

Degree College
**Computer Journal
CERTIFICATE**

SEMESTER II UID No. _____

Class FYBSC-CS Roll No. 1861 Year 2020

This is to certify that the work entered in this journal
is the work of Mst. / Ms. Vaidhi Rajesh
Dahanu

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

MV 03/03/20
Teacher In-Charge

Head of Department

Date : _____

Examiner

No.

1

2

3

4

5

6

7

8

9

P

★ ★ INDEX ★ ★

| No. | Title | Page No. | Date | Staff Member's Signature |
|-----|--|----------|----------|--------------------------|
| 1. | Demonstrate the use of different file accessing mode, different attributes and read method | 29 | 05/12/17 | mm 24/12/17 |
| 2. | Demonstrate the use of iteration and iterators | 34 | 10/12/17 | |
| 3. | Demonstrate the use of exception handling | 38 | 17/12/17 | |
| 4. | Demonstrate the use of regular expression | 41 | 24/12/17 | mm 07/01/18 |
| 5. | GUI components (a) | 47 | | mm 05/03/18 |
| | GUI components (b) | 49 | | |
| | GUI components (c) | 53 | | |
| | GUI components (d) | 55 | | |
| | GUI components (e) | 57 | | |
| 6. | Creating a human face using GUI and conversion of Celsius to Fahrenheit. | 59 | | |
| 7. | Factorial of a number and arithmetic operations using GUI | 61 | | |

Practical 01

* Objective:- Demonstrate the use of different file accessing mode, different attributes of read method.

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 // 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.

Step 4:- Now open the file object in write mode write some another content close subsequent then again open the file object in 'wt' mode that is the update mode and write content.

File object = file object (f) = file object (mode)
 f = open("computer science subject.txt", "w")
 f.write("computer science subject")
 f.close()

f = open("computer science subject.txt", "r")
 str = f.read()
 print("The content of read method: ", str)

f.close()
 The content of read method: computer science
 subject
 subject
 subject

f = open("computer science subject.txt", "r")
 str = f.readline()
 print("The content of readline method: ", str)

The content of readline method: computer science subject

f = open("computer science subject.txt", "r")
 str = f.readlines()

print("The content of readlines method: ", str)

The content of readlines method: computer science
 subject
 subject
 subject

f.close()
 f = open("computer science subject.txt", "w")
 f.write("computer science subject")
 f.close()

Step 5 - Open file object in read mode, display the updated written contents and close.
 - Then again open the file object in 'a+' mode that is the update mode & write content with parameters passed & display the output subsequently.

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.

file attributes

a = file obj name
 print("Name of file (name attribute):", a)
 => Name of file (name of attribute, attribute)
 b = file obj. closed
 print("closed attribute:", b)
 => ("closed" attribute: ; True)

c = file obj. mode
 print("file mode:", c)
 => ("file mode", 'a+')
 d = file obj. softspace
 print("softspace:", d)
 => ("softspace:", 0)

write mode

file obj = open("abc.txt", "w")
 file obj. write("Saurabh")
 file obj. close()

write mode
 file obj = open("abc.txt", "w")
 file obj. write("Saurabh")
 file obj. close()

read mode

file obj = open("abc.txt", "r")
 str1 = file obj. read()
 print("output of read mode", str1)
 file obj. close()
 => ("output of read mode", 'Saurabh')

read mode
 file obj = open("abc.txt", "r")
 str2 = file obj. read()
 print("output of read mode", str2)
 => ("output of read mode", 'Saurabh')

Append mode

```
fileobj = open("abc.txt", "a")
fileobj.write("Data structure")
fileobj.close()
str1 = open("abc.txt", "r")
str3 = fileobj.read()
print("Output of append mode", str3)
fileobj.close()
```

⇒ ("Output of append mode", "Sumanth", "Data structure")

```
# 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)
str8 = fileobj.read()
print("The beginning of the line is =", str8)
```

Step 7:- Open the file object in read mode, declare a variable & assign it the tell method and store the output consequently in variable.

Step 8:- Use the seek method with the arguments with opening the file object in read mode & doing subsequently.

Step 9:- Open file object with read mode & use the readlines method & store the output consequently in list or the same for counting the length the for condition statement and display the length.

Q.16
Q.17:-

Ques - Demonstrate the use of Iterator & Iterator.

