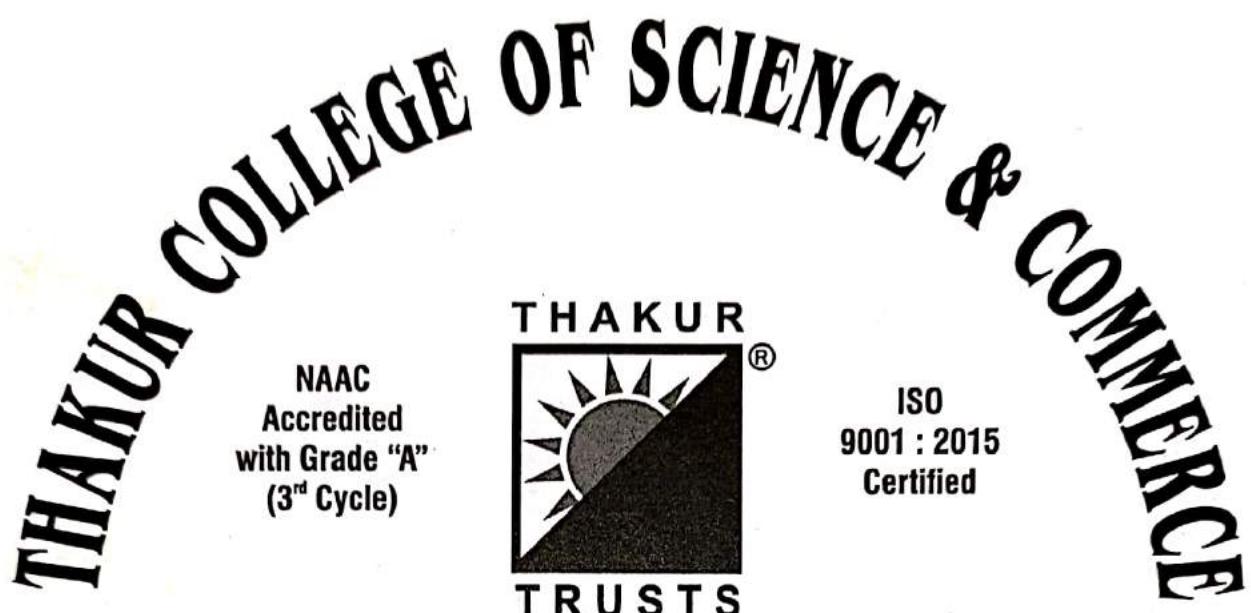


Exam Seat No. _____



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



ISO
9001 : 2015
Certified

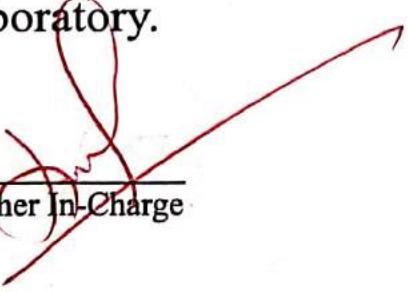
Degree College
Computer Journal
CERTIFICATE

SEMESTER II UID No. _____

Class FY BSC CS Roll No. 1730 Year 2019 - 20

This is to certify that the work entered in this journal
is the work of Mst. / Ms. Dinesh K. Lohar

who has worked for the year 2019 - 20 in the Computer
Laboratory.


Teacher In-Charge


Head of Department

Date : _____

Examiner



INDEX



SEM II.

| No. | Title | Page No. | Date | Staff Member's Signature |
|-----|--|----------|----------------|--------------------------|
| 1. | PRACTICAL - 1. Demonstrate the use of different file accessing modes diff. attributes read mode methods. | 25-28 | 26-11-19 | Jm 26-11-19 |
| 2 | PRACTICAL - 2 Iterators | good | 29-32 03-12-19 | Jm 03-12-19 |
| 3. | PRACTICAL - 3 Exception | good | 32-35 17/12/19 | Jm 17/12/19 |
| 4 | PRACTICAL - 4 Regular Expression | 36 - 40 | 24/12/19 | Jm 24/12/19 |
| 5. | PRACTICAL - 5(1) GUI (Graphical User Interface) | 40-41 | 7/01/20 | Jm 7/01/20 |
| 6. | PRACTICAL - 5(2) GUI , Radiobutton and Scrollbar widget | 42-44 | 14/01/20 | Jm 14/01/20 |

INDEX

| No. | Title | Page No. | Date | Staff Member's Signature |
|-----|--|----------|----------|--------------------------|
| 7. | PRACTICAL - 5(3) Messagebox, Relief, and Traverse | 45-48 | 21/01/20 | Dr. K. S. J. |
| 8. | PRACTICAL - 5(4) Frame and PhotoImage | 49-51 | 28/01/20 | Dr. K. S. J. |
| 9. | PRACTICAL - 5(5) Entry, Canvas, Spinbox, PanedWindow | 52-54 | 11/02/20 | Dr. K. S. J. |
| 10 | PRACTICAL - 6 Database Connectivity using sqlite3 | 55-59 | 25/02/20 | Dr. K. S. J. |
| 11. | PROJECT B.M.I Calculator | 60-63 | 3/03/20 | Dr. K. S. J. |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Coding :- Python 2.7

PSO

```
print "DINESH K LOHAR \n 1730-BATCH A \n 26-11-2019  
  \n PRACTICAL 1 \n FILE HANDLING"
```

WRITE

```
f = open("abc.txt", "w")  
f.write("Python" + "\n")  
f.write(" Smart Language \n AI Features \n Interpreter  
 Language")  
f.close()
```

READ

```
print "\n * READ"  
f = open("abc.txt", "r")  
str1 = f.read()  
str2 = f.readline()  
str3 = f.readlines()  
print "The read method is : \n", str1  
print "The readline method is : \n", str2  
print "The readlines method : \n", str3  
f.close()
```

APPEND

```
print "\n * APPEND"  
f = open("abc.txt", "a")  
f.write("\n Object Oriented Language")  
f.close()  
f = open("abc.txt", "r")  
append = f.read()  
print append  
f.close()
```

BASIC FILE ATTRIBUTES

```
print "\n * BASIC FILE ATTRIBUTES"  
f = open("abc.txt", "r")  $\hookrightarrow$  f.close()  
a = f.name  
b = f.closed  
c = f.mode  
d = f.softspace
```

```
print "Name Attribute : ", a  
print "closed Attribute : ", b  
print "Mode Attribute : ", c  
print "Softspace Attribute : ", d
```

PRACTICAL - 1

Aim :- File Handling ; Write , Read , Append ,
Basic Attributes , seek and Tell method &
Input methods , length of lines .

Algorithm :-

Step1. - create a file object by using the open() and use the write access mode followed by writing some content to it & then close the file.

Step2 - Now open the file in the read mode & use the read() , readline(), and readlines() method and show the output in the variable and finally display the content of the file.

Step3. - Use the file object for finding the name of the file, mode in which the file is been opened , wether the file is still open or closed & finally the output of the softspace attribute.

Step4 - Now append the file in the append mode & use the write()

mode followed by writing content to add in the file, then close the file and display the content of the file using read mode.

Step 5 - create a file object and use the r+ access mode.

Step 6 - use the read method with no. of bytes to be read and store it in a variable & display the contents of the variable.

Step 7 - Open file in the read mode, declare a variable and perform file object. dot tellmethod and store the output consequently in variable.

Step 8 - open file object with read mode also use the readlines method to store the output consequently in and print the same for counting the length use the for conditional statement display the length.

SEEK & TELL METHOD

```
print "\n* SEEK & TELL"
f=open("abc.txt","r")
t=f.tell()
print "TELL : \n",t
f.close()

f=open("abc.txt","r")
s=f.seek(0,0)
print "SEEK (0,0) : \n",s
s1=f.seek(0,1)
print "SEEK (0,1) : \n",s1
f.close()

f=open("abc.txt","r")
f.seek(11)
print "SEEK : \n",f.readline()
f.close()
```

026

LENGTH OF THE LINES

```
print "\n* LENGTH OF THE LINES"
f=open("abc.txt","r")
s=f.readlines()
for line in s:
    print len(line)
f.close()
```

INPUT METHODS

```
print "\n* INPUT METHODS"
print "RAW INPUT"
search=raw_input("Enter the raw input : ")
print search
print "INPUT"
search1=input("Enter the input : ")
print search1
```

WRITE +

```
f=open("abc.txt","w+")
f.write("\nIt is FOSS")
f.close()
```

READ +

```
print "\n* READ +"
f=open("abc.txt","rt")
r=f.read()
print r
```

Output :- Python 2.7

```
>>> 250  
DINESH KLOHAR  
1730 BATCH A  
26-11-2019  
PRACTICAL  
FILE HANDLING
```

* READ

The read method is :

Python
Smart Language

AI Features

Interpreter language

The readline method :

The readlines method : []

* APPEND

Python

Smart Language

AI Features

Interpreter language

Object oriented Language

* BASIC FILE ATTRIBUTES

Name Attribute : abc.txt

Closed Attribute : True

Mode Attribute : r

Softspace Attribute : 0

* SEEK & TELL

TELL:

0

SEEK (0,0) :

None

SEEK (0,1) :

None

SEEK :

Smart Language

Step 9 - The raw_input method reads one line at a time and returns the string. Accept the input from the user in raw_input() method and display the output.

Step 10 - The input method will accept the valid python expression from the user and returns the evaluated results. Accept the input from the user in input() method and display the output.

Jm
3/2/19

* LENGTH OF DIFFERENT LINES

7
15
12
21
24

028

* INPUT METHODS

RAW INPUT

Enter the raw input : Python

Python

INPUT

Enter the input: [x*10 for x in range(5,10,5)]

[50]

* READ +

It is FOSS

Coding :- Python 2.7

```
850  
print "DINESH K LOHAR\n1730 BATCH A\n03-12-19\nPRACTICAL 2\nITERATORS"
```

#ITER AND NEXT

```
print "\nITER AND NEXT"  
mytuple1 = ("Samsung", "Apple", "Oneplus")  
print mytuple1  
myiter1 = iter(mytuple1)  
print next(myiter1)  
print next(myiter1)  
print next(myiter1)
```

#LOOPING THROUGH AN ITERATOR

```
print "\n*LOOPING THROUGH AN ITERATOR"  
mytuple2 = ("Exception", "Iterators", "File Handling")  
print mytuple2  
for x in mytuple2:  
    print x
```

#SQUARE AND CUBE

```
print "\n * SQUARE AND CUBE"  
def square(x):  
    return x**2  
def cube(x):  
    return x**3  
fun1 = [square, cube]  
S = int(input("Enter the number : "))  
for r in range(S):  
    values = list(map(lambda x: x(r), fun1))  
    print values
```

#MAP

```
print "\n * MAP"  
listnum = [0, 6, 8, 9, 12, 13, 15, 16, 18, 21]  
listnm = list(map(lambda x: x*.5, listnum))  
print listnm  
def even(x):  
    if x*.2 == 0:  
        return "EVEN"  
    else:  
        return "ODD"
```

PRACTICAL - 2

Aim :- Iterations

Algorithm :-

Step 1 - Create a tuple with elements that we need to iterate using the `iter()` and `next()` method. The number of time we use the `iter()` & `next()` method we will get the next iterating element in the tuple. Display the output.

Step 2 - The similar output can be obtained by using for conditional statement looping. An iterable variable is to be declared in for loop which will iterate.

Step 3 - Define a function 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 3 & return the same.

Step 4 - Call the declared function using function call.

PSO

Step 5 - Using for conditional statement

Specifying the range use the list typecasting with map method declare a 'lambda' ie, anonymous function and print the same.

