

```

file obj = open("abc.txt", "w")
file obj . write ("computer science subjects" + "\n")
file obj . write ("DBMS in python\n")
file obj . close()

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

file obj = open("abc.txt", "r")
str2 = file obj . read_line()
print ("The output of read_line method : ", str2)
file obj . close()

file obj = open("abc.txt", "r")
str3 = file obj . reading()
print ("The output of reading method : ", str3)

a = file obj . name
print ("Name of file (name attribute) : ", a)

b = file obj . closed
print ("attribute : ", b)

```

PRACTICAL NO-1

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

Step 1: Create a file objective using open methods and use the write access mode followed 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() and readlines() and store the output in variables and finally display the content of variable.

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

Step 4:- Now open the file obj in write mode save writer content class subsequently then again open the file obj in ('wt') mode that is the update mode and with write content.

Step 5:- Then file obj in read mode display the update written contents and close open again in ('rt') mode with parameter passed and display the output subsequently.

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

```
c = file obj - mode  
fobj("file mode", c)
```

```
d = file obj - separator  
print ("sept separator", d)
```

```
fobj = open('abc.txt', "wt")  
file - obj - write ("lastin in")  
file obj - close()
```

```
fobj = open("abc.txt", "rt")  
S1 = file obj - read (6)  
print ("output of rt", S1)  
file obj - close()
```

```
fobj = open ("abc.txt", "a")  
file obj - write ("data structure")  
file obj = open ("abc.txt", "r")  
S3 = file obj - read()  
print ("output of append mode", S3)  
S = file obj - close()
```

```

fileobj = open("abc.txt", "r")
pos = fileobj.readline()
print(pos)
fileobj.close()

fileobj = open("abc.txt", "r")
st1 = fileobj.readline(0, 0)
print(st1[0, 0])
fileobj.close()

fileobj = open("abc.txt", "r")
st1 = fileobj.readline(0, 1)
print(st1[0, 1])
fileobj.close()

fileobj = open("abc.txt", "r")
st1 = fileobj.readline(0, 2)
print(st1[0, 2])
fileobj.close()

fileobj = open("abc.txt", "r")
st1 = fileobj.readline(0, 3)
print(st1[0, 3])
fileobj.close()
    
```

- Step 7:- Open file in read mode, declare a variable and perform file object tell method and store the output consequently in variable.
- Step 8:- Use the seek method with the assignments with opening the file object in read mode and closing subsequently.
- Step 9:- Open file object with readmode also use the read lines() method and store the output consequently in and print the same for counting the length use the for conditional statement and display the length.

Date
Topic

Q8

```
# iter() and next()
my tuple1 = iter((my tuple))
print(next(my tuple1))
my tuple2 = iter((my tuple))
print(next(my tuple2))
my tuple3 = iter((my tuple))
print(next(my tuple3))

Output
>> banana
orange
apple

# for loop
my tuple = ("Kevin", "Steve", "Jack")
for x in my tuple:
    print(x)
>> Kevin
Steve
Jack

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

Q9

PRACTICAL NO. 2

Algebraic & Iterators

Step 1: Create a tuple with elements stated need to iterate using the iter and next method the number of time we use the iter and next method we will get the next iterating element in the tuple displaying its name.

Step 2: The similar output can be obtained by using for conditional statement in iterable variable is to be declared as for loop will iterate.

Step 3: Define a function name square with a parameter which will obtain with a square value of the given number. In similar fashion values can be get the value raised to and others the same.

Step 4: Call the above function using the function calling with list [square, cube] and

2.3. Iteration

Step 5: Using for conditional statement specify the range use the list object certain with map method declares a "lambda" which is no anonymous function and prints the same elements or values like

Step 6: Define a variable named value and define some elements then use the map method with help of lambda function give two argument display the output like

Step 7: Define a function even() with a parameter for using conditional statement to check whether the number is even and odd and return respectively.

Step 8: Define a class and another that defines the iter() method which will extract the first element within the container object.

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

```

my obj = add(7)
my list = my obj()
x = st (input("Enter a number"))
for i in my list:
    if (i % 2) == 0:
        print(i)
for n in range(5):
    value = list (map (lambda x : x * 2, first))
    print (value)
    print (list)
    print (first)
    print (list)
    print (first)
    print (list)
    print (first)
# def():
listnum = [0, 4, 8, 7, 9, 11, 13, 22, 19, 25]
list num = list (map (lambda x : x * 5, listnum))
print (list num)
def even(x):
    if (x % 2 == 0):
        return "even"
    else:
        return "odd"
list (map (even, listnum))
# odd numbers:
class odd:
    def __iter__(self):
        self.nums = 1
        return self.

```

188