- Ans - In python, Iterator is an object which implements Iterator class which has 2 methods namely `--iter()` and `--next()`. List, tuple, dictionary & the set all naturally 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 and initialize the value and return that value.

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

```
4 code:
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
```

iterator:

class power:

def __iter__(self):

self = 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

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

Q2

Write a program using an iterator for calculating the power of a given no. for instance number entered is 2 then value calculated should be $1, 2, 2^2, 2^3, 2^4$.

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 & pass this object in the ite method.

Q1 Write a program using iterable concept to find factorial of 10 in range 1 to 10:

Algorithm -
 Step 1 - Define a iter() with argument & 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.

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

Q2 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 and return the value.

!! code:

```
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(self.f - 1, "!= ", fac)
        else:
            raise StopIteration
```

```
f = fact()
x = iter(f)
x = next()
1 != 1
x = next()
2 != 2
x = next()
3 != 6
```

code:

```
class mult:
    def __init__(self):
        self.m = 1
        self.n = 1
    def next(self):
        if self.m <= 10:
            num = self.m
            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()
```

```
2 * 3 = 6
```

```
>>> n.next()
```

```
2 * 4 = 8
```

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 & pass this object in the iter method.

Practical: 5

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

Q Write a program to check the range of the age of the students in given class and if age does not fall in given range, the ValueError exception otherwise return the valid no.

Algorithm -

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

Code :

```
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")
```

```
7/7 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
```

Not use

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: 17.2

Not a valid number! Try again.

>>> Enter a number: 12

Valid number.



Step 2:- Use if conditional to check whether the input age falls in range if 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 and terminate the looping condition.

Step 4:- Use except with ValueError and print the message, not a valid age.

Q2 Write a program to check whether the number is 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 to accept the input using input() and convert it into integer datatype and subsequently terminate the block.

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

Q5 Write a program to demonstrate use of zerodivision error.

Algorithm:-

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

Step 2:- Define a function with 2 parameters to divide the number given by user.

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

Step 4:- Use except with zerodivision error & 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)
```

```
    except:
```

```
        print("Error!")
```

```
    break
```

⇒ Enter first number: 1

⇒ Enter second number: 1

Division of 1 & 1 is 1

⇒ Enter first number: 1

⇒ Enter second number: 0

24/12/19 Error!

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']
```

Practical 4:-

41

Aim:- To 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. Common usage of regular expression involved 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.

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 "1A Python" and string as two parameters.

Step 3:- Now display the output.

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

CODE 2:-

```
import re
string = "Python is an important language"
result = re.search("1A Python", string)
print(result)
if result:
    print("Match found")
else:
    print("Match not found")
```

Output:-

```
>>> re match object : span = (0, 6) ;
>>> match found
```

Code 3:

```
import re
li = ["9876543210", "8765432109",
      "7654321098", "6543210987"]
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.
```

43

Q5] Write a regular expression to check whether the given mobile number starts with 8 or 9 & the total length of digit should be atleast 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 be 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() to display the output and if we have use & don't display incorrect mobile

Q1) Write a regular expression for extracting a word from given string along with space character in this the word and subsequently extract the word without space characters.

Algorithm:-

Step 1:- Import re module and apply a string.

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

Step 3:- Use ".*" to extract word along with space and use "word" to extract word without space.

Step 4:- Now display the output.

// CODE 4:-

```
import re
string = "Python is important"
result1 = re.findall(".*", string)
result2 = re.findall("word", string)
print(result1)
print(result2)
```

// output:-

⇒ ['Python', ' ', 'is', ' ', 'important']

✓ ['Python', 'is', 'important']

CODE 5 -

```
import re
string = "python is important"
result = re.findall('!wt', string)
result = re.findall('!wt$', string)
print(result)
print(len(result))
```

Output:-

```
>>> ['python']
>>> ['!important']
```

CODE 6 -

```
import re
string = "Amit 201 24-12-2019"
result = re.findall('!d ?23-!d?23-!d?4')
print(result)
```

Output:-

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

45

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

Algorithm:-

Step 1:- Import re module and apply a string.
Step 2:- Use findall() in which use "!"wt" as one parameter to find first word of string, then use "!"wt\$" as parameter to find last word of string.

Step 3:- Now display the result.

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

Algorithm:-

Step 1:- Import re module and apply string.

Step 2:- Use findall method and use '!d ?23-!d?23-!d?4' as a parameter.

Step 3:- Now display the output.

