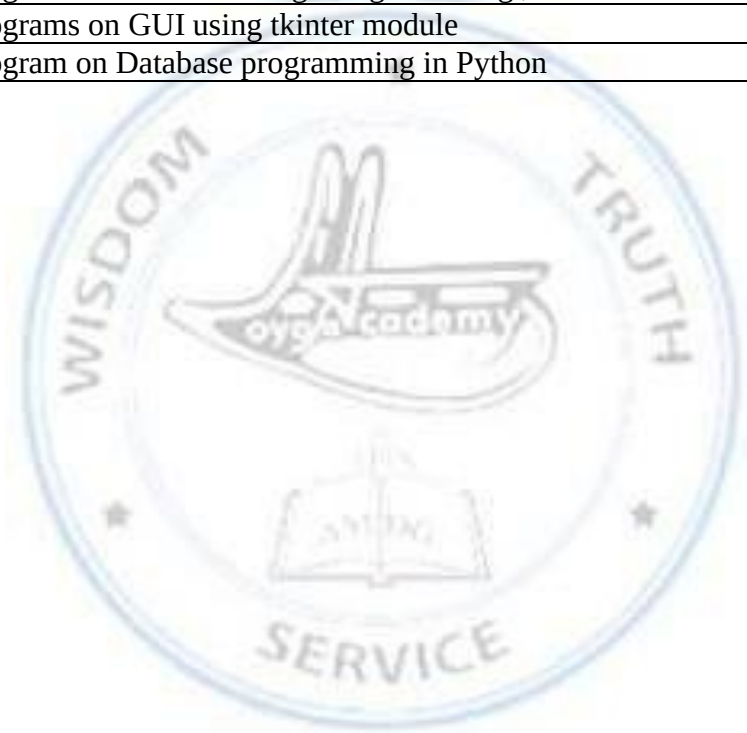
INDEX

S.NO	PROGRAM NAME	Date	SIGN
1	Program to add two integers by taking user inputs		
2	Program to print Student information		
3	Program to demonstrate if a number is even or odd.		
4	Program to demonstrate basic arithmetic operations		
5	Program to demonstrate if else statement		
6	Program to demonstrate if else elif Statement		
7	Program to find the greatest of 3 numbers		
8	Program to find the average of first n natural numbers using FOR loop		
9	Program to print multiplication table using FOR loop		
10	Program to calculate factorial of number using FOR loop		
11	Program to find the sum of the digits of a number using WHILE loop		
12	Program to Create a LIST and search an element in the list		
13	Program to demonstrate LIST operations		
14	Program that displays the first and second largest numbers in a LIST		
15	Program to read the list of numbers and print the lists in reverse order		
16	Program to demonstrate a string is palindrome or not		
17	Program to count the number of vowels and consonants in a String		
18	Program to demonstrate string methods and functions		
19	Program to demonstrate tuple operations		
20	Program to perform adjacent element multiplication in a tuple		
21	Program to demonstrate dictionary built in functions and methods		
22	Program to sort the keys of a Dictionary		
23	Program to find out factorial of a given number using Functions		
24	Program to demonstrate multiple functions		
25	Program to get Current Date and Time Using the Datetime Module		
26	Program to demonstrate math Module.		
27	Program to demonstrate calendar module		
28	Program to handle divide by zero EXCEPTION		
29	Program to demonstrate multiple except block		
30	Program to raise an exception		
31	Program to demonstrate basic File Operations		
32	Program to rename a file and delete a file		
33	Program to read the contents of a text file and display it line wise with a line number		

INDEX

S.NO	PROGRAM NAME	Date	SIGN
34	Program to print each line of a file in reverse Order.		
35	Program to demonstrate Classes and Objects.		
36	Program to demonstrate Single Inheritance		
37	Program to demonstrate Multiple Inheritance		
38	Program to demonstrate Method Overriding		
39	Program to demonstrate Method Overload		
40	Program on Multithreading using threading module		
41	Program on Multithreading using threading , time modules		
42	Programs on GUI using tkinter module		
43	Program on Database programming in Python		



1. Write a program to add two integers by taking user inputs.

```
num1 = int(input("enter the number 1: "))
num2 = int(input("enter the number 2: "))
Result = num1 + num2
print("The Sum of ",num1,"and ",num2,"is ",Result)
```

Output :

```
Enter the number 1: 15
Enter the number 2: 67
The Sum of  15 and  67 is  82
```

```
In [ ]:
```

