace

print ("softspace : ", d)

>>> ('softspace : ', 0)

# w+ mode

fileobj = open ("abc.txt", "w+")

fileobj.write ("skiller")

fileobj.close()

# a+ mode

fileobj = open ("abc.txt", "a+")

s1 = fileobj.read (5)

print ("output of r+", s1)

fileobj.close()

>>> ('output of r+', 'skill')

# append mode

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

fileobj.write ("Ragga bomb")

fileobj.close()

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

s2 = fileobj.read ()

print ("Output of append mode : ", s2)

fileobj.close()

>>> ('Output of append mode : ', 'skiller Ragga bomb')

22

# write mode

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

fileobj.write ("Alpha")

fileobj.close()

# read mode

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

s3 = fileobj.read ()

print ("output of read mode : ", s3)

>>> ('output of read mode : ', 'Alpha')

```

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

>>> ('tell() : ', 0)

# seek()
fileobj = open('abc.txt', "r")
st = fileobj.seek(0, 0)
print ("seek(0,0) is : ", st)
fileobj.close()

>>> ('seek(0,0) is : ', None)

fileobj = open('abc.txt', "r")
pos1 = fileobj.seek(0, 1)
st1 = fileobj.readline(8)
print ("seek(0,1) is : ", st1)
fileobj.close()

>>> ('seek(0,1) is : ', 'stilllex')

# finding length of different lines exist
fileobj = open('abc.txt', "r")
stat = fileobj.readlines()
print ("output : ", stat)
for line in stat:
    print (len(line))
fileobj.close()

>>> ('output : ', [stilllex, Ragga Bomb])
>>> 9

```

Step 7: Open the fileobj in read mode, declare a variable & perform fileobject dot tellmethod & store the output consequently in variable.

Step 8: Use the seek method with the arguments with opening the file obj in read mode & closing subsequently.

Step 9: Open fileobj with read mode also use the readlines method & store the output consequently in st, print the same & for counting the length use the for conditional statement & display the length. Date: 12/12/19

## Practical 2

Objective : Iterators

Step1 : Create a tuple with element that we need to iterate using the `iter` and `next` method no. of the time we use the `iter` and `next` method we will get the next iterating element in the tuple. Display the same:

Step2 : The similar output can be obtained by using for conditional statement. An iterable variable is to be declared in for loop which will iterate. It is a function.

Step3 : Define a func name `square` with a parameter which will obtain output of square value of the given number - In similar fashion declare `cube` to get the value raised to 3 & return the same.

Step4 : Call the declared function using function call

```
# iterator & next()
mytuple = ("banana", "orange", "apple")
myiterator1 = iter(mytuple)
print(next(myiterator1))
myiterator2 = iter(mytuple)
print(next(myiterator2))
myiterator3 = iter(mytuple)
print(next(myiterator3))
>>> banana
orange
apple
```

```
# for loop
mytuple1 = ("Cole", "KJ Apa", "Charles")
for x in mytuple1:
    print(x)
```

```
>>> Cole
KJ Apa
Charles
```

```
# square & cube
def square(x):
    y = x * x
    return y
def cube(x):
    z = x * x * x
    return z
unit1 = [square, cube]
```

```

for r in range(s):
    value = list(map(lambda x: x**r, funct))
    print(value)

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

# map[]
list num = [0,4,5,7,9,11,13,15,20,19,25]
list num = [list(map(lambda x: x%5, list num))]
print(list num)

def even(x)
    if (x%2==0)
        return "Even"
    else:
        return "Odd"
list (map(even, list num))

# odd number
class odd:
    def __iter__(self):
        self.num=1
        return self
    def __next__(self):
        num=self.num
        num+=2
        self.num+=2
        return num

```

25

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

6: Declare a list num variable & declare some element then use the map method with help of lambda function give two argument display the output.

7: Define a function given with a parameter then using conditional statement do check whether the number is odd, even & return respectively.