Q7 write a re for extracting the ① username from email, ② hostname from email & ③ both username & hostname from email id.

Algorithm:-

Step 1:- Import re module and apply a string.

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

Step 3:- Use "^[wt]" for username. Use "[wt].[wt]" for hostname and use "[wt.-]+" for both as parameter in findall().

Step 4:- Display the output.

CODE 7 :-

```
import re
string = "abc@tesc.edu"
result1 = re.findall("^[wt]", string)
result2 = re.findall("[wt].[wt]", string)
result3 = re.findall("[wt.-]+", string)
print(result1)
print(result2)
print(result3)
```

Output:-

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

m
07/10/20

Creation of parent window

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

Output:-



Label attributes

```
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 = "DS", bg = "lightblue", fg = "black", font = "10")
l2.pack(side = LEFT, padx = 20)
l3 = Label(root, text = "GT", bg = "yellow", fg = "black", font = "10")
l3.pack(side = TOP, pady = 40)
```

Practical-5

47

Topic:- GUI components.

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

Step:- Create an object using the Tk()

Step:- Create a variable using the widget label and use the text method.

Step:- Use the mainloop() for triggering of the corresponding above mentioned events.

2:-

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

Step:- Create a variable from the text method and position it on the parent window.

13

Step 1: Use the pack() along with the object created from the text() and use the constructor.

1. side = LEFT, text = 20
 2. side = LEFT, text = 50
 3. side = TOP, text = 40
 4. side = TOP, text = 50

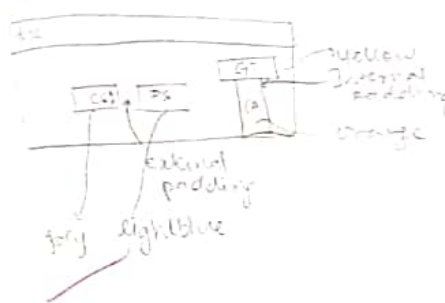
Step 2: Use the mainloop() for the triggering of the corresponding event.

Step 3: Now create above style with the label() which takes the following arguments:

- 1. Name of the parent window
- 2. Text attribute which defines the string.
- 3. The background color (bg)
- 4. The foreground color (fg) and you can use the pack() with a relevant padding attributes.

1. label (root, text = "CA", bg = "orange", fg = "black", font = "10")
 2. pack (side = TOP, pady = 50)
 3. mainloop()

Output:-

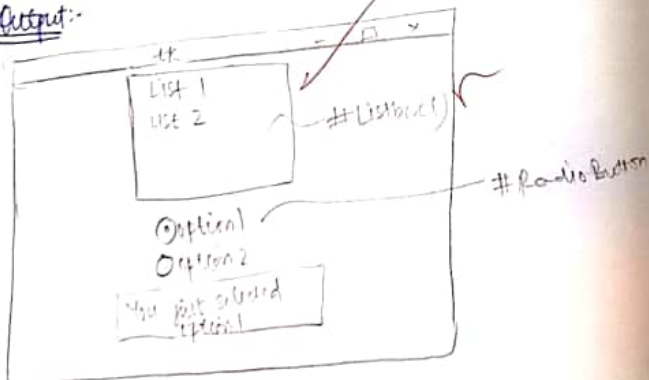


```

#1:-
#Radio button:-
from tkinter import *
root = Tk()
root.geometry("500x500")
def select():
    selection = "You just selected" + str(var1.get())
    t1 = Label(text=selection, bg="white", fg="green")
    t1.pack(side=TOP)
var1 = StringVar()
l1 = Listbox()
l1.insert(1, "List 1")
l1.insert(2, "List 2")
l1.pack(anchor=N)
r1 = Radiobutton(text="option 1", variable=var1, value="option 1",
                 command=select)
r2 = Radiobutton(text="option 2", variable=var1, value="option 2",
                 command=select)
r1.pack(anchor=N)
r2.pack(anchor=N)
root.mainloop()

```

#Output:-



Practical - 5CB

49

Aim: GUI components

1:-

- Step 1:- Import the relevant methods from the tkinter library. Create an object with the parent window.
- Step 2:- Use the parent window object along with the geometry() declaring specific pixel size of the parent window.
- Step 3:- Now define a fn which tells the user about the given selection made from multiple option available.
- Step 4:- Now define the parent window and define the option with control variable.
- Step 5:- Use the listbox() and insert options on the parent window along with the pack() with specifying anchor attribute.
- Step 6:- Create an object from radiobutton which may take following arguments (parent window object, text variable which will take the value option 1, 2, 3, ... variable arguments, corresponding value and trigger the function declared).
- Step 7:- Now call the pack() for radio object so created and specify the argument using anchor attribute.
- Step 8:- Finally make use of the mainloop() along with parent object.