Step 6 - Declare a list number variable and declare some elements then use the map method with help of lambda function give two arguments, display the output.

Step 7 - Define a function even with a parameter then using conditional statements do check whether the number is even and odd and return respectively.

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

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

```
d = list(map(lambda, listnum))
```

ODD NUMBERS

```
print "\n*ODD NUMBERS"
```

```
class odd:
```

```
    def __iter__(self):
```

```
        self.num = 1
```

```
        return self
```

```
    def next(self):
```

```
        num = self.num
```

```
        self.num += 2
```

```
    def next(self):
```

```
        num = self.num
```

```
        self.num += 2
```

```
        return num
```

```
myobj = odd()
```

```
myiter = iter(myobj)
```

```
g = int(input("Enter a Number : "))
```

```
for j in myiter:
```

```
    if j < g:
```

```
        print j
```

030

Output :— Python 2.7

>>> 080

DINESH K LOHAR
1730 BATCH A
03-12-19
PRACTICAL 2
ITERATORS

* ITER AND NEXT

('Samsung', 'Apple', 'Oneplus')

Samsung
Apple
Oneplus

* LOOPING THROUGH AN ITERATOR

('Exception', 'Iterators', 'File Handling')

Exception
Iterators
File Handling

* SQUARE AND CUBE

Enter a number : 4

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

* MAP

[0,1,3,4,2,3,0,1,3,1]

['EVEN', 'EVEN', 'ODD', 'EVEN', 'EVEN', 'ODD', 'EVEN', 'EVEN', 'ODD']

* ODD NUMBERS

Enter a number : 10

1
3
5
7
9

Step 10 - Define an object of a class.

Step 11 - Accept an number from the user till which we want to display the odd numbers.

~~no 12 has stored in variable = 1012~~

~~function return in off~~

~~1011 odd function base~~

~~variable define square with int = 0~~

~~if loop with condition for i <= n~~

~~odd no. of sum~~

~~if condition true then add odd no. to square~~

~~else increment i by 1~~

~~else increment i by 1~~

~~return square~~

~~and square has initialized with zero is quite~~

~~but this may give error~~

~~so we can use int square = 0~~

~~if loop with condition for i <= n~~

~~odd no. of sum~~

~~else increment i by 1~~

17-12-19

180

PRACTICAL - 3

Aim :- Exception Handling

Algorithm :-

IO Error

Step 1 - Use the try block and open an file in write mode & write some content to the file.

Step 2 - use the except block with 'IOERROR' block and convey the appropriate message to the user.

Step 3 - use else block and display the message that the operation is conveyed out successfully.

Value error

Step 4 - use the try block and accept the integer input from the user.

Step 5 - use the except block with 'ValueError' block and convey the appropriate message to the user.

Step 6 - use else and display the message that the operation is successful.

Coding :- Python 3.8

```
print("DINESH K LOHAR\n1730 BATCH A\n17-12-2019\nPRACTICAL 3\nEXCEPTION HANDLING")  
  
# IOError  
print("\n* IOError")  
try:  
    fo=open("abc.txt","w")  
    fo.write("A Sample Paragraph")  
except IOError:  
    print("Errors : 1\nIOError : Enter a valid MODE for  
File Operation")  
else:  
    print("Errors : 0\nFile Operation is Successful")  
finally:  
    fo.close()  
  
# ValueError  
print("\n* VALUE ERROR")  
try:  
    a=int(input("Enter a Integer Value :"))  
except ValueError:  
    print("Errors : 1\nValueError : Enter a valid  
Integer value")  
else:  
    print("Errors : 0\nOperation is Successful")  
  
# ZeroDivisionError  
print("\n* ZERODIVISION ERROR")  
try:  
    b=int(input("Enter a Integer Value :"))  
    c=10/b  
    print("The Division by 10 is",c)  
except ZeroDivisionError:  
    print("Errors : 1\nZeroDivisionError : 0 is not  
Divisible")  
else:  
    print("Errors : 0\nOperation is Successful")
```

~~# IndexError~~

```
print("\n * INDEX ERROR")
try:
    mylist = [5,7,2,12,6,4,3]
    print(mylist)
    i = int(input("Enter a Index Number : "))
    print(mylist[i])
except IndexError:
    print("Errors : 1\nIndexError : List index
          out of range")
else:
    print("Errors : 0\n Operation is Successful")
```

~~#AssertionError~~

```
print("\n * ASSERTION ERROR")
try:
    x = int(input("Enter a Numerator : "))
    y = int(input("Enter a Denominator : "))
    assert y != 0, 'Invalid Operation'
    print("Division : ", x/y)
except AssertionError:
    print("Errors : 1\nAssertionError : Enter a
          valid Number , Greater than Zero")
else:
    print("Errors : 0\nOperation is Successful")
```

~~# Raise & Finally~~

```
print("\n * RAISE AND FINALLY")
try:
    s = float(input("Enter the length of side : "))
    voc = s**3
    if s > 0: print("Volume of Cube : ", voc)
    else: raise IOError
except IOError:
    print("Errors : 1\nIOError : Enter a valid Input")
else:
    print("Errors : 0\nOperation is successful")
Finally:
    print("Area of Cube : ", 6*s*s)
```

ZeroDivisionError

Step 7 - use tryblock and accept the input from the user.

Step 8 - Initialize a variable for dividing 10 by the input taken from the user.

Step 9 - Use except block with 'ZeroDivisionError' and display the appropriate message to the user.

Step 10 - Use else to display the message 'operation is successful'.

Index Error

Step 11 - use tryblock and initialize a variable mylist and create a list and display the created list.

Step 12 - Use Accept the indexing number from the user to access the index of a list. Display the output 'print(mylist[1]).

Step 13 - Use except block with 'IndexError' and display the appropriate message else display the operation is successful.

AssertionError

Step 14 - Use tryblock and Accept the two inputs from the user one is numerator and another is denominator.

Step 15 - Use assert keyword Statement that denominator is not equal to zero with a message Invalid Operation. Then print the output with a message the Division is.

Step 16 - Use except block with 'AssertionError' and display the appropriate message else display the operation is successful message.

Raise and Finally

Step 17 - Use try block and accept the length of a side of a cube from the user. Use the formula of volume of cube.

Step 18 - Use conditional statement , if side is greater than zero , display the volume of cube else raise

Output 1 :- Python 3.8 without Errors

>>>

DINESH K LOHAR
1T30 BATCH A
17-12-2019
PRACTICAL 3
EXCEPTION HANDLING

034

* IO ERROR

Errors : 0

File Operation is Successful

* VALUE ERROR

Enter a Integer Value : 12

Errors : 0

Operation is Successful

* ZERO DIVISION ERROR

Enter a Integer Value : 2

The Division by 10 is 5.0

Errors : 0

Operation is Successful

* INDEX ERROR

[5, 7, 2, 12, 6, 4, 3]

Enter a Index Number : 5

4

Errors : 0

Operation is Successful

* ASSERTION ERROR

Enter a Numerator : 10

Enter a Denominator : 4

Division : 2.5

Errors : 0

Operation is Successful

* RAISE AND FINALLY

Enter the length of side : 4

Volume of Cube : 64.0

Errors : 0

Operation is Successful

Area of Cube : 96.0

Output 2 :- Python 3.8 with Errors

>>>

DINESH K LOHAR

T30 BATCH A

17-12-2019

PRACTICAL 3

EXCEPTION HANDLING

* IO ERROR

Errors: 1

IOError: Enter a valid MODE for file Operation

* VALUE ERROR

Enter a Integer Value : d

Errors: 1

ValueError: Enter a Valid Integer Value

* ZERO DIVISION ERROR

Enter a Integer Value : 0

Errors: 1

ZeroDivisionError: 0 is not Divisible

* INDEX ERROR

[5, 7, 2, 12, 6, 4, 3]

Enter a Index Number : 9

IndexError: List Index out of range

* ASSERTION ERROR

Enter a Numerator : 9

Enter a Denominator : 0

Errors: 1

AssertionError: Enter a valid number, Greater than 0

* RAISE AND FINALLY

Enter the length of side :-2

Errors: 1

IOError: Enter a valid Input

Area of Cube : 24.0

IOError.

Step 19 - Use the except block with 'IOError' and display the appropriate message to the user else display a message 'Operation is successful'.

Step 20 - Use the finally block and find the area of cube and display the output.

P.E.O
24. Dec. 2019

PRACTICAL - 4

Aim :- Regular Expression

Algorithm :-

Step 1 - import re module for using regular expression functions. and use library.

Step 2 - MATCH:

Write a string or a sequence & initialize it in a variable.

Step 3 - Initialize a variable pattern to r'FYCS'

Step 4 - Use if conditional statement to check whether the pattern is matching to the string using re.match function, if it matches then print Match found else print match not found.

Step 5 - FIND ALL:

Initialize a string or sequence to a variable.

Step 6 - Use \d+ pattern for integers & use re.findall() method for finding

Coding :- Python 3.8

036

```
print ("DJNESH K LOHAR \n 1730 BATCH A \n 24-12-2019 \n PRACTICAL \n")  
import re  
# MATCH  
print ("\n * MATCH")  
seq1 = 'FYCS represents First Year in Computer Science'  
pat1 = r'FYCS'  
if re.match (pat1, seq1):  
    print ("Match Found for FYCS")  
else:  
    print ("NO Match Found for FYCS")  
# FINDALL  
print ("\n * FINDALL")  
Seq2 = 'Regular Expression 21369'  
pat2 = r'\d+'  
op1 = re.findall (pat2, seq2)  
print ("Output using dt : ", op1)  
pat3 = r'\D+'  
op2 = re.findall (pat3, seq2)  
print ("Output using Dt : ", op2)  
# SPLIT  
print ("\n * SPLIT")  
Seq3 = 'hrv12, wbv42, tbh13'  
Pat4 = r'\d+'  
op3 = re.split (pat4, seq3, maxsplit=0)  
pats = r'\D+'  
op4 = re.split (pats, seq3, maxsplit=0)  
print ("Output using dt : ", op3)  
print ("Output using Dt : ", op4)  
# SUB  
print ("\n * SUB")  
Str1 = 'P Y T H O N'  
Pat6 = r'\S+'  
Replace = ''  
op5 = re.sub (pat6, replace, str1)  
Print (op5)
```

STARTING OF A NUMBER

```
Print("\n* STARTING OF A NUMBER")
Count = 0
list = ['9988776655', '8976012345', '7777151134', '6377280001']
for value in list:
    if re.match(r'[8-9]{2}[0-9]{9}', value):
        if len(value) == 10:
            Count += 1
    Print(Count, "Cell Numbers Starting with
          8 & 9 are", value)
else:
    Count += 1
Print(Count, "Cell Numbers Not Starting with
      8 & 9 are", value)
```

FINDING A STRING IN PARAGRAPH

```
Print("\n* FINDING A STRING IN PARAGRAPH")
Str2 = ("CAA Stands for The Citizenship Amendment Act,
        2019 \n was passed by the Parliament of India
        on 11 December 2019.")
print(Str2)
result1 = re.findall(r'[aeiou AEIOU]\w+', Str2)
Print("Output : \n", result1)
result2 = re.findall(r'\b[aeiou AEIOU]\w+', Str2)
Print("Output using b : \n", result2)
```

HOSTNAME AND DOMAIN NAME

```
Print("\n* HOSTNAME AND DOMAIN NAME")
Str3 = ('log on to tesc@gmail.com, single@tinder.in,
        mata.pita.com')
Print(re.findall(r'[\w\.-]+@[\\w\.-]+', Str3))
```

EXTRACTION OF CHARACTERS

```
Print("\n* EXTRACTION OF CHARACTERS")
Str4 = ("python is an AI language and it is very
        optimized language")
print(Str4)
print("Output : \n", re.findall(r'\b[a-z]{2}', Str4))
```

the integers from the given string.