2. Program to print Student information

```
name=input("Enter your Name :")
full_addr=input("Enter the full address :")
mobile_num=input("Enter your Moblie number :")
clg_name=input("Enter College name :")
course_name=input("Enter Course Name :")


print("**The Student Information**")
print("Name:",name)
print("Full Address :",full_addr)
print("Mobile number :",mobile_num)
print("College Name :",clg_name)
print("Course Name :",course_name)
```

Output

```
Enter your Name :Raghav Reddy
Enter the full address ::House no 3, Street No 40, Alwal, Hyderabad -500010
Enter your Moblie number :2222222222
Enter College name :Loyola Academy
Enter Course Name :B.Com Information Systems
**The Student Information**
Name: Raghav Reddy
Full Address : :House no 3, Street No 40, Alwal, Hyderabad -500010
Mobile number : 2222222222
College Name : Loyola Academy
Course Name : B.Com Information Systems
```

In [ ]:

3. Program to demonstrate if a number is even or odd.

```
n= int(input("Enter a number"))
if(n%2==0):
    print("The number is even")
else:
    print("The number is odd")
```

Output

```
Enter a number45
The number is odd
```

```
Enter a number62
The number is even

In [ ]:
```

4. Program to demonstrate basic arithmetic operations.

```
a=int(input('Enter the value of a :'))
b=int(input('Enter the value of b :'))
add=a+b
diff=a-b
mul=a*b
div=a/b
print('SUM : ',add)
print('DIFFERENCE :  ',diff)
print('MULTIPLICATION : ',mul)
print('DIVISION : ',div)
```

Output

```
Enter the value of a :25
Enter the value of b :40
SUM :  65
DIFFERENCE :   -15
MULTIPLICATION :  1000
DIVISION :  0.625
```

```
In [ ]:
```

5. Program to demonstrate if else statement.

```
age=int(input("Enter the age : "))
if(age>=18):
    print("You are eligible to vote")
else:
    yrs=18-age
    print("You have to wait for another "+str(yrs)+" years to vote")
```

Output

```
Enter the age : 20
You are eligible to vote
```

In [ ]:

```
Enter the age : 16
You have to wait for another 2 years to vote
```

In [ ]:

6 .Program to demonstrate if else elif Statement.

```
ch=input("Enter the character : ")
if(ch=="A" or ch=="E" or ch=="I" or ch=="O" or ch=="U"):
    print(ch," is a vowel")
elif(ch=="a" or ch=="e" or ch=="i" or ch=="o" or ch=="u"):
    print(ch," is a vowel")
else:
    print(ch," is not a vowel")
```

Output :

```
Enter a character : b
b  is not a vowel
```

In [ ]:

```
Enter a character : o
o  is a vowel
```

In [ ]:

7. Program to find the greatest of 3 numbers.

```
n1=int(input("Enter the first number :"))
n2=int(input("Enter the second number :"))
n3=int(input("Enter the third number :"))
if(n1>n2):
    if(n1>n3):
        print(n1," is greater than ",n2," and ",n3)
    else:
        print(n3," is greater than ",n1," and ",n2)
elif(n2>n3):
    print(n2," is greater than ",n1," and ",n3)
else:
    print("The three numbers are equal")
```

Output

```
Enter the first number :12
Enter the second number :56
Enter the third number :19
56  is greater than  12  and  19
```

```
In [ ]:
```

```
Enter the first number :35
Enter the second number :35
Enter the third number :35
The three numbers are equal
```

```
In [ ]:
```

8. Program to find the average of first n natural numbers using FOR loop

```
n=int(input("Enter the value of n : "))
sum1=0
for i in range(1,n+1,1):
    sum1=sum1+i
    avg=sum1/n
print("The sum of first ",n," natural numbers is ",sum1)
print("The average of first ",n," natural numbers is ",avg)
```

Output :

```
Enter the value of n : 8
The sum of first  8  natural numbers is  36
The average of first  8  natural numbers is  4.5
```

In [ ]:

9. Program to print multiplication table using FOR loop

```
num=int(input("Enter the value of n : "))
for i in range(1,11):
    print(num,'*',i,'=',num*i)
```

Output :

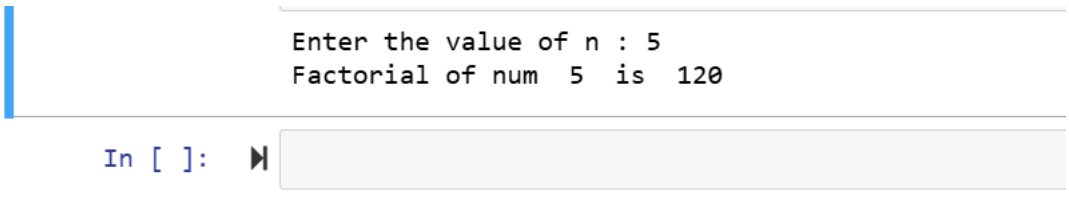
```
Enter the value of n : 6
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
6 * 9 = 54
6 * 10 = 60
```

```
In [ ]:
```