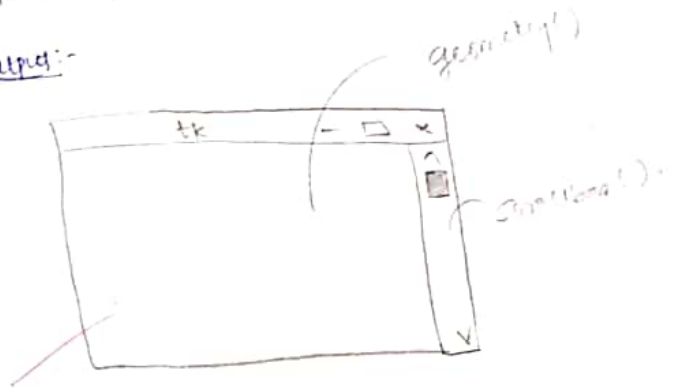
#1:-

- Step 1: Import relevant methods from the tkinter library.
- Step 2: Create a parent object corresponding to the parent window.
- Step 3: Create an object and use the scrollbar().
- Step 4: Use the geometry() for laying of the window.
- Step 5: Use the pack() along with the scrollbar object with side and fill attributes.
- Step 6: Use the mainloop with the parent object.

#2:-
#Scrollbar:-

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

output:-

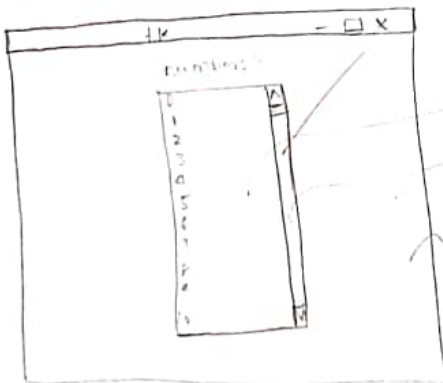


#3:-

widget 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", 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:-



listbox
scrollbar
parent window

51

#3:-

- Step 1:- Import the relevant libraries from the tkinter method.
- Step 2:- Create a 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 and subsequently use the pack method.
- Step 5:- Use the frame widget along with the parent object created and subsequently use the pack method.
- Step 6:- Use the listbox method along with the attributes like width, height, font. Do create a listbox method's obj. Use pack () for same.
- Step 7:- Use the scrollbar () with an object. Use the attribute vertical, then configure the same with object created from the scrollbar () and use pack () .
- Step 8:- Trigger the events using mainloop () .

14. 12
- 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 termed from the method and place it on 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 button object placed onto the given frame with the attribute as text, active background and foreground.
- Step 6: Now 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 with object & place on to the RIGHT frame & label the button as ADD.
- Step 9: Add another button & put it on the top of frame and label it as EXIT.
- Step 10: Use the pack() simultaneously for all the objects & finally use the mainloop().

14. 52

4:-

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

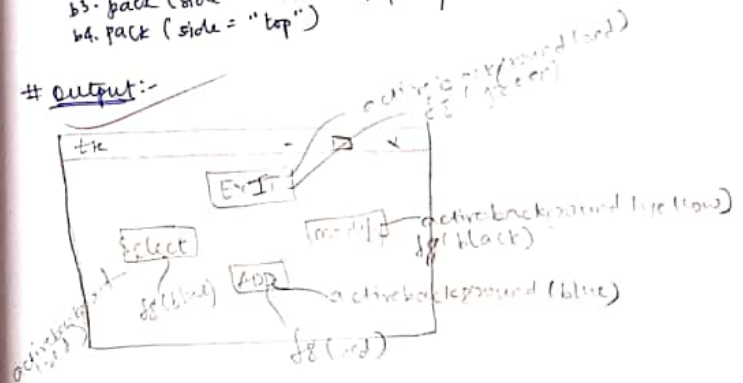
leftframe = Frame(window)
leftframe.pack(side="left")

rightframe = Frame(window)
rightframe.pack(side="right")