Step 7 - Use \D+ pattern for anything but a integer and then use re.findall() method for finding all the characters of the string.

Step 8 - SPLIT:

Initialize a sequence ~~in~~ ~~initialized~~ in a variable.

Step 9 - Use the \d+ pattern and re.split() method, for splitting the characters in a sequence with maxsplit

Step 10 - Use the \D+ pattern and re.split() method for splitting the integers present in a sequence with maxsplit.

Step 11 - SUB:

Initialize a string with spaces in a variable.

Step 12 - Use the \s+ pattern and initialize a variable replace with no space.

Step 13 - Use re.sub() method for removing whitespace from the string.

80

Step 14 - Starting Of a Number:

Initialize a variable list1 and write phone number in it.

Step 15 - Use for loop and in for loop use if conditional statement to find the numbers starting with 8 and 9 using re.match() method and in the if conditional statement use another if conditional statement to check the length of the numbers and Display the output.

Step 16 - Finding a string in Paragraph:

Initialize a paragraph (string) to a variable str2.

Step 17 - Use \w+ pattern and re.findall() method for finding the words in a paragraph containing 'aeiou' and print the output.

Step 18 - Use \b and \w+ pattern and re.findall() method for finding the words starting with 'aeiou' and print the output.

FINDING MALES AND FEMALES

```
Print("\n* FINDING MALES AND FEMALES")
strs = ("Mr. Kaminari, Mrs. Nobisuke, Mr. Pablo,
        Mrs. Kaminari, Mr. Nobisuke")
print(re.findall(r'\b[Mr.] \{3\}', strs))
print(re.findall(r'\b[Mrs.] \{4\}', strs))
print("Males : ", len(re.findall(r'\b[Mr.] \{3\}', strs)))
print("Females : ", len(re.findall(r'\b[Mrs.] \{4\}', strs)))
```

038

Output : Python 3.8

>>>

DINESH K LOHAR
1730 BATCH A

24-12-2019

PRACTICAL 4

REGULAR EXPRESSION

* MATCH

Match found for F4CS

* FINDALL

Output using dt : ['21369']

Output using Dt : ['Regular Expression']

* SPLIT

Output using dt : ['hru', 'wbu', 'tbh', '']

Output using Dt : ['', '12', '42', '13']

* SUB

PYTHON

* STARTING OF A NUMBER

1 Cell Number Starting with 8 & 9 are 9988776655

2 Cell Number Starting with 8 & 9 are 8976012345

3 Cell Number Not Starting with 8 & 9 are 77715134

4 Cell Number Not Starting with 8 & 9 are 6377280001

* FINDING A STRING IN PARAGRAPH

CAA stands for The Citizenship Amendment Act, 2019
was passed by the Parliament of India on 11 Dec 2019.

Output:

```
['AA', 'ands', 'or', 'itizenship', 'Amendment', 'Act', 'as',
 'ssed', 'arliament', 'of', 'India', 'on', 'ember']
```

Output using b:

```
['Amendment', 'Act', 'of', 'India', 'on']
```

* HOSTNAME AND DOMAIN NAME

```
['tcsc@gmail.com', 'single@tinder.in']
```

* EXTRACTION OF CHARACTERS

python is an AI language and it is very optimized language

Output:

```
['py', 'is', 'an', 'la', 'an', 'it', 'is', 've', 'op', 'la']
```

* FINDING MALES AND FEMALES

```
['Mr!', 'Mr!', 'Mr!']
```

```
['Mrs!', 'Mrs!']
```

Males : 3

Females : 2

Step 18+1 - Finding Males and Females :

Initialize a string with names of males and females to variable.

Step 20 - Use \b[Mr.]\{3} pattern for finding the no. of males in a string using `int re.findall()` method and print the output.

Step 21 - Use \b[Mrs.]\{4} pattern and repeat the above step.

Step 22 - Print the no. of males and females.

On 11/2020

E&O
7.01.2020.

PRACTICAL - 5 (1)

Aim :- GUI (Graphical User Interface)

Algorithm :-

Parent Window

Step 1 - Import the tkinter module and import features for the text widget.

Step 2 - Create an object root using TK() method.

Step 3 - Create a variable from the text method & position it onto the parent window.

Step 4 - Use the mainloop() method for triggering of the corresponding events.

Padding & Pack Method

Step 5 - Use the tkinter library for importing features of the text widget.

Step 6 - Create a variable from the text method & position it onto the parent window.

Coding :- Python 3.6

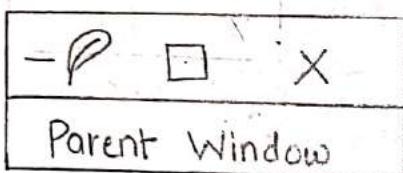
1. Parent Window :-

040

Parent Window

```
from Tkinter import *
root = TK()
l = Label(root, text = "Parent Window")
l.pack()
root.mainloop()
```

Output :-



Coding :- Python 3.6

2. Padding & Pack Method :-

Padding & Pack Method

```
from Tkinter import *
```

```
root = TK()
```

```
l = Label(root, text = "CRA Stands for Citizenship  
Amendment Act, 2019", bg = "light blue",  
fg = "black", font = "Algerian 20", height = 30,  
width = 20)
```

```
l.pack(side = 'left', padx = 30, pady = 20)
```

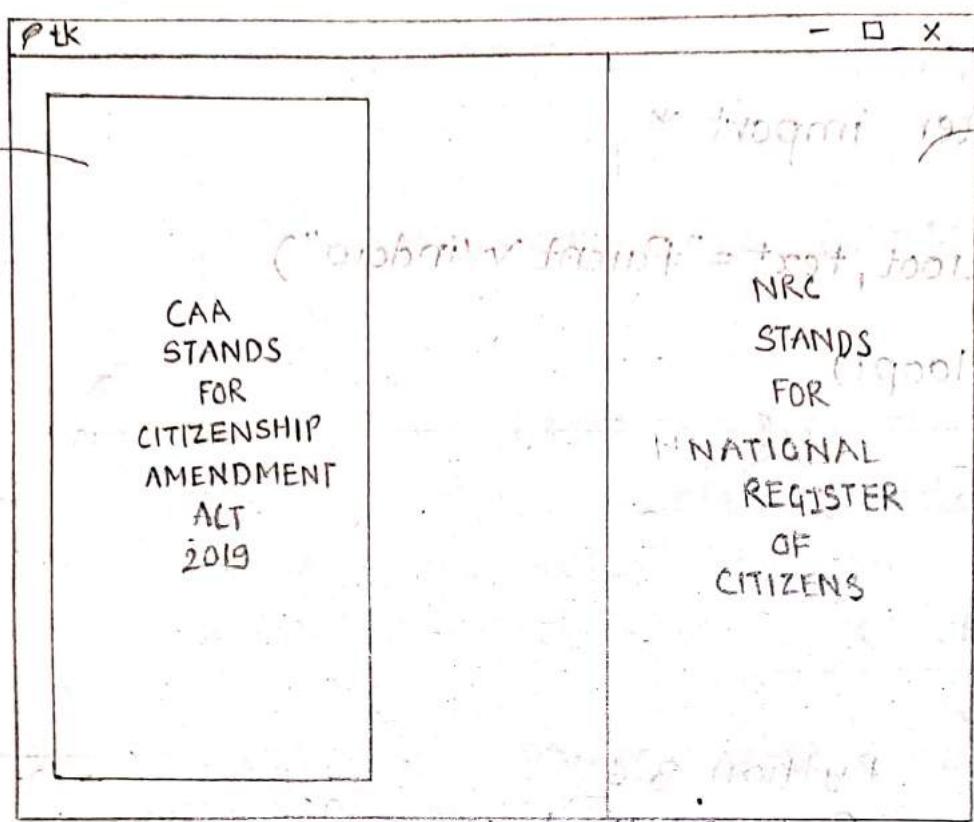
```
l1 = Label(root, text = "NRC Stands for National  
Register of Citizens", bg = "grey", fg = "black",  
font = "Algerian 20", height = 30, width = 20)
```

```
l1.pack(side = 'right', ipadx = 30, ipady = 20)
```

```
root.mainloop()
```

Coding Output :- Python 3.6

ON



Step 7 - Use the pack method along with object created from the text method & use the parameters.

- 1) side = LEFT , padx = 20
- 2) side = LEFT , pady = 30
- 3) side = TOP , ipadx = 40
- 4) side = TOP , ipady = 50

Step 8 - Use the mainloop() method for triggering of the corresponding events.

Step 9 - Now, repeat the above steps with the Label method which takes the following arguments 1) The name of the parent window 2) Text attribute which defines the string 3) The background color 4) Foreground color (fg) & then use the pack method with the relevant padding attributes.

*Jm
19/10*

140

14. Jan. 20

PRACTICAL - 5 (2)

Aim :- RadioButton and Scrollbar using tkinter (GUI).

Algorithm :-

Step 1 - Use the tkinter module to import the relevant methods.

Step 2 - Define a function which tells the user about the given selection made from the multiple option available.

Step 3 - Use the config() method along with the label object and call the variable as an argument within the method.

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

Step 5 - Now create an object from the RadioButton which will take the following argument:-

- i) Positioning on the parent window.
- ii) Defining the text variable which

* RADIOBUTTON

042

Coding :- Python 3.7

```
#RADIOBUTTON
from tkinter import*
def caa():
    Sel = "\nCITIZENSHIP AMENDMENT ACT 2019 - " + str(var.get())
    label.config(text=sel, font='Algerian 30')
def nrc():
    Sel = "\nNATIONAL REGISTER CITIZEN - " + str(var.get())
    label.config(text=sel, font='Algerian 30')
def ias():
    Sel = "\nINDIAN ADMINISTRATIVE SERVICE - " + str(var.get())
    label.config(text=sel, font='Algerian 30')
def ips():
    Sel = "\nINDIAN POLICE SERVICE - " + str(var.get())
    label.config(text=sel, font='Algerian 30')
def pcs():
    Sel = "\nPUBLIC SERVICE COMMISSION - " + str(var.get())
    label.config(text=sel, font='Algerian 30')
def upsc():
    Sel = "\nUNION PUBLIC SERVICE COMMISSION - " + str(var.get())
    label.config(text=sel, font='Algerian 30')
root = Tk()
var = IntVar()
L = Label(root, text="FULL FORMS OF SOME INDIAN SERVICES AND ACTS", font='Times 40', width=70, bg="#33ccff")
L.pack(anchor='c', side='top')
r = Radiobutton(root, text="CAA", variable=var, value=1, bg='orange', activebackground='green', fg='black', height=2, width=70, font='Algerian 25', command=caa)
r.pack(anchor='w')
r = Radiobutton(root, text="NRC", variable=var, value=2, bg='orange', activebackground='green', fg='black', height=2, width=70, font='Algerian 25', command=nrc)
r.pack(anchor='w')
```

```
r = Radiobutton(root, text = "JAS", variable=var, value=3,  
                bg='white', activebackground='blue', fg='black',  
                height=2, width=10, font='Algerian 25', command=jas)  
r.pack(anchor='w')
```

```
r = Radiobutton(root, text = "IPS", variable=var, value=4,  
                bg='white', activebackground='blue', fg='black',  
                height=2, width=10, font='Algerian 25', command=ips)  
r.pack(anchor='w')
```

```
r = Radiobutton(root, text = "PSC", variable=var, value=5,  
                bg='green', activebackground='orange', fg='black',  
                height=2, width=10, font='Algerian 25', command=psc)
```

```
r.pack(anchor='w')
```

```
r = Radiobutton(root, text = "UPSC", variable=var, value=6,  
                bg='green', activebackground='orange', fg='black',  
                height=2, width=10, font='Algerian 25', command=upsc)
```