```

def __init__(self):
    self.nums = []

def __add__(self, other):
    num = self + other
    return num

def __str__(self):
    nom = self.nums
    self.nums += other
    return str(self.nums)

my_obj = odd()
my_other = other(my_obj)
x = int(input("Enter a number:"))
for i in my_other:
    if (i % 2) != 0:
        print(i)

```

Output

```

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

```

029

Step 10: Define an object of a class.

Step 11: Accept an number from the user like which we want to display the odd numbers in numbers. pass this number to the object of the class.

Date: Jan 16/10

Q30

Factorial :

Step 1:- Define a class with argument and initialize its 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 and pass this object in the iter method.

Q30

1) Factorial

```
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
```

```
            fac = fac * i
```

```
        else:
```

```
            raise StopIteration
```

Output :

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

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

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

```
1! = 1
```

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

```
2! = 2
```

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

```
! = 6
```

```
# Program :-  
while True  
    try:  
        x = int(input("Enter class"))  
        break  
    except ValueError:  
        print("enter integer value")  
  
Output  
Enter class 457
```

Program :-

```
try:  
    f0 = open("abc.txt", "w")  
    f0.write("Archit Goyal")  
except IOError:  
    print("Error writing on the file")  
  
else:  
    print("Operation carried out successful")  
    f0.close()  
  
Output  
Operation carried out successfully.
```

PRACTICAL NO.3

Aim: Program to demonstrate exception handling

- Q) Write a program using the exception method of the native arithmetic errors.

Step 1:- Use the try block and accept the input using the raw_input method and convert it into the integer data type and 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.

- Q) Write a program for accepting the file on a given mode and use the environment errors as an exception for the given input

Step 1:- Within the try block open the file using the write mode and write the same content on the file.

Step 2:- Use the except block with IOError and display the message regarding missing of the file or unavailability of the mode.

CLASS LIBRARY

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

Step 4:- We accept user value errors or prints the message not a valid orange.

```

# program:
def assert_cn():
    assert (len(cn) == 0)
    print ("list is empty")
    var1 = []
    print (assert_cn (var1))

output:
list is empty
# program:
def accept_age():
    age = int (input ("Enter age :"))
    if age > 30 or age < 16:
        raise ValueError
    return age

valid = False
while not valid:
    try:
        age = accept_age()
        valid = True
    except ValueError:
        print ("Not a valid age")

output
Enter age : 4
Not a valid age
Enter age : 18

```

3. Write a program using the assert() to check if the list element are empty.
- Step 1:- Define a function which accepts an argument and check using the assert statement whether the given list is empty list and accordingly return the message.
- Step 2:- Class the functional and in the body of program and define certain elements in list and take appropriate.
4. Write a program to check the range of age of students in given class and if the age do not fall in given range else the raise error exception otherwise return the valid number.
- Step 1:- Define a function which will accept the age of the student from the standard input.
- Step 2:- Use the if condition to check whether the input age falls in the range and so return the age else raise the value errors exception.

680

step 3:- Before the while loop to check whether the boolean expression holds true use the try block to seek accept the age of student and terminate the looping condition.

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

PRIN

PRACTICAL - 4

Topic :- Regular Expression.

Step 1:- Import re module declare pattern and declare sequence use match method with declare arguments.

If arguments matched it has print the same otherwise print pattern NOT FOUND!

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

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

Step 4:- import re module declare string and accordingly declare pattern replace the blank space with no space. Use sub() with 3 arguments and print the string without spaces.

Step 5 - Import re module declarations sequence use search method for finding subsequently I use the group() with double dot operator as search() gives returning location using group() and it will show up the matched string.

Step 6 - Import re module declare list with numbers. Use the conditional statement here we have need up the for condition statement. Use if condition for checking first number is either 8 or 9 and next number are in range of 0 to 9 and check whether the entered numbers are equal to 10. If criteria matches first all number matches otherwise print failed.