b1 = Button(frame, text="select", activebackground="red", fg="blue")
b2 = Button(frame, text="modify", activebackground="yellow", fg="black")
b3 = Button(frame, text="ADD", activebackground="blue", fg="red")
b4 = Button(frame, text="EXIT", activebackground="red", fg="green")

b1.pack(side="LEFT", padx=20)
b2.pack(side="RIGHT", padx=30)
b3.pack(side="bottom", pady=20)
b4.pack(side="top")
```

Output:-



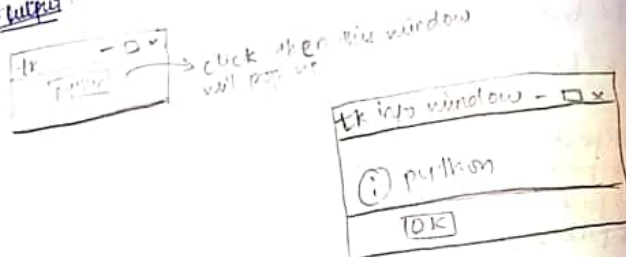
```

# message box
from tkinter import *
import tkinter.messagebox
root = Tk()

def function():
    tkinter.messagebox.showinfo("info window", "python")
    b1 = Button(root, text = "python", command = function)
    b1.pack()
    root.mainloop()

```

Output :-



Practical-5(C)

53

Aim - GUI components.

Step 1 - Import the relevant methods from tkinter library.

Step 2 - Import tkinter.messagebox.

Step 3 - Define a parent window object along with the parent window.

Step 4 - Define a function which will use tkinter.messagebox with showinfo method along with info window attribute.

Step 5 - Declare a button with parent window object along with the command attribute.

Step 6 - Place the button widget onto the parent window and finally call the mainloop() method for triggering the events mentioned above.

Step 1 Import the relevant methods from the tkinter library along with parent window object declared.

Step 2 Use parentwindow object alongwith minsize function for window size.

Step 3 Define a function main, declare parent window object and use config(), title(), minsize(), label() as well as button() and use pack() & mainloop() simultaneously.

Step 4 Similarly, define the function second and use the attributes accordingly.

Step 5 Declare another function button along with parent object and declare button with attributes like PLAT, RISE, GLOWE, RAISED, SUNKEN along with the relief widget.

Step 6 Finally call the mainloop() for event driven programming.

Multiple window
Efficient button (Relief()).

from tkinter import *
root = Tk()
root.minsize(300,300)

def main():

top = Tk()
top.config(bg="black")
top.title("HOME")
top.minsize(300,300)

L = Label(top, text="SAN FRANCISCO \n Places of Interest:
 \n Golden Gate Bridge \n Lombard street \n
 ChinaTown \n Leet Tower")

L.pack()

b1 = Button(top, text="next", command=second)

b1.pack(side=RIGHT)

b2 = Button(top, text="exit", command=terminate)

b2.pack(side=LEFT)

top.mainloop()

def second():

top2 = Tk()

top2.config(bg="orange")

top2.title("Moo ut us!")

top2.minsize(300,300)

L = Label(top2, text="Created by: Richa Mishra \n For
more details contact to our official account")

L.pack()

b3 = Button(top2, text="prev", command=main)

b3.pack(side=LEFT)

b2 = Button(top2, text="exit", command=terminate)

b2.pack(side=RIGHT)

top2.mainloop()

```

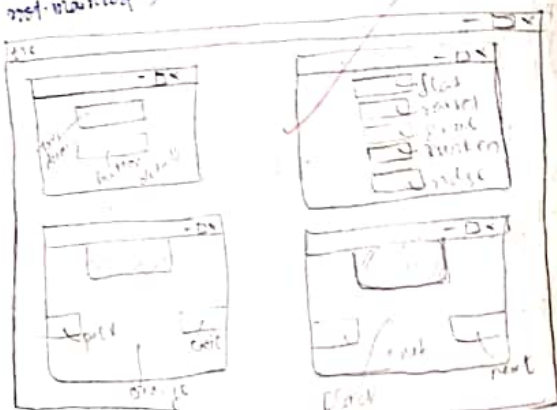
def button():
    top3 = Tk()
    top3.geometry("400x300")
    b1 = Button(top3, text="flat button", relief=FLAT)
    b1.pack()
    b2 = Button(top3, text="groove button", relief=GROOVE)
    b2.pack()
    b3 = Button(top3, text="raised button", relief=RAISED)
    b3.pack()
    b4 = Button(top3, text="sunken button", relief=SUNKEN)
    b4.pack()
    b5 = Button(top3, text="ridge button", relief=RIDGE)
    b5.pack()
    top3.mainloop()

```