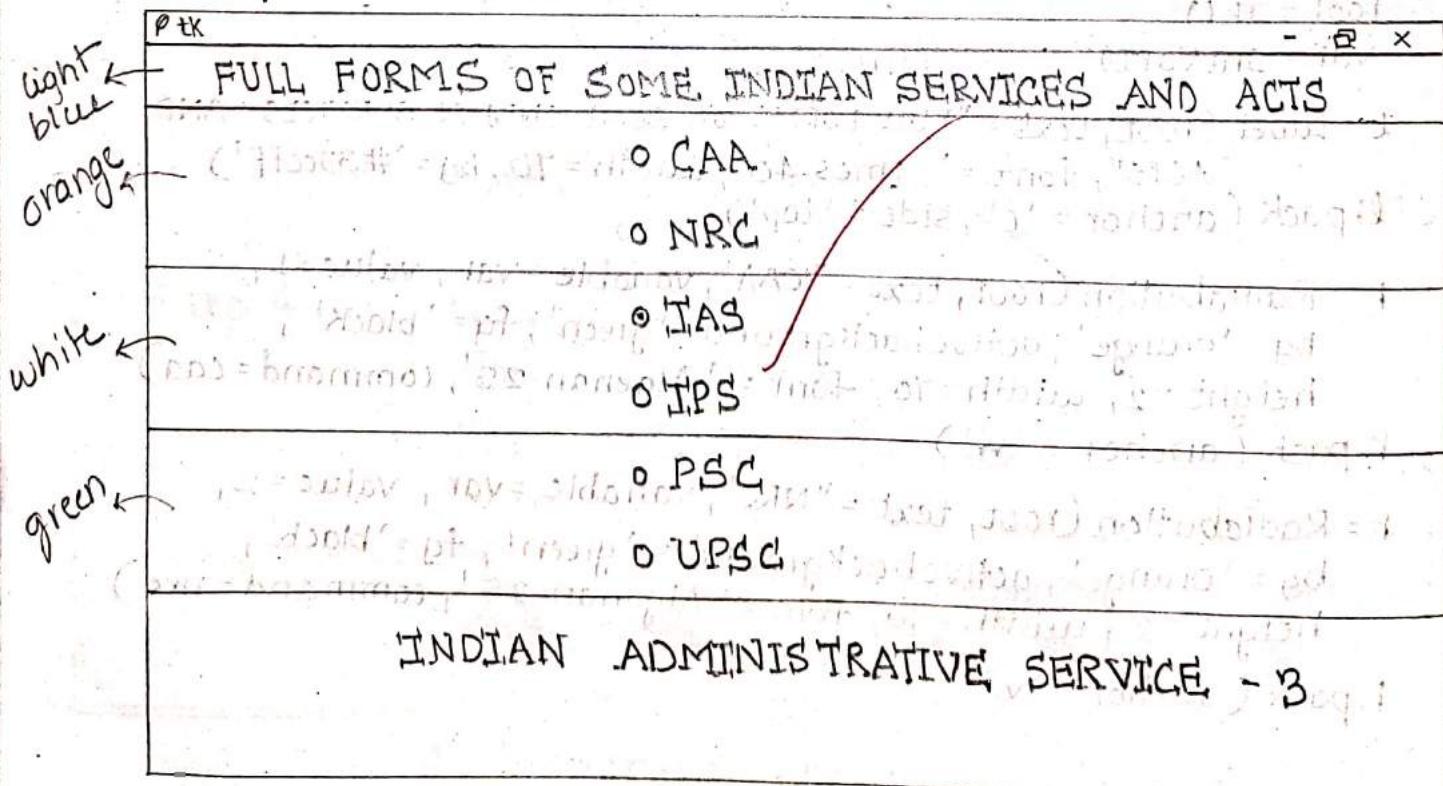
```
r.pack(anchor='w')
```

```
label = Label(root)
```

```
label.pack()
```

```
root.mainloop()
```

Output - Python 3.7



will take the values option 1, 2, 3, ...

iii) Define the variable argument.

iv) The corresponding value and trigger the given function.

Step 6 - Use the pack method for the corresponding Radio object so created and specify the argument as an anchor attribute.

Step 7 - Now define the label object from the corresponding and place it onto the parent window and subsequently use the pack method for this widget and finally call the mainloop() method.

Scrollbar

Step 8 - Import the relevant methods from the tkinter library.

Step 9 - Create an object corresponding to the parent window & create an object from the scrollbar() method and place it onto the parent window so created.

Step 10 - Create an object from the text method and place it onto the

EDO

parent window within the height and width specified.

Step 11 - Use the pack method along with the object of the scrollbar() method and use the argument as size and fill.

Step 12 - Now use the text object along with the pack method and again use the side and fill attribute.

Step 13 - Now use the config() method along with the object and the scrollbar and use the command attribute.

Step 14 - Similarly, use the config method along with the text object and use the yscroll command argument.

Step 15 - Now define the textual information in terms of paragraph and use the insert method with the two arguments and call the loop method at last.

* SCROLLBAR

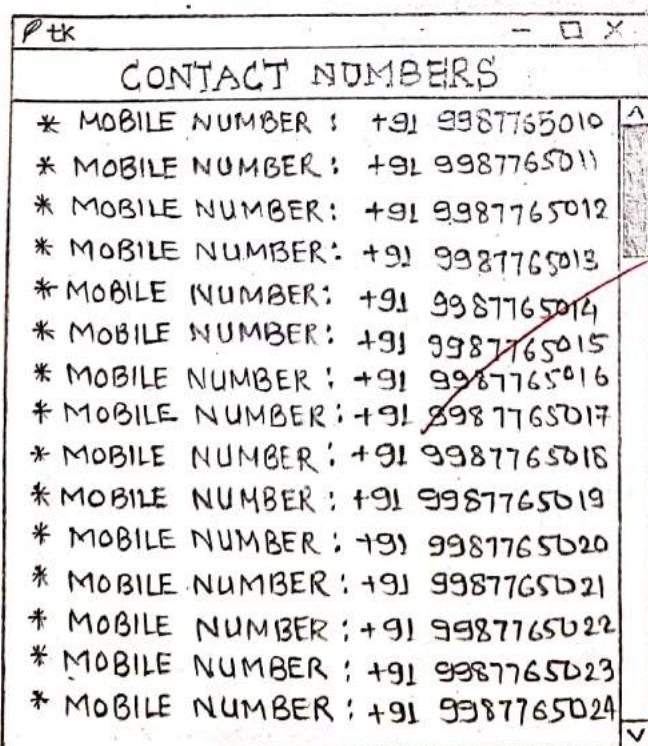
Coding :- Python 3.7

044

#SCROLLBAR

```
from tkinter import *
root = Tk()
root.geometry('450x400')
l = Label(root, text = "CONTACT NUMBERS", font = 'courier 20 bold',
          width = 100, bg = 'peach puff')
l.pack()
Scroll = Scrollbar(root)
Scroll.pack(side = RIGHT, fill = Y)
mylist = Listbox(root, font = 'courier 15', yscrollcommand = Scroll.set,
                 width = 400, bg = 'light blue')
for num in range(10, 100):
    mylist.insert(END, "* MOBILE NUMBER : +91 99877650" + str(num))
mylist.pack(side = LEFT, fill = BOTH)
scroll.config(command = mylist.yview)
root.mainloop()
```

Output :- Python 3.7



Jm
2/1

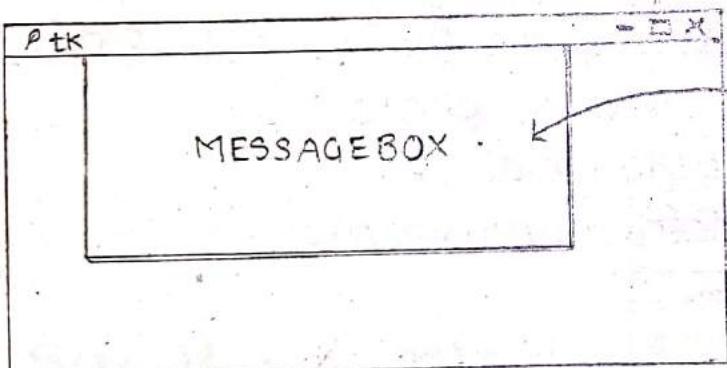
★ Messagebox

Coding :- Python 3.7

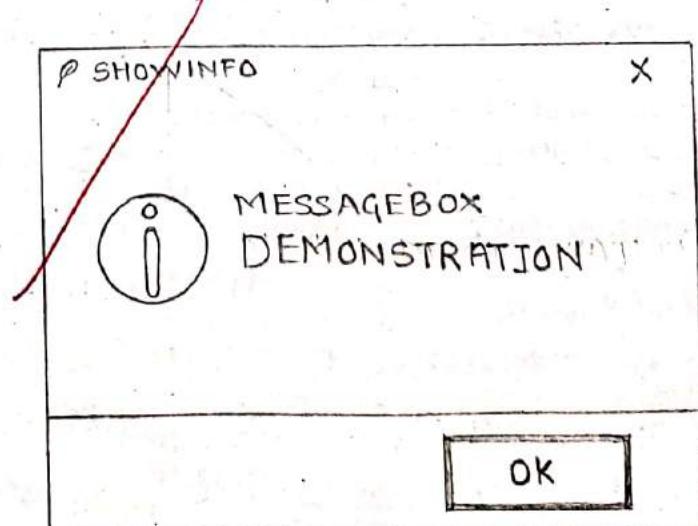
MESSAGE BOX

```
from tkinter import *
from tkinter import messagebox
def msgbox():
    messagebox.showinfo("SHOWINFO", "MESSAGE BOX\nDEMONSTRATION")
root = Tk()
root.config(bg='grey')
Button(root, text='MESSAGEBOX', bg='light blue',
       activebackground='light green', height=4,
       width=30, font='Verdana 10 bold', command=msgbox).pack()
root.mainloop()
```

Output:- Python 3.7



Click on this



PRACTICAL - 5(3)

Aim :- Demonstrating the use of MessageBox, Relief and Traversing and making use of geometry layout.

Message box

Algorithm :- To demonstrate the messagebox library.

Step 1 - Import the relevant method from tkinter library.

Step 2 - Def a function and use the message box along with different methods available are showinfo(), showwarning(), showerror(), askyesno(), askquestion(), and askokcancel().

Step 3 - Thus different options which are available in messagebox library.

Step 4 - Create object from button method and place it onto the parent window with the title of the button specified and the corresponding event called for triggering.

240

Step 5 - Use the pack method to display the button widget and finally use mainloop method.

Step 6 - If the user wants to hide the parent window and only the info window should be visible corresponding to the six options given above the withdraw method is used.

Relief Style

Algorithm :-

Step 1 - Use the button object with the following attributes

1. The parent window
2. Text attribute
3. Relief style

Step 2 - Use the corresponding pack method for the respective button objects and trigger the corresponding event.

Step 3 - Finally use the mainloop method.

