PYTHON CORE PROJECT

1.

AIM :

Python | Printing different messages by using different variations of print() method

CODE :

```
print("My name is Animesh.")
print('My name is Animesh.')
print("""My name is Animesh.""")
print('My name is Animesh.', end = '***')
print("My name is Animesh.")
```

OUTPUT :

```
My name is Animesh.
My name is Animesh.
My name is Animesh.
My name is Animesh.***My name is Animesh.
```

2.

AIM :

Python | Printing different values (integer, float, string, Boolean)

CODE :

```
a = 12
b = 12.43
c = 'say'
d = False
e = complex(21,32)
print(a)
print(type(a))
print(id(a))
print(b)
print(type(b))
print(id(b))
print(c)
print(type(c))
print(id(c))
print(d)
```

```
print(type(d))
print(id(d))
print(e)
print(type(e))
print(id(e))
```

OUTPUT :

```
12
<class 'int'>
140732199363568
12.43
<class 'float'>
2004317617296
say
<class 'str'>
2004359033608
False
<class 'bool'>
140732198844784
(21+32j)
<class 'complex'>
2004358466224
```

3.

AIM :

Python | Declare different types of variables, print their values, types and Ids

CODE :

```
a = print(21)
b = print('sad')
c = print(12.23)
d = print(False)
```

OUTPUT :

```
21
sad
12.23
False
```

4.

AIM :

Create number variables (int, float and complex) and print their types and values in Python

CODE :

```
a = 12
b = 12.43
e = complex(21,32)
print(a)
print(type(a))
print(id(a))
print(b)
print(type(b))
print(id(b))
print(e)
print(type(e))
print(id(e))
```

OUTPUT :

```
12
<class 'int'>
140732199363568
12.43
<class 'float'>
2196677770384
(21+32j)
<class 'complex'>
2196689259184
```

5.

AIM :

Determine the type of an object in Python

CODE :

```
a = 10
b = 10.23
c = "Hello"
print("type(a): ", type(a))
print("type(b): ", type(b))
print("type(c): ", type(c))
```

OUTPUT :

```
type(a):  <class 'int'>
type(b):  <class 'float'>
type(c):  <class 'str'>
```

6.

AIM :

Create an integer variable by assigning binary value in Python

CODE :

```
print(bin(23))
a = 0b10111
print(a)
```

OUTPUT :

```
0b10111
23
```

7.

AIM :

Create an integer variable by assigning octal value in Python

CODE :

```
print(oct(23))
a = 0o27
print(a)
```

OUTPUT :

```
0o27
23
```

8.

AIM :

Create an integer variable by assigning hexadecimal value in Python

CODE :

```
print(hex(23))
a = 0x17
print(a)
```

OUTPUT :

```
0x17
23
```

9.

AIM :

Python | Typecasting Input to Integer, Float

CODE :

```
a = int(input("Enter number: "))
print(a)
```

OUTPUT :

```
Enter number: 5
5
```

10.

AIM :

Python | Program to define an integer value and print it

CODE :

```
a = 12
print(type(a))
print(a)
```

OUTPUT :

```
<class 'int'>
12
```

11.

AIM :

Python | Input two integers and find their addition

CODE :

```
a = 10
b = 5
c = a+b
print("Your sum is : ",c)
```

OUTPUT :

```
Your sum is :  15
```

12.

AIM :

Python program to find sum of two numbers

CODE :

OUTPUT :

13.

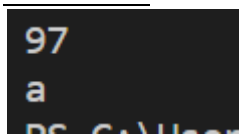
AIM :

Python program to print ASCII value of a character

CODE :

```
print(ord('a'))
print(chr(97))
```

OUTPUT :



14.

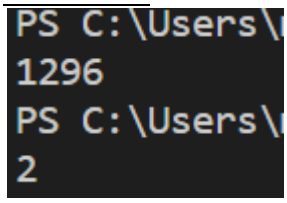
AIM :

Python program to find power of a number using exponential operator

CODE :

```
print(pow(6,4))
```

OUTPUT :



15.

AIM :

Python program to find floor division

CODE :

```
a = 25//12
print(a)
```

OUTPUT :

```
PS C:\Users\
1296
PS C:\Users\
2
```

16.

AIM :

Python | Some of the example of simple if else.

CODE :

```
a = 23
b = 34

if (a>b):
    print("First value is greater than Second.")
else:
    print("Second value is greater than first."
```

OUTPUT :

```
PS C:\Users\my laptop> python -u "c:\
Second value is greater than first.
PS C:\Users\my laptop> python -u "c:\
```

17.

AIM :

Python | Input age and check eligibility for voting.

CODE :

```
a = int(input("Enter your Age : "))
print("You are eligibale for voting.") if a>=18 else print("You are not eligibale for voting.")
```

OUTPUT :

```
Enter your Age : 18
You are eligibale for voting.
```

18.

AIM :

Python | Find largest of three number using nested if else.

CODE :

```
a=int(input("Enter A =  "))
b=int(input("Enter B = "))
c=int(input("Enter C = "))
if(a>b):
    if(a>c):
```

```
        print(a," is Greatest number.")
    else:
        print(c," is Greatest number.")
else:
    if(b>c):
        print(b," is Greatest number.")
```

OUTPUT :

```
Enter A =  12
Enter B = 2
Enter C = 1
12  is Greatest number.
```

19.

AIM :

Python | Calculate discount based on the sale amount.

CODE :

```
a = int(input('Enter total bill amount :'))
b = 25*a/100
c = a - b
print(f'Your discount is {b} ')
print(f'Your final bill is {c}')
```

OUTPUT :

```
Enter total bill amount :10000
Your discount is 2500.0
Your final bill is 7500.0
```

20.

AIM :

Python | Calculate discount based on the sale amount using Nested if else.

CODE :

```
a = int(input('Enter total bill amount :'))
if(a<=1000):
    b = 25*a/100
    c = a - b
    print(f'Your discount is {b} ')
    print(f'Your final bill is {c}')
    if(a<250):
        print("You are not get discount.")
        print(f'your final bill is {a}')
```

```
else :
    b = 50*a/100
    c = a-b
    print(f'Your discount is {b} ')
    print(f'Your final bill is {c}')
```

OUTPUT :

```
Enter total bill amount :1
Your discount is 5000.0
Your final bill is 5000.0
```

21.

AIM :

Python | Example of Ternary Operator

CODE :

```
a = int(input('Enter first number : '))
b = int(input('Enter second number : '))
print('both are same.' if a==b else f'{a} number is greatest'
    if a>b else f'{b} number is greatest.')
```

OUTPUT :

```
Enter first number : 12
Enter second number : 23
23 number is greatest.
```

22.

AIM :

Python | Design a simple calculator using if elif (just like switch case)

CODE :

```
a=int(input("Enter value of A : "))
d=input("Enter your sign : ")
b=int(input("Enter value of B : "))
```

```
if(d == '+'):
    c=a+b
    print("Your Sum is ",c)

elif(d == '-'):
    c=a-b
    print("Your Substraction is  ",c)

elif(d == '*'):
    c=a*b
    print("Your mutification is ",c)

elif(d == '/'):
    c=a/b
    print("Your sum is ",c)

elif(d == '//'):
    c=a//b
    print("Your sum is ",c)

elif(d == '%'):
    c=a%b
    print("Modular is  ",c)

else :
    print("You Enter invalid syntex.")
```

OUTPUT :

```
Enter value of A : 23
Enter your sign : 34
Enter value of B : 4
You Enter invalid syntex.
```

23.

AIM :

Python | Demonstrate an example of for loop

CODE :

```
or i in range(1,6):
    for j in range(1,i+1):
        print("*" , end = " ")
    print(" ")
```

OUTPUT :

24.

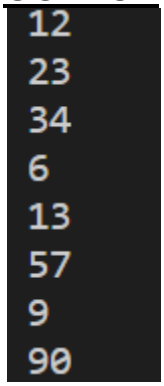
AIM :

Python | Demonstrate an example of for each loop

CODE :

```
values = [12,23,34,6,13,57,9,90]
for fruit in values:
    print(fruit)
```

OUTPUT :



25.

AIM :

Python | Examples of loops (based on their control)

CODE :

```
a=1
while a<=10:
```

```
    print(a)
    a=a+1
```

OUTPUT :

```
1
2
3
4
5
6
7
8
9
10
```

26.

AIM :

Python | Some of the Examples of loops

CODE :

```
n=int(input("Enter N: "))

for i in range(1,n+1):
    print(i)
```

OUTPUT :

```
Enter N: 10
1
2
3
4
5
6
7
8
9
10
```

27.

AIM :

Python | Demonstrate an Example of break statement

CODE :

```
for i in 'animesh':
    if(i=='m'):
        break
    print(i)

print('loop is break.')
```

OUTPUT :

```
a
n
i
loop is break.
```

28.