8: Define a class and within that define the iter() method which will initialize the first element within the container object

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

```

myobj = odd()
myiter = iter(myobj)
x = int(input("Enter a number "))
for i in myiter
    if (i < x):
        print(i)
    
```

>> Enter a number : 15

13  
11  
9  
7  
5  
3  
1

J-T 2011-11

```
# arithmetic error
while True:
    try:
        x = int(input("Enter class"))
        break
    except ValueError:
        print("Enter numeric char:")

```

Output:

```
Enter class : fycs
Enter numeric char: 123
Enter class : 13
```

# environment error

```
try:
    f = open("abc.txt", "w")
    f.write("sakshi")
except IOError:
    print("error writing on the file")
else:
    print("operation carried out successful")
    f.close()
```

Output:

Operation carried out successful

### Practical 3

27

Aim: To demonstrate exception handling

Write a program using the exception method of the nature Arithmetic error

Step 1: Use the try block & except the input using the raw input method and converts it into the integer datatype & subsequently terminate the block.

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

2) Write a program for accepting the file in a given mode & use the environment error as an exception for the given input:

Step 1: Within the try block open the file using the write mode & write some content on the file.

Step 2: Use the except block with IO error & display the message regarding missing of the file or incompatibility of the mode use the else block to display to display a message that the operation is carried out successfully.

TS

3) WAP using the assert () to check if the last elements are empty.

S1: Define a function which accepts an argument & check using fix assert statement whether the given list is empty list & accordingly return the message.

S2: Close the func & in the body of program & define certain elements in the list & take some appropriate action.

4) WAP to check the range of the age of the students in given class & if the age do not fall in given range else raise the value error exception otherwise return the valid no.

Step3: 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 conditions.

4: Use except with valueError & print the message

Not a valid range

28

```
# assert error
def assert_(n)
    assert (len(n) == 0)
    print ("list is empty")
Var1 = []
print (assert_(Var1))

Output:
List is empty
None
```

1) def acceptage ():

```
age = int (input ("Enter your age :"))
if age >= 30 or age < 16 :
    raise ValueError
```

```
return age
```

```
valid = False
```

```
while not valid :
```

try :

```
    age = acceptage ()
    valid = True
```

```
except ValueError:
```

```
    print ("not valid age")
```

Output:

```
Enter your age : 18
```

```
#code
import re
string = "Hello 1234 abc 4567"
result = re.findall ('\d', string)
result = re.findall ('[d]', string)
print (result)
print (result)
```

```
# output
>>> ['1234', '4567']
>>> ['Hello', 'abc']
```

### Practical +

Aim: Demonstrate the use of regular expression

Theory: Regular expression represents the sequence of characters which is mainly used for finding & replacing the given pattern in a string & for this we import module re. It involves following functionalities.

Write a regular expression segregating numeric & alphabetic values from a given string.

Algorithm:

- 1: Now apply string & pattern in findall() & display the output & display the output
- 2: \d is used for matching all decimal digits whereas \D is used to match non decimal digits

15

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

# code 2

```
import re
string = "Python is an important language"
result = re.search('P', 'in Python', string)
print(result)
if result:
    print("match found")
else:
    print("match not found")
```

- 3) write a regular expression for finding the match string at the beginning of given sequence.

1: import re module & apply a string

2: use search() write 're python' & string as two parameters.

3: Now display the output

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

\* output  
=> re match object . span(0,6)  
match = 'Python'  
=> match not found.

code

```
108.  
import re  
li = ["9876543210", "8765432109", "76543210981", "65432109"]  
for elements in li:  
    result = re.match("[8-9]-[0-9][0-9] [0-9]", element)  
    if result:  
        print("correct mobile no.")  
        print(result.group(1))  
    else:  
        print("incorrect mobile no")
```

# output

```
>>> correct mobile no
```

```
98675543210
```

```
correct mobile no
```

```
87654632109
```

```
incorrect mobile no.
```

31

3) write a regular expression to check whether the given mobile number starts with 8 or 9 & the total length is 10.

-Algorithm:

1: Import the re module & apply a string of mobile no &

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

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

```
#1 code 4
import re
string = "Python is important"
result_1 = re.findall("\w*", string)
result_2 = re.findall("\w+", string)
print(result_1)
print(result_2)
```

- 4) write a regular expression for extracting a word from given string along with space characters in between the words & subsequently extract the word without space del character.