10. Program to calculate factorial of number using FOR loop.

```
num=int(input("Enter the value of n : "))
if(num==0):
    fact=1
fact=1
for i in range(1,num+1):
    fact=fact*i
print("Factorial of num ",num," is ",fact)
```

Output



11. Program to find the sum of the digits of a number using WHILE loop

```
number = int(input("Enter a positive integer: "))
sum_of_digits = 0
if number < 0:
    print("Please enter a positive integer.")
else:
    while number > 0:
        # Extract the last digit
        digit = number % 10
        sum_of_digits += digit
        # Remove the last digit from the number
        number = number // 10
print(f"The sum of the digits is: {sum_of_digits}")
```

Output

```
Enter a positive integer: 7865
The sum of the digits is: 26
```

```
In [ ]:
```

12. Program to Create a LIST and search an element in the list.

```
mylist = []
print("Enter 5 elements for the list: ")
for i in range(5):
    val = int(input())
    mylist.append(val)
print("Enter an element to be search: ")
elem = int(input())
for i in range(5):
    if elem == mylist[i]:
    print("\nElement found at Index:", i)
    print("Element found at Position:", i+1)
```

Output

```
Enter 5 elements for the list:
23
14
67
8
54
Enter an element to be search:
8

Element found at Index: 3
Element found at Position: 4
```

In [ ]:

13. Program to demonstrate LIST operations.

```
list=[1,2,3,4,5,6,7,8]
print('The length of the list is ',len(list))
list.append(9)
print('The new list is ',(list))
print('The element 4 occured ',list.count(4),' times in list')
print('Maximun value in the list is ',max(list))
print('Minimum value in the list is ',min(list))
```

Output

```
The length of the list is  8
The new list is  [1, 2, 3, 4, 5, 6, 7, 8, 9]
The element 4 occured  1  times in list
Maximun value in the list is  9
Minimum value in the list is  1
```

```
In [ ]:
```

14.Program that displays the first and second largest numbers in a LIST

```
L1 = []
print("Enter 10 elements for the list: ")
for i in range(10):
    val = int(input())
    L1.append(val)
m=max(L1)
ind=L1.index(m)
p=L1.pop(ind)
print("the first largest number is ",p)
print("the second largest number is ",max(L1))
```

Output

```
Enter 10 elements for the list:
23
6
89
4
99
156
54
88
230
9
the first largest number is  230
the second largest number is  156
```

```
In [ ]:
```

15.Program to read the list of numbers and print the lists in reverse order

```
List1 = []
num = int(input("Enter Number of elements in the list : "))
print("Enter the elements of the list :")
for i in range(0,num):
    val = int(input())
    List1.append(val)
print ("The Original list is : ")
print(List1)
List2=[]
List2 = List1[::-1]
print("The reverse order of List is ",List2)
```

Output

```
Enter Number of elements in the list : 8
Enter the elements of the list :
34
6
89
2
45
90
21
7
The Original list is :
[34, 6, 89, 2, 45, 90, 21, 7]
The reverse order of List is  [7, 21, 90, 45, 2, 89, 6, 34]
```

In [ ]:

16. Program to demonstrate a string is palindrome or not

```
s=(input('Enter a string : '))
st=s[:: -1]
print(st)
if(st==s):
    print('Palindrome')
else:
    print('Not palindrome')
```

Output

```
Enter a string : This is Python
nohtyP si sihT
Not palindrome
```

In [ ]:

```
Enter a string : MALAYALAM
MALAYALAM
Palindrome
```

In [ ]:

17. Program that accepts a string as an argument and returns the number of vowels and consonants the string contains

```
str = input("Enter any string: ")
vCount = 0
cCount = 0
vowels = ['A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o', 'u']
for i in str:
    if i in vowels:
        vCount = vCount + 1
    else:
        cCount = cCount + 1
print(f"Vowels = {vCount}")
print(f"Consonents = {cCount}")
```

Output:

```
Enter any string: This is my first Program using Strings
Vowels = 8
Consonents = 30
```

```
In [ ]:
```