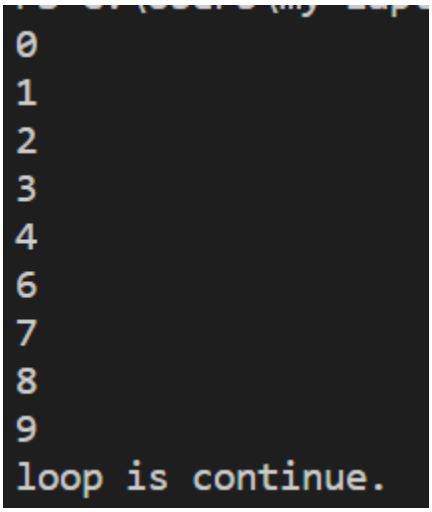
AIM :

Python | Demonstrate an Example of continue statement

CODE :

```
for i in range(0,10):
    if(i==5):
        continue
    print(i)
print('loop is continue.')
```

OUTPUT :



29.

AIM :

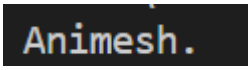
Python | Demonstrate an Example of pass statement

CODE :

```
def get():
    pass

print("Animesh.")
```

OUTPUT :



30.

AIM :

Python | Program to print numbers from N to 1 (use range() with reverse order).

CODE :

```
a = int(input("Enter your starting position of reverse chain : "))

for i in range(a,0,-1):
    print(i)
```

OUTPUT :

```
Enter your starting position of reverse chain : 13
13
12
11
10
9
8
7
6
5
4
3
2
1
```

31.

AIM :

Python | Print all numbers between 1 to 1000 which are divisible by 7 and must not be divisible by 5.

CODE :

```
for i in range(0,1001):
    if (i%7==0):
        if(i%5!=0):
            print(i)
```

OUTPUT :

7	224	441	658	798
14	231	448	672	812
21	238	462	679	819
28	252	469	686	826
42	259	476	693	833
49	266	483	707	847
56	273	497	714	854
63	287	504	721	861
77	294	511	728	868
84	301	518	742	882
91	308	532	749	889
98	322	539	756	896
112	329	546	763	903
119	336	553	777	917
126	343	567	784	924
133	357	574	791	931
147	364	581	798	938
154	371	588	812	952
161	378	602	819	959
168	392	609	826	966
182	399	616	833	973
189	406	623	847	987
196	413	637	854	994
203	427	644	861	
217	434	651	868	

32.

AIM :

Python | Calculate square of a given number (3 different ways).

CODE :

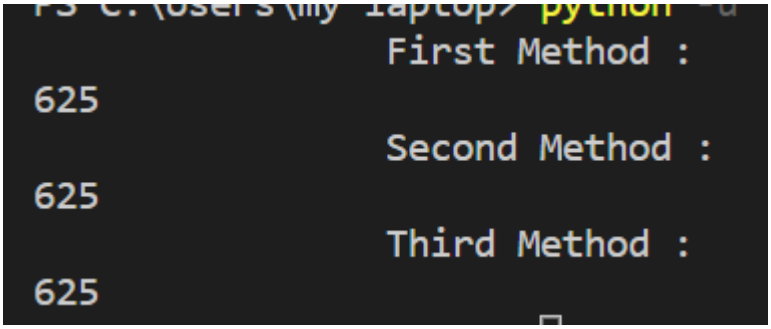
```
a = int(25)
```

```
print("\t\tFirst Method : \t\t")
print(a*a)

print("\t\tSecond Method : \t\t")
print(a**2)

print("\t\tThird Method : \t\t")
import math
print(int(math.pow(a,2)))
```

OUTPUT :



33.

AIM :

Python | Find factorial of a given number (2 different ways).

CODE :

```
n = int(7)
print("\t\tFirst Methadod of find Factorial")
def Factorial(a):
    if(a==1):
        return a
    else:
        return a*Factorial(a-1)

print(f"the factorial of {n} is ",Factorial(n))

print("\t\tSecond Methadod of find Factorial")
import math
print(f"the factorial of {n} is ",math.factorial(n))
```

OUTPUT :

```
                First Methadod of find Factorial
the factorial of 7 is  5040
                Second Methadod of find Factorial
the factorial of 7 is  5040
```

34.

AIM :

Python | Write functions to find square and cube of a given number.

CODE :

```
def calc(self):
    a = int(7)
    b = a**2
    print(f"Your {a} square is ",b)
    c = a**3
    print(f"Your {a} square is ",c)

calc()
```

OUTPUT :

```
Your 7 square is  49
Your 7 square is  343
```

35.

AIM :

Python | Declare any variable without assigning any value.

CODE :

```
a = None
b = 0
print(a)
print(b)
```

OUTPUT :

```
None
0
```

36.

AIM :

Python | Write a function to find sum of two integral numbers in string format.

CODE :

```
def sum():
    a = '12'
    b = '13'
    c = int(a)+int(b)
    print(f"Sum is {c}")

sum()
```

OUTPUT :

```
 value.py"
Sum is 25
```

37.

AIM :

Python | BMI (Body Mass Index) calculator.

CODE :

```
a = float(input("Input your hight in meters: "))
b = float(input("Input your weight in kilogram: "))
c = b / (a * a)
print(f"Your body mass index is {c} ")
```

OUTPUT :

```
Input your hight in meters: 132
Input your weight in kilogram: 67
Your body mass index is 0.0038452708907254 36
```

38.

AIM :

Python | Program to print Odd and Even numbers from the list of integers.

CODE :

```
a = [12,23,4,56,7,9,10,13,15,57,75,76,1,14]
print("Even numbers : ")
for i in a :
    if(i%2==0):
        print(i, end = ' ')
print("\nOdd numbers : ")
for i in a :
    if(i%2!=0):
        print(i, end = ' ')
```

OUTPUT :

```
Even numbers :
12 4 56 10 76 14
Odd numbers :
23 7 9 13 15 57 75 1
```

39.

AIM :

Python | Program to print Palindrome numbers from the given list.

CODE :

```
n=[11,2,32,33,44,1551,67,7676,1313,99]
print("Palindrome numbers are:")
for i in n:
    num=str(i)
    if("".join(reversed(num))==num):
        print(i)
```

OUTPUT :

```
11
2
33
44
1551
99
```

40.

AIM :

Python | Compute the net amount of a bank account based on the transactions.

CODE :

```
    class bank :
    name = None
    pin = 0
    ac = 0
    amount = 0
    deposit = 0
    b_amount = 0

    def __init__(self,name,pin,ac,amount):
        self.name = name
        self.pin = pin
        self.ac = ac
        self.amount = amount
        b_amount += self.amount

        print ('************************************')

    def printdata(self):
        print('NAME : ',self.name)
        print('ACCOUNT NUMBER : ',self.ac)
        print('AMOUNT : ',self.amount)

        print ('************************************')


class ATM(bank) :
    def atm(self):
        e = int(input("Enter your pin : "))
        if (e == self.pin) :
            print("If you diposit money press 1.")
            print("If you withdraw money press 2.")

            f = int(input('Enter your option : '))
            if (f == 1) :
```

```
                self.deposit = int(input("Enter your diposit Emount:"))
                self.amount = self.deposit + self.amount
                self.b_amount += self.deposit
                print(f"Your total Amount : {self.amount}")
            elif (f==2):
                self.deposit = int(input("Enter your withdraw amount :"))
                if (self.deposit<=self.amount):
                    self.amount = self.amount - self.deposit
                    self.b_amount -= self.deposit
                    print(f"Your Balance {self.amount}")
                else :
                    print("You have not sufficent balance.")
            else :
                print("You pass wrong number.")
        else :
            print("Your pin is not valid.")

    def grand_amount(self):
        print(f"Total Bank balance : {self.b_amount}")

if (__name__ == '__main__'):
    n = int(input ("Enter number of account holder name : "))
    c =list()

    for i in range(n):
        name = input("Enter Account Holder Name : ")
        ac = int(input('Enter Account Number : '))
        amount = int(input('Enter Amount : '))
        pin = int(input('Set PIN : '))
        c.append(ATM(name,pin,ac,amount))

    for i in range(n):
        c[i].printdata()
        print('\n')

    for i in range(n):
        c[i].atm()
```

OUTPUT :

41.

AIM :

Python | Count total number of bits in a number.

CODE :

```
a = str(bytes(10))
print(len(a))
```

OUTPUT :

42.

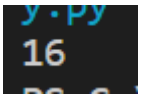
AIM :

Python | Generate random number using numpy library.

CODE :

```
import numpy
a = numpy.random.randint(12,23)
print(a)
```

OUTPUT :



43.