Algo:

- 1: Import re module & apply a string
- 2: Use findall() to extract word from given string
- 3: Use "\w\*" to extract word along with space
- 4: use "\w+" to extract word without space.

- 4: Now display the output

88

```
# code 5
import re
string = "Python is important"
result = re.findall ("n\w+", string)
result1 = re.findall ("\w+\$ ", string)
print (result)
print (result1)

# output
» ['Python']
» ['Important']
```

88

5) write a regular expression for extracting first or last word from a string

Algo

- 1: Import re module & apply a string
- 2: use.findall() in which use "\c\t" as one parameter to find first word of string then use "\w+" as parameter to find last word to string.
- 3: Display the output.

38

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

- Type  
1: Import re module & apply string  
2: Use.findall module & use \d{2} - \d{2} - \d{4} as a parameter  
3: Now display the output.

39

# code 6

```
import re
string = "Amit 201 204-12-2019"
result = re.findall (" \d{2} - \d{2} - \d{4} ", string)
print (result)
```

# output

```
24-12-2019
```

import re

```
string = "abc @ tesc. edu"
```

```
result1 = re.findall ("^\\w+", string)
```

```
result2 = re.findall ("^+\\w+ \\w+ $", string)
```

```
result3 = re.findall ("[\\w . -] +", string)
```

- 1) write a re for extracting the username from email id
- 2) Extract name & last name from email id
- 3) Extract last name

Ans :

1: Import re module & apply a string, last name & last as email id.

# output

```
print (result1)
print (result2)
print (result3)
```

3: Use "\w+" for username use "+\w+ \w+\\$" for both no

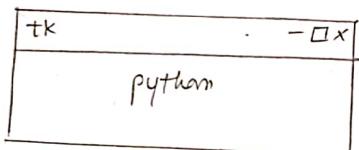
```
>>> [ 'abc',  
>>> [ 'tesc. edu',  
>>> [ 'abc', 'tesc. edu' ]
```

4: Display the output.

# creation of parent window

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

output:



#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)
```

## Practicals

36

### Part A

Topic: GUI components

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

#2

- 1: Use the Tkinter library for importing the features of the text widget
- 2: Create a variable from the text method & position it on the parent window.
- 3: Use the pack() along with the object created from the text() & use the parameter.
  - 1) side = LEFT, padx = 20  
 side = LEFT, pady = 30  
 side = TOP, ipadx = 40  
 side = TOP, ipady = 50

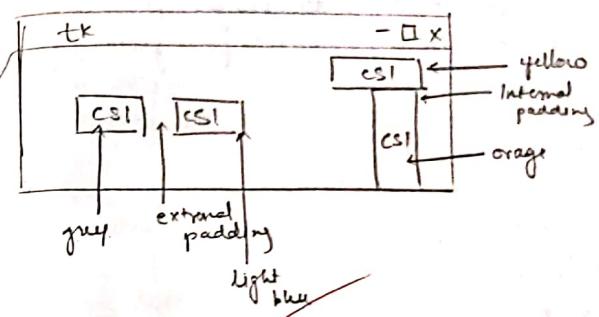
24. label (root, text = "csl", bg = "orange", fg = "black", font = "10")

37

14. pack (side = TOP, ipady = 50)

root.mainloop()

output:



```

#1:
# Radio button
from tkinter import *
root = Tk()
root.geometry("500x500")
def select():
    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, "List1")
l2 = insert(2, "List2")
l1.pack(anchor=N)
r1 = Radiobutton(root, text="option1", variable=var, value="option1", command=select)
r1.pack(anchor=N)
r2 = Radiobutton(root, text="option2", variable=var, value="option2", command=select)
r2.pack(anchor=N)
root.mainloop()

```

### Practical B

38

Aim: GUI components

#1:

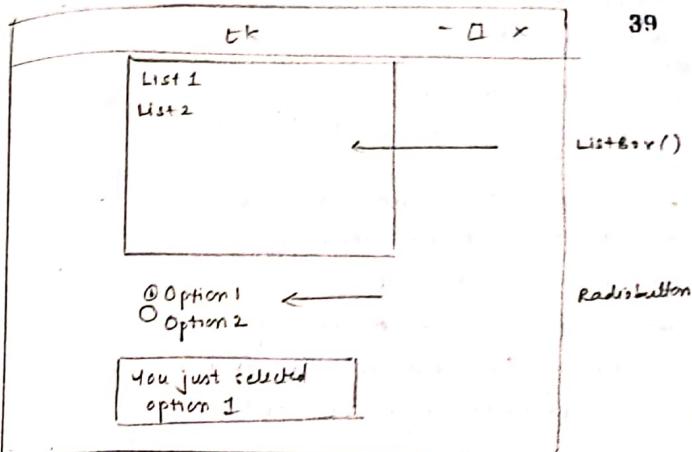
- 1: Import the relevant method from the tkinter library. Create an object with the parent window.
- 2: Use the parent window object along with the geometry() declaring specific pixel size of the parent window.
- 3: Now define a function which tells the user about the given selection mode from multiple option available.
- 4: Now define the parentwindow & define the option with control variable
- 5: Use the Listbox() & insert options on the parent window along with the pack() with specifying anchor attribute.
- 6: Create an object from radio button which will take following arguments a parent window object, text variable which will take the values option no 1, 2, 3 ... variable argument, corresponding value & triggers the function declared.
- 7: Now all the pack() for radio object so created & specify the argument using anchor attribute.
- 8: Finally make use of the mainloop() along with parent object

188

- #2

  - 1: Import relevant method from the tkinter library.
  - 2: Create a parent obj corresponding to the parent window.
  - 3: Create an object & use the scrollbar().
  - 4: Create an object & use the scrollbar().
  - 5: Use the pack() along with the scrollbar() obj with side & fill attributes.
  - 6: Use the mainloop with the parent obj.

## Output

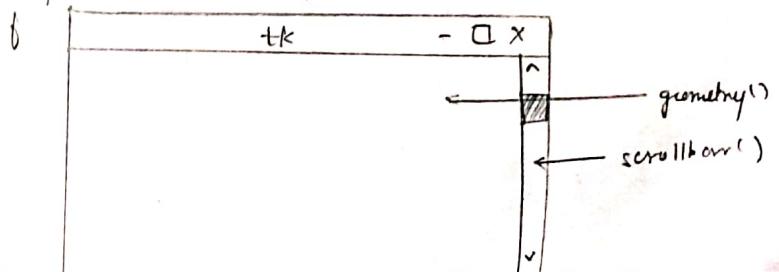


```

# 2
scrollbar()
from tkinter import *
root = Tk()
root.geometry("500x500")
s = scrollbar()
s.pack(side = "right", fill = "y")
root.mainloop()

```

Output:

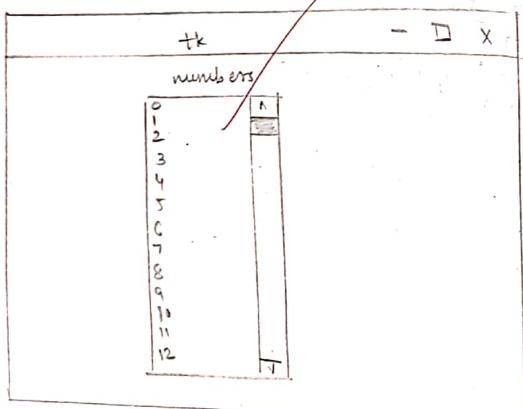


```

#3
#using frame widget
from tkinter import *
window = Tk()
window.geometry("650x500")
Label(window, text="numbers :").pack()
frame = Frame()
frame.pack()
listnodes = Listbox(frame, width=20, height=20,
                    font=("Times New Roman", 10))
listnodes.pack(side="left", fill="y")
scrollbar = scrollbar(frame, orient="vertical")
scrollbar.config(command=listnodes.yview)
scrollbar.pack(side="right", fill="y")
for x in range(100):
    listnodes.insert(END, str(x))
window.mainloop()

```

Output:



43

- 1: Import the relevant libraries from the tkinter method.
- 2: Create an corresponding object of the parent window
- 3: Use the geometry manager with pixel size (650x500) or any other suitable pixel value.
- 4: Use the label widget along with the parent object created & subsequently use the pack method.
- 5: Use the frame widget along with the parent object created and else the pack method.
- 6: Use the listbox method along with the attributes like width, height font. Do create a listbox method's object use pack() for the same.
- 7: Use the scrollbar() with an object use the attribute of vertical. Then configure the same with object created from the scrollbar() & use pack().
- 8: Trigger the events using mainloop.

40

#4:

1: Import relevant methods from tkinter library

```
114
from tkinter import *
window = Tk()
window.geometry ("680x500")
frame = Frame (window)
frame.pack()
```

2: Define the object corresponding to parent window & define the size of parent window in terms of no of pixels

& place it on the parent window

3: Now define the frame object from the method & place it on the parent window

```
leftframe = Frame (window)
leftframe . pack (side = "left")
rightframe = Frame (window)
rightframe . pack (side = "right")
```

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

```
b1 = Button (frame, text = "select", active background
= "red", fg = "blue")
```

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

```
b2 = Button (frame, text = "modify", active background
= "yellow", fg = "black")
```

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

```
b3 = Button (frame, text = "ADD", active background
= "red", fg = "green")
```

7: Similarly, create the button object corresponding to the modify operation put it into frame object on side "right".

```
b4 = Button (frame, text = "exit", active background
= "blue", fg = "red")
```

```
b1 . pack (side = "left", padx = 20)
```

```
b2 . pack (side = "right", padx = 30)
```

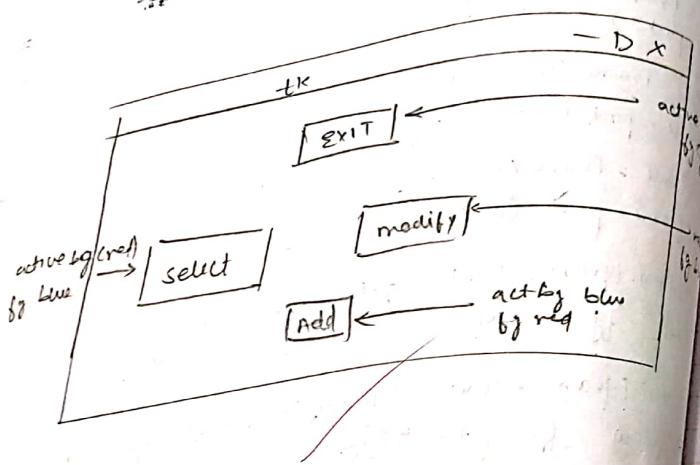
```
b3 . pack (side = "bottom", pady = 20)
```

```
b4 . pack (side = "top")
```

8: Create another button object & place it on the right frame & label the button as EXIT

9: Add another button & puts it on the top of frame

& label it as EXIT



10) use the pack() simultaneously for all the objects &  
finally use the mainloop().

Jmn

## Practical 6

Aim: Demonstrate the use of GUI by creating a human face & converting ellipses into function.

- Q1) write a program to draw human face using GUI.
- Algorithm:
  - 1: Import relevant methods from Tkinter library
  - 2: Create an obj corresponding to the parent window from tk()
  - 3: Create an object from canvas() & place into parent window along with height & width.
  - 4: Now use pack() for positng of widget onto parent window
  - 5: Now create object face & use object create\_oval() with co-ordinates 50,50,350,350 & outline = "black", fill = "yellow" as attribute to create face.
  - 6: Now create eye1 object & again use object create\_oval() with appropriate co-ordinates along with fill as attribute to create left eye.
  - 7: Now repeat same step 6 to create right eye.
  - 8: Create an object mouth & use object create\_oval() with appro. co-ordinates
  - 9: Finally use the mainloop()

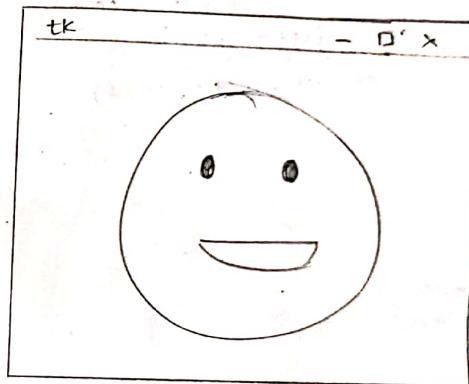
\* code :

```

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(225,125,275,175, fill="black")
mouth = c.create_oval(125,125,275,275, start=0, extent=180, width=5, fill="red")
root.mainloop()

```

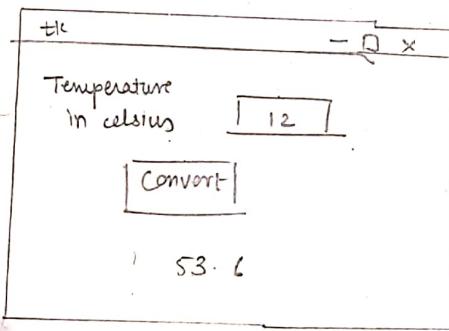
\* Output :



# code

```
from tkinter import *
window = Tk()
fahrenheit = DoubleVar()
fahrenheit.set(32.0)
def convert(celsius):
    fahrenheit.set((9.0/5.0)*celsius + 32)
l1 = Label(window, text="Temperature in celsius")
l1.grid(row=0, column=0)
e = Entry(window, textvariable=celsius)
e.grid(row=0, column=1)
celsius = IntVar()
l2 = Label(window, textvariable=fahrenheit)
l2.grid(row=2, column=0, columnspan=2)
B = Button(window, text="Calculate", command=lambda:
            convert(celsius.get()))
B.grid(row=1, column=0, columnspan=2)
mainloop()
```

# Output:



Q1) Write a prog to convert celsius into fahrenheit using GUI

Algorithm:

- 1: Import all the relevant methods in the tkinter library
- 2: Create object corresponding to the parent window from Tk()
- 3: Now initialize fahrenheit as DoubleVar() & set it to 32.0
- 4: Now define a function 'convert' with argument celsius to convert celsius into fahrenheit using .set()
- 5: Now create an object l2 using label() & place it onto parent window & use text attribute as enter a no.
- 6: Now use grid() for position the object onto the parent window
- 7: Initialize celsius as integer using IntVar()
- 8: Create another object & use entry widget to enter the i/p & place it into the parent window
- 9: Now use grid() for position the object into parent window with text variable attribute.
- 10: Now again use label() along with text variable using attribute to display output & use grid() for position.
- 11: Finally use mainloop():

```

from tkinter import *
import random
import sqlite3

def submit():
    def database():
        name1=str(name.get())
        contact1=int(con.get())
        email1=str(email.get())
        din=str(ds.get())
        dout=str(dc.get())
        nadult=int(s1.get())
        nchild=int(s2.get())
        room=str(c1.get())

        conn = sqlite3.connect('tajlands.db')
        with conn:
            cursor=conn.cursor()
            cursor.execute('CREATE TABLE IF NOT EXISTS "bookings" ("booking_no" INTEGER
NOT NULL PRIMARY KEY AUTOINCREMENT , "Name" TEXT, \
>Contact" INTEGER, "Email" TEXT, "Check_In" TEXT, "Check_Out" TEXT, "Adults"
INTEGER, "Children" INTEGER, "Room_Suite" TEXT)')
            cursor.execute('INSERT INTO bookings
(Name,Contact,Email,Check_In,Check_Out,Adults,Children,Room_Suite)
VALUES(?,?,?,?,?, ?, ?, ?)', \
                (name1,contact1,email1,din,dout,nadult,nchild,room))
        conn.commit()

        rec=cursor.execute('SELECT *FROM bookings')
        for row in rec:
            print(row)