* Relief

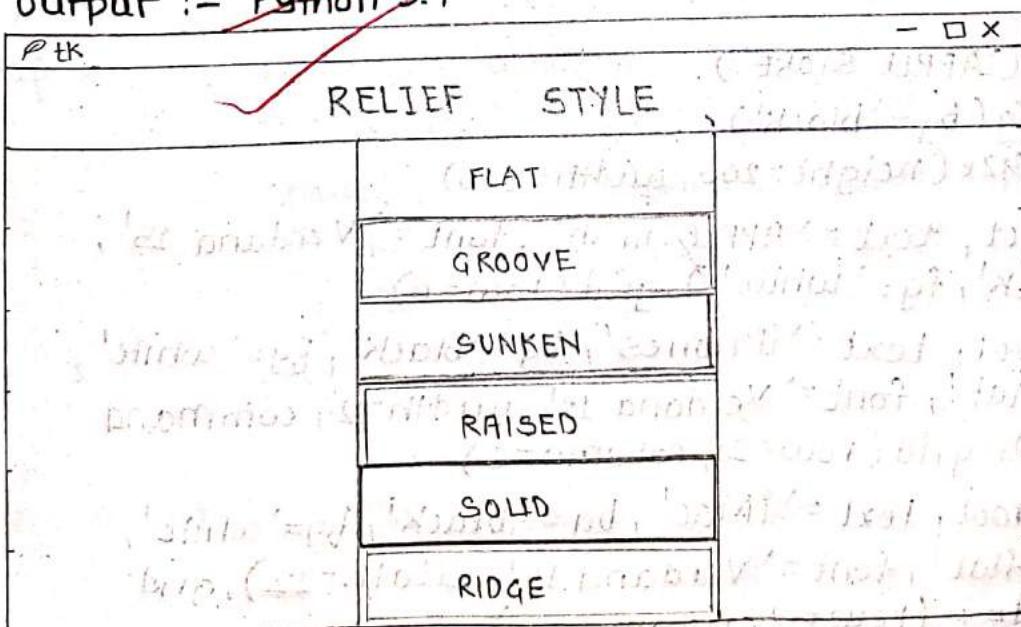
Coding :- Python 3.7

046

RELIEF

```
from tkinter import *
root = TK()
root.config(bg='grey')
Label (root, text='RELIEF STYLE', height=4, width=70,
       bg='peach puff', font='Times 12 bold').pack()
Button (root, text='FLAT', relief='flat', bg='light blue',
        height=4, width=15).pack()
Button (root, text='GROOVE', relief='groove', bg='light blue',
        height=4, width=15).pack()
Button (root, text='SUNKEN', relief='sunken', bg='light blue',
        height=4, width=15).pack()
Button (root, text='RAISED', relief='raised', bg='light blue',
        height=4, width=15).pack()
Button (root, text='SOLID', relief='solid', bg='light blue',
        height=4, width=15).pack()
Button (root, text='RIDGE', relief='ridge', bg='light blue',
        height=4, width=15).pack()
root.mainloop()
```

output :- Python 3.7



★ Traversing

Coding :- Python 3.7

TRAVERSING

```
from tkinter import *
def iphone():
    x = TK()
    x.title('iPhone')
    x.minsize(height=200, width=200)
    x.config(bg='black')
    Label(x, text='iPhone', width=50, font='Verdana 15', bg='black', fg='white').pack()
    Label(x, text='\n * iPhone 11 ProMax\n * iPhone 11 Pro\n * iPhone 11\n * iPhone XS Max\n * iPhone XS\n * iPhone XR', bg='black', fg='white', font='Verdana 12').pack(side=LEFT, anchor='n')
    Button(x, text='BACK', command=x.destroy).pack(anchor='s', side=RIGHT, padx=5, pady=5)
```

```
root = TK()
root.title('APPLE STORE')
root.config(bg='black')
root.minsize(height=200, width=200)
Label(root, text='APPLE.in\n', font='Verdana 15', bg='black', fg='white').grid(row=0)
Button(root, text='iPhones', bg='black', fg='white', relief='flat', font='Verdana 12', width=12, command=iphone).grid(row=2, column=0)
Button(root, text='iMac', bg='black', fg='white', relief='flat', font='Verdana 12', width=12).grid(row=2, column=1)
```

Traversing and making use of geometry layout manager method.

Algorithm:

Step 1 - Define a function and create a object of the given window by using the three methods namely config, title and minsize.

Step 2 - Create a button object and use the text and command attribute for triggering the given event and used grid method along with internal and external padding specified, similarly create another button object which will allow application to terminate.

Step 3 - Define second function corresponding to second window with attributes config, title, minsize for the window object and define one button object which will shift the focus onto the third window.

Step 4 - Create third window object and in this create two button object for moving on to the first window for restarting the process

540

and second button for terminating.

Step 5 - Define a function for terminating and call the quit method and finally call the first function with pre-allocated and trigger main loop quit, python method.

```
Button(root, text='Macbooks', bg='black', fg='white',  
relief='flat', font='Verdana 12', width=12).grid(row=2, column=2)
```

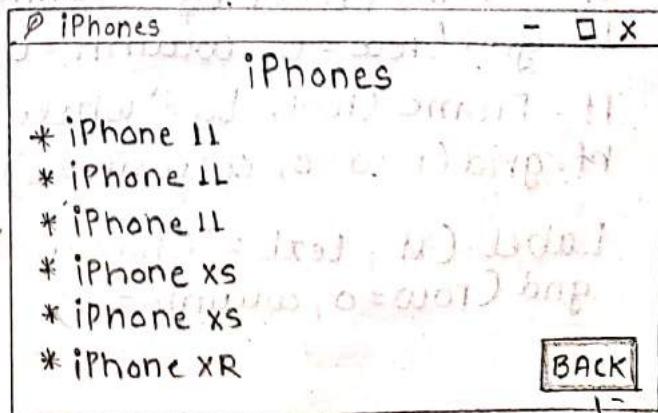
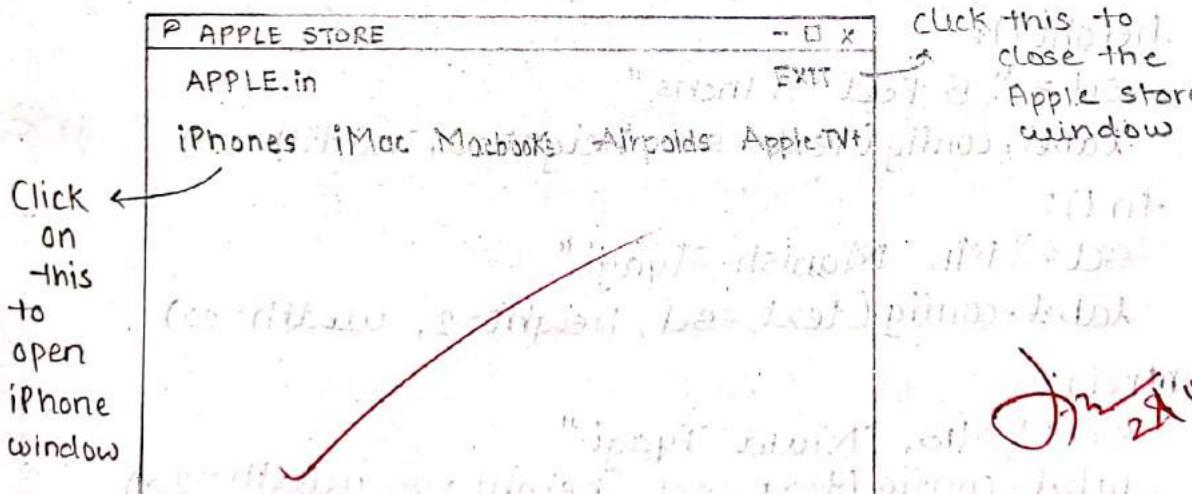
```
Button(root, text='Airpods', bg='black', fg='white', relief='flat', font='Verdana 12', width=12).grid(row=2, column=3)
```

```
Button(root, text='Apple TV+', bg='black', fg='white', relief='flat', font='Verdana 12', width=12).grid(row=2, column=4)
```

```
Button(root, text='EXIT', bg='black', fg='white', relief='flat', font='Verdana 12', width=12, command=root.destroy).grid(row=0, column=5)
```

mainloop()

output:- Python 3.7



Click on this to go back to home window

★ FRAME AND PHOTO IMAGE

84)

Coding :- Python 3.7

```
from tkinter import *
root = Tk()
root.title('Profile')
root.maxsize(height=650, width=900)
root.config(bg='white')
def name():
    sel = "Mr. Abhinav Tyagi"
    label.config(text=sel, height=2, width=20)
def dob():
    sel = "22nd November 1990"
    label.config(text=sel, height=2, width=20)
def add():
    sel = "Lucknow"
    label.config(text=sel, height=2, width=20)
def height():
    sel = "5 Feet 8 Inches"
    label.config(text=sel, height=2, width=20)
def fn():
    sel = "Mr. Manish Tyagi"
    label.config(text=sel, height=2, width=20)
def mn():
    sel = "Mrs. Mala Tyagi"
    label.config(text=sel, height=2, width=20)
lf = Frame(root, bg='white', height=400, width=200)
lf.grid(row=0, column=0)
rf = Frame(root, bg='white', height=400, width=250)
rf.grid(row=0, column=1)
Label(lf, text='Original Photo', height=2, width=20).grid(row=0, column=0)
```

28.01.20

PRACTICAL - 5(4)

Aim :- Demonstrating use of Frame object and PhotoImage.

Algorithm :-

Step 1 - Create an object corresponding to the parent window and use the following three methods :-

- i) title()
- ii) maxsize()
- iii) config()

Step 2 - Create a left frame object from the Frame() method and place it onto the parent window with the height, width, and bg attribute specified, subsequently use the grid method with row, column, padx, pady attribute specified.

Step 3 - Now create a right frame object from the frame() method with the width & height specified and the row and the column value should be specified.

E&O

Step 4 - Create a Label object from the label() method and place it onto the leftframe with the text attribute , denoting 'Original Photo' with relief attribute used as raised value & subsequently use the grid() with row and column value (0,0) with some external padding values.

Step 5 - Now use the PhotoImage() method with the file attribute specified.

Step 6 - Use the subsample() method with the object of the image and give the x and y coordinate values

Step 7 - Use the label() method and position it onto the leftframe and placing the image after the sampling and use the grid() method for positioning in the first row.

Step 8 - Create another label object positioning onto the rightframe and specifying the image and bg attribute with the row and column (0,0)

```
photo1 = PhotoImage (file='kartik.png')
photo1.subsample (3,4) (It's width 050)
photo2 = PhotoImage (file='kartik1.png') (width=100, height=100)
photo2.subsample (3,4)
photo3 = PhotoImage (file='india.png') (width=100, height=100)
photo3.subsample (4,3)

Label (lf, image=photo1, bg='white').grid (row=1,
                                            column=0, padx=10,
                                            pady=10)
Label (rf, image=photo2, bg='white').grid (row=1,
                                            column=0, padx=10,
                                            pady=10)
Label (rf, image=photo3, bg='white').grid (row=0,
                                            column=0, padx=10,
                                            pady=10)
```

```
toolbar = Frame (lf, width=200, height=100, bg='white')
toolbar.grid (row=2, column=0, padx=10, pady=10)
```

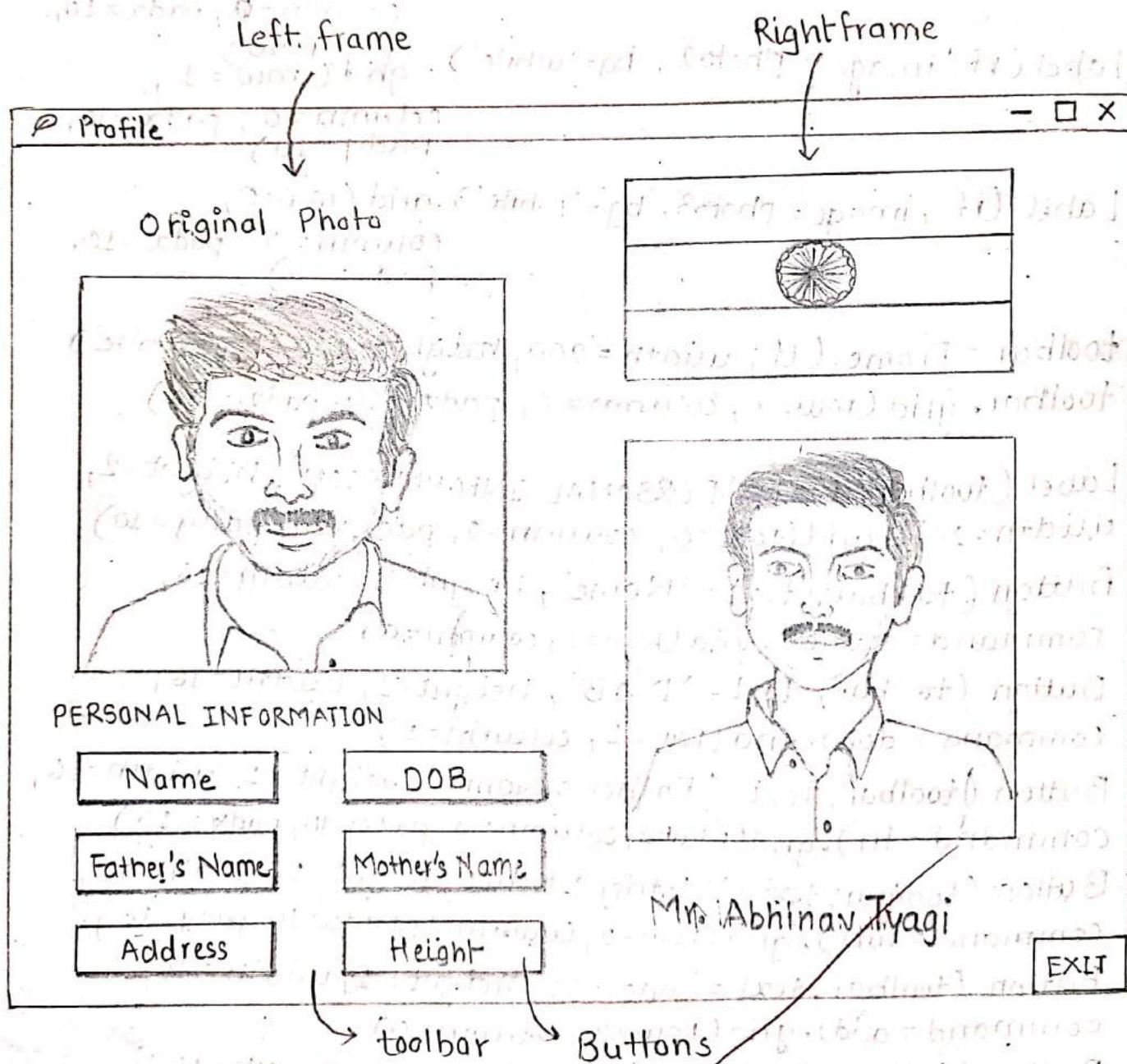
```
Label (toolbar, text='PERSONAL INFORMATION', height=2,
       width=20).grid (row=0, column=0, padx=20, pady=10)
Button (toolbar, text='Name', height=1, width=16,
        command=name).grid (row=1, column=0)
Button (toolbar, text='D O B', height=1, width=16,
        command=dob).grid (row=1, column=1)
Button (toolbar, text='Father's Name', height=1, width=16,
        command=fn).grid (row=2, column=0, padx=10, pady=10)
Button (toolbar, text='Mother's Name', height=1, width=16,
        command=mn).grid (row=2, column=1, padx=10, pady=10)
Button (toolbar, text='Address', height=1, width=16,
        command=add).grid (row=3, column=0)
Button (toolbar, text='Height', height=1, width=16,
        command=height).grid (row=3, column=1)
Button (rf, text='EXIT', height=1, width=5, command=
root.destroy).grid (row=4, column=1)
```

label(Label(rf))

label.grid(row=2, column=0)

mainloop()

Output: - Python 3.7



Step 9 - Now create a toolbar object from the Frame() and position it onto the leftframe with the height and width specified and position it onto the 2nd row.

Step 10 - Now define the various function for the different toolbar option provided in the leftframe.

Step 11 - From the label() method position the text onto the toolbar use relief attribute and the corresponding grid value and incorporate the internal padding as well.

Step 12 - Create label() method position it onto the toolbar with the next title as personal information and position it onto the same row but next column.

Drill

120
11/02/20

PRACTICAL - 5 (5)

Aim - Demonstrating the use of Canvas, Entry, Spinbox and PanedWindow widgets.

Algorithm -

Spinbox()

Step 1 - Create an object from the TK() method and subsequently create an object from the spinbox() method.

Step 2 - Make the object so situated onto the parent window.

PanedWindow()

Step 3 - Create an object from the PanedWindow() method and use the pack or grid method with the attribute fill and expand.

Step 4 - Create an object from the Label() method and put it onto the panedwindow with text attribute and use the add() method to embed the new object.

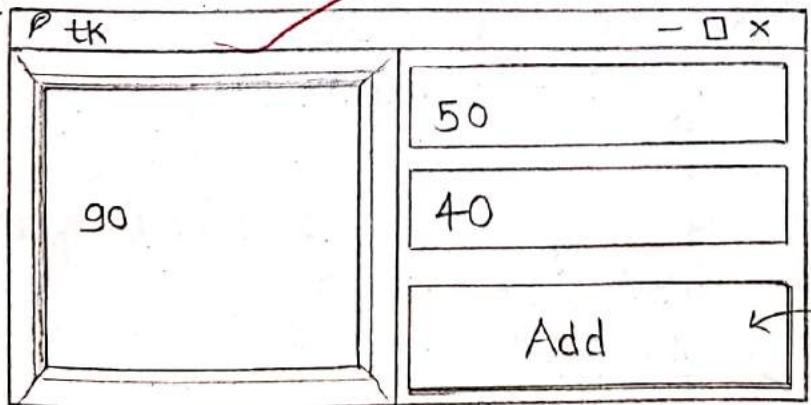
* PanedWindow & Entry

coding : Python 3.8

052

```
from tkinter import *
def add():
    a = int(e1.get())
    b = int(e2.get())
    leftdata = str(a+b)
    leftdata.insert(1, leftdata)
w1 = PanedWindow()
w1.pack(fill=BOTH, expand=1)
left = Entry(w1, bd=5)
w1.add(left)
w2 = PanedWindow(w1, orient=VERTICAL)
w1.add(w2)
e1 = Entry(w2)
e2 = Entry(w2)
w1.add(e1)
w1.add(e2)
bottom = Button(w2, text="Add", command=add)
w2.add(bottom)
mainloop()
```

Output : Python 3.8



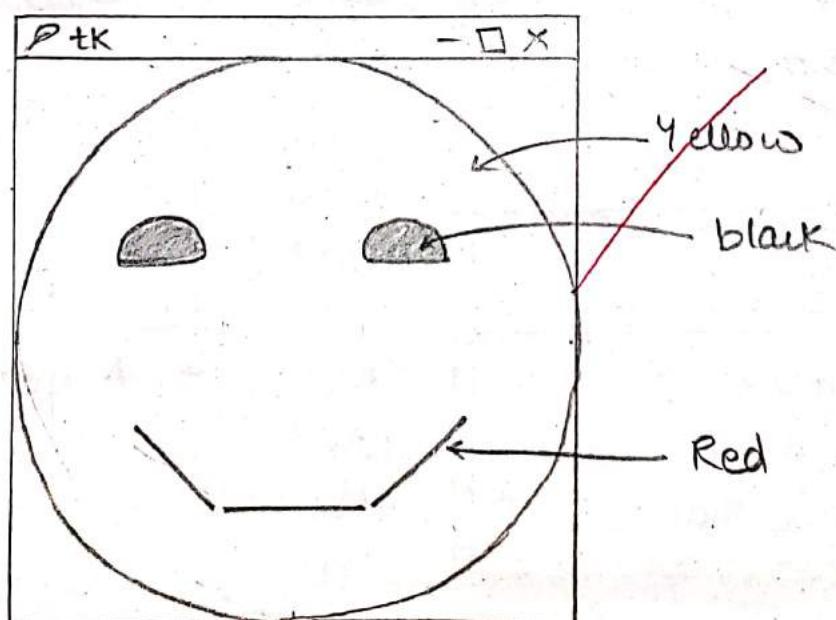
click
on
this

★ Canvas

Coding : Python 3.8

```
from tkinter import *
root = Tk()
c = Canvas(root, bg="White", width=302, height=302)
c.pack()
c.create_oval(2, 2, 300, 300, fill="Yellow")
c.create_arc(200, 100, 250, 150, extent=180, fill="black")
c.create_arc(50, 100, 100, 150, extent=180, fill="black")
c.create_line(50, 200, 110, 240, fill="red", width=5)
c.create_line(110, 240, 190, 240, fill="red", width=5)
c.create_line(190, 240, 250, 200, fill="red", width=5)
root.mainloop()
```

Output : Python 3.8



Step 5 - Similarly, create a second paned window object and add it onto the P1 window with orientation specified.

Step 6 - Now create another label object and place it onto the second paned window object and add it onto the second pane (P2).

Step 7 - Trigger the mainloop.

Canvas()

Step 8 - Create an object from the canvas() method and use the attribute, height, width, bg and parent window object.

Step 9 - Use the method create-line(), create_oval, create_arc(), etc. along with the canvas() object so created and use the coordinate values.

Step 10 - Similarly, use the other method and call the pack method and then mainloop.

Entry()

Step 11 - Create an object from the Entry() method and put it onto the parent window.

Step 12 - Define a function which display the entered value inside the Entrybox using get() method.

Step 13 - Create a Button object using Button() method and call the defined function and display the content.