AIM :

Python | Program to calculate n-th term of a Fibonacci Series

CODE :

```
def febonacci(n):
    if(n <= 1):
        return n
    else:
        return (febonacci(n-1)+febonacci(n-2))

n = int(input("Enter number of febonacci series :"))

for i in range(n):
    print(febonacci(i))
```

OUTPUT :

```
Enter number of febonacci series :5
0
1
1
2
3
```

44.

AIM :

Python program to calculate prime numbers (using different algorithms) upto n

CODE :

```
a=int(input("Enter upper limit: "))
for i in range(2,a+1):
    b=0
    for n in range(2,i//2+1):
        if(i%n==0):
            b=b+1
    if(b<=0):
        print(i)
```

OUTPUT :

```
Enter upper limit: 45
2
3
5
7
11
13
17
19
23
29
31
37
41
43
```

45.

AIM :

Python program for not None test

CODE :

```
= input("Enter value : ")
if (a != None):
    print("variable is None")
else:
    print(a)
```

OUTPUT :

```
nter value : None
ariable is None
```

46.

Python program to check whether a string contains a number or not

CODE :

```
a = input("Enter the value : ")
```

```
try:
    int(a)
    print("Yes String is intiger.")
except:
    print("No String is Not intiger.")
```

OUTPUT :

```
Enter the value : 23.6
No String is Not intiger.
```

47.

AIM :

Python program for pass statement

CODE :

```
def get():
    pass

print("Animesh.")
```

OUTPUT :

```
Animesh.
```

48.

AIM :

Python program to define an empty function using pass statement

CODE :

```
def get():
    pass

print("Animesh.")
```

OUTPUT :

```
Animesh.
```

49.

AIM :

Python program to convert temperature from Celsius to Fahrenheit and vice-versa

CODE :

```python
print("""Please select option :-
            1. Celsius
            2. Faherenhit""")
option = int(input("Enter your option : "))
temp =  float(input("Enter your Temprature : "))

if (option == 1):
    faherenhit = (temp*1.8)+32
    print(f"Temprature in Faherenhit is {faherenhit}")
elif(option == 2):
    celsius = (temp-32)/1.8
    print(f"Temprature in Celsius is {celsius}")
else :
    print("Please Enter Valid Number.")
```

OUTPUT :

```
Please select option :-
            1. Celsius
            2. Faherenhit
Enter your option : 1
Enter your Temprature : 23
Temprature in Faherenhit is 73.4
```

50.

AIM :

Python program to count number of trailing zeros in Factorial of number N

CODE :

```
        def trailling(n):
    c = 0
    i = 5
    while (n/i>=1):
        c += int(n/i)
        i *= 5

    return int(c)

n = int(input("Enter factorial number : "))
print(f"Count of Zero in {n} ! are {trailling(n)}.")
```

OUTPUT :

```
Enter factorial number : 34
Count of Zero in 34 ! are 7.
```

51.

AIM :

Python program for swapping the value of two integers

CODE :

```
        a =12
b = 13
c = 0
c=a
a=b
b=c
print(f"after swaping value is {a}")
print(f"after swaping value is
```

OUTPUT :

```
after swaping value is 13
after swaping value is 12
```

52.

AIM :

Python program for swapping the value of two integers without third variable

CODE :

```
a = 12
b = 13
a,b = b,a
print(f"after swaping value is {a}")
print(f"after swaping value is {b}")
```

OUTPUT :

```
after swaping value is 13
after swaping value is 12
```

53.

AIM :

Python program to find the matched characters in a given string

CODE :

```
string = str("Hell , My name is Animesh And I am student, My Father is Business man.")
a = str("My")
print(string.find(a))
```

OUTPUT :

```
7
```

Python array Programs

AIM :

Python program to find the sum of all elements of an array

CODE :

```
a = [12,23,45,67,12,34,7,89]
sum =0
for i in a:
    print(i)
    sum = sum+i
```

```
print("Sum : ",sum)
```

OUTPUT :

```
12
23
45
67
12
34
7
89
Sum :  289
```

AIM:

Python program to find a series in an array consisting of characters

CODE :

```
def array_series(ch,se):
    for i in range(len(ch)):
        for n in range(len(se)):
            if (ch[i] == se[n]):
                return True
            else :
                return False
ch = input("Enter your array : ")
se = input("Enter your sub Array : ")
if(array_series(ch,se)):
    print("Yes it is a part of string.")
else :
    print("No it is not a part of string.")
```

OUTPUT :

```
nter your array : 23,45,5,23,12,54,6,78,98
nter your sub Array : 23,3
es it is a part of string.
```

AIM :

Python program to find the occurrence of a particular number in an array

CODE :

```
def ocur(num,c):
    a = 0
    for i in num :
        if (i == 12):
            a = a+1
    return a

    print (a)

num = list(input("Enter your array's List : "))
print(num)
c = int(input("Enter your number : "))
print(ocur(num,c))
```

OUTPUT :

```
Enter your array's List : 12,23,4,67,78,89,23,34
['1', '2', ',', '2', '3', ',', '4', ',', '6', '7', ',', '7', '8', ',', '8', '9', ',', '2', '3', ',', '
3', '4']
Enter your number : 4
0
```


Python programs on int class

AIM :

Python program to find number of bits necessary to represent an integer in binary

CODE :

```
number = int(input("Enter an integer number: "))
bits = number.bit_length()
print(f"bits required to store  {number} is {bits}")
print(f"binary value of {number} is ", bin(number))
```

OUTPUT :

```
Enter an integer number: 5
bits required to store  5 is 3
binary value of 5 is  0b101
```

AIM :

Python program to print number of bits to store an integer and also print number in Binary format

CODE :

```
number = int(input("Enter an integer number: "))
bits = number.bit_length()
print(f"bits required to store  {number} is {bits}")
```

OUTPUT :

```
Enter an integer number: 45
bits required to store  45 is 6
```

AIM :

Python program to print an array of bytes representing an integer

CODE :

```
size = int(input("Enter number : "))
array = bin(size)
print(array)
```

OUTPUT :

```
Enter number : 34
0b100010
```

AIM :

Python program to print the binary value of the numbers from 1 to N

CODE :

```
n = int(input("Enter the value of N: "))
for i in range(1, n+1):
    print(f"Binary value of {i} is {bin(i)}")
```

OUTPUT :

```
Enter the value of N: 5
Binary value of 1 is 0b1
Binary value of 2 is 0b10
Binary value of 3 is 0b11
Binary value of 4 is 0b100
Binary value of 5 is 0b101
```

Lists

AIM :

Python | Program to declare and print a list.

CODE :

```
a = [12,34,56,897,122,'asdf']

print("List : ",a)
```

OUTPUT :

```
PS C:\Users\my laptop> python -u "c:\Users\my laptop\Desktop\Python\python projects\print list.py"
List :  [12, 34, 56, 897, 122, 'asdf']
PS C:\Users\my laptop> python -u "c:\Users\my laptop\Desktop\Python\python projects\list in diffrent w
ay.py"
['Amit', 'Abhi', 'Radib', 21, 22, 37]
Amit
['Radib', 21, 22]
['Abhi', 'Radib', 21, 22, 37]
[100, 200, 'Hello', 'World', 100, 200, 'Hello', 'World']
['Amit', 'Abhi', 'Radib', 21, 22, 37, 100, 200, 'Hello', 'World']
PS C:\Users\my laptop> python -u "c:\Users\my laptop\Desktop\Python\python projects\tempCodeRunnerFile
.py"
List elements:  [10, 20, 30, 'New Delhi', 'Mumbai', 40]
List elements:  [10, 20, 30, 'New Delhi', 'Mumbai']
```

AIM :

Python program to print list elements in different ways.

CODE :

```
list1 = ["Amit", "Abhi", "Radib", 21, 22, 37]
list2 = [100, 200, "Hello", "World"]

print (list1)
```

```
print (list1[0])
print (list1[2:5])
print (list1[1:])
print (list2 * 2)
print (list1 + list2)
```

OUTPUT :

```
PS C:\Users\my laptop> python -u "c:\Users\my laptop\Desktop\Python\python projects\print list.py"
List :  [12, 34, 56, 897, 122, 'asdf']
PS C:\Users\my laptop> python -u "c:\Users\my laptop\Desktop\Python\python projects\list in diffrent w
ay.py"
['Amit', 'Abhi', 'Radib', 21, 22, 37]
Amit
['Radib', 21, 22]
['Abhi', 'Radib', 21, 22, 37]
[100, 200, 'Hello', 'World', 100, 200, 'Hello', 'World']
['Amit', 'Abhi', 'Radib', 21, 22, 37, 100, 200, 'Hello', 'World']
PS C:\Users\my laptop> python -u "c:\Users\my laptop\Desktop\Python\python projects\tempCodeRunnerFile
.py"
List elements:  [10, 20, 30, 'New Delhi', 'Mumbai', 40]
List elements:  [10, 20, 30, 'New Delhi', 'Mumbai']
```

AIM :

Python | Program for Adding, removing elements in the list.