18.Program to demonstrate string methods and functions

```
str= input("Enter a String :")
print('The capitalized string is ',str.capitalize())
print('The length of string is ',len(str))
print('The maximum of string is ',max(str))
print('The centered string is ',str.center(30,'*'))
print('The number of times msg occurred in the main string
is',str.count("he",0,len(str)))
print('Is string ending with rld :',str.endswith("rld",0,len(str)))
print('Is string ending with hon :',str.endswith("hon",0,len(str)))
print('Is come present in the main string :',str.find("come",0,len(str)))
print('What is index number of m in main string',str.index('m',0,len(str)))
print('The join method is used in this way ','-'.join([str]))
print('Lower case of the string is ',str.lower())
print('Upper case of the string is ',str.upper())
```

Output:

```
Enter a String :hello ! Welcome to the World of Python
The capitalized string is  Hello ! welcome to the world of python
The length of string is  38
The maximum of string is  y
The centered string is  hello ! Welcome to the World of Python
The number of times msg occured in the main string is 2
Is string ending with rld : False
Is string ending with hon : True
Is come present in the main string : 11
What is index number of m in main string 13
The join method is used in this way  hello ! Welcome to the World of Python
Lower case of the string is  hello ! welcome to the world of python
Upper case of the string is  HELLO ! WELCOME TO THE WORLD OF PYTHON
```

```
In [ ]:
```

19.Program to demonstrate tuple operations

```
Tup1 = ('Maths','Chemistry','Physics','Biology')
Tup2 = (456,800,990)
print("Printing tuple elments using Index:\n")
print(Tup1)
print(Tup1[0])
print(Tup1[1:3])
print(Tup1[2: ])
print(Tup1+Tup2)
print("\nUsing tuple Operations:\n")
print('Max value element of Tup1 ',max(Tup1))
print('Max value element of Tup2 ',max(Tup2))
print('Min value element of Tup1 ',min(Tup1))
print('Min value element of Tup2 ',min(Tup2))
print('len value element of Tup1 ',len(Tup1))
print('len value element of Tup2 ',len(Tup2))
```

Output:

```
Printing tuple elments using Index:

('Maths', 'Chemistry', 'Physics', 'Biology')
Maths
('Chemistry', 'Physics')
('Physics', 'Biology')
('Maths', 'Chemistry', 'Physics', 'Biology', 456, 800, 990)

Using tuple Operations:

Max value element of Tup1  Physics
Max value element of Tup2  990
Min value element of Tup1  Biology
Min value element of Tup2  456
len value element of Tup1  4
len value element of Tup2  3
```

In [ ]:

20. Program to perform adjacent element multiplication in a tuple

```
Tup1 = (1, 5, 7, 8, 10, 12, 6, 0)
print(f"The original tuple :  {Tup1}")
res = []
for i in range(len(Tup1) - 1):
    res.append(Tup1[i] * Tup1[i+1])
res = tuple(res)
print(f"Resultant tuple after multiplication :  {res}")
```

Output :

```
The original tuple :  (1, 5, 7, 8, 10, 12, 6, 0)
Resultant tuple after multiplication :  (5, 35, 56, 80, 120, 72, 0)
```

In [ ]:

21.Program to demonstrate dictionary built in functions and methods

```
d1= {'Name': 'Ravi', 'Age': '18', 'Gender': 'M', 'Country': 'India'}
print(d1.get('Name'))
print(d1.get('Gender'))
print(d1.get('Address'))

#Remove the item whose key is Age
d1.pop('Age')
print(d1)

#Remove the last item
d1.popitem()
print(d1)

d2 = {'Name': 'Neha', 'Gender': 'F','Age': '22'}
d1.update(d2)
print(d1)
```

Output :

```
Ravi
M
None
{'Name': 'Ravi', 'Gender': 'M', 'Country': 'India'}
{'Name': 'Ravi', 'Gender': 'M'}
{'Name': 'Neha', 'Gender': 'F', 'Age': '22'}
```

```
In [ ]:
```

22.Program to sort the keys of a Dictionary

```
D1 = {
    86: 'Ravi',
    22: 'Shruthi',
    19: 'Vani',
    6: 'Hari',
    34: 'Radha',
    65: 'Sagar'
}


sorted_keys = sorted(D1.keys())
Result = {}
for key in sorted_keys:
    Result[key] = D1[key]
# Print the sorted dictionary
print("Original Dictionary:", D1)
print("Sorted Dictionary:", Result)
```

Output:

```
Original Dictionary: {86: 'Ravi', 22: 'Shruthi', 19: 'Vani', 6: 'Hari', 34: 'Radha', 65: 'Sagar'}
Sorted Dictionary: {6: 'Hari', 19: 'Vani', 22: 'Shruthi', 34: 'Radha', 65: 'Sagar', 86: 'Ravi'}
```

In [ ]:

23.Program to find out factorial of a given number using Functions

a) Using normal function

```
def factorial(num):
    fact=1
    for i in range(1, num+1):
        fact=fact*i
    return fact
number=int(input("Please enter any number to find factorial: "))
result=factorial(number)
print(f"The factorial of {number} = {result}")
```

Output:

```
Please enter any number to find factorial: 6
The factorial of 6 = 720

In [ ]:
```

b. Using recursive function

```
def recur_factorial(n):
    if n == 1:
        return n
    else:
        return n*recur_factorial(n-1)
num = int(input("Enter the number: "))
if num < 0:
    print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    print("The factorial of", num, "is", recur_factorial(num))
```

Output:

```
Enter the number: 9
The factorial of 9 is 362880
```

```
In [ ]:
```

24. Program to demonstrate multiple functions

```
def calc_square(number):
    return number ** 2
def calc_cube(number):
    return number ** 3
def calc_power(Base, Expo):
    if Expo == 0:
        return 1
    else :
        return (Base*power(Base, Expo-1))


num = int(input("Enter an Number : "))
base = int(input("Enter the value of base : "))
exponent = int(input("Enter the value of exponent : "))
print(f"The square of {num} is {calc_square(num)}")
print(f"The cube of {num} is {calc_cube(num)}")
print(f"{base} raised to the power of {exponent} is {calc_power(base, exponent)}")
```

Output :

```
Enter an Number : 5
Enter the value of base : 4
Enter the value of exponent : 3
The square of 5 is 25
The cube of 5 is 125
3 raised to the power of 4 is 81
```

```
In [ ]:
```

25. Program to get Current Date and Time Using the Datetime Module

```
import datetime
current_time = datetime.datetime.now()
print("The Current date and Time are :")
print("Year :", current_time.year)
print("Month : ", current_time.month)
print("Day : ", current_time.day)
print("Hour : ", current_time.hour)
print("Minute : ", current_time.minute)
print("Second :", current_time.second)
print("Microsecond :", current_time.microsecond)
```

Output:

```
The Current date and Time are :
Year : 2024
Month :  12
Day :  30
Hour :  20
Minute :  0
Second : 21
Microsecond : 378667
```

```
In [ ]:
```

26. Program to demonstrate math Module.

```
import math
number = 16
sqrt_result = math.sqrt(number)
print(f"Square root of {number} is {sqrt_result}")
n = 5
factorial_result = math.factorial(n)
print(f"Factorial of {n} is {factorial_result}")
angle_in_degrees = 45
sine_result = math.sin(math.radians(angle_in_degrees))
print(f"Sine of {angle_in_degrees} degrees is {sine_result}")
cosine_result = math.cos(math.radians(angle_in_degrees))
print(f"Cosine of {angle_in_degrees} degrees is {cosine_result}")
x = 10
log_result = math.log(x, 2)
print(f"Logarithm of {x} base 2 is {log_result}")
```

Output :

```
Square root of 16 is 4.0
Factorial of 5 is 120
Sine of 45 degrees is 0.7071067811865476
Cosine of 45 degrees is 0.7071067811865476
Logarithm of 10 base 2 is 3.3219280948873626
```

```
In [ ]:
```

27.Program to demonstrate calendar module

```
import calendar
text_cal = calendar.TextCalendar(firstweekday = 0)
# printing prmonth
print(text_cal.prmonth(2024, 12, w = 5))
```

Output :