        name.delete(0, END)
        con.delete(0, END)
        email.delete(0, END)
        ds.delete(0, END)
        dc.delete(0, END)
        s1.delete(0, END)
        s2.delete(0, END)
        c1.set('Select Your Room or Suite')

    def exit1():
        end.destroy()
        win.destroy()

    end=Tk()
    end.config(bg='#edb92b')
    end.geometry('+150+250')
    end.title("Confirmation !")
    end.maxsize(height=90, width=500)

```

1. *Antennulae* = 2 pairs of long setae, 1 pair of short setae.  
2. *Chelipeds* = 4 pairs of long setae, 1 pair of short setae.  
3. *Abdomen* = 1 pair of long setae, 1 pair of short setae.  
4. *Gills* = 1 pair of long setae, 1 pair of short setae.

W.H. = "3518151"

1988 (cont'd) 1989 (cont'd) 1990 (cont'd) 1991 (cont'd) 1992 (cont'd) 1993 (cont'd)

卷之三

1886 (1886), 1887 (1887), 1888 (1888), 1889 (1889)

*Georgian Gothic* and *Norman Gothic* were established, and "Gothic Revival" was

144801/64 08/02 "Map of East Asia" MURKIN MURKIN  
144802/64 08/02 "Map of Europe" MURKIN MURKIN

```

    Label(win, text=dout, width=20, bg='edb92b', font='Constantia 14
bold').grid(row=5, column=1, columnspan=2, padx=40, pady=10)

    Label(win, text="Number Of Adults", width=20, bg='edb92b', font='Constantia 14
bold').grid(row=6, column=0, padx=20, pady=10)
    Label(win, text=nadult, width=20, bg='edb92b', font='Constantia 14
bold').grid(row=6, column=1, columnspan=2, padx=40, pady=10)

    Label(win, text="Number Of Children", width=20, bg='edb92b', font='Constantia 14
bold').grid(row=7, column=0, padx=20, pady=10)
    Label(win, text=nchild, width=20, bg='edb92b', font='Constantia 14
bold').grid(row=7, column=1, columnspan=2, padx=40, pady=10)

    Label(win, text="Room & Suite", width=20, bg='edb92b', font='Constantia 14
bold').grid(row=8, column=0, padx=20, pady=10)
    Label(win, text=room, width=20, bg='edb92b', font='Constantia 14
bold').grid(row=8, column=1, columnspan=2, padx=40, pady=10)

    Button(win, text='Confirm', fg='white', command=database, width=10, bg='a87d07',
font='Constantia 14 bold').grid(row=9, column=2, padx=20, pady=20)

win.mainloop()

root = Tk()
root.config(bg='edb92b')
root.geometry('+100+20')
root.title('Taj Lands End')

Label(root, text="\nTAJ LANDS END MUMBAI\nBOOK A STAY\n", font="Constantia 21
bold", width=50, bg='white', fg='edb92b').grid(row=0, columnspan=3)

pic=PhotoImage(file='logo.png')
pic.subsample(4,4)
Label(root, image=pic, bg='white').place(x=10, y=10)

Label(root, text="Full Name", width=20, bg='edb92b', font='Constantia 14
bold').grid(row=1, column=0, padx=20, pady=10)
name = Entry(root, width=40, font='Arial 12 bold')
name.grid(row=1, column=1, columnspan=2, sticky=W, padx=40, pady=10)

Label(root, text="Contact", width=20, bg='edb92b', font='Constantia 14
bold').grid(row=2, column=0, padx=20, pady=10)
con = Entry(root, width=40, font='Arial 12 bold')
con.grid(row=2, column=1, columnspan=2, sticky=W, padx=40, pady=10)

Label(root, text="Email", width=20, bg='edb92b', font='Constantia 14
bold').grid(row=3, column=0, padx=20, pady=10)
email = Entry(root, width=40, font='Arial 12 bold')
email.grid(row=3, column=1, columnspan=2, sticky=W, padx=40, pady=10)

```