CODE :

```
list = [10, 20, 30, "New Delhi", "Mumbai"]
list.append (40)
print ("List elements: ", list)
list.pop ()
print ("List elements: ", list)
```

OUTPUT :


```
PS C:\Users\my laptop> python -u "c:\Users\my laptop\Desktop\Python\python projects\print list.py"
List :  [12, 34, 56, 897, 122, 'asdf']
PS C:\Users\my laptop> python -u "c:\Users\my laptop\Desktop\Python\python projects\list in diffrent w
ay.py"
['Amit', 'Abhi', 'Radib', 21, 22, 37]
Amit
['Radib', 21, 22]
['Abhi', 'Radib', 21, 22, 37]
[100, 200, 'Hello', 'World', 100, 200, 'Hello', 'World']
['Amit', 'Abhi', 'Radib', 21, 22, 37, 100, 200, 'Hello', 'World']
PS C:\Users\my laptop> python -u "c:\Users\my laptop\Desktop\Python\python projects\tempCodeRunnerFile
.py"
List elements:  [10, 20, 30, 'New Delhi', 'Mumbai', 40]
List elements:  [10, 20, 30, 'New Delhi', 'Mumbai']
```


AIM :

Python | Program to print a list using ‘FOR and IN’ loop.

CODE :

```
list = [10, 20, 30, 40, 50]
print ("List elements are: ", list)
print ("List elements are: ")
for i in list:
    print (i)
```

OUTPUT :

```
List elements are:  [10, 20, 30, 40, 50]
List elements are:
10
20
30
40
50
```

AIM :

Python | Program to add an element at specified index in a list.

CODE :

```
list = [10, 20, 30]
print (list)
list.insert (1, 5)
print (list)
```

OUTPUT :

```
[10, 20, 30]
[10, 5, 20, 30]
```

AIM :

Python | Program to remove first occurrence of a given element in the list.

CODE :

```
list = [10, 20, 30, 40, 30]
list.remove(30);
print ("List element after removing 30:")
```

```
for i in range(len(list)):
    print (list[i])
```

OUTPUT :

```
List element after removing 30:
10
20
40
30
```

AIM :

Python | Remove all occurrences a given element from the list.

CODE :

```
list = [10, 20, 30, 40, 10, 30,12, 34, 10, 34, 56, 10]

for i in list :
    if (i== 10):
        list.remove(i)

print ("List element after removing 10:")
print (list)
```

OUTPUT :

```
List element after removing 10:
[20, 30, 40, 30, 12, 34, 34, 56]
```

AIM :

Python | Program to remove all elements in a range from the List.

CODE :

```
list = [10, 20, 30, 40, 10, 30,12, 34, 10, 34, 56, 10]
del list[2:5]
print ("List element : ")
print (list)
```

OUTPUT :

```
List element :
[10, 20, 30, 12, 34, 10, 34, 56, 10]
```

AIM :

Python | Program to sort the elements of given list in Ascending and Descending Order.

CODE :

```
list = [10, 20, 30, 40, 10, 30,12, 34, 10, 34, 56, 10]
list.sort()
print(list)
```

OUTPUT :

```
[10, 10, 10, 10, 12, 20, 30, 30, 34, 34, 40, 56]
PS C:\Users\my laptop>
```

AIM :

Python | Program to find the differences of two lists.

CODE :

```
list1 = [10, 20, 30, 40, 10, 30,12, 34, 10, 34, 56, 10]
list2 = [10, 20, 30, 10, 30, 12, 34, 10, 4, 5, 10]
lis1 = set(list1)
lis2 = set(list2)
diffrance = list(lis1.difference(lis2))
print(diffrance)
```

OUTPUT :

```
[40, 56]
```

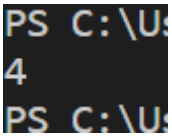
AIM :

Python | Program to Print the index of first matched element of a list.

CODE :

```
list = [10, 20, 10, 20, 30, 40, 50, 30]
print (list.index (30))
```

OUTPUT :



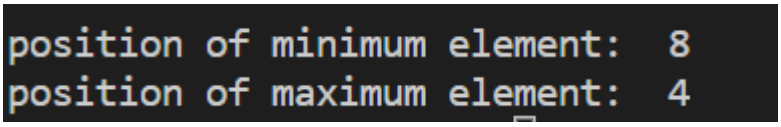
AIM :

Python | Program to find the position of minimum and maximum elements of a list.

CODE :

```
list = [10, 20, 10, 20, 231, 49.8, 50, 30, 5, 12]
min = list.index (min(list))
max = list.index (max(list))
print("position of minimum element: ", min)
print("position of maximum element: ", max)
```

OUTPUT :



AIM :

Python | Program to input, append and print the list elements.

CODE :

```
list1 = [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
n = 2
m = 5
print(list1)
for i in list1 :
    if(i%n == 0 and i%m == 0):
        print(f"{i} is dividable with {n} and {m}.")
```

OUTPUT :

```
Enter limit of the list: 4
Enter the value: 23
Enter the value: 23
Enter the value: 34
Enter the value: 45
 list  :
[23, 23, 34, 45]
```

AIM :

Python | Program to remove duplicate elements from the list.

CODE :

```
list1 = [10, 20, 10, 20, 231, 49.8, 50, 30, 5, 12]
list2 = []
print(list1)
for i in list1:
    if (i not in list2):
        list2.append(i)
print(list2)
```

OUTPUT :

```
[10, 20, 10, 20, 231, 49.8, 50, 30, 5, 12]
[10, 20, 231, 49.8, 50, 30, 5, 12]
```

AIM :

Python | Program to Create two lists with EVEN numbers and ODD numbers from a list.

CODE :

```
list1 = [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
even = []
odd = []
for i in list1 :
    if(i%2 != 0):
        odd.append(i)
    else:
        even.append(i)
print("Even : ",even)
print("Odd : ",odd)
```

OUTPUT :

```
Even :  [10, 20, 10, 20, 50, 30, 12]
Odd :  [231, 49, 5]
```

AIM :

Python | Program to print all numbers which are divisible by M and N in the List.

CODE :

```
list1 = [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
n = 2
m = 5
print(list1)
for i in list1 :
    if(i%n == 0 and i%m == 0):
        print(f"{i} is dividable with {n} and {m}.")
```

OUTPUT :

```
[10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
10 is dividable with 2 and 5.
20 is dividable with 2 and 5.
10 is dividable with 2 and 5.
20 is dividable with 2 and 5.
50 is dividable with 2 and 5.
30 is dividable with 2 and 5.
```

AIM :

Python | Create a list from the specified start to end index of another list.

CODE :

```
list = [10, 20, 30, 40, 50, 60]
start = int(input("Enter your starting point : "))
end = int(input("Enter your Ending point : "))
if ( start < 0 and end > len(list)):
    print ("Invalid start index")
list1 = list[start:end+1]
print ("Given list : ", list)
print ("Coppy List : ", list1)
```

OUTPUT :

```
Enter your starting point : 3
Enter your Ending point : 4
Given list :  [10, 20, 30, 40, 50, 60]
Coppy List :  [40, 50]
```

AIM :

Python | Create three lists of numbers, their squares and cubes.

CODE :

```
numbers = []
squares = []
cubes = []
start = int(input("Enter your starting point : "))
end = int(input("Enter your Ending point : "))
for i in range (start, end+1) :
    numbers.append (i)
    squares.append (i**2)
    cubes.append (i**3)
print ("numbers: ",numbers)
print ("squares: ",squares)
print ("cubes  : ",cubes)
```

OUTPUT :

```
Enter your starting point : 3
Enter your Ending point : 4
numbers:  [3, 4]
squares:  [9, 16]
cubes  :  [27, 64]
```

AIM :

Python | Create two lists with first half and second half elements of a list.

CODE :

```
list = [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
list1 = list[:5]
list2 = list[5:]
print(f"Orignal = {list}")
```

```
print(f"First Half = {list1}")
print(f"Second Half = {list2}")
```

OUTPUT :

```
Orignal = [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
First Half = [10, 20, 10, 20, 231]
Second Half = [49, 50, 30, 5, 12]
```

AIM :

Python | Iterate a list in reverse order.

CODE :

```
list = [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
print("List: ", list)
list1 = list[::-1]
print("list in reverse order: ", list1)
```

OUTPUT :

```
List:  [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
list in reverse order:  [12, 5, 30, 50, 49, 231, 20, 10, 20, 10]
PS C:\Users\my laptop>
```


AIM :

Python | print list after removing EVEN numbers.

CODE :

```
list = [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
print(" list:",list)
for i  in list:
    if(i%2 == 0):
        list.remove(i)
print("EVEN numbers list :",list)
```

OUTPUT :

```
.py"
 list: [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
EVEN numbers list : [20, 20, 231, 49, 30, 5]
PS C:\Users\my laptop> python -u "c:\Users\my laptop\Des
.py"
 list: [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
Odd numbers list : [10, 20, 10, 20, 49, 50, 30, 12]
PS C:\Users\my laptop>
```

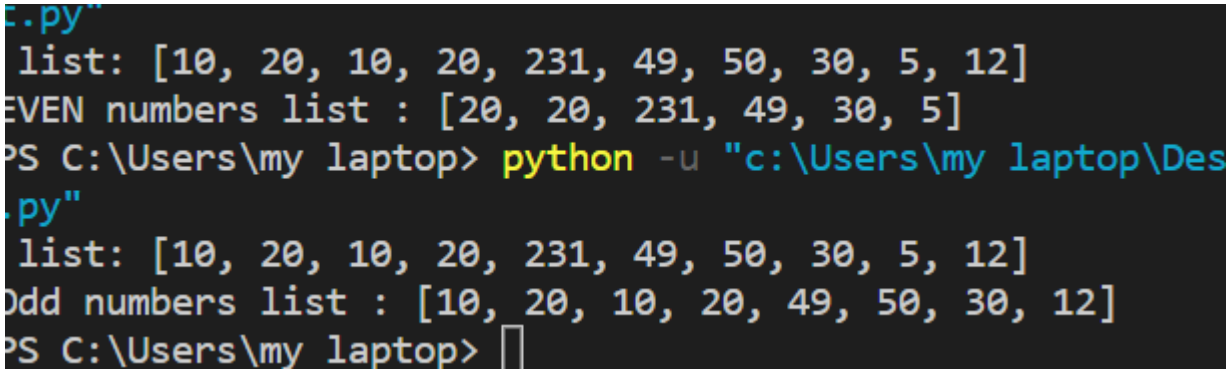
AIM :

Python | print list after removing ODD numbers.

CODE :

```
list = [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
print(" list:",list)
for i  in list:
    if(i%2 != 0):
        list.remove(i)
print("Odd numbers list :",list)
```

OUTPUT :



AIM :

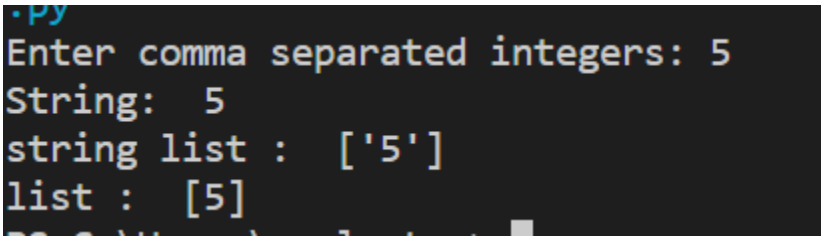
Python | Input comma separated elements, convert into list and print.

CODE :

```
str = str(input ("Enter comma separated integers: "))
print("String: ", str)
lis = str.split (",")
print("string list : ", lis)
list = []
```

```
for i in lis:
    list.append(int(i))
print ("list : ", list)
```

OUTPUT :



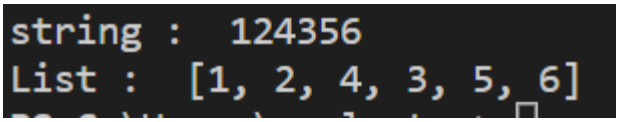
AIM :

Python | Convert a string to integers list.

CODE :

```
str = "124356"
list =[]
for i in str:
    list.append(int(i))
print("string : ",str)
print("List : ",list)
```

OUTPUT :



AIM :

Using List as Stack in Python.

CODE :

```
stack = [10, 20, 30]
print ("elements: ",stack)
stack.append(50)
print ("new stack : ",stack)
print (stack.pop (), " is removed")
```

OUTPUT :

```
elements:  [10, 20, 30]
new stack :  [10, 20, 30, 50]
50  is removed
```

AIM :

Python | Extend a list using + Operator.

CODE :

```
list = [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
print("List : ",list)
list += [13,45,76]
print("Updated List : ",list)
```

OUTPUT :

```
List :  [10, 20, 10, 20, 231, 49, 50, 30, 5, 12]
Updated List :  [10, 20, 10, 20, 231, 49, 50, 30, 5, 12, 13, 45, 76]
```

AIM :

Python program to find N largest and smallest elements from the list

CODE :

```
import heapq
def largest_ele(l,n):
    s=[]
    for i in range(n):
        s.append(max(l))
        l.remove(max(l))
    print('Largest Number : ',s)
def smallest_ele(m,n):
    t=[]
    for i in range(n):
        t.append(min(m))
        m.remove(min(m))
    print('Smallest Function : ',t)
l=[2,4,6,8,10]
m=[0,1,2,3,4,5,6]
n=2
largest_ele(l,n)
smallest_ele(m,n)
nums = [1, 8, 2, 23, 7, -4, 18, 23, 42, 37, 2]
print('BY heapq.nlargest: ',heapq.nlargest(3, nums))
print('BY heapq.nsmallest: ',heapq.nsmallest(3, nums))
```

OUTPUT :

```
Largest Number :  [10, 8]
Smallest Function :  [0, 1]
BY heapq.nlargest:  [42, 37, 23]
BY heapq.nsmallest:  [-4, 1, 2]
```

Python string programs

AIM :

Python | Declare, assign and print the string (Different ways).

CODE :

```
str1 = 'Hello world, How are you?'
str2 = "Hello world, How are you?"
str3 = '''Hello world, How are you?'''
str4 = """Hello world, How are you?"""
print ("str1: ", str1)
print ("str2: ", str2)
print ("str3: ", str3 )
print ("str4: ", str4)
```

OUTPUT :

```
str1:  Hello world, How are you?
str2:  Hello world, How are you?
str3:  Hello world, How are you?
str4:  Hello world, How are you?
```

AIM :

Python | Access and print characters from the string.

CODE :

```
str = "hello world."
print(str)
print ("character : ", str[6])
print (str[2:5])
```

OUTPUT :

```
hello world.
character :  w
llo
```

AIM :

Python program to print a string, extract characters from the string.

CODE :

```
str = "hello world."
print(str)
print ("character : ", str[6])
print (str[2:5])
```

OUTPUT :

```
hello world.
character :  w
llo
```

AIM :

Python | Program to print words with their length of a string.

CODE :

```
def splitString (str):
    str = str.split (' ')
    for i in str:
        print(f"{i}({len (i)})")
str = "Hello World How are you?"
splitString(str)
```

OUTPUT :

```
Hello(5)
World(5)
How(3)
are(3)
you?(4)
```

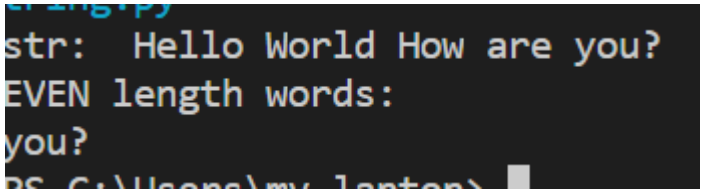
AIM :

Python | Print EVEN length words.

CODE :

```
str = "Hello World How are you?"
words = list(str.split(' '))
print("str: ", str)
print ("EVEN length words:")
for i in words:
    if(len(i)%2==0 ):
        print(i)
```

OUTPUT :



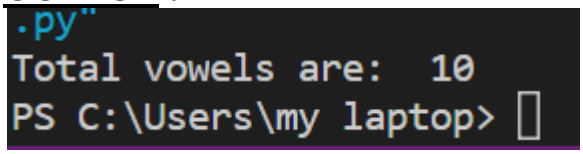
AIM :

Python | Count vowels in a string.

CODE :

```
str = "My name is Animesh Kakadiya."
n = 0
for i in str:
    if( i=='A' or i=='a' or i=='E' or i=='e'
    or i=='I' or i=='i' or i=='O' or i=='o'
    or i=='U' or i=='u'):
        n +=1;
print ("Total vowels are: ", n)
```

OUTPUT :



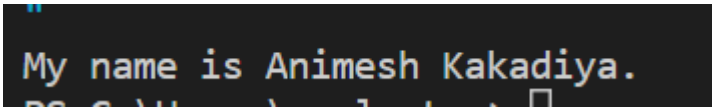
AIM :

Python | Passing string value to the function.

CODE :

```
def printMsg(str):
    print (str)
printMsg("My name is Animesh Kakadiya.")
```

OUTPUT :



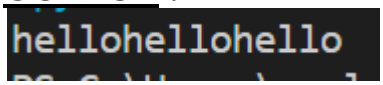
AIM :

Python | Create multiple copies of a string by using multiplication operator.

CODE :

```
str = "hello"
str1 = str * 3
print(str1)
```

OUTPUT :



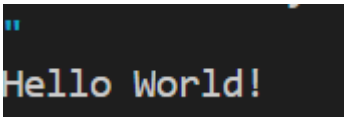
AIM :

Python | Appending text at the end of the string using += Operator.

CODE :

```
str = 'Hello'
str += ' '
str += 'World'
str += '!'
print(str)
```

OUTPUT :



AIM :

Python | Concatenate two strings and assign in another string by using + operator.

CODE :

```
str = "Hello, My name is Animesh."
sub_str = input("Enter your substring part : ")
if sub_str in str:
    print("Yes, substring presents in the string.")
else:
    print("No, substring does not present in the string.")
```

OUTPUT :

```
Enter your substring part : Animesh
Yes, substring presents in the string.
```

AIM :

Python | Check if a substring presents in a string using 'in' operator

CODE :

```
str1 = "Hello"
str2 = "World"
str3 = str1 + ' ' + str2
print ("str1: ",str1)
print ("str2: ",str2)
print ("str3: ",str3)
```

OUTPUT :

```
str1:  Hello
str2:  World
str3:  Hello World
```

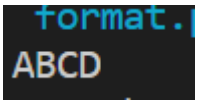
AIM :

Python | Assign Hexadecimal values in the string and print it in the string format

CODE :

```
str = "\x41\x42\x43\x44"
print(str)
```

OUTPUT :



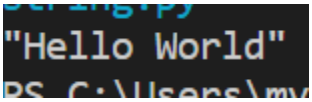
AIM :

Python | How to print double quotes with the string variable?

CODE :

```
str = "Hello World"
print(f'"{str}"')
```

OUTPUT :



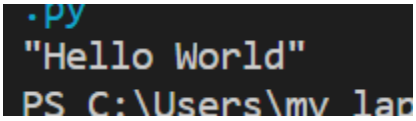
AIM :

Python | Ignoring escape sequences in the string

CODE :

```
str = "\"Hello World\""
print(str)
```

OUTPUT :



AIM :

Python program to calculate the number of all possible substrings of a string

CODE :

```
str = "My name is Animesh Kakadiya."
n = len(str)
substr = n*(n+1)//2
print("Number of Sub strings are",substr)
```

OUTPUT :

```
Number of Sub strings are 406
```

AIM :

Python program to reverse a string using stack and reversed method

CODE :

```
def createStack():
    stack=[]
    return stack
def size(stack):
    return len(stack)
def isEmpty(stack):
    if (size(stack) == 0):
        return True
def push(stack,item):
    stack.append(item)
def pop(stack):
    if isEmpty(stack): return
    return stack.pop()
def reverse(string):
    n = len(string)
    stack = createStack()
    for i in range(0,n,1):
        push(stack,string[i])
    string=""
    for i in range(0,n,1):
        string+=pop(stack)
    return string
string= "My name is Animesh Kakadiya."
string1 = reverse(string)
print("Reversed string is " + string1)
string2 = reverse(string)
print("Reversed string is " + string2)
```

OUTPUT :

```
Reversed string is .ayidakaK hseminA si eman yM
Reversed string is .ayidakaK hseminA si eman yM
PS C:\Users\my lapton>
```

AIM :

Split a string into array of characters in Python

CODE :

```
def split(str):
    return[i for i in str]
str =  "My name is Animesh Kakadiya."
print(split(str))
```

OUTPUT :

```
.py
['M', 'y', ' ', 'n', 'a', 'm', 'e', ' ', 'i', 's', ' ', 'A', 'n', 'i', 'm', 'e', 's', 'h', ' ', 'K', '
a', 'k', 'a', 'd', 'i', 'y', 'a', '.']
```


AIM :

Python program for slicing a string

CODE :

```
def slice(str, n):
    if len(str) < n:
        n = len(str)
    sliceing = str[:n]
    return sliceing
print(slice('Chocolate', 5))
print(slice("Hello",7))
```

OUTPUT :

```
Choco
Hello
```

AIM :

Python program to repeat M characters of a string N times

CODE :

```
def repetation(str, m, n):
    s_len = m
    if s_len > len(str):
        s_len = len(str)
    sub_str = str[:s_len]
    result = ''
    for i in range(n):
        result = result + sub_str
    return result
```

```
print (repetation('Animesh', 4, 3))
print (repetation('Hello', 3, 7))
```

OUTPUT :

```
AnimAnimAnim
HelHelHelHelHelHelHel
```

AIM :

Python program to swap characters of a given string

CODE :

```
str = "Hello World"
reverse = str[1:len(str)-1]
swap = str[len(str)-1] + reverse + str[0]
print(swap)
```

OUTPUT :

```
dello WorlH
```

AIM :

Python program to remove a character from a specified index in a string

CODE :

```
def remove_char(str, n):
    front = str[:n]
    back = str[n + 1:]
    return front + back
print (remove_char('Hello World', 7))
```

OUTPUT :

```
Hello Wrld
```

AIM :

Python program for adding given string with a fixed message

CODE :

```
str = input("Enter Your String : ")
if len(str)>=5 and str[:5] == 'Hello':
    print (str)
else :
    print('Hello '+str)
```

OUTPUT :

```
Enter Your String : My name is Animesh.
Hello My name is Animesh.
```

Python dictionary programs

AIM :

Python | Generate dictionary of numbers and their squares (i, i*i) from 1 to N

CODE :

```
n = int(input("Enter Number : "))
numbers = {}
for i in range(1, n+1):
    numbers[i] = i * i
print(numbers)
```

OUTPUT :

```
Enter Number : 23
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 1
96, 15: 225, 16: 256, 17: 289, 18: 324, 19: 361, 20: 400, 21: 441, 22: 484, 23: 529}
```


Python Class and Object programs

AIM :

Python | Demonstrate an example of Class and Object

CODE :

```
class Student:
    __Class = 0
    __name = None
```

```
    __rn = 0

    def __init__(self,ClassName,name,rn):
        self.Class=ClassName
        self.name=name
        self.rn=rn

    def getdata(self):
        print(self.Class , end = '\t\t')
        print(self.name, end = '\t\t')
        print(self.rn, end = '\t\t')

if(__name__ == "__main__"):
    n=int(input("Enter your class size : "))

    st=list()

    for i in range(n):
        name=input("Enter Student name: ")

        rn=int(input("Enter Student roll number: "))

        ClassName=int(input("Enter Student Divison: "))

        st.append(Student(ClassName,rn,name))

    print('**************************************************')
    print('\t\tDetails of Students ')
    print('Divison\t\troll_number\t\tname')
    for i in range(n):
        st[i].getdata()
        print('\n')
```

OUTPUT :

```
Enter your class size : 2
Enter Student name: Animesh
Enter Student roll number: 16
Enter Student Divison: 12
Enter Student name: Tirth
Enter Student roll number: 24
Enter Student Divison: 8
**************************************************
                Details of Students
Divison         roll_number             name
12              16              Animesh

8               24              Tirth
```

AIM :

Python | Simple program of a class (Input and print a number)

CODE :

```
class A :
    def input_data(self):
        age = int(input("Enter your Age : "))
    def getdata(self):
        year = 2019 - self.age
        print("Your birth Year : ",year)
a = A()
a.input_data()
a.getdata()
```

OUTPUT :

```
Enter your Age : 17
Your birth Year :  2002
```

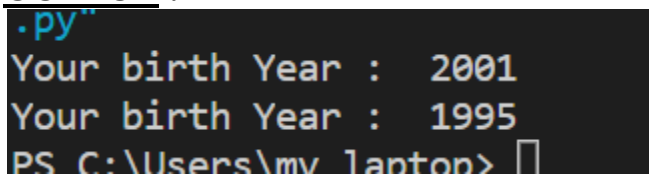
AIM :

Public variables in Python

CODE :

```
class A :
    def __init__(self):
        self.age = 18
    def getdata(self):
        year = 2019 - self.age
        print("Your birth Year : ",year)
a = A()
a.getdata()
a.age = 24
a.getdata()
```

OUTPUT :



AIM :