* Spinbox

054

Coding : Python 3.8

```
from tkinter import *
import calendar

def cal():
    yy = int(s2.get())
    mm = int(s1.get())
    cal = calendar.month(yy, mm)
    t = Text(root, height=8, width=22)
    t.insert(END, cal)
    t.grid(row=4, columnspan=2, padx=20, pady=20)

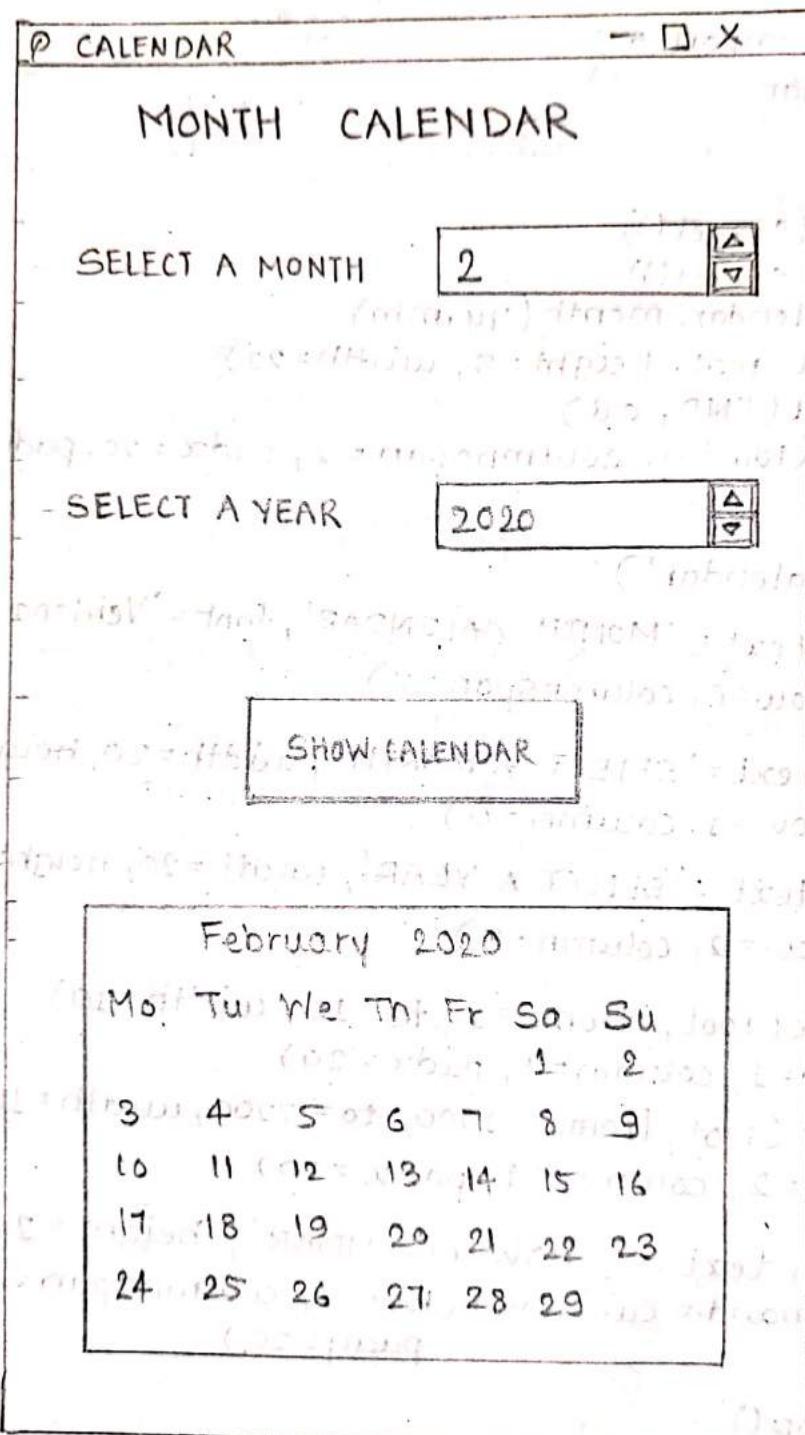
root = Tk()
root.title('Calendar')
Label(root, text='MONTH CALENDAR', font='Verdana 12 bold').grid(row=0, columnspan=2)
Label(root, text='SELECT A MONTH', width=20, height=5).grid(row=1, column=0)
Label(root, text='SELECT A YEAR', width=20, height=5).grid(row=2, column=0)
s1 = Spinbox(root, from_=1, to=12, width=10)
s1.grid(row=1, column=1, padx=20)
s2 = Spinbox(root, from_=1900, to=2200, width=10)
s2.grid(row=2, column=1, padx=20)
Button(root, text='SHOW CALENDAR', height=2, width=20,
       command=cal).grid(row=3, columnspan=2, padx=20, pady=20)

root.mainloop()
```

Jn 25/10

Practical 5 (5)

Output: Python 3.8



25/02/20

PRACTICAL-6

① Aim - Database Connectivity using SQLite3.

Algorithm -

Step 1 - Import the corresponding library for making the database connectivity.

Step 2 - Now create the connection object using sqlite3 library and then connect method for creating the new database.

Step 3 - Now create the cursor object using the cursor method from the connection object so created in the earlier step.

Step 4 - Now use the execute method for creating the table with the column and respective datatype.

Step 5 - Now with the cursor object use the insert statement for entering the values corresponding to the fields considering the datatype.

220

Step 6 - Now use the commit method to complete the transaction using the connection object.

Step 7 - Use the execute statement along with the cursor object for accessing the values from the database "SELECT * FROM review".

Step 8 - Finally, use the fetch() method for displaying the values from the table using the cursor object.

Step 9 - Use the execute method and the DROP TABLE syntax for terminating the database and use the close() method.

* Creating Database and Table :-

056

Coding : Python 3.8 Shell

```
>>> import sqlite3  
>>> sqlite3.connect('sleepless.db')  
<sqlite3.Connection object at 0x0037A767428>  
>>> conn=sqlite3.connect('sleepless.db')  
>>> cur=conn.cursor()  
>>> cur.execute('CREATE TABLE "review" ('  
    "Customer_No" INTEGER NOT NULL PRIMARY  
    KEY AUTOINCREMENT UNIQUE,  
    "Name" TEXT NOT NULL,  
    "Contact_No" INTEGER NOT NULL,  
    "Email_ID" TEXT NOT NULL,  
    "Service" TEXT NOT NULL,  
    "Ratings" INTEGER NOT NULL)  
<sqlite3.Cursor object at 0x000000378276EFGAHT>
```

* Coding : Python 3.8 (using GUI) Insert & Display

```
from tkinter import *  
import sqlite3  
  
def submit():  
    name = str(e1.get())  
    contact = str(e2.get())  
    email = str(e3.get())  
    service = str(s1.get())  
    rating = int(s2.get())  
  
    conn = sqlite3.connect('sleepless.db')  
    cur = conn.cursor()  
  
    cur.execute('INSERT INTO review (Name, ContactNo,  
        Email_ID, Service, Ratings) VALUES  
        (?, ?, ?, ?, ?)', (name, contact, email,  
        service, rating))
```

```
#6) for row in cur.execute('SELECT * FROM review'):  
    print(row)
```

```
conn.commit()
```

```
x = Tk()
```

```
Label(x, text = 'Café SLEEPLESS\nReview Form',  
      font = 'Verdana 12 bold').grid(row=0,  
      columnspan = 2)
```

```
e1 = Entry(x, width = 40)
```

```
e2 = Entry(x, width = 40)
```

```
e3 = Entry(x, width = 40)
```

```
s2 = Scale(x, from_ = 1, to = 5, orient = HORIZONTAL)
```

```
Label(x, text = 'Name', width = 20, height = 3).  
    grid(row = 1, column = 0)
```

```
e1.grid(row = 1, column = 1, padx = 20)
```

```
Label(x, text = 'Contact-No', width = 20, height = 3).  
    grid(row = 2, column = 0)
```

```
e2.grid(row = 2, column = 1, padx = 20)
```

```
Label(x, text = 'Email-ID', width = 20, height = 3).  
    grid(row = 3, column = 0)
```

```
e3.grid(row = 3, column = 1, padx = 20)
```

```
Label(x, text = 'Taste & Service', width = 20,  
      height = 3).grid(row = 4, column = 0)
```

```
s1 = StringVar()
```

```
s1.set('How Was The Food & Service')
```

```
list1 = ['Poor', 'Average', 'Good', 'Better', 'Best', 'Excellent']
```

```
l1 = OptionMenu(x, s1, *list1)
```

```
l1.grid(row = 4, column = 1)
```

```
Label(x, text = 'Overall Ratings', width = 20,  
      height = 3).grid(row = 5, column = 0)
```

```
s2.grid(row=5, column=1)
Button(x, text = 'submit', command = submit).grid(
    row=6, column=1, padx=20, pady=20, sticky=E)
x.mainloop()
```

Output : Python 3.8

Pytk

Café SLEEPLESS
Review Form

Name

Contact-No

Email

Taste & Service

Overall Ratings

Submit

CLICK on this

```
>>> (1, 'Dinesh K Lohar', '9892980666', 'dk123streetest@gmail.com',
      'Excellent', 5)
```

★ Alteration :-

```
>>> cur.execute('ALTER TABLE review ADD "Total"  
TEXT')
```

```
<sqlite3.Cursor object at 0x000037864A6A>
```

★ Drop Table :-

```
>>> cur.execute('DROP TABLE review')
```

```
<sqlite3.Cursor object at 0x0000004784A6B2>
```

★ USE OF DBM :-

Case 1 : (32-bit) Version

```
import dbm  
db= dbm.open("db1","n")  
db["www.wikipedia.org"] = "Encyclopedia"  
if db["www.wikipedia.org"] != None:  
    print("Page Found")  
else:  
    print("Page Not Found")
```

Output :

```
>>> Page Found
```

Case 2 : (32-bit) Version

```
import dbm  
db= dbm.open ("db1","n")  
db["www.patreon.com"] = "Fans"  
if db["www.patreon.com"] == None:  
    print ("Page Found")  
else:  
    print ("Page Not Found")
```

Output :

```
>>> Page Not Found
```

② Aim :- Use of database library

Algorithm :-

Step 1 - Import the dbm library to implement various features of database.

Step 2 - Use the open method creating the database by specifying the name of the database by specifying the name of the database along with the corresponding flag.

Step 3 - Use the object so created for accessing the given website & the corresponding regular name for the website.

Step 4 - Check whether the given URL address with the regular name of the page is not equal to none, then display the message, that URL is found or else not found.

Step 5 - Use the close() to terminate the database.

060

PROJECT

SOURCE CODE -

```
from tkinter import *
import sqlite3

def calculate():
    nam=str(e1.get())
    gen=str(g1.get())
    con='\n\nYour Contact : '+str(e4.get())
    h1=int(e2.get())
    w1=int(e3.get())
    h2=(h1/100)
    h=h2*h2
    bmi=w1/h
    bmix=' Your B.M.I. is '+str(round(bmi))+" kg/m\u00b2"

    if gen=='Male':
        t.insert(END, '\nMr. ')
    else:
        t.insert(END, '\nMs. ')

    t.insert(END, nam)
    t.insert(END, bmix)

    if bmi<=18 :
        t.insert(END, '\n\nWhich is Under Weight')
    elif bmi>=18 and bmi<=25 :
        t.insert(END, '\n\nWhich is Normal Weight')
    elif bmi>=25 and bmi<=30 :
        t.insert(END, '\n\nWhich is Over Weight')
    elif bmi>=30 and bmi<=40 :
        t.insert(END, '\n\nWhich is Obesity')
    else:
        t.insert(END, '\n\nWhich is Morbid Obesity')

    t.insert(END, con)

    name=str(e1.get())
    gender=str(g1.get())
    height=int(e2.get())
    weight=int(e3.get())
    h2=(height/100)
    h=h2*h2
    bmi=weight/h
    bmis=str(round(bmi))+" kg/m\u00b2"
    contact=int(e4.get())

    conn=sqlite3.connect('goldgym.db')

    cur=conn.cursor()
    '''cur.execute('CREATE TABLE "bmi" ("Sr.No" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE, "Name" TEXT NOT NULL,\n"Gender" TEXT NOT NULL, "Height" TEXT NOT NULL, "Weight" TEXT NOT\nNULL,"BMI" TEXT NOT NULL, "Contact_No" INTEGER NOT NULL)''''

    cur.execute('INSERT INTO bmi\n(Name,Gender,Height,Weight,BMI,Contact_No) VALUES\n(?, ?, ?, ?, ?, ?)', (name,gender,height,weight,bmis,contact))

    for row in cur.execute('SELECT *FROM bmi'):
        print(row)

    conn.commit()
```

PROJECT

Aim :- Develop project on real life applications using database operations.

Topic :- B.M.I Calculator for Gyms and personal use.