```
# no-space:
import re
string = 'abc def ghi'
pattern = r'.'*15
replace = ''
V1 = re.sub(pattern, replace, string)
print(V1)
>>> abcdefghi
# group()
import re
sequence = '# python is an interesting language'
V = re.search('python', sequence)
first(V)
V1 = V.group()
print(V1)
>>> c_SRE_Match object at 0x0251DFOO>
python
# verifying the given set of phone numbers
import re
list1 = ['8004567891', '91+5673210', '7865032981',
         '9876543201']
for value in list1:
    if re.match(r'[8-9][\d]{13}[\d-9]\d{3}', value):
        value or len(value) == 10:
            print("Criteria matched for all numbers!")
    else:
        print("Criteria failed!")
```

080

if (V == ins'):
 f = f + 1.

else:

 m = m + 1
 first ("No. of males is: ", m)
 print ("No. of females is: ", f)
 print ("m1, ins', m5", m)
 >> [m1, ins', m5], m
 ("No. of males is: 2")
 ("No. of females is: 2")

087

Step 7:- Import re module declare a string
use the module with findall() for
finding the vowels in the string and
declare the same.

Step 8:- import re module declare the host
and domain name declare pattern
for separating the host and domain
name. use the findall() and
print the output respectively.

Practical 5

Ques 5 GUI components

Step 1:- See the Tkinter library for importing a feature of set widget.

Step 2:- Create a variable from set method and position it on parent window.

Step 3:- Use the pack method along with the object created from set method.

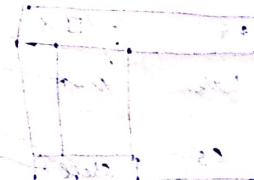
Step 4:- See the main loop method for triggering of corresponding event.

Step 5:- See the Tkinter library for importing its feature of set widget.

TEXT

```
CODE :-  
from tkinter import *  
root = Tk()  
l = Label(root, text = "ANISH")  
l.pack()  
root.mainloop()
```

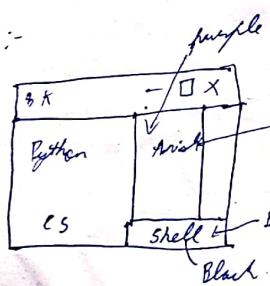
Output :-



Code :-

```
from tkinter import*
root = Tk()
l = Label (root, text = "Python")
l . pack (side = LEFT, padx = 30)
l1 = Label (root, text = "ANISH", bg = "purple", fg = "red")
l1 . pack (side = TOP, ipady = 40)
m = Label (root, text = "CS")
m . pack (side = LEFT, padx = 20)
m1 = Label (root = "shell", bg = "black blue", fg = "black")
m1 . pack (side = left, ipadx = 50)
root . mainloop()
```

Output :-



Step 6:- Create a variable upon `root` method and position it onto `tkinter` window.

Step 7:- Use the `pack` method along with the object created from `root` method not use its parameter.

Step 8:-
 SIDE = LEFT , padx = 20
 SIDE = LEFT , pady = 30
 SIDE = TOP , padx = 40
 SIDE = TOP , pady = 50

Step 9:- Use do to main loop method for triggering of corresponding events.

Step 10:- Now repeat the above steps with the `label` method with which takes the following arguments:

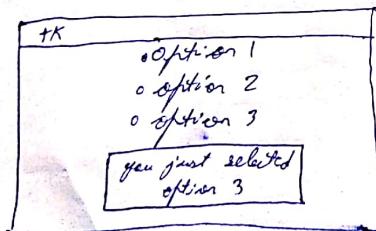
- 1) Text attribute with which defines string
- 2) Bg (background) colour
- 3) Fg (for ground) colour
- 4) Name of the parent window
- 5) Use `pack` method with the `ipadx` attribute.

Dr. M/S

Radio button