Python | Create Employee class with some attributes and methods

CODE :

```
class worker:
    __work_year = 0
    __name = None
    __id = 0

    def __init__(self,work_year,name,id):
        self.work_year=work_year
        self.name=name
        self.id=id

    def getdata(self):
        print(self.work_year , end = '\t\t')
        print(self.name, end = '\t\t')
        print(self.id, end = '\t\t')

if(__name__ == "__main__"):
    n=int(input("Enter your Office staff number : "))

    st=list()

    for i in range(n):
        name=input("Enter worker name: ")

        id=int(input("Enter worker's id number: "))
```

```
        work_year=int(input("Enter worker's working years: "))

        st.append(worker(work_year,id,name))

    print('************************************************')
    print('\t\tDetails of workers ')
    print('work_year\t\tid\t\tname')
    for i in range(n):
        st[i].getdata()
        print('\n')
```

OUTPUT :

```
Enter your Office staff number : 2
Enter worker name: computer Engineering
Enter worker's id number: 23
Enter worker's working years: 2
Enter worker name: Animesh
Enter worker's id number: 21
Enter worker's working years: 3
************************************************
                Details of workers
work_year               id              name
2               23              computer Engineering

3               21              Animesh
```

AIM :

Python program to calculate student grade

CODE :

```
class Student:
    def __init__(self):
        self.__roll=0
        self.__name=""
        self.__marks=[]
        self.__total=0
        self.__per=0
        self.__grade=""
        self.__result=""
    def setStudent(self):
        self.__roll=int(input("Enter Roll: "))
```

```
        self.__name=input("Enter Name: ")
        print("Enter marks of 5 subjects: ")
        for i in range(5):
            self.__marks.append(int(input("Subject "+str(i+1)+": ")))
    def calculateTotal(self):
        for x in self.__marks:
            self.__total+=x
    def calculatePercentage(self):
        self.__per=self.__total/5
    def calculateGrade(self):
        if self.__per>=85:
            self.__grade="S"
        elif self.__per>=75:
            self.__grade="A"
        elif self.__per>=65:
            self.__grade="B"
        elif self.__per>=55:
            self.__grade="C"
        elif self.__per>=50:
            self.__grade="D"
        else:
            self.__grade="F"
    def showStudent(self):
        self.calculateTotal()
        self.calculatePercentage()
        self.calculateGrade()
        print(f"Roll number : {self.__roll}\n",f"Name : {self.__name}\n",f"Total Marks :
{self.__total}\n",f"Percentage : {self.__per}\n",f"Grade : {self.__grade}")
s=Student()
s.setStudent()
s.showStudent()
```

OUTPUT :

```
Enter Roll: 23
Enter Name: Animesh
Enter marks of 5 subjects:
Subject 1: 78
Subject 2: 76
Subject 3: 89
Subject 4: 87
Subject 5: 90
Roll number : 23
 Name : Animesh
 Total Marks : 420
 Percentage : 84.0
 Grade : A
```

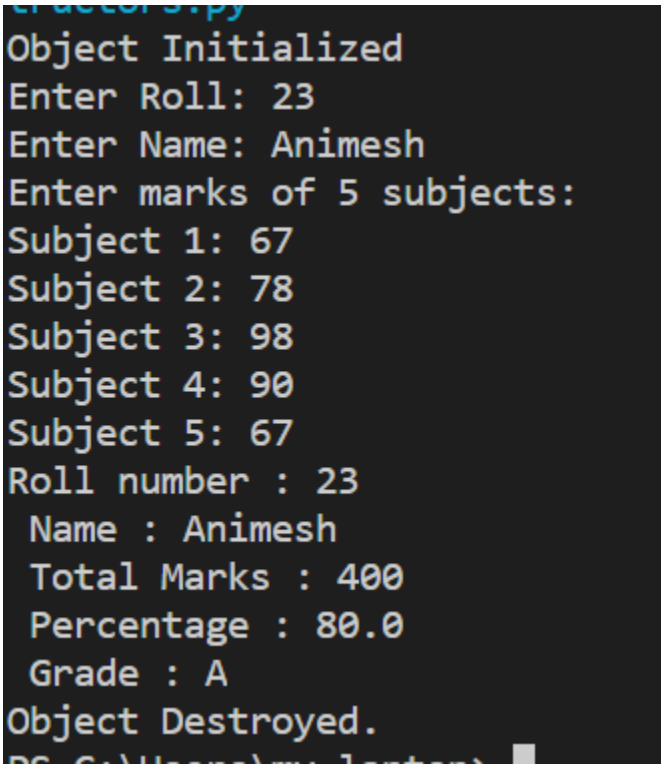
AIM :

Python | Example to implement destructor and constructors using __del__() and __init__()

CODE :

```
class Student:
    def __init__(self):
        self.__roll=0
        self.__name=""
        self.__marks=[]
        self.__total=0
        self.__per=0
        self.__grade=""
        self.__result=""
        print("Object Initialized")
    def setStudent(self):
        self.__roll=int(input("Enter Roll: "))
        self.__name=input("Enter Name: ")
        print("Enter marks of 5 subjects: ")
        for i in range(5):
            self.__marks.append(int(input("Subject "+str(i+1)+": ")))
    def calculateTotal(self):
        for x in self.__marks:
            self.__total+=x
    def calculatePercentage(self):
        self.__per=self.__total/5
    def calculateGrade(self):
        if self.__per>=85:
            self.__grade="S"
        elif self.__per>=75:
            self.__grade="A"
        elif self.__per>=65:
            self.__grade="B"
        elif self.__per>=55:
            self.__grade="C"
        elif self.__per>=50:
            self.__grade="D"
        else:
            self.__grade="F"
    def showStudent(self):
        self.calculateTotal()
        self.calculatePercentage()
        self.calculateGrade()
        print(f"Roll number : {self.__roll}\n",f"Name : {self.__name}\n",f"Total Marks :
{self.__total}\n",f"Percentage : {self.__per}\n",f"Grade : {self.__grade}")
    def __del__(self):
        print("Object Destroyed.")
s=Student()
s.setStudent()
s.showStudent()
```

OUTPUT :



AIM :

Python | Example to implement Getters and Setters in a class

CODE :

```
class Student:
    def __init__(self):
        self.__roll=0
        self.__name=""
        self.__marks=[]
        self.__total=0
        self.__per=0
        self.__grade=""
        self.__result=""
    def setStudent(self):
        self.__roll=int(input("Enter Roll: "))
        self.__name=input("Enter Name: ")
        print("Enter marks of 5 subjects: ")
        for i in range(5):
            self.__marks.append(int(input("Subject "+str(i+1)+": ")))
    def calculateTotal(self):
        for x in self.__marks:
            self.__total+=x
```

```
    def calculatePercentage(self):
        self.__per=self.__total/5
    def calculateGrade(self):
        if self.__per>=85:
            self.__grade="S"
        elif self.__per>=75:
            self.__grade="A"
        elif self.__per>=65:
            self.__grade="B"
        elif self.__per>=55:
            self.__grade="C"
        elif self.__per>=50:
            self.__grade="D"
        else:
            self.__grade="F"
    def showStudent(self):
        self.calculateTotal()
        self.calculatePercentage()
        self.calculateGrade()
        print(f"Roll number : {self.__roll}\n",f"Name : {self.__name}\n",f"Total Marks :
{self.__total}\n",f"Percentage : {self.__per}\n",f"Grade : {self.__grade}")
s=Student()
s.setStudent()
s.showStudent()
```

OUTPUT :

```
Enter Roll: 23
Enter Name: Animesh
Enter marks of 5 subjects:
Subject 1: 78
Subject 2: 76
Subject 3: 89
Subject 4: 87
Subject 5: 90
Roll number : 23
 Name : Animesh
 Total Marks : 420
 Percentage : 84.0
 Grade : A
```

AIM :

Python | Implementing setters and getters with the help of properties

CODE :

```
class Employee:
    def __init__(self): #Constructor
        self.__id = 0
        self.__name = ""
        self.__gender = ""
        self.__city = ""
        self.__salary = 0
    def id(self,value):
        self.__id=value
    def name(self, value):
        self.__name = value
    def gender(self, value):
        self.__gender = value
    def city(self, value):
        self.__city = value
    def salary(self, value):
        self.__salary = value
    def quiting(self):
        quit("Distroy Function.")

i = int(input("Enter Id\t:"))
n = input("Enter Name\t:")
g = input("Enter Gender:")
c = input("Enter City\t:")
s = int(input("Enter Salary:"))

employe =Employee()
employe.id=i
employe.name=n
employe.gender=g
employe.city=c
employe.salary=s
employe.quiting
```

OUTPUT :

```
.py"
Enter Id        :23
Enter Name      :animesh
Enter Gender:Male
Enter City      :surat
Enter Salary:2300
```

AIM :

Python | Implement Abstraction using Abstract class

CODE :

```
class Vehicle:
    def start(self,name=""):
        print(name,"is Started")
    def acclerate(self,name=""):
        pass
    def park(self,name=""):
        pass
    def stop(self,name=""):
        print(name,"is stopped")
class Bike(Vehicle):
    def acclerate(self, name=""):
        print(name,"is accelrating @ 60kmph")
    def park(self, name=""):
        print(name,"is parked at two wheeler parking")
class Car(Vehicle):
    def acclerate(self, name=""):
        print(name,"is accelrating @ 90kmph")
    def park(self, name=""):
        print(name,"is parked at four wheeler parking")
def main():
    print("Bike Object")
    b=Bike()
    b.start("Bike")
    b.acclerate("Bike")
    b.park("Bike")
    b.stop("Bike")
    print("\nCar Object")
    c = Car()
    c.start("Car")
    c.acclerate("Car")
    c.park("Car")
    c.stop("Car")
if __name__=="__main__":main()
```

OUTPUT :

```
Bike Object
Bike is Started
Bike is accelrating @ 60kmph
Bike is parked at two wheeler parking
Bike is stopped

Car Object
Car is Started
Car is accelrating @ 90kmph
Car is parked at four wheeler parking
Car is stopped
```

AIM :

Python | Implement Interface using class

CODE :

```
class Vehicle:
    def start(self,name=""):
        print(name,"is Started")
    def acclerate(self,name=""):
        pass
    def park(self,name=""):
        pass
    def stop(self,name=""):
        print(name,"is stopped")
class Bike(Vehicle):
    def acclerate(self, name=""):
        print(name,"is accelrating @ 60kmph")
    def park(self, name=""):
        print(name,"is parked at two wheeler parking")
class Car(Vehicle):
    def acclerate(self, name=""):
        print(name,"is accelrating @ 90kmph")
    def park(self, name=""):
        print(name,"is parked at four wheeler parking")
def main():
    print("Bike Object")
    b=Bike()
    b.start("Bike")
    b.acclerate("Bike")
    b.park("Bike")
    b.stop("Bike")
    print("\nCar Object")
    c = Car()
    c.start("Car")
    c.acclerate("Car")
    c.park("Car")
    c.stop("Car")
if __name__=="__main__":main()
```

OUTPUT :

```
Bike Object
Bike is Started
Bike is accelrating @ 60kmph
Bike is parked at two wheeler parking
Bike is stopped

Car Object
Car is Started
Car is accelrating @ 90kmph
Car is parked at four wheeler parking
Car is stopped
```

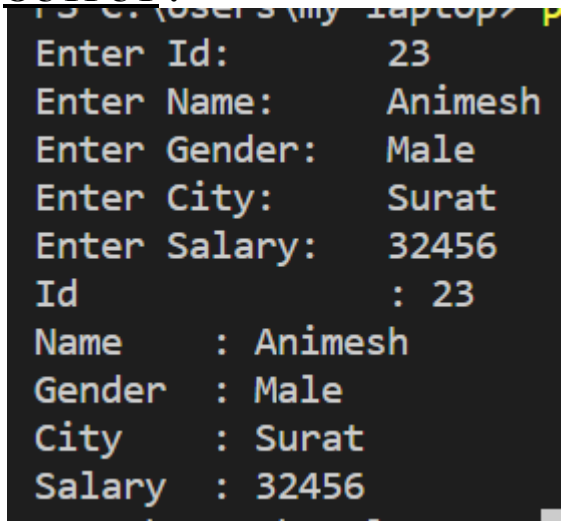
AIM :

Python | Create Employee Class

CODE :

```
class Employee:
    __id=0
    __name=""
    __gender=""
    __city=""
    __salary=0
    def setData(self):
        self.__id=int(input("Enter Id:\t"))
        self.__name = input("Enter Name:\t")
        self.__gender = input("Enter Gender:\t")
        self.__city = input("Enter City:\t")
        self.__salary = int(input("Enter Salary:\t"))
    def showData(self):
        print("Id\t\t:",self.__id)
        print("Name\t:", self.__name)
        print("Gender\t:", self.__gender)
        print("City\t:", self.__city)
        print("Salary\t:", self.__salary)
def main():
    emp=Employee()
    emp.setData()
    emp.showData()
if __name__=="__main__":
    main()
```

OUTPUT :



AIM :

Python | Create Employee Class with Constructor and Destructor

CODE :

```
class Employee:
    def __init__(self): #Constructor
        self.__id = 0
        self.__name = ""
        self.__gender = ""
        self.__city = ""
        self.__salary = 0
        print("Object Initialized.")
    def __del__(self):
        print("Object Destroyed.")
    def setData(self):
        self.__id=int(input("Enter Id\t:"))
        self.__name = input("Enter Name\t:")
        self.__gender = input("Enter Gender:")
        self.__city = input("Enter City\t:")
        self.__salary = int(input("Enter Salary:"))
    def __str__(self):
        data =
"["+str(self.__id)+","+self.__name+","+self.__gender+","+self.__city+","+str(self.__salary)+"]"
        return data
    def showData(self):
        print("Id\t\t:",self.__id)
        print("Name\t:", self.__name)
        print("Gender\t:", self.__gender)
        print("City\t:", self.__city)
        print("Salary\t:", self.__salary)
def main():
```

```
    emp=Employee()
    emp.setData()
    emp.showData()
    print(emp)
if __name__=="__main__":
    main()
```

OUTPUT :

```
.py
Object Initialized.
Enter Id         :23
Enter Name       :Animesh
Enter Gender:Male
Enter City       :Surat
Enter Salary:234346
Id               : 23
Name     : Animesh
Gender   : Male
City     : Surat
Salary   : 234346
[23,Animesh,Male,Surat,234346]
Object Destroyed.
```

AIM :

Example of single inheritance in Python

CODE :

```
class A:
    a=0
    b=0
    def A(self):
        print("Enter A : ")
        a=input()
        self.a=a
class B(A):
    def B(self):
        super().A()
        super().b
        print("Enter B : ")
```

```
        b=input()
        self.b=b
```

OUTPUT :

```
itance.py
Enter A :
23
Enter B :
34
```

AIM :

Example of inheritance with two child (derived) classes in Python

CODE :

```
class A:
    a=0
    b=0
    def A(self):
        print("Enter A : ")
        a=input()
class B(A):
    def B(self):
        super().A()
        print("Enter B : ")
        b=input()
        self.b=b
class C(A):

    def C(self):
        super().A()
        print(self.a)
b1 = B()
b1.B()
c1 = C()
c1.C()
```

OUTPUT :

```
Enter A :
23
Enter B :
34
```

AIM :

Example of multiple inheritance in Python

CODE :

```
class A:
    def __init__(self):
        print("MY name is Animesh. ")
    def __del__(self):
        print("Program over.")
class B:
    def __init__(self):
        A.__init__(self)
        print("I am Student.")
class C(B,A):
    def __init__(self):
        super().__init__()
        print("I am 18 year old.")
c=C()
del c
```

OUTPUT :

```
MY name is Animesh.
I am Student.
I am 18 year old.
Program over.
```

AIM :

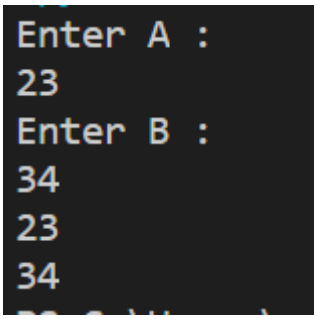
Example of multilevel inheritance in Python

CODE :

```
class A:
    a=0
    b=0
    def A(self):
        print("Enter A : ")
        a=input()
        self.a=a
class B(A):
    def B(self):
        super().A()
        super().b
        print("Enter B : ")
        b=input()
```

```
        self.b=b
class C(B):
    def C(self):
        super().B()
        print(self.a)
        print(self.b)
c1 = C()
c1.C()
```

OUTPUT :



AIM :

Example of hierarchical inheritance in Python

CODE :

```
class A:
    def __init__(self):
        print("MY name is Animesh. ")
class B(A):
    def __init__(self):
        A.__init__(self)
        print("I am Student.")
class C(A):
    def __init__(self):
        super().__init__()
        print("I am 18 year old.")
class D(A):
    def __init__(self):
        super().__init__()
        print("I am Boy.")
b = B()
c = C()
d = D()
```

OUTPUT :

```
I am Student.
MY name is Animesh.
I am 18 year old.
MY name is Animesh.
I am Boy.
```

Python Regex programs

AIM :

Python Regex | program to Remove leading zeros from an IP address

CODE :

```
import re
regex = '\.[0]*'
def Zeros(ip):
    modified_ip = re.sub(regex, '.', ip)
    print(modified_ip)
if __name__ == '__main__' :
    ip = "216.08.094.196"
    Zeros(ip)
```

OUTPUT :

```
.py
216.8.94.196
```