```

Label(root, text="Date Of Stay and Checkout",width=20,bg='#edb92b',font='Constantia
14 bold').grid(row=4, column=0, padx=20, pady=10)
ds=Entry(root,width=17, font='Arial 12 bold')
ds.grid(row=4, column=1, padx=40, pady=10, sticky=W)
Label(root, text="TO",bg='#edb92b',font='Constantia 14 bold').grid(row=4, column=1,
sticky=E)
dc=Entry(root, width=17, font='Arial 12 bold')
dc.grid(row=4, column=2, padx=40, pady=10, sticky=W)

Label(root, text="Number Of Adults",width=20,bg='#edb92b',font='Constantia 14
bold').grid(row=5, column=0, padx=20, pady=10)
s1=Spinbox(root, from_=0, to=10, width=20, font='Constantia 14 bold')
s1.grid(row=5, column=1, columnspan=2, padx=40, pady=10, sticky=W)

Label(root, text="Number Of Children",width=20,bg='#edb92b',font='Constantia 14
bold').grid(row=6, column=0, padx=20, pady=10)
s2=Spinbox(root, from_=0, to=10, width=20, font='Constantia 14 bold')
s2.grid(row=6, column=1, columnspan=2, padx=40, pady=10, sticky=W)

Label(root, text="Rooms and Suites",width=20,bg='#edb92b',font='Constantia 14
bold').grid(row=7, column=0, padx=20, pady=10)
list1 = ['Luxury Room','Delux Room','Serene Infinity Sea View','Grand Luxury
Room','Taj Club Room','Executive Suite Pool View King Bed','Luxury Suite Sea View','Grand Luxury
Suite Sea View','Presidential Suite'];
c1=StringVar()
droplist=OptionMenu(root,c1, *list1)
droplist.config(width=25,bg='white',activebackground='gold',font='Constantia 14
bold')
c1.set('Select Your Room or Suite')
droplist.grid(row=7, column=1, columnspan=2, padx=40, pady=10, sticky=W)

Button(root, text='Submit',fg='white',command=submit, width=10,bg='#a87d07',
font='Constantia 14 bold').grid(row=8, column=2, padx=20, pady=20)

root.mainloop()

```

```

Python 3.8.1 (tags/v3.8.1:0fb95c2, Mar 18 2019, 13:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:\PROJECT\PRINT\PRJ 5\PROJECT.py =====
(1, 'Shamika Chavan', 7788665544, 'shamikachavan', '16/03/2020', '20/03/2020', 3, 2, 'Taj Club Room')
(2, 'Alia Bhatt', 6655553311, 'aliaabhatt@gmail.com', '22/03/2020', '25/03/2020', 3, 0, 'Presidential Suite')
(3, 'Ranveer Singh', 7894561230, 'ranveersingh@gmail.com', '22/03/2020', '24/03/2020', 1, 0, 'Presidential Suite')
(4, 'Jasmeet Kaur', 6655442233, 'jasmeetkaur@gmail.com', '12/04/2020', '15/04/2020', 2, 0, 'Delux Room')
>>>

```

Taj Lands End

**TAJ LANDS END MUMBAI  
BOOK A STAY**

Full Name	Shamika Chavan		
Contact	7788665544		
Email	shamikachavan		
Date Of Stay and Checkout	16/03/2020	TO	20/03/2020
Number Of Adults	3		
Number Of Children	2		
Rooms and Suites	Taj Club Room		
<b>Submit</b>			



# TAJ LANDS END MUMBAI

## BOOK A STAY

**Name**

Shamika Chavan

**Contact**

7788665544

**Email**

shamikachavan

**Date of Check In**

16/03/2020

**Date of Check Out**

20/03/2020

**Number Of Adults**

3

**Number Of Children**

2

**Room & Suite**

Taj Club Room

**Confirm**

# TAJ LANDS END MUMBAI

## BOOK A STAY

**Name**

Shamika Chavan

**Contact**

7788665544

Confirmation!

Your Room&Suite is Successfully Booked !  
Enjoy your Luxurious Stay at Taj Lands End !

OK

**Date****Date of Check Out**

20/03/2020

**Number Of Adults**

3

**Number Of Children**

2

**Room & Suite**

Taj Club Room

**Confirm**