```

def terminate():
    quit()
    b6 = Button(top, text="TOUR DETAILS", command=main)
    b6.pack()
    b7 = Button(top, text="BUTTON DETAILS", command=button)
    b7.pack()
    not.mainloop()

```



Practical - SCD

55

Aim GUI components.

Step 1 - Import relevant methods from the tkinter library.

Step 2 - Create parent window object and use the config method along with background color attribute specified.

Step 3 - Define a function finish with the messagebox widget which will display a message box i.e. a warning message & subsequently terminate the program.

Step 4 - Define a function intro. Use a listbox widget along with the object of the same. Use the listbox object along with insert method and insert the same and finally use the grid() with ipadx attributes.

Step 5 - Define a function about us with label widget and text attribute and subsequently use the grid()

Step 6 - Use photoImage widget with file and filename with gif attribute.

Step 7 - Create a frame object along with the frame() along with parent window object height & width specified & subsequently use the grid() with row & column attribute specified.

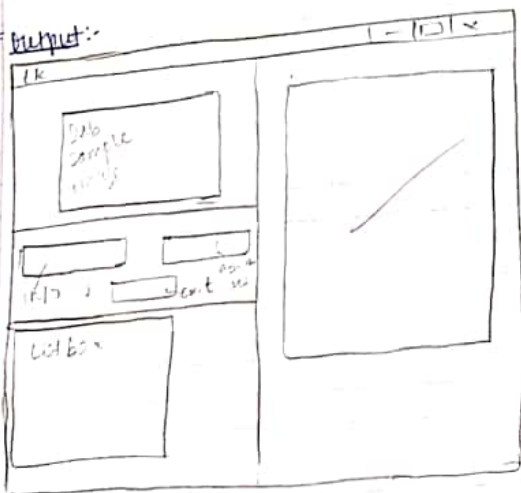
Step 8 - Similarly create another frame object as declared by step 7.

Step 9 - Create another object and use the subclass (S/Y)

Expt 1:- Use label widget along with the frame object, relief attribute and subsequently use the grid().

Expt 2:- Now create button object dealing with different sections of frame.

* Output :-

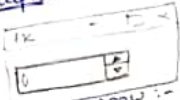


```
from tkinter import *
root = Tk()
root.config(bg="grey")
def finish():
    messagebox.askokcancel("Warning", "This will end the program")
    quit()
def info():
    list1 = Listbox()
    list1.insert(1, "Co. Name = Apple")
    list1.insert(2, "Products = iPhone")
    list1.insert(3, "Language = Swift")
    list1.insert(4, "OS = IOS")
    list1.grid(ipadx=50)
def about_us():
    list2 = Label(text="About us")
    list2.grid(ipadx=30)
    list3 = Label(text="Steve Jobs theatre March 2020")
    list3.grid(ipadx=24)
p1 = PhotoImage(file="download.gif")
f1 = Frame(root, height=55, width=5)
f1.grid(row=1, column=0)
f2 = Frame(root, height=250, width=500)
f2.grid(row=1, column=1)
p2 = p1.subsample(5, 4)
l1 = Label(f1, image=p2, relief=FLAT)
l1.grid(row=1, column=0, padx=20, pady=15)
l2 = Label(f2, image=p1, relief=SUNKEN)
l2.grid(padx=25, pady=10)
b1 = Button(f1, text="Information", relief=SUNKEN, command=info)
b1.grid(row=1, column=0)
b2 = Button(f1, text="About us", relief=SUNKEN, command=about_us)
b2.grid(row=1, column=2, padx=5)
b3 = Button(f1, text="Exit", relief=RAISED, command=finish)
b3.grid(row=2, column=1, ipadx=15)
root.mainloop()
```


Spinbox

```
from tkinter import *
master = Tk()
s = Spinbox(master, from_ = 0, to = 10)
s.pack()
master.mainloop()
```

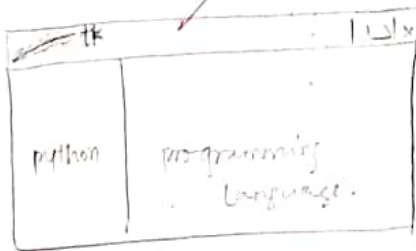
Output :-



PANELED WINDOW :-

```
from tkinter import *
root = Tk()
p = PanedWindow()
p.pack(fill=BOTH, expand=10)
l = Label(p, text="PYTHON")
p.add(l)
p1 = PanedWindow(p, orient=VERTICAL, bg="black")
p1.add(p1)
l1 = Label(p1, text="Programming language")
p1.add(l1)
root.mainloop()
```

Output :-



Practical - 5CE)

57

Aim: GUI Components.

Spinbox :

- Step 1:- Import relevant ()'s from the tkinter library.
- Step 2:- Create parent window object along with Tk()
- Step 3:- Create an object from spinbox method and use attributes pass parent window object, from - and to attributes.
- Step 4:- Subsequently call the pack method along with spinbox object & call the mainloop().

Paned Window:

- Step 1:- Import relevant methods from the tkinter library Also create a parent window object.
- Step 2:- Create an object along with paned window and subsequently use the pack() along with the paned window object along with attributes like fill & expand.
- Step 3:- Create a label object along with Label(). Use paned window object, orient & background color.
- Step 4:- Similarly create another label method and use the add subsequently.
- Step 5:- Finally call the mainloop() for event triggeration.

#3: canvas:

Step 1: Import relevant (i.e. from tkinter library) declare a parent window object.

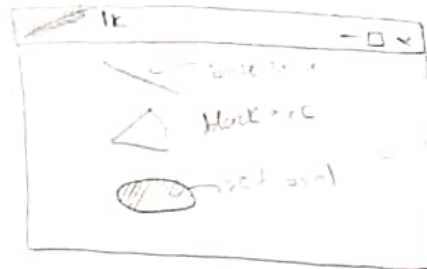
Step 2: Create a canvas object along with the canvas() with attributes parent window object, height, width, background color.

Step 3: Use create_arc() along with canvas object declared along with start & extent & co-ordinates.

Step 4: Similarly for oval & line use the pack() and call mainloop() for event driven programming.

#2: tkinter import

```
root = Tk()
c = Canvas (root, height = 100, width = 200, bg = "orange")
arc = c.create_arc (10, 70, 30, 40, start = 10, extent = 50, fill = "black")
oval = c.create_oval (20, 31, 45, 50, fill = "red")
line = c.create_line (10, 15, 30, 70, fill = "blue")
c.pack()
root.mainloop()
```



Canvas

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")
left_eye = c.create_oval(125, 125, 175, 175, fill="black")
right_eye = c.create_oval(325, 125, 375, 175, fill="black")
mouth = c.create_arc(175, 325, 325, 375, start=0, extent=180, width=5, fill="red")
root.mainloop()
```

Output -



Project 6

59

Aim - Demonstrate the use of GUI by creating a human face and converting Celsius into Fahrenheit.

Step 1 - 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() & place onto parent window along with height & width.

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

Step 5 - Now create an object face & use object ~~create~~ create_oval() with co-ordinates 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 same step 6 to create right eye.

Step 8 - Create an object mouth & use 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 mainloop()

Q2

Write a program to convert Celsius into Fahrenheit using GUI.

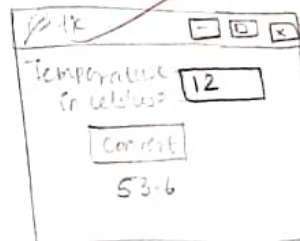
Algorithm:-

- Step 1: Import all the relevant methods in the 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 f2 using label() & place it onto parent window & use text attribute as enter a no.
- 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 for use entry widget to enter the input & 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 text variable using attribute to display output & use grid() for positioning.
- Step 11: Finally use mainloop().

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)
c = Entry(window, textvariable=celsius)
c.grid(row=0, column=1)
celsius = IntVar()
f2 = Label(window, textvariable=fahrenheit)
f2.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()
window
```

Output:-



```

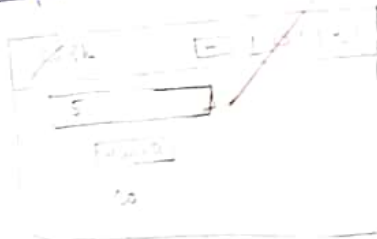
# code :-
import tkinter
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n-1)

def calculate():
    result = factorial(int(entry_text.get()))
    info.config(text = result)

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

```

Output :-



Practical 7

61

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

Ans:- 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().

Write a program to perform arithmetic operation on 2 no. of

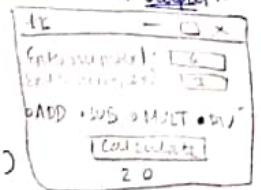
Algorithm -

1. Import relevant methods from tkinter library.
2. Now create an object corresponding to parent window.
3. Now define a function 'calculate' to carry out arithmetic operation on 2 numbers.
4. Now create object with label() as num1 & num2 and use grid() to place it onto parent window.
5. Create object with entry() to take inputs from user().
6. Now initialize v as integer using IntVar()
7. Now create 4 objects with Radio Button() to choose any one of arithmetic operation & use grid() for positioning onto parent window.
8. Now create an object with button() along with command attribute to carry out the arithmetic operation by way of v.
9. Now create an object with label() to show output.
10. Finally use the mainloop().

Code -

Code :-

```
from tkinter import *
def calculate():
    if int(v.get()) == 1:
        res = int(e1.get()) + int(e2.get())
    elif int(v.get()) == 2:
        res = int(e1.get()) - int(e2.get())
    elif int(v.get()) == 3:
        res = int(e1.get()) * int(e2.get())
    elif int(v.get()) == 4:
        res = int(e1.get()) / int(e2.get())
    else:
        res = int(e1.get()) / int(e2.get())
    l3.config(text=res)
root = Tk()
f1 = Label(root, text="Enter a no.")
f1.grid(row=0, column=0)
e1 = Entry(root)
e1.grid(row=0, column=1)
f2 = Label(root, text="Enter 2nd no.")
f2.grid(row=1, column=0)
e2 = Entry(root)
e2.grid(row=1, column=1)
v = 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)
b1 = Button(root, text="Calculate", command=calculate)
b1.grid(row=3, column=1, columnspan=2)
l3 = Label(root)
l3.grid(row=4, column=1)
root.mainloop()
```



#1 code:-

```
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()
```

Output:-

```
$python 3.6 socket - server.py
connection from: ('127.0.0.1', 57822)
from connected user: Hi
→ Hello
from connected user: Awesome!
→ ok user, bye!
```

Practical-8

63

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

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

Algorithm:-

- Step1:- Import the socket module to import relevant methods.
- Step2:- Define a fn as server-program to get hostname.
- Step3:- Now get value for port variable to initialize port no. about 1024.
- Step4:- Use .socket() to get instance.
- Step5:- Now use bind() function to bind host address and port together to configure how many client the server can list simultaneously.
- Step6:- Now use accept() to accept new connection.
- Step7:- Now print the address.
- Step8:- Use while loop as True to receive data stream.
- Step9:- Now close the program.

Handwritten:

Ex: Import socket module to import methods that are related

Ex: Write a function client_program() get the hostname & give print a value.

Ex: Now again initiate by using socket.socket().

Ex: We want to connect the server.

Ex: Now we use input() →

Ex: We will add additional msg to send a message.

Ex: Now we decide to receive response.

Ex: Now show the data.

Ex: Close the input.

Ex: Close the program by using close().

Code:

```
import socket
client = program()
host = socket.gethostname()
port = 5100

client_socket = socket.socket()
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:-

```
$ python s-6 socket-client.py
→ hi
Received from server: Hello
→ How are you?
Received from server: Good
→ Hrushee!
Received from server: Ok then, Bye!
→ Bye
```

code:-

```

>>> 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, dob date)')
>>> cur.execute('insert into student values(
              101, "Rishi", "19/08/2001")')
>>> cur.execute('insert into student values(
              102, "Sohu", "13/09/2001")')
>>> cur.execute('insert into student values(
              103, "Kandivali", "13/09/2001")')
>>> cur.execute('select * from student')
>>> cur.fetchall()
[(101, 'Rishi', '19/08/2001'),
 (102, 'Sohu', '13/09/2001'),
 (103, 'Kandivali', '13/09/2001')]
>>> cur.execute('update student set dob = "14/09/2002"
              where roll-no = 101')
>>> cur.execute('select * from student
              where address = "Kandivali"')
>>> cur.fetchall()
[(103, 'Sohu', 'Kandivali', '13/09/2001')]
>>> cur.execute('commit')
>>> conn.close()

```

Practical-1

65

1. 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 statement 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.

13/09/2001