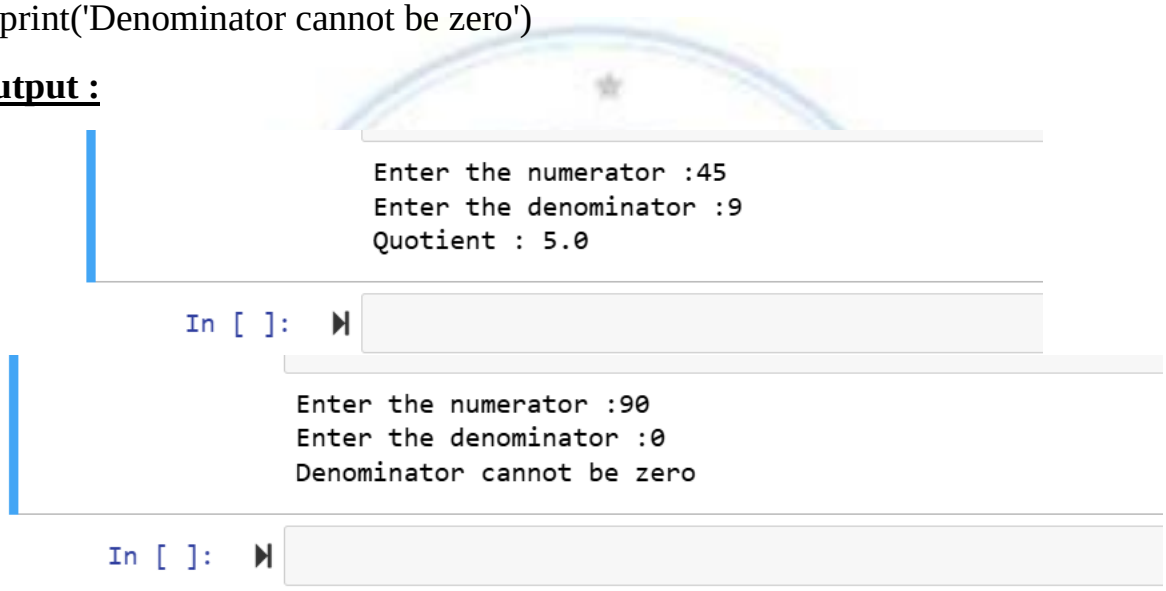
```
                December 2024
  Mon    Tue    Wed    Thu    Fri    Sat    Sun
                                              1
    2      3      4      5      6      7      8
    9     10     11     12     13     14     15
   16     17     18     19     20     21     22
   23     24     25     26     27     28     29
   30     31
None
```

In [ ]:

28. Program to handle divide by zero exception

```
num=int(input('Enter the numerator :'))
demo=int(input('Enter the denominator :'))
try:
    quo=num/demo
    print('Quotient :',quo)
except ZeroDivisionError:
    print('Denominator cannot be zero')
```

Output :



29. Program to demonstrate multiple except block

a.) In separate blocks

```
try:
    num=int(input('Enter the numerator :'))
    print(num**2)
except KeyboardInterrupt:
    print('You should have entered a number....Program terminating')
except ValueError:
    print('Please check before you enter....Program terminating')
print('END')
```

Output :

```
Enter the numerator :75
5625
END
```

In [ ]:

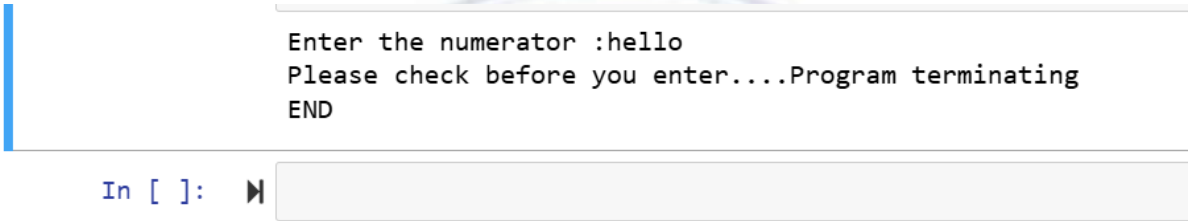
```
Enter the numerator :
Please check before you enter....Program terminating
END
```

In [ ]:

b.)In single block

```
try:
    num=int(input('Enter the numerator :'))
    print(num**2)
except (KeyboardInterrupt , ValueError):
    print('Please check before you enter....Program terminating')
print('END')
```

Output:



30.Program to raise an exception

```
def check_age(age):
    if age < 0:
        raise ValueError("Age cannot be negative!")
    print(f"Valid age: {age}")
try:
    user_age = int(input("Enter Age"))
    check_age(user_age)
except ValueError as e:
    print(f"An error occurred: {e}")
```

Output:

```
Enter Age-65
An error occurred: Age cannot be negative!
```

```
In [ ]:
```

31.Program to demonstrate basic File Operations

```
with open("example.txt", "w") as file:
    file.write("Hello, this is a file operation example.\n")
    file.write("This is the first line.\n")

# Read from the file
with open("example.txt", "r") as file:
    print("Name of the file :",file.name)
    print("File has been opened in ",file.mode,"mode")
    print("Reading the file content:")
    content = file.read()
    print(content)

# Append to the file
with open("example.txt", "a") as file:
    file.write("This is an appended line.\n")

# Read the updated file content
with open("example.txt", "r") as file:
    print("Updated file content:")
    content = file.read()
    print(content)
    file.close()
```

Output:

```
Name of the file : example.txt
File has been opened in  r mode
Reading the file content:
Hello, this is a file operation example.
This is the first line.

Updated file content:
Hello, this is a file operation example.
This is the first line.
This is an appended line.
```

In [ ]:

32. Program to rename a file and delete a file.

```
Import os
def rename_file(old_name, new_name):
    try:
        os.rename(old_name, new_name)
        print(f"File '{old_name}' renamed to '{new_name}'")
    except FileNotFoundError:
        print(f"Error: File '{old_name}' not found.")
def delete_file(file_name):
    try:
        os.remove(file_name)
        print(f"File '{file_name}' deleted successfully.")
    except FileNotFoundError:
        print(f"Error: File '{file_name}' not found.")
old_file_name = "old_file.txt"
new_file_name = "new_file.txt"
with open(old_file_name, 'w') as file:
    file.write("This is a test file.")
rename_file(old_file_name, new_file_name)
delete_file(new_file_name)
```

Output:

```
File 'old_file.txt' renamed to 'new_file.txt'
File 'new_file.txt' deleted successfully.

In [ ]:
```

33.Program to read the contents of a text file and display it on the screen line wise with a line number

```
file=open("file1.txt","r+")
count =0
for I in file.readlines():
    count+=1
    print(count,":",i)
```

Output :

```
1 : A directory is a collection of files and subdirectories.

2 : A directory inside a directory is known as a subdirectory.

3 : Python has the os module.

4 : It provides us with many useful methods to work with directories.

5 : We can get the present working directory using the getcwd() method of the os module.

6 : This method returns the current working directory in the form of a string.
```

```
In [ ]:
```

34.Program to print each line of a file in reverse

```
import os
file=open("file1.txt","r+")
for I in file.readlines():
    print(i[::-1])
```

Output :

```
 .seirotceridbus dna selif fo noitcelloc a si yrotcerid A

.yrotceridbus a sa nwonk si yrotcerid a edisni yrotcerid A

.eludom so eht sah nohtyP

.seirotcerid htiw krow ot sdohtem lufesu ynam htiw su sedivorp tI

.eludom so eht fo dohtem )(dwcteg eht gnisu yrotcerid gnikrow tneserp eht teg nac eW
.gnirts a fo mrof eht ni yrotcerid gnikrow tnerruc eht snruter dohtem sihT
```

In [ ]:

35.Program to demonstrate Classes

```
class Circle:
    def area(self,radius):
        print("Area of the Circle is:",3.14*radius*radius)
    def perimetre(self,radius):
        print("Perimeter of the Circle is:",2*3.14*radius)

radius=float(input("Enter the radius: "))
cir=Circle()
cir.area(radius)
cir.perimetre(radius)
```

Output :

```
Enter the radius: 10
Area of the Circle is: 314.0
Perimeter of the Circle is: 62.800000000000004
```

```
In [ ]:
```

36.Program to demonstrate Single Inheritance

```
class Person:
    def __init__(self, name, idnumber):
        self.name = name
        self.idnumber = idnumber
    def display(self):
        print("Name : ", self.name)
        print("ID : " ,self.idnumber)

# Child Class
class Employee(Person):
    def __init__(self, name, idnumber, salary, post):
        super().__init__(name, idnumber)
        self.salary = salary
        self.post = post

emp = Employee("Suryakanth",250, 55000, "Manager")
print ("Employee Details:")
emp.display()
print("Salary : ",emp.salary)
print("Post : " ,emp.post)

emp1 = Employee("Niharika",150, 72000, "CEO")
print ("\nEmployee Details:")
emp1.display()
print("Salary : ",emp1.salary)
print("Post : " ,emp1.post)
```

Output :

```
Employee Details:
Name :  Suryakanth
ID :  250
Salary :  55000
Post :  Manager

Employee Details:
Name :  Niharika
ID :  150
Salary :  72000
Post :  CEO
```

In [ ]:

37. Program to demonstrate Multiple Inheritance

```
class Employee():
    def __init__(self, name, Id,salary):
        self.name = name
        self.Id = Id
        self.salary = salary
    def display(self):
        print("Name :", self.name)
        print("ID :", self.Id)
        print("Salary :", self.salary)
class Job:
    def __init__(self, salary, Post):
        self.salary = salary
        self.Post = Post
    def show(self):
        print("Post : ", self.Post)
class EmployeeJob(Employee, Job):  # Inherits from both Employee and Job
    def __init__(self, name,Id, salary, Post):
        Employee.__init__(self, name,Id, salary)  # Initialize Employee
        Job.__init__(self, salary, Post)


emp = EmployeeJob("Alice", 100, 50000, "Salesman")
emp.display()
emp.show()
```

Output :

```
Name : Alice
ID : 100
Salary : 50000
Post :  Salesman
```

```
In [ ]:
```

38.Program to demonstrate Method Overriding

```
class Animal:
    def sound(self):
        print("Animal makes a sound")
class Dog(Animal):
    def sound(self): # Method is overridden
        print("Dog barks")
dog = Dog()
dog.sound()
```