```
source code:  
from tkinter import *  
root = Tk()  
obj = Label():  
selection = "you selected the option " + str(obj)  
L = Label().config(text=selection, justify="center")  
L.pack(anchor="center")  
v = IntVar()  
r1 = Radiobutton(text="option 1", variable=v,  
value=1, command=sel)  
r2 = Radiobutton(text="option 2", variable=v,  
value=2, command=sel)  
r3 = Radiobutton(text="option 3", variable=v,  
value=3, command=sel)  
r1.pack()  
r2.pack()  
r3.pack()  
root.mainloop()
```

Output :



041

Radio button

A: Write a program making use of the control variable and button widget for selection of the given option.

Step 1: Use the `tkinter` method to import the relevant method.

Step 2: Use the configuration method along with the label object and all the variables as an argument within the method.

Step 3: Now define the parent window and define the option using control variable.

Step 4: Now define the parent window and define the option using control variable.

Step 5: Now create an object from radio button method which will take the roll arguments

- (1) Positioning of the parent window
- (2) Defining the text variable [1, 2, 3, 4]
- (3) Define the variable arguments
- (4) Corresponding value and trigger the given operation.

140

Step 6:- Pack method for the corresponding window objects so created and specify its attribute as an anchor attribute.

Step 7:- Now define the label object for the corresponding method and place it on parent window.

Subsequently use pack method for window and make use of window method.

Step 8:- Import the relevant method from Tkinter library.

Step 9:- Define the object corresponding to the object window and define the size of parent window in terms of the number of pixels.

Step 10:- Now define the frame object for method and place it onto the parent window.

Step 11:- Create another frame object for LEFT frame and put it onto window in its LEFT side.

142

Frame object

source code:

Frame object

from Tkinter import *

top = Tk()

top.geometry('100x200')

frame = Frame(top)

frame.pack()

left_frame = Frame(top)

left_frame.pack(side = LEFT)

b1 = button(frame, text = "select", activebackground = "red", fg = "black")

b1.pack()

b2 = button(frame, text = "modify", activebackground = "blue", fg = "purple")

b2.pack()

b3 = button(frame, text = "Add", activebackground = "green", fg = "yellow")

b3.pack()

b4 = button(frame, text = "exit", activebackground = "green", fg = "yellow")

b4.pack()

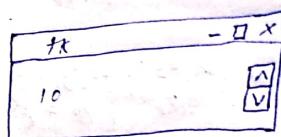
b4 = button(frame, text = "EXIT", activebackground = "Yellow", fg = "red");

b4.pack()

top.mainloop()

540

General code:-
Spinbox :-
from tkinter import *
raster = Tk()
S1 = spinbox(raster, from_ = 0, to = 10)
S1 . pack ()
raster . mainloop ()



043

PRACTICAL No. 5C17

- Write a program to make use of spinbox widget
- Step 1:- Use of Tkinter library to import the relevant methods
- Step 2:- Create the parent window object
- Step 3:- Create an object from the spinbox and place it onto the parent window with the option specified.
- Step 4:- Now use the pack method to now make the object visible onto the parent window and call the mainloop method.

Farad window

Step 1:- Create an object from the farad window() and use the pack() to make this object visible.

Step 2:- Now create an object from entry widget and place it on to the farad window and use the add() method.

Step 3:- Similarly create an object of the farad window and add it onto the existing window.

Step 4:- Create an object from the scale method and place it on to the preceding farad window and note add method accordingly.

At Step 5:- Create a button widget onto the farad window and define a functionality along with its event.

Step 6:- Use the pack and mailloop methods corresponding and to be triggered.

Save code

Farad window
from tkinter import *

root = Tk()

root.title("Farad window")

m1 = Farad()

m1.pack(expand = 100)

l1 = Entry(m1, bd = 10)

m1.add(l1)

m2 = Farad(m1, orient = VERTICAL)

m1.add(m2)

top = scale(m2, orient = HORIZONTAL)

m2.add(top)

def cell():

first = Tk()

first.title("New Page")

first.minsize(200, 200)

first.config(bg = "RED")

ll = Label(first, text = "NEW PAGE")

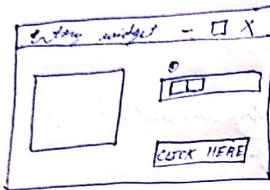
ll.pack()

first.mainloop()

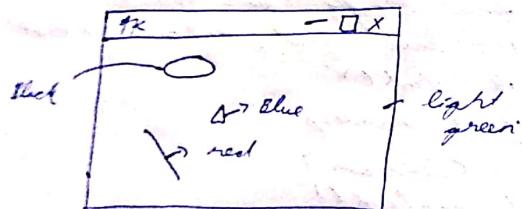
m2.add(ll)

root.mainloop()

44



```
# canvas
from Tkinter import *
root = Tk()
c = canvas(root, height=100, width=200, bg="light yellow")
arc = c.create_oval(10, 20, 30, 40, start=0, extent=50, fill="red")
line = c.create_line(10, 30, 45, 50, fill="red")
oval = c.create_oval(10, 15, 30, 20, fill="blue")
c.pack()
root.mainloop()
```



45

canvas :-

Step 1 :- Create an object from the canvas widget by using its attribute height, width, background colour and the parent window object.

Step 2 :- Use the corresponding method for drawing the simple geometrical steps like arc, oval and line. Specify the coordinate values.

Step 3 :- Finally use pack and its master method.