Synopsis :- This B.M.I calculator is created by using python 3.8.1 and using tkinter and sqlite3 modules or library of python.

* This B.M.I calculator takes the values from the user :

- 1) Name
- 2) Contact
- 3) Gender
- 4) Height (in cms)
- 5) Weight (in kgs)

* After taking these above values from the user this B.M.I calculator calculates the B.M.I (Body Mass Index) using formula :

$$\text{B.M.I} = \frac{\text{Weight (Kg)}}{(\text{Height})^2 (\text{m}^2)}$$

* After calculating the B.H.I it will show the output in the given Textbox with the results showing that you are:

- 1) Underweight
- 2) Normal
- 3) Overweight
- 4) Obesity

* After displaying the results, The entry and B.H.I of user will be stored in the database.

```

x=Tk()
x.title('BMI Calculator')
x.geometry('650x700')
x.maxsize(height=700, width=630)
x.config(bg='black')

Label(x, text="\n\n\n\n", font='Verdana 13 bold', fg='gold',
bg='black').grid(row=0, columnspan=2)

photo=PhotoImage(file='banner.png')
photo.subsample(4,3)

head=PhotoImage(file='head.png')
head.subsample(4,3)

logo=PhotoImage(file='logo.png')
logo.subsample(4,4)

Label(x, image=photo, bg='black').place(x=110, y=15)
Label(x, image=head, bg='black').place(x=200, y=70)
Label(x, image=logo, bg='black').place(x=10, y=20)

e1=Entry(x, width=30, font='Verdana 11 bold')
e2=Entry(x, width=30, font='Verdana 11 bold')
e3=Entry(x, width=30, font='Verdana 11 bold')
e4=Entry(x, width=30, font='Verdana 11 bold')

Label(x, text='Name', width=20, height=3, fg='gold', bg='black',
font='Verdana 11 bold').grid(row=1, column=0)
e1.grid(row=1, column=1, padx=20)

Label(x, text='Gender', width=20, height=3, fg='gold', bg='black',
font='Verdana 11 bold').grid(row=2, column=0)
gen=["Male","Female"]
g1=StringVar(x)
g1.set("Select Your Gender")
g=OptionMenu(x,g1, *gen)
g.config(width=25, relief='flat', fg='gold', bg='black', font='Verdana
11 bold')
g.grid(row=2, column=1, padx=20)

Label(x, text='Height(in cm)', width=20, height=3, fg='gold',
bg='black', font='Verdana 11 bold').grid(row=3, column=0)
e2.grid(row=3, column=1, padx=20)

Label(x, text='Weight(in kg)', width=20, height=3, fg='gold',
bg='black', font='Verdana 11 bold').grid(row=4, column=0)
e3.grid(row=4, column=1, padx=20)

Label(x, text='Contact', width=20, height=3, fg='gold', bg='black',
font='Verdana 11 bold').grid(row=5, column=0)
e4.grid(row=5, column=1, padx=20)

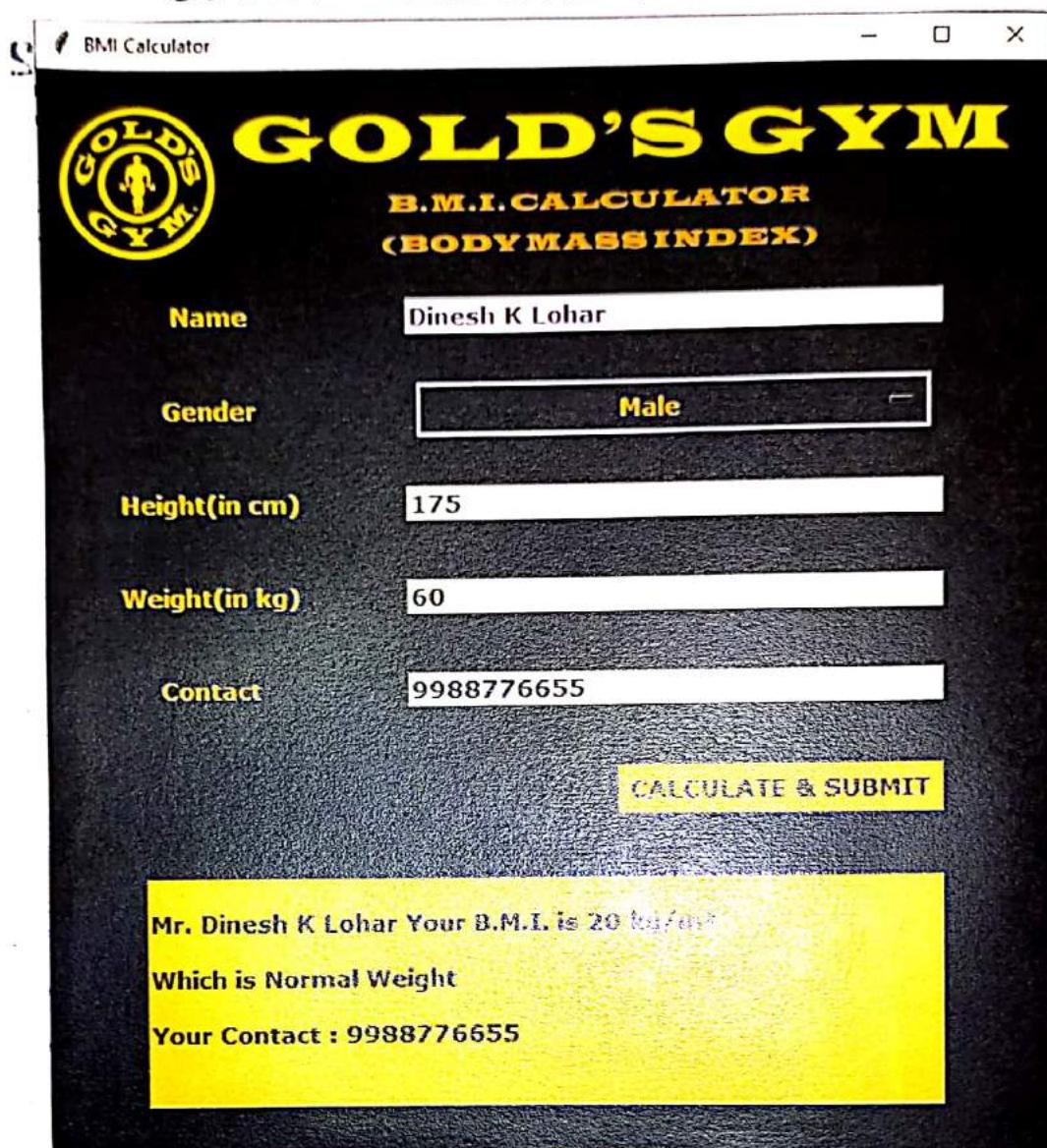
Button(x, text='CALCULATE & SUBMIT', command=calculate, bg='gold',
relief=FLAT, font='Verdana 11 bold').grid(row=6, column=1, pady=20,
padx=20, sticky=E)

t=Text(x, height=8, width=45, bg='gold', font='Verdana 11 bold')
t.grid(row=7, columnspan=2, padx=20, pady=20, sticky=E)

mainloop()

```

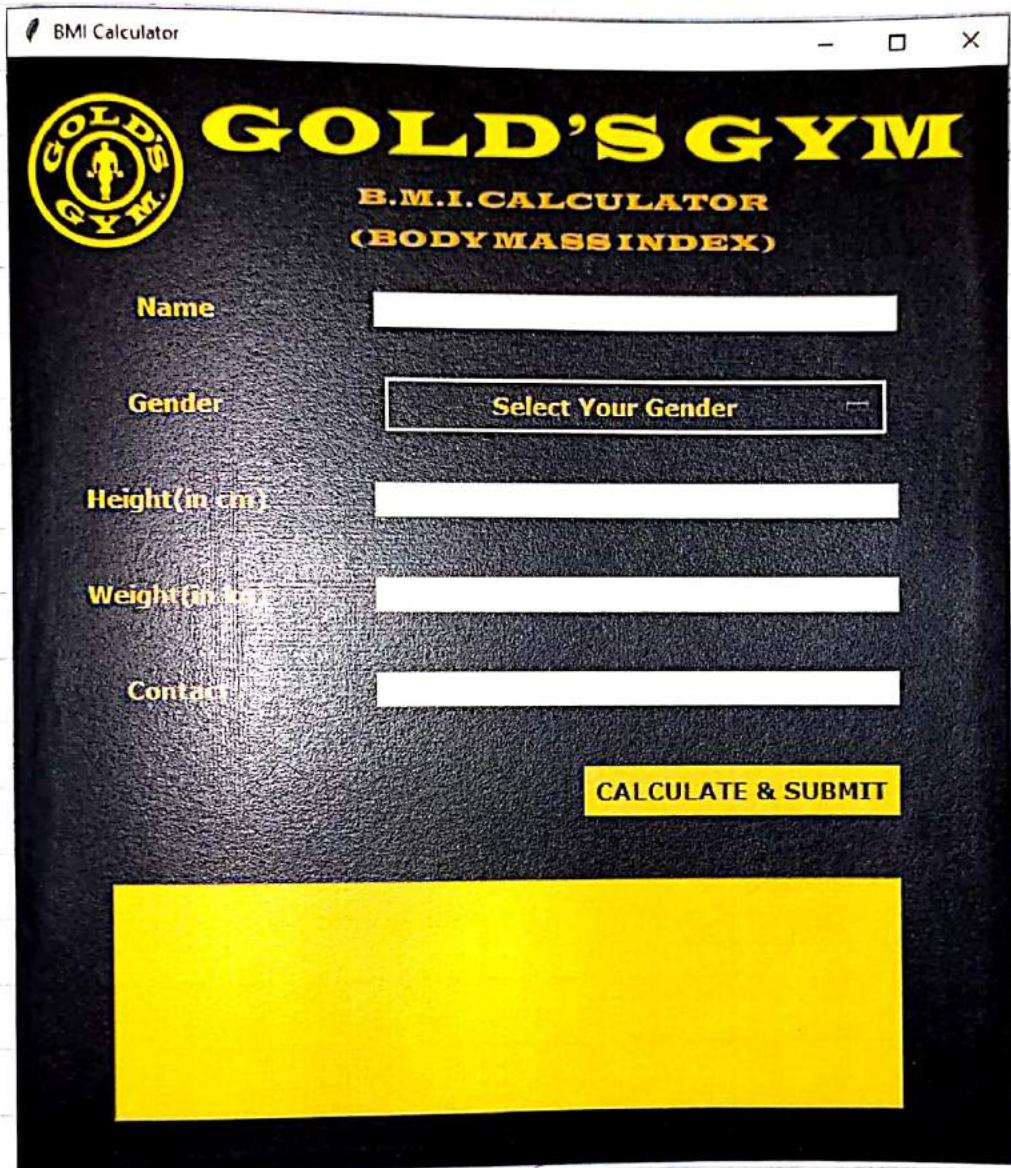
OUTPUT: After filling the Entries and Clicking on Button "CALCULATE & SUBMIT"



Displaying the total entries from the database.

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\GINESH\Desktop\BMI CALULATOR\bmical.py =====
(1, 'Dinesh K Lohar', 'Male', '180', '65', '20 kg/m²', 9892980614)
(2, 'Dharan Shah', 'Male', '175', '55', '18 kg/m²', 7788994455)
(7, 'Ashish Trivedi', 'Male', '165', '55', '20 kg/m²', 8855221144)
(8, 'Dinesh K Lohar', 'Male', '175', '60', '20 kg/m²', 9988776655)
>>> |
```

Output : When we run the program



Dr. GJ