Output :

```
Dog barks

In [ ]:
```

39.Program to demonstrate Method Overload

```
class OverloadExample:
    def add(self, a, b, c=0, d=0):
        return a + b + c + d
oe = OverloadExample()
result1 = oe.add(5, 10)
print(result1)
result2 = oe.add(5, 10, 15)
print(result2)
result3 = oe.add(5, 10, 15, 4)
print(result3)
```

Output:

```
15
30
34
```

```
In [ ]:
```

40. Program on Multithreading using threading module

```
import threading
def print_cube(num):
    print(“Cube : “, num**3)
def print_square(num):
    print(“Square:”, num**2)
if __name__ ==”__main__”:
    t1 = threading.Thread(target=print_square, args=(10,))
    t2 = threading.Thread(target=print_cube, args=(5,))
    t1.start()
    t2.start()
    t1.join()
    t2.join()
print(“Done!”)
```

Output:

```
Square: 100
Cube :  125
Done!
```

```
In [ ]:
```

41. Program on Multithreading using threading , time modules

```
import threading
import time


def print_numbers():
    for i in range(5):
        print(f"Thread 1: {i}")
        time.sleep(1)


def print_letters():
    for letter in 'abcde':
        print(f"Thread 2: {letter}")
        time.sleep(1)
thread1 = threading.Thread(target=print_numbers)
thread2 = threading.Thread(target=print_letters)
thread1.start()
thread2.start()
thread1.join()
thread2.join()
print("Done!")
```

Output :

```
Thread 1: 0
Thread 2: a
Thread 1: 1
Thread 2: b
Thread 1: 2
Thread 2: c
Thread 1: 3
Thread 2: d
Thread 1: 4
Thread 2: e
Done!
```

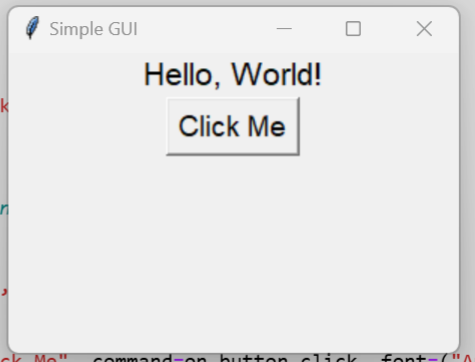
```
In [ ]:
```

42. Programs on GUI using tkinter module

```
import tkinter as tk

# Function when button is clicked
def on_button_click():
    label.config(text="Button clicked!")


root = tk.Tk()
root.title("Simple GUI")
root.geometry("300x200")  # Set window size
label = tk.Label(root, text="Hello, World!", font=("Arial", 16))
label.pack()
button = tk.Button(root, text="Click Me", command=on_button_click, font=("Arial", 14))
button.pack()
root.mainloop()
```

Output:

```
In [*]: import tkinter as tk

# Function when button is clicked
def on_button_click():
    label.config(text="Button click

root = tk.Tk()
root.title("Simple GUI")
root.geometry("300x200")  # Set win


label = tk.Label(root, text="Hello,
label.pack()

button = tk.Button(root, text="Click Me", command=on_button_click, font=("Arial", 14))
button.pack()

root.mainloop()
```



43. Program on Database programming in Python

```
import sqlite3
connection = sqlite3.connect('example.db')
cursor = connection.cursor()
connection.execute('''CREATE TABLE customer_address
        (ID INT PRIMARY KEY     NOT NULL,
        NAME           TEXT    NOT NULL,
        AGE            INT     NOT NULL,
        ADDRESS        CHAR(50)); ''')
connection.execute(
   "INSERT INTO customer_address VALUES (1, 'nikhil teja', 22, 'hyderabad' )")
connection.execute(
   "INSERT INTO customer_address VALUES (2, 'karthik', 25, 'khammam')")
connection.execute(
   "INSERT INTO customer_address VALUES (3, 'sravan', 22, 'ponnur' )")
connection.execute(
   "INSERT INTO customer_address VALUES (4, 'deepika', 25, 'chebrolu' )")
connection.execute(
   "INSERT INTO customer_address VALUES (5, 'jyothika', 22, 'noida')")

cursor = connection.execute("SELECT * FROM customer_address ")
for i in cursor:
   print(i)
connection.close()
```

Output :

```
(1, 'nikhil teja', 22, 'hyderabad')
(2, 'karthik', 25, 'khammam')
(3, 'sravan', 22, 'ponnur')
(4, 'deepika', 25, 'chebrolu')
(5, 'jyothika', 22, 'noida')
```

In [ ]: