Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294 - Jay Vegad

Lab - 1

01) WAP to print “Hello World”

In [1]:
```
print("Hello World")
```

```
Hello World
```

02) WAP to print addition of two numbers with and without using input().

In [2]:
```
a = 5;
b = 10;
print(a + b)
```

```
15
```

03) WAP to check the type of the variable.

In [3]:
```
a = 'Darshan'
type(a)
```

Out[3]:
```
str
```

04) WAP to calculate simple interest.

In [4]:
```
P = int(input("Enter principle value: "))
r = int(input("Enter rate of interest: "))
t = int(input("Enter Time interval: "))
```

```
A = (P*r*t)/100
print(A)
```

```
Enter principle value: 5000
Enter rate of interest: 9
Enter Time interval: 2
900.0
```

05) WAP to calculate area and perimeter of a circle.

In [8]:
```
import math

r = int(input("Enter radius : "))
p = 2* math.pi *r

print("parameter: ",p)
```

```
Enter radius : 5
parameter:  31.41592653589793
```

06) WAP to calculate area of a triangle.

In [9]:
```
b = int(input("Enter base : "))
h = int(input("Enter height : "))
a = (b*h)/2
print("area : ",a)
```

```
Enter base : 3
Enter height : 4
area :  6.0
```

07) WAP to compute quotient and remainder.

In [14]:
```
import math
dividend = int(input("Enter no dividend : "))
divisor = int(input("Enter no divisor : "))

print("quotient: ",(dividend//divisor))
print("remainder: ",(dividend%divisor))
```

```
Enter no dividend : 5
Enter no divisor : 4
quotient:  1
remainder:  1
```

08) WAP to convert degree into Fahrenheit and vice versa.

In [16]:
```
C = float(input("Enter temp in celsius : "))
F = (C * 9/5) + 32
print(f"Temp: {F}*F")
```

```
Enter temp in celsius : 0
Temp: 32.0*F
```

In [18]:
```
F = float(input("Enter temp in fahrenheit : "))
C = (F - 32) * 5/9
print(f"Temp: {C}*C")
```

```
Enter temp in fahrenheit : 32
Temp: 0.0*C
```

09) WAP to find the distance between two points in 2-D space.

In [3]:
```
import math
x1 = int(input("Enter x1 : "))
y1 = int(input("Enter y1 : "))
x2 = int(input("Enter x2 : "))
y2 = int(input("Enter y2 : "))
d=math.sqrt((x2 - x1)^2 + (y2 - y1)^2)
print(d)
```

```
Enter x1 : 0
Enter y1 : 0
Enter x2 : 2
Enter y2 : 2
2.0
```

10) WAP to print sum of n natural numbers.

In [21]:
```
C = int(input("Enter number : "))
sum = 0;
i = 1
for i in range(C) :
    sum = sum + i
print(sum)
```

```
Enter number : 5
10
```

11) WAP to print sum of square of n natural numbers.

In [23]:
```
C = int(input("Enter number : "))
sum = 0;
i = 1
for i in range(C) :
    sum = sum + i*i
print(sum)
```

```
Enter number : 5
30
```

12) WAP to concate the first and last name of the student.

In [24]:
```
firstName = 'First'
lastName = 'Last'
```

```
print(firstName+ " "+lastName)
```

```
First Last
```

13) WAP to swap two numbers.

In [25]:
```
a = int(input("Enter 1st number : "))
b = int(input("Enter 2nd number : "))
temp = a
a=b
b=temp
print(a)
print(b)
```

```
Enter 1st number : 1
Enter 2nd number : 2
2
1
```

14) WAP to get the distance from user into kilometer, and convert it into meter, feet, inches and centimeter.

In [26]:
```
a = float(input("distance in km : "))
print(f'{a*100} m')
print(f'{a*3280.84} ft')
print(f'{a*39370.1} inches')
print(f'{a*1000} cm')
```

```
distance in km : 5
500.0 m
16404.2 ft
196850.5 inches
5000.0 cm
```

15) WAP to get day, month and year from the user and print the date in the given format: 23-11-2024.

In [27]:
```
d = int(input("enter date : "))
m = int(input("enter month : "))
y = int(input("enter year: "))
print(f'{d}-{m}-{y}')
```

```
enter date : 28
enter month : 12
enter year: 2005
28-12-2005
```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294 - Jay Vegad

Enrollment: 23010101294

if..else..

01) WAP to check whether the given number is positive or negative.

In [2]:
```
a = int(input("Enter Number one"))
if (a > 0):
    print("Number is positive... ")
else:
    print("Number is negaive")
```

```
Enter Number one12
Number is positive...
```

02) WAP to check whether the given number is odd or even.

In [3]:
```
a = int(input("Enter Number:  "))
if(a % 2 == 0):
    print("Number is even")
else:
    print("Number is odd")
```

```
Enter Number:  2
Number is even
```

03) WAP to find out largest number from given two numbers using simple if and ternary operator.

In [4]:
```
num1 = int(input("Enter First Number:  "))
num2 = int(input("Enter Second Number:  "))
if(num1 > num2):
    print(f"{num1} is greater...")
else:
    print(f"{num2} is greater")
```

```
Enter First Number:  12
Enter Second Number:  23
23 is greater
```

04) WAP to find out largest number from given three numbers.

In [6]:
```
num1 = int(input("Enter First Number:  "))
num2 = int(input("Enter Second Number:  "))
num3 = int(input("Enter Third Number:  "))

if(num1 > num2 and num1 > num3):
    print(f"{num1} is greater...")
elif(num2 > num1 and num2 > num3):
    print(f"{num2} is greater")
else:
    print(f"{num3} is greater")
```

```
Enter First Number:  12
Enter Second Number:  23
Enter Third Number:  34
34 is greater
```

05) WAP to check whether the given year is leap year or not.

[If a year can be divisible by 4 but not divisible by 100 then it is leap year but if it is divisible by 400 then it is leap year]

In [8]:
```
year = int(input("enter Year: "))
if(year % 4 == 0):
    print("Year is leap year.. ")
else:
    print("Year is not year.. ")
```

```
enter Year: 2004
Year is leap year..
```

06) WAP in python to display the name of the day according to the number given by the user.

In [20]:
```
day = int(input("Enter Day"))
if(day<=0 and day > 7):
    print("INvalid day")
else:
    match(day):
        case 1:print("Sunday")
        case 2:print("monday")
        case 3:print("Tuesday")
        case 4:print("Wednesday")
        case 5:print("Turshday")
        case 6:print("friday")
        case 7:print("saturday")
        case _:print("Not valid")
```

```
Enter Day4
Wednesday
```

07) WAP to implement simple calculator which performs (add,sub,mul,div) of two no. based on user input.

In [9]:
```
num1 = int(input("Enter Number 1: "))
num2 = int(input("Enter Number 2: "))
sign = input("Enter Sign")
match(sign):
    case '+':
        print(f"Here is your result: {num1 + num2}")
    case '-':
        print(f"Here is your result: {num1 - num2}")
    case '*':
        print(f"Here is your result: {num1 * num2}")
    case '/':
        print(f"Here is your result: {num1 / num2}")
    case _:
        print("Not a valid value: ")
```

```
Enter Number 1: 12
Enter Number 2: 23
Enter Sign+
Here is your result: 35
```

08) WAP to read marks of five subjects. Calculate percentage and print class accordingly.

Fail below 35 Pass Class between 35 to 45 Second Class between 45 to 60 First Class between 60 to 70 Distinction if more than 70

In [12]:
```
maths = float(input("Enter Maths Marks: "))
se = float(input("Enter SE Marks: "))
flutter = float(input("Enter Flutter Marks: "))
py = float(input("Enter Python Marks: "))
pc = float(input("Enter pc Marks: "))
```

```
avg  = (maths + se + flutter + py + pc) / 5

if(avg < 35):
    print("FAIL.........................")
elif(avg > 35 and avg < 45):
    print("PASS CLASS")
elif(avg > 45 and avg < 60):
    print("SECOND Class")
elif(avg > 60 and avg < 70):
    print("First CLASS")
elif(avg > 70):
    print("DISTINCTION")
```

```
Enter Maths Marks: 50
Enter SE Marks: 50
Enter Flutter Marks: 50
Enter Python Marks: 50
Enter pc Marks: 50
SECOND Class
```

09) Three sides of a triangle are entered through the keyboard, WAP to check whether the triangle is isosceles, equilateral, scalene or right-angled triangle.

In [14]:
```
import math
side1 = float(input("Enter Side One: "))
side2 = float(input("Enter Second Side: "))
side3 = float(input("Enter Theird Side "))
if(side1 == side2 or side2 ==  side3 or side3 == side1):
    print("isosceles")
elif(side1 == side2 == side3):
    print("equilateral")
else:
    if(side1 > side2 and side1 > side3):
        if((side1 ** side1) > ((side2 ** side2) + (side3 ** side3))):
            print("Right Angle")
    elif(side2 > side3 and side2 > side1):
        if((side2 ** side2) > ((side1 ** side1) + (side3 ** side3))):
            print("Right Angle")
    else:
        if((side3 ** side3) > ((side1 ** side1) + (side2 ** side2))):
            print("right angle")
```

```
Enter Side One: 3
Enter Second Side: 4
Enter Theird Side 5
right angle
```

In [ ]:
```

```

10) WAP to find the second largest number among three user input numbers.

In [15]:
```
num1 = int(input("Enter First Number.. "))
num2 = int(input("Enter Second Number.. "))
num3 = int(input("Enter Third Number.. "))
if(num1 > num2):
    if(num2 > num3):
        print(f"{num2} is second largest")
    else:
        print(f"{num1} is Second Largest")
else:
    if(num2 > num3):
        print(f"{num3} is second large")
    else:
        print(f"{num2} is second large")
```

```
Enter First Number.. 12
Enter Second Number.. 23
Enter Third Number.. 43
23 is second large
```

11) WAP to calculate electricity bill based on following criteria. Which takes the unit from the user.

a. First 1 to 50 units – Rs. 2.60/unit b. Next 50 to 100 units – Rs. 3.25/unit c. Next 100 to 200 units – Rs. 5.26/unit d. above 200 units – Rs. 8.45/unit

In [19]:
```
unit = int(input("Enter Unit"))

if(unit > 1 and unit < 50):
    print(f"{unit * 2.60} is bill")
elif(unit > 50 and unit < 100):
    print(f"{unit * 3.25} is bill")
elif(unit > 100 and unit < 200):
    print(f"{unit * 5.26} is bill")
elif(unit > 200):
    print(f"{unit * 8.45} is bill")
```

```
Enter Unit60
195.0 is bill
```

In [ ]:
```

```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294 - Jay Vegad

Lab - 3

for and while loop

01) WAP to print 1 to 10.

In [ ]:
```
i = 0;
while i != 10:
    print(i)
    i = i + 1
```

02) WAP to print 1 to n.

In [ ]:
```
a = int(input("Enter number... "))
i = 1
while(i != a):
    print(i)
    i+=1
```

03) WAP to print odd numbers between 1 to n.

In [ ]:
```
a = int(input("Enter number... "))
for i in range(1,a + 1,2):
    print(i)
```

In [ ]:
```
### 04) WAP to print numbers between two given numbers which is divisible by
```

In [1]:
```
number1 = int(input("Enter First Number... "))
number2 = int(input("Enter Second Number... "))
for i in range(number1,number2 + 1):
    if(i % 2 == 0 and i % 3 != 0):
        print(i)
```

```
Enter First Number... 1
Enter Second Number... 100
2
4
8
10
14
16
20
22
26
28
32
34
38
40
44
46
50
52
56
58
62
64
68
70
74
76
80
82
86
88
92
94
98
100
```

In [ ]:
```
### 05) WAP to print sum of 1 to n numbers.
```

In [2]:
```
number = int(input("Enter Number... "))
sum = 0
for i in range(0,number + 1):
    sum += i
print(sum)
```

```
Enter Number... 10
55
```

06) WAP to print sum of series 1 + 4 + 9 + 16 + 25 + 36 + ...n.

In [7]:
```
number = int(input("Enter Number.. "))
sum = 0
for i in range(0,number+1):
    sum += i*i
print(sum)
```

```
Enter Number.. 3
14
```

07) WAP to print sum of series 1 – 2 + 3 – 4 + 5 – 6 + 7 ... n.

In [9]:
```
number = int(input("Enter Number.. "))
sum = 0
for i in range(0,number+1):
    if(i%2==0): sum-=i
    else: sum+=i
print(sum)
```

```
Enter Number.. 10
-5
```

08) WAP to print multiplication table of given number.

In [11]:
```
number = int(input("Enter Number:... "))
for i in range(1,11):
    print(f"{number} * {i} = {number * i}")
```

```
Enter Number:... 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

09) WAP to find factorial of the given number.

In [2]:
```
number = int(input("Enter Number:... "))
fac = 1
for i in range(1,number+1):
    fac*=i
print(fac)
```

```
Enter Number:... 4
24
```

10) WAP to find factors of the given number.

In [13]:
```
number = int(input("Enter Number:... "))
for i in range(1,number):
    if(number%i==0): print(i, end=',')
else: print(i + 1)
```

```
Enter Number:... 10
1,2,5,10
```

11) WAP to find whether the given number is prime or not.

In [15]:
```
num = int(input("Enter Number:... "))
if num > 1:
    for i in range(2, (num//2)+1):
        if (num % i) == 0:
            print(num, "is not a prime number")
            break
    else:
        print(num, "is a prime number")
else:
    print(num, "is not a prime number")
```

```
Enter Number:... 10
10 is not a prime number
```

12) WAP to print sum of digits of given number.

In [21]:
```
number = (input("Enter Number:...  "))
sum = 0;
for i in number:
    sum += int(i)
print(sum)
```

```
Enter Number:...  19
10
```

13) WAP to check whether the given number is palindrome or not

In [25]:
```
s = input("Enter String... ")

def isPalindrome(s):
    return s == s[::-1]

ans = isPalindrome(s)

if ans:
    print("Yes")
```

```
else:
    print("No")
```

```
Enter String... helpo
No
```

14) WAP to print GCD of given two numbers.

In [26]:
```
import math

number1 = int(input("Enter Number:.."))
number2 = int(input("Enter Second Number:..."))
print (math.gcd(number1,number2))
```

```
Enter Number:..10
Enter Second Number:...30
10
```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294 - Jay Vegad

Lab - 4

String

01) WAP to check whether the given string is palindrome or not.

In [3]:
```
s = input("Enter String:...")
rev = s[::-1]
if s == rev:
    print("Palindrome")
else:
    print("Not Palindrom")
```

```
Enter String:...nayan
Palindrome
```

02) WAP to reverse the words in the given string.

In [4]:
```
s = input("Enter String:...")
rev = s[::-1]
print(rev)
```

```
Enter String:...kuldip
pidluk
```

03) WAP to remove ith character from given string.

In [4]:
```
s = input("Enter String:.. ")
idx = int(input("Enter Char:.. "))
modifiedString = s[0:idx] + s[idx+1:len(s)]
print(modifiedString)
```

```
Enter String:.. lkjh
Enter Char:.. 1
ljh
```

In [ ]:
```

```

In [ ]:
```
# 04) WAP to find length of string without using len function.
```

In [5]:
```
count = 0
s = 'akshdajhsd'
for i in s:
    count += 1
print(count)
```

```
10
```

05) WAP to print even length word in string.

In [8]:
```
s = "This is Beautiful Bird"
strArr = s.split(" ")
for i in strArr:
    if len(i) % 2 == 0:
        print(i)
```

```
This
is
Bird
```

06) WAP to count numbers of vowels in given string.

In [19]:
```
s = 'hello world'
count = 0
vowels = 'aeiou'
for i in s:
    if i in "aeiou":
        count += 1
print(count)
```

```
3
```

07) WAP to capitalize the first and last character of each word in a string.

In [101…
```
s = "hello world Kem cho"
splitedArr = s.split(" ")
for i in splitedArr:
    print(i[0].capitalize() + i[1:len(i)-1] + i[len(i)-1:len(i)].capitalize(
```

```
HellO WorlD KeM ChO
```

08) WAP to convert given array to string.

In [4]:
```python
arr = ["Welcome to python"]
s = ''.join(arr)
print(s)
```

```
Welcome to python
```

09) Check if the password and confirm password is same or not.

In case of only case's mistake, show the error message.

In [7]:
```python
pass1 = input("Enter Password: ")
pass2 = input("Enter Confirmed Password: ")
if pass1.lower() == pass2.lower():
    print("Go Ahed")
else:
    print("Password is wrong... ")
```

```
Enter Password: JAY
Enter Confirmed Password: jay
Go Ahed
```

10) : Display credit card number.

card no. : 1234 5678 9012 3456

display as : **** **** **** 3456

In [2]:
```python
card_number = "1234 4567 8901 2345"

parts = card_number.split()

masked_parts = ['****' for _ in parts[:-1]] + [parts[-1]]

masked_card_number = ' '.join(masked_parts)

print(masked_card_number)
```

```
**** **** **** 2345
```

11) : Checking if the two strings are Anagram or not.

s1 = decimal and s2 = medical are Anagram

In [4]:
```python
str1 = "decimal"
str2 = "medical"
```

```
str1 = str1.replace(" ", "").lower()
str2 = str2.replace(" ", "").lower()

is_anagram = sorted(str1) == sorted(str2)

print("Anagram" if is_anagram else "Not an Anagram")
```

```
Anagram
```

12) : Rearrange the given string. First lowercase then uppercase alphabets.

input : EHlsarwiwhtwMV

output : lsarwiwhtwEHMV

In [5]:
```
input_string = "EHlsarwiwhtwMV"

lowercase = ''.join([char for char in input_string if char.islower()])
uppercase = ''.join([char for char in input_string if char.isupper()])

output_string = lowercase + uppercase

print("Input :", input_string)
print("Output:", output_string)
```

```
Input : EHlsarwiwhtwMV
Output: lsarwiwhtwEHMV
```

In [ ]:
```

```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294 - Jay Vegad

Lab - 5

List

01) WAP to find sum of all the elements in a List.

In [11]:
```
sum = 0
l1 = [10,20,30,40]
for i in range(0,len(l1)):
    sum = sum + l1[i]
print(sum)
```

```
100
```

02) WAP to find largest element in a List.

In [18]:
```
l1 = [10,20,30,40,90,420,102,402]
large = l1[0]
for i in l1:
    if i > large:
        large = i
print(large)
```

```
420
```

03) WAP to find the length of a List.

In [19]:
```
l1 = [10,20,301,30,203,21012,302]
print(len(l1))
```

```
7
```

04) WAP to interchange first and last elements in a list.

In [26]:
```
l1 = [1102,20102,102,3000102,30010]
first = l1[0]
last = l1[len(l1) - 1]
l1[0] = last
l1[len(l1) - 1] = first
print(l1)
```

```
[30010, 20102, 102, 3000102, 1102]
```

05) WAP to split the List into two parts and append the first part to the end.

In [24]:
```
l1 = [10,20,30,20,10,201,201,201,203,30]
firstPart = []
lastPart = []
for i in range(0,len(l1)//2):
    firstPart.append(l1[i])
for i in range(len(l1)//2, len(l1)):
    lastPart.append(l1[i])
print(lastPart + firstPart)
```

```
[201, 201, 201, 203, 30, 10, 20, 30, 20, 10]
```

06) WAP to interchange the elements on two positions entered by a user.

In [29]:
```
l1 = [10,20,30,40,50,60,20]
number1 = int(input("Enter First Position:..  "))
number2 = int(input("Enter Second Position:... "))
temp = l1[number1]
l1[number1] = l1[number2]
l1[number2] = temp
print(l1)
```

```
Enter First Position:..  0
Enter Second Position:... 4
[50, 20, 30, 40, 10, 60, 20]
```

07) WAP to reverse the list entered by user.

In [9]:
```
l1 = list(map(int, input("Enter numbers separated by space: ").split()))
l1 = l1[::-1]
print(l1)
```

```
Enter numbers separated by space: 1 3 4 2 3 2 4 4 5 2
[2, 5, 4, 4, 2, 3, 2, 4, 3, 1]
```

08) WAP to print even numbers in a list.

In [11]:
```
l1 = [1,2,3,4,2,1,2,1,2,1,2]
for i in l1:
    if i % 2 == 0:
        print(i,end = " ")
```

```
2 4 2 2 2 2
```

09) WAP to count unique items in a list.

In [35]:
```
l1 = [1,2,3,2,1,2,4,7]
l2 = []
count = 0
print(l1)
for i in l1:
    if i not in l2 : l2.append(i)
print(len(l2))
```

```
[1, 2, 3, 2, 1, 2, 4, 7]
5
```

10) WAP to copy a list.

In [41]:
```
l1 = [10,20,40,20]
l2 = []
l2 = l1.copy()
print(l2)
```

```
[10, 20, 40, 20]
```

11) WAP to print all odd numbers in a given range.

In [2]:
```
l1 = [10,20,30,201,201,2000304,2030,2310]
number1 = int(input("Enter First Number:... "))
number2 = int(input("Enter Second Number:... "))
for i in range(number1,number2):
    if l1[i] % 2 != 0:
        print(l1[i])
```

```
Enter First Number:... 3
Enter Second Number:... 6
201
201
```

12) WAP to count occurrences of an element in a list.

In [4]:
```
l1 = [10,20,30,2,1,2,112]
number = int(input("Enter Number That You Find In List:.. "))
count = 0
for i in range(0,len(l1)):
```

```
    if l1[i] == number: count += 1
print(count)
```

```
Enter Number That You Find In List:.. 2
2
```

13) WAP to find second largest number in a list.

In [3]:
```
l1 = [203,32,5,32,65,65,68,4,12,45]
l1.sort()
max = l1[len(l1)-1]
secondLarge = l1[0]
for i in l1:
    if secondLarge < i and i != max:
        secondLarge = i
print(secondLarge)
```

```
68
```

14) WAP to extract elements with frequency greater than K.

In [7]:
```
l1 = [10,20,302,10,20,101010,20]
l1.sort()
l2=[]
count = 0
number = int(input("Enter Number:... "))
for i in l1:
    if l1.count(i)>number and i not in l2:l2.append(i)
print(l2)
```

```
Enter Number:... 2
[20]
```

15) WAP to create a list of squared numbers from 0 to 9 with and without using List Comprehension.

In [11]:
```
l1 = []
for i in range(0,10):
    l1.append(i**2)
print(l1)
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

16) WAP to create a new list (fruit whose name starts with 'b') from the list of fruits given by user.

In [20]:
```
fruits = ["Apple","Banana","Centerfruit","Fruit","Graps","Babli"]
secondFruitList = []
for i in fruits:
    if i.startswith("B"):
        secondFruitList.append(i)
print(secondFruitList)
```

```
['Banana', 'Babli']
```

17) WAP to create a list of common elements from given two lists.

In [23]:
```
l1 = [10,20,302,3,12,3,2]
l2 = [2,334,23,55,67,8,3,302]
l3 = []
for i in l1:
    for j in l2:
        if i == j:
            l3.append(i)
print(l3)
```

```
[302, 3, 3, 2]
```

In [ ]:
```

```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294 - Jay Vegad

Lab - 6

Tuple

01) WAP to find sum of tuple elements.

In [2]:
```
t = (1,2,3,4,5,6,23,1,)
print(sum(t))
```

```
21
```

02) WAP to find Maximum and Minimum K elements in a given tuple.

In [1]:
```
tup = (1,6,3,2,9,6)
k = 2
t1 = list(tup)
t1.sort()

for i in range(k):
    print(t1[i])
    print(t1[-i+1])
```

```
1
2
2
1
```

03) WAP to find tuples which have all elements divisible by K from a list of tuples.

In [3]:
```
l1 = [(1,2,3,4),(5,6,7,8),(2,4,6,8),(1,2,5,7),(5,10)]
k = 2
count = 0

for i in l1:
    for j in i:
        if j % k != 0:
            count += 1

    if count == 0:
        print(i)
    count = 0
```

```
(2, 4, 6, 8)
```

04) WAP to create a list of tuples from given list having number and its cube in each tuple.

In [4]:
```
l1 = [1,2,3,4,5,6]

ans = [(i,i**3) for i in l1]
ans
```

Out[4]:
```
[(1, 1), (2, 8), (3, 27), (4, 64), (5, 125), (6, 216)]
```

05) WAP to find tuples with all positive elements from the given list of tuples.

In [6]:
```
l1 = [(1,2,3,4),(5,-6,7,8),(2,4,6,8),(1,-2,5,7),(5,-10)]
count = 0

for i in l1:
    if all(j>0 for j in i):
        print(i)
```

```
(1, 2, 3, 4)
(2, 4, 6, 8)
```

06) WAP to add tuple to list and vice – versa.

In [20]:
```
tuple_example = (1, 2, 3)
list_example = [5, 6, 7]

original_list = [123, True, "Ramji Premji", ['a', 's', 'd', 'f']]
original_list.append(tuple_example)

original_tuple = (1, 2, 4, 6, 8, 4)
```

```
original_tuple = list(original_tuple)
original_tuple.append(list_example)
original_tuple = tuple(original_tuple)
original_tuple
```

Out[20]: (1, 2, 4, 6, 8, 4, [5, 6, 7])

07) WAP to remove tuples of length K.

In [7]:
```
l1 = [(1,2,3,6),(1,2,5,7),(5,-6,7,8,9),(2,4,6,8),(5,-10),(1,1,1,1),(2,2,2,2)]
l2 = l1.copy()
k = 4

for i in l2:
    if len(i) == k:
        l1.remove(i)
print(l1)
```

```
[(5, -6, 7, 8, 9), (5, -10)]
```

08) WAP to remove duplicates from tuple.

In [8]:
```
tup = (1,1,2,3,4,3,6,8,9,9)
tup = set(tup)
tup = tuple(tup)
tup
```

Out[8]: (1, 2, 3, 4, 6, 8, 9)

09) WAP to multiply adjacent elements of a tuple and print that resultant tuple.

In [58]:
```
tup1 = (1,2,3,4,5)
tup1 = list(tup1)

ans = [(i*(i+1)) for i in range (1,len(tup1))]
ans
```

Out[58]: [2, 6, 12, 20]

10) WAP to test if the given tuple is distinct or not.

In [9]:
```
l1 = [(1,2,3,6),(1,2,5,7),(5,-6,7,8,9),(1,2,3,6),(5,-10),(1,1,1,1),(2,2,2,2)]

tup = (1,2,3,4,5)
t1 = list(tup)
t2 = set(tup)

if len(t1) == len(t2):
    print("No dups")
```

```
else:
    print("has dups")
```

```
No dups
```

In [10]:
```
l1 = [(1,2,3,6),(1,2,5,7),(5,-6,7,8,9),(1,2,3,6),(5,-10),(1,1,1,1),(2,2,2,2)]

for i in range(len(l1)):
    for j in l1[i+1:]:
        if l1[i] == j:
            print(l1[i])
```

```
(1, 2, 3, 6)
```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294 - Jay Vegad

Python Programming - 2301CS404

Lab - 7

Set & Dictionary

01) WAP to iterate over a set.

In [3]:
```
a = {1,2,3,4,5,123,34,5,43,3,2,3,5,43,2,23,2,1,2,3,4,3,2,1,2}

for i in a:
    print(i)
```

```
1
2
3
4
5
34
43
23
123
```

02) WAP to convert set into list, string and tuple.

In [2]:
```
a = {1,2,3,4,5,12,3,2,12,123,3,12,3,123,123,123,1,23,123,12,3}

print(list(a))
print("".join(map(str,a)))
print(tuple(a))
```

```
[1, 2, 3, 4, 5, 12, 23, 123]
123451223123
(1, 2, 3, 4, 5, 12, 23, 123)
```

03) WAP to find Maximum and Minimum from a set.

In [1]:
```
b = {1,2,3,4,5,6,7,87,8,45,63,4,5,343,324}

b = list(b)
print(f'Min: {b[0]}')
print(f'Max: {b[-1]}')
```

```
Min: 1
Max: 63
```

04) WAP to perform union of two sets.

In [4]:
```
a = {1,2,3,4,5,6,7,8,9,123}
b = {10,11,12,13,4,5,16,18,17,19,20}
c = a.union(b)
print(c)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 16, 17, 18, 19, 20, 123}
```

05) WAP to check if two lists have at-least one element common.

In [8]:
```
a = [1,2,3,4,5,6,7,8,9,123]
b = [10,11,12,13,4,5,16,18,17,19,20]
c = set(a).intersection(set(b))
print(list(c))
```

```
[4, 5]
```

06) WAP to remove duplicates from list.

In [10]:
```
a = [10,20,30,40,10,402,102,1022,3011]
b = set(a)
print(b)
```

```
{3011, 102, 40, 10, 402, 20, 1022, 30}
```

07) WAP to find unique words in the given string.

In [13]:
```
str = "Jay shree ram"
words = str.split(" ")
unique = set(words)
print(unique)
```

```
{'ram', 'shree', 'Jay'}
```

08) WAP to remove common elements of set A & B from set A.

In [16]:
```
a = {10,20,20,10,20,30}
b = {20,10,20,30,40,50}
a.union(b)
print(a)
```

```
{10, 20, 30}
```

09) WAP to check whether two given strings are anagram or not using set.

In [9]:
```
str1 = "add"
str2 = "aad"
str1Set = set(str1)
str2Set = set(str2)
combineStrSet = str1Set.union(str2Set)
orignalString = ','.join(combineStrSet)
if str1 == orignalString:
    print("Anagram")
else:
    print("Not Anagram")
```

```
Not Anagram
```

10) WAP to find common elements in three lists using set.

In [19]:
```
l1 = [10,20,30,40]
l2 = [20,30,20,10,20,30]
l3 = [50,60,30,20,10,]
combineSet = set(l1).union(set(l2)).union(set(l3))
print(combineSet)
```

```
{40, 10, 50, 20, 60, 30}
```

11) WAP to count number of vowels in given string using set.

In [68]:
```
str1 = "Jay shree ram"
count = 0
for i in str1:
    if i in {'a','e','i','o','u'}:
```

```
        count += 1
print(count)
```

```
4
```

12) WAP to check if a given string is binary string or not.

In [44]:
```
str = "abababababc"
stringSet = set(str)
if len(stringSet) == 2:
    print("binary string.. ")
else:
    print("not binary string")
```

```
not binary string
```

13) WAP to sort dictionary by key or value.

In [66]:
```
dict = {
    "name":"Ramlal",
    "age":12,
    "residence":"Darshan"
}
keys = list(dict.keys())
keys.sort()
sd = {i: dict[i] for i in keys}
print(sd)
```

```
{'age': 12, 'name': 'Ramlal', 'residence': 'Darshan'}
```

14) WAP to find the sum of all items (values) in a dictionary given by user. (Assume: values are numeric)

In [8]:
```
records = int(input("Enter Number of records:... "))
dictonary = {}
sum1 = 0
for i in range(records):
    key = input("Enter Value:.. ")
    value = input("Enter Value:... ")
    dictonary[key] = value
list1 = list(dictonary.values())
for i in list1:
    sum1 += int(i)
print(sum1)
```

```
Enter Number of records:... 2
Enter Value:.. jay
Enter Value:... 12
Enter Value:.. asd
Enter Value:... 23
35
```

15) WAP to handle missing keys in dictionaries.

Example : Given, dict1 = {'a': 5, 'c': 8, 'e': 2}

if you look for key = 'd', the message given should be 'Key Not Found', otherwise print the value of 'd' in dict1.

In [17]:
```
dict1 = {'a': 5, 'c': 8, 'e': 2,'d':10}
keys1 = dict1.keys()
keyList = list(keys1)
flag = True;
for i in keyList:
    if 'd' in i:
        flag = False
if flag == True:
    print("Key not found")
else:
    print(dict1['d'])
```

```
10
```

In [ ]:
```

```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294 - Jay Vegad

Lab - 8

User Defined Function

01) Write a function to calculate BMI given mass and height. (BMI = mass/h**2)

In [1]:
```
def bmi(m,h):
    return m/(h**2)

print(bmi(45,180))
```

```
0.001388888888888889
```

02) Write a function that add first n numbers.

In [4]:
```
def addn(n):
    add = 0
    for i in range(n+1):
        add+=i
    return add

print(addn(10))
```

```
55
```

03) Write a function that returns 1 if the given number is Prime or 0 otherwise.

In [10]:
```
def isPrime(n):
    if n <= 1:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return 0
    return 1

print(isPrime(7))
```

```
1
```

04) Write a function that returns the list of Prime numbers between given two numbers.

In [16]:
```
def isPrime(n1,n2):
    l = []
    for i in range (n1,n2+1):
        if i < 2:
            continue
        for j in range(2, int(i ** 0.5) + 1):
            if i % j == 0:
                break
        else:
            l.append(i)
    return l
print(isPrime(10,40))
```

```
[11, 13, 17, 19, 23, 29, 31, 37]
```

05) Write a function that returns True if the given string is Palindrome or False otherwise.

In [19]:
```
def isPalindrome(s):
    rev = s[::-1]
    if rev == s:
        return True
    else:
        return False

print(isPalindrome("helleh"))
```

```
True
```

06) Write a function that returns the sum of all the elements of the list.

In [20]:
```
def sumOfList(l):
    temp = sum(l)
    return temp

l = [1,2,4,6,8,9]
print(sumOfList(l))
```

```
30
```

07) Write a function to calculate the sum of the first element of each tuples inside the list.

In [23]:
```
def sumOfTpl(l):
    sum = 0
    for i in l:
        if i:
            sum+=i[0]
    return sum

l = [(1,2,3),(4,5),(6,7,8),(9,)]
print(sumOfTpl(l))
```

```
20
```

08) Write a recursive function to find nth term of Fibonacci Series.

In [9]:
```
def findNthFibb(n):
    if n<=1:
        return n

    return findNthFibb(n-1) + findNthFibb(n-2)

findNthFibb(6)
```

Out[9]: 8

09) Write a function to get the name of the student based on the given rollno.

Example: Given dict1 = {101:'Ajay', 102:'Rahul', 103:'Jay', 104:'Pooja'} find name of student whose rollno = 103

In [14]:
```
def findStu(roll,d):
    return d[roll]

d = {101:'Ajay', 102:'Rahul', 103:'Jay', 104:'Pooja'}
findStu(104,d)
```

Out[14]: 'Pooja'

10) Write a function to get the sum of the scores ending with zero.

Example : scores = [200, 456, 300, 100, 234, 678]

Ans = 200 + 300 + 100 = 600

In [21]:
```
def sumOfScore(l):
    sum = 0
    for i in l:
        if i%10 == 0:
            sum += i
    return sum

l = [200, 456, 300, 100, 234, 678]
sumOfScore(l)
```

Out[21]: 600

11) Write a function to invert a given Dictionary.

hint: keys to values & values to keys

Before : {'a': 10, 'b':20, 'c':30, 'd':40}

After : {10:'a', 20:'b', 30:'c', 40:'d'}

In [24]:
```
def invertDict(d1):
    d2 = {v:k for k,v in d1.items()}
    return d2

d1 = {'a': 10, 'b':20, 'c':30, 'd':40}
invertDict(d1)
```

Out[24]: {10: 'a', 20: 'b', 30: 'c', 40: 'd'}

12) Write a function to check whether the given string is Pangram or not.

hint: Pangram is a string containing all the characters a-z atlest once.

"the quick brown fox jumps over the lazy dog" is a Pangram string.

In [53]:
```
def isPan(s1):
    alphas = "abcdefghijklmnopqrstuvwxyz"
    check = True
    for i in alphas:
        if i not in s1:
            check = False
            break
    return check

s = "the quick brown fox jumps over the lazy dog"
isPan(s)
# alphas = "qwertyuiopasdfghjklzxcvbnm"
# new_str = "".join(sorted(alphas))
# print(new_str)
```

Out[53]: True

13) Write a function that returns the number of uppercase and lowercase letters in the given string.

example : Input : s1 = AbcDEfgh ,Ouptput : no_upper = 3, no_lower = 5

In [57]:
```
def findLowHigh(s1):
    l=0
    u=0
    for i in s1:
        if i.islower():
            l+=1
        elif i.isupper():
            u+=1
    l = [l,u]
    return l

findLowHigh("AbcDEfgh")
```

Out[57]: [5, 3]

14) Write a lambda function to get smallest number from the given two numbers.

In [59]:
```
x = lambda a,b : max(a,b)
print(x(10,20))
```

20

15) For the given list of names of students, extract the names having more that 7 characters. Use filter().

In [ ]:

16) For the given list of names of students, convert the first letter of all the names into uppercase. use map().

In [ ]:

17) Write udfs to call the functions with following types of arguments:

1. Positional Arguments
2. Keyword Arguments
3. Default Arguments

4. Variable Legngth Positional(*args) & variable length Keyword Arguments (**kwargs)
5. Keyword-Only & Positional Only Arguments

In [ ]:

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294- Jay Vegad

Lab - 9

File I/O

01) WAP to read and display the contents of a text file. (also try to open the file in some other directory)

- in the form of a string

- line by line

- in the form of a list

In [6]:
```
fp = open('a.txt','r')
print(fp.readlines())
fp.close()
```

```
['qwertyuiop asdfghjkl zxcvbnm\n', 'mnbvcxz lkjhgfdsa\n', 'poiuytrewq']
```

02) WAP to create file named "new.txt" only if it doesn't exist.

In [7]:
```
fp = open('new.txt','w')
fp.close()
```

03) WAP to read first 5 lines from the text file.

In [11]:
```
fp = open('a.txt','r')

for i in range(5):
    print(fp.readline())
fp.close()
```

```
qwertyuiop asdfghjkl zxcvbnm

mnbvcxz lkjhgfdsa

poiuytrewq

qwertyuiop asdfghjkl zxcvbnm

mnbvcxz lkjhgfdsa
```

04) WAP to find the longest word(s) in a file

In [ ]:
```
fp1 = open('a.txt','r')
l1 = fp1.read().split()

max = 0

for i in l1:
    if len(i) > max:
        max = len(i)

for i in l1:
    if len(i) == max:
        l2.append(i)

print(l2)
l2.clear()
```

05) WAP to count the no. of lines, words and characters in a given text file.

06) WAP to copy the content of a file to the another file.

In [44]:
```
fp1 = open('a.txt','r')
l1 = fp1.read()
fp1.close()

fp2 = open('b.txt','w')
fp2.write(l1)
fp2.close()
```

07) WAP to find the size of the text file.

In [45]:
```
fp = open('a.txt','r')
print(f'chars:{len(fp.read()) * 50}')
```

```
chars:1900
```

08) WAP to create an UDF named frequency to count occurances of the specific word in a given text file.

In [48]:
```
def frq(word):
    fp = open('a.txt','r')
    l = fp.read().split()
    cnt = 0

    for i in l:
        if i == word:
            cnt+=1

    return cnt

print(frq('bro'))
```

```
2
```

09) WAP to get the score of five subjects from the user, store them in a file. Fetch those marks and find the highest score.

In [64]:
```
l = [10,52,48,97,63]

fp1 = open('marks.txt','w')
l2 = [fp1.write(str(i)+" ") for i in l]
fp1.close()

fp2 = open('marks.txt','r')
l3 = fp2.read().split()
l4 = [int(i) for i in l3]
l4.sort()
l4[-1]
```

Out[64]:
```
97
```

10) WAP to write first 100 prime numbers to a file named primenumbers.txt

(Note: each number should be in new line)

In [13]:
```
fp = open('prime.txt','w+')
import math
def isPrime(n):
    for i in range(2,int(math.sqrt(n))+1):
        if n % i == 0:
            break
```

```
    else:
        return(n)

l = []
i = 2
while(len(l) != 100):
    if isPrime(i):
        l.append(i)
    i+=1

for i in l:
    fp.write(str(i)+'\n')
fp.close()
```

11) WAP to merge two files and write it in a new file.

In [ ]:

12) WAP to replace word1 by word2 of a text file. Write the updated data to new file.

In [ ]:

13) Demonstrate tell() and seek() for all the cases(seek from beginning-end-current position) taking a suitable example of your choice.

In [ ]:

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294 - Jay Vegad

Lab - 10

Exception Handling

01) WAP to handle following exceptions:

1. ZeroDivisionError
2. ValueError
3. TypeError

Note: handle them using separate except blocks and also using single except block too.

In [2]:
```
try:
    print(369/0)
except ZeroDivisionError:
    print("zero div error")

try:
    n = 'ramukaka'
    print(100/int(n))
except ValueError:
    print('Value Error')

try:
    lol = "halvai"
    num = 2
    print(lol + num + lol)
```

```
except TypeError:
    print('Type Error')
```

```
zero div error
Value Error
Type Error
```

02) WAP to handle following exceptions:

1. IndexError
2. KeyError

In [2]:
```
try:
    a = [1,2,3,4]
    print(a[5])
except IndexError:
    print('Index error')

try:
    a = {'name' : 'lol'}
    print(a['lol'])
except KeyError:
    print('Key error')
```

```
Index error
Key error
```

03) WAP to handle following exceptions:

1. FileNotFoundError
2. ModuleNotFoundError

In [5]:
```
try:
    fp = open('lol.txt','r')
except FileNotFoundError:
    print('FIle not found')
try:
    import Module_Not_Found_Mate
except ModuleNotFoundError:
    print('Moule not found')
```

```
FIle not found
Moule not found
```

04) WAP that catches all type of exceptions in a single except block.

In [12]:
```
try:
    print(369/0)
except:
    print('something went Wrong !')
```

```
something went Wrong !
```

05) WAP to demonstrate else and finally block.

In [15]:
```
try:
    print(100/0)
except ZeroDivisionError:
    print('Zero devision')
else:
    print('NO error accures')
finally:
    print('This is last Statement')
```

```
Zero devision
This is last Statement
```

06) Create a short program that prompts the user for a list of grades separated by commas.

Split the string into individual grades and use a list comprehension to convert each string to an integer.

You should use a try statement to inform the user when the values they entered cannot be converted.

In [ ]:
```
try:
    a = input('Enter grades').split(',')
    a2 = [int(i) for i in a]
except ValueError:
    print('Grade cannot converted into int')
else:
    print('You enterd right numbers')
```

07) WAP to create an udf divide(a,b) that handles ZeroDivisionError.

In [1]:
```
def divide(a,b):
    try:
        print(a/b)
    except ZeroDivisionError:
        print("Can not divede by zero")

divide(10,0)
```

```
Can not divede by zero
```

08) WAP that gets an age of a person form the user and raises ValueError with error message: "Enter Valid Age" :

If the age is less than 18.

otherwise print the age.

In [4]:
```
class SimpleClass(Exception):
   def __init__(self,msg):
        self.msg = msg

try:
    age = int(input("Enter a Age: "))

    if age < 18:
        raise SimpleClass("You are underage")
    else:
        print(age)

except SimpleClass as er:
    print(er)
```

```
You are underage
```

09) WAP to raise your custom Exception named InvalidUsernameError with the error message : "Username must be between 5 and 15 characters long":

if the given name is having characters less than 5 or greater than 15.

otherwise print the given username.

In [6]:
```
class InvalidUsernameError (Exception):
    def __init__(self,msg):
        self.msg = msg

try:
    s = input("Enter a Username")

    if len(s)<5 or len(s)>15:
        raise InvalidUsernameError("Invalid Name")
    else:
        print(s)

except InvalidUsernameError as e:
    print(e)
```

```
Invalid Name
```

10) WAP to raise your custom Exception named NegativeNumberError with the error message : "Cannot

calculate the square root of a negative number" :

if the given number is negative.

otherwise print the square root of the given number.

In [7]:
```
class NegativeNumberError(Exception):
    def __init__(self,msg):
        self.msg = msg

try:
    n = int(input("Enter a number: "))

    if n < 0:
        raise NegativeNumberError("No valid....")
    else:
        print(n ** (1/2))

except NegativeNumberError as e:
    print(e)
```

```
3.4641016151377544
```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294 - Jay Vegad

Lab - 11

Modules

01) WAP to create Calculator module which defines functions like add, sub,mul and div.

Create another .py file that uses the functions available in Calculator module.

In [5]:
```
import cal as fn
add = fn.add(2,3)
print(add)
```

```
5
```

02) WAP to pick a random character from a given String.

In [20]:
```
import random
str = 'Hello world from darshan university'
ran = random.randint(0,len(str))
print(str[ran])
```

```
v
```

03) WAP to pick a random element from a given list.

In [19]:
```
import random
list = [10,20,30,20,10,20,30,40,50,60,70,30,20,430,233]
ran = random.randint(0,len(list))
print(list[ran])
```

```
30
```

04) WAP to roll a dice in such a way that every time you get the same number.

In [26]:
```
import random
random.seed(10)
print(int(random.random() * 10))
```

```
5
```

05) WAP to generate 3 random integers between 100 and 999 which is divisible by 5.

In [55]:
```
import random
import math
ran1 = random.randint(100,999)
ran2 = random.randint(100,999)
ran3 = random.randint(100,999)
firstNum = math.ceil(ran1 / 5) * 5
secondNum = math.ceil(ran2 / 5) * 5
thirdNum = math.ceil(ran3 / 5) * 5
print(firstNum, secondNum, thirdNum)
```

```
300 760 705
```

06) WAP to generate 100 random lottery tickets and pick two lucky tickets from it and announce them as Winner and Runner up respectively.

In [75]:
```
import random
import math
lottery_tickets = []
for i in range(0,100):
    ran = int(random.random() * 10**10)
    lottery_tickets.append(ran)
winner1 = int(random.randint(0,100))
winner2 = int(random.randint(0,100))
print(f"First Winner is {lottery_tickets[winner1]} and Runner up is {lottery
```

```
First Winner is 1287108290 and Runner up is 1717449134
```

07) WAP to print current date and time in Python.

In [74]:
```
import datetime
now = datetime.datetime.now()
print(now)
```

```
2025-02-14 12:55:34.189472
```

08) Subtract a week (7 days) from a given date in Python.

In [83]:
```
from datetime import datetime, timedelta
now = datetime.now() + timedelta(days=-7)
print(now.date())
```

```
2025-02-07
```

09) WAP to Calculate number of days between two given dates.

In [89]:
```
from datetime import datetime

strdate1 = '14/02/2025'
strdate2 = '20/02/2022'

date1 = datetime.strptime(strdate1, "%d/%m/%Y")
date2 = datetime.strptime(strdate2, "%d/%m/%Y")


print(date1 - date2)
```

```
1090 days, 0:00:00
```

10) WAP to Find the day of the week of a given date.(i.e. wether it is sunday/monday/tuesday/etc.)

In [93]:
```
import datetime

date = datetime.date(2024, 7, 17)

day = date.strftime("%A")
sort_day = date.strftime("%a")

print(day_name, sort_day)
```

```
Wednesday Wed
```

11) WAP to demonstrate the use of date time module.

In [115…
```
from datetime import datetime, timedelta
strdate1 = '14/02/2025'

print(d1.date(2024, 7, 17))
print(d1.datetime(2024, 7, 17))
print(d1.time())
print(datetime.strptime(strdate1, "%d/%m/%Y"))
```

```
2024-07-17
2024-07-17 00:00:00
00:00:00
2025-02-14 00:00:00
```

12) WAP to demonstrate the use of the math module.

In [125…

```python
import math as m1

print(m1.pi)
print(m1.e)
print (math.cos(0.00))
print(math.sin(0.00))
print(math.atan(0.00))
print(math.tan(1.00))
```

```
3.141592653589793
2.718281828459045
1.0
0.0
0.0
1.5574077246549023
```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

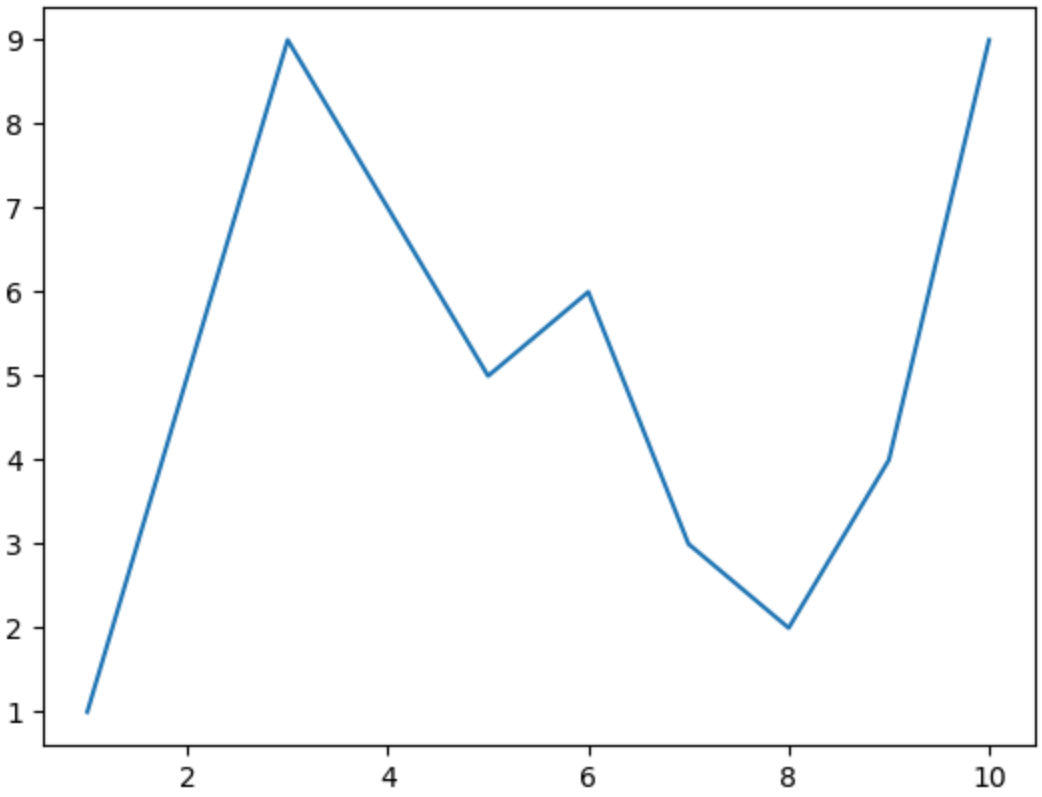
Python Programming - 2301CS404

23010101294 - Jay Vegad

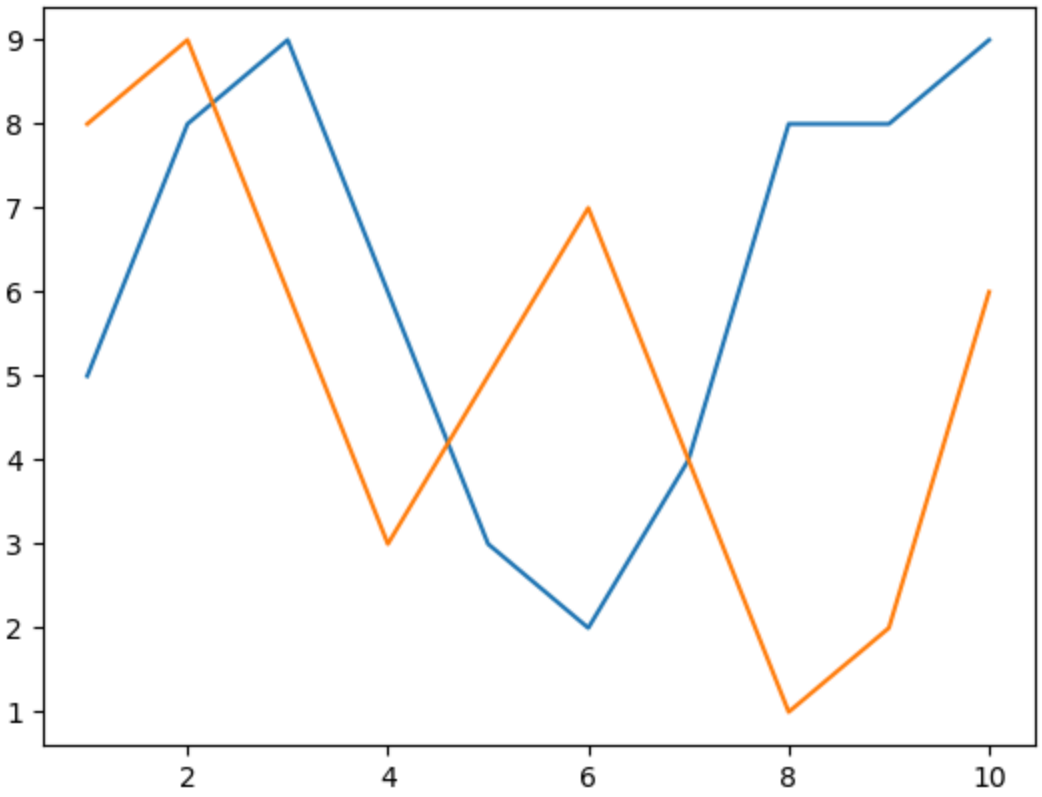
Lab - 12

In [2]:
```
#import matplotlib below
import matplotlib.pyplot as plt
```

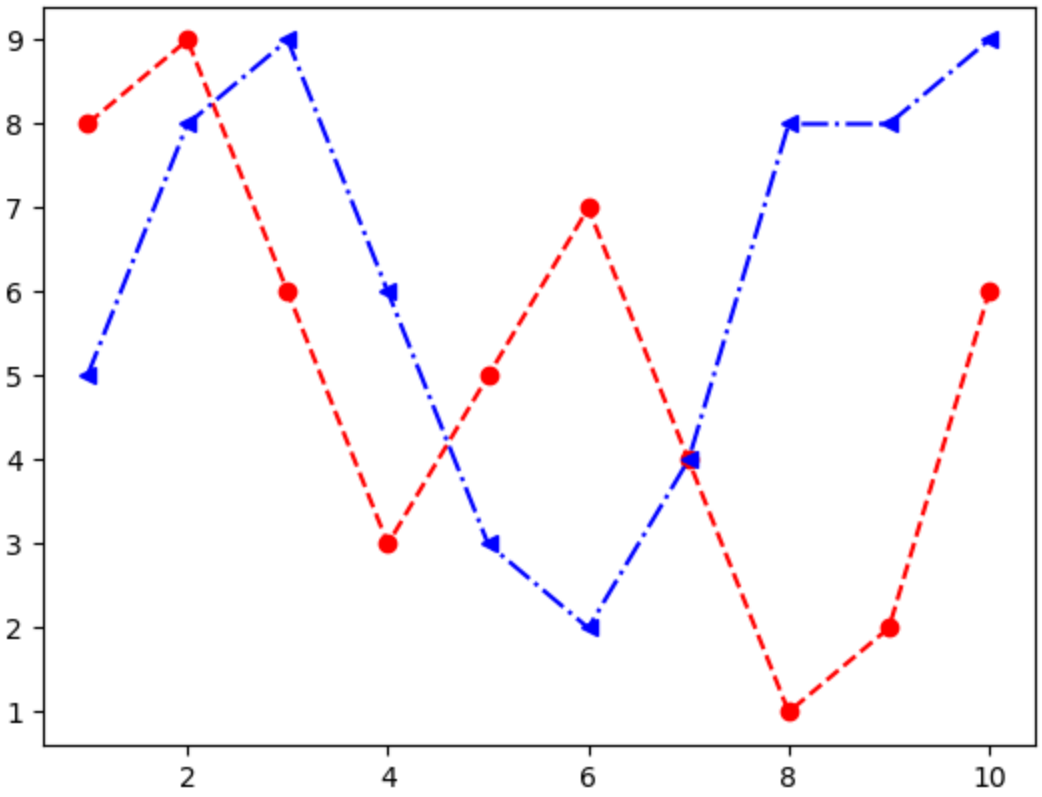
In [3]:
```
x = range(1,11)
y = [1,5,9,7,5,6,3,2,4,9]

# write a code to display the line chart of above x & y

plt.plot(x,y)
plt.show()
```



In [4]:
```
x = [1,2,3,4,5,6,7,8,9,10]
cxMarks = [5,8,9,6,3,2,4,8,8,9]
cyMarks = [8,9,6,3,5,7,4,1,2,6]

# write a code to display two lines in a line chart (data given above)
plt.plot(x,cxMarks)
plt.plot(x,cyMarks)
plt.show()
```



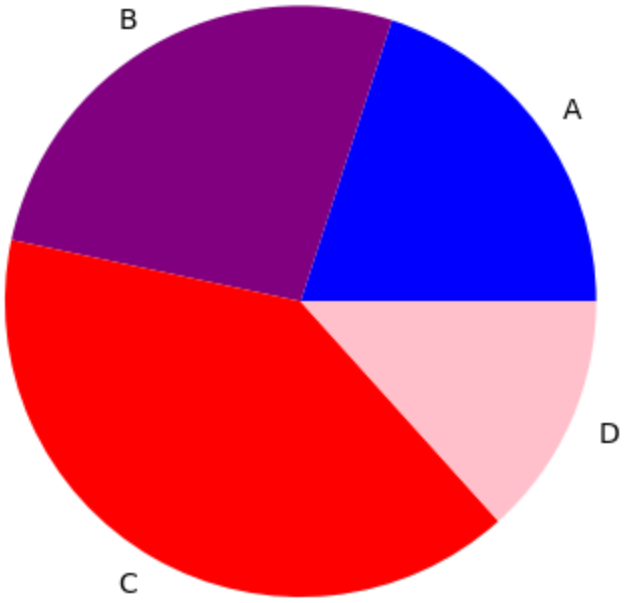
In [9]:
```
x = range(1,11,1)
cxMarks= [8,9,6,3,5,7,4,1,2,6]
cyMarks= [5,8,9,6,3,2,4,8,8,9]


# write a code to generate below graph
plt.plot(x,cxMarks,marker='o',color='r',label='cxMarks',linestyle='--')
plt.plot(x,cyMarks,marker='<',color='b',label='cyMarks',linestyle='-.')
plt.show()
```



In [ ]:

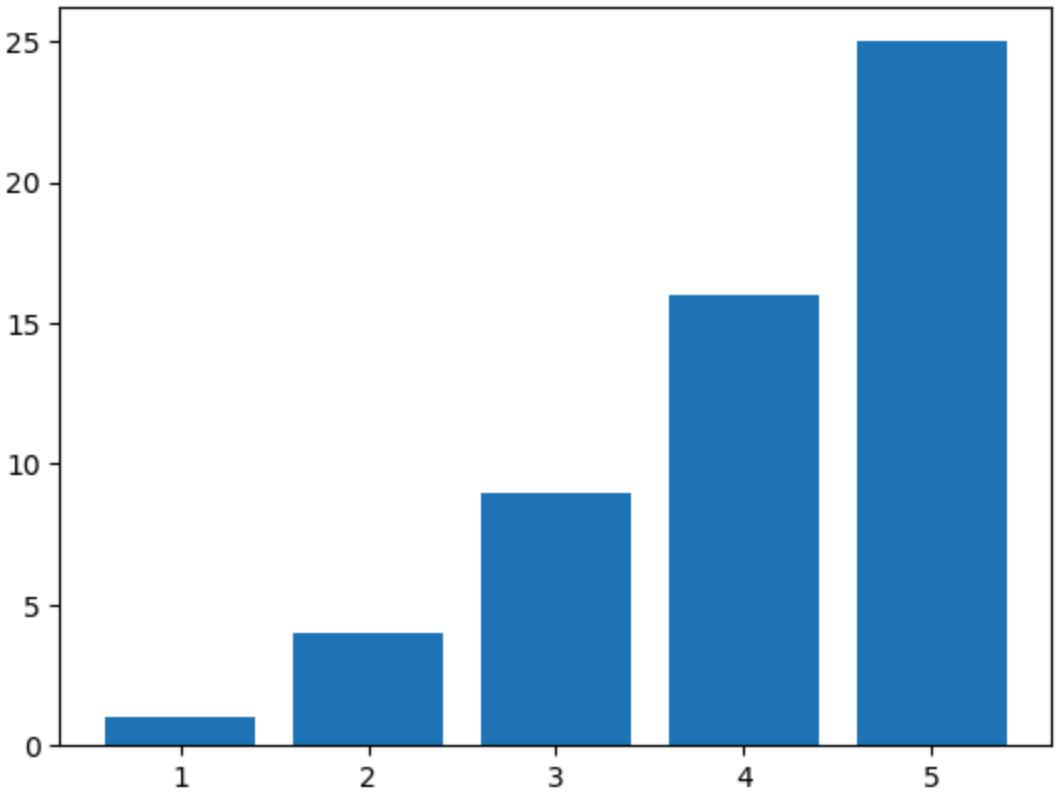
04) WAP to demonstrate the use of Pie chart.

In [12]:
```
#pie chart

sizes = [3,4,6,2]
labels = ['A','B','C','D']
colors = ['blue','purple','red','pink']
plt.pie(sizes,labels=labels,colors=colors)
plt.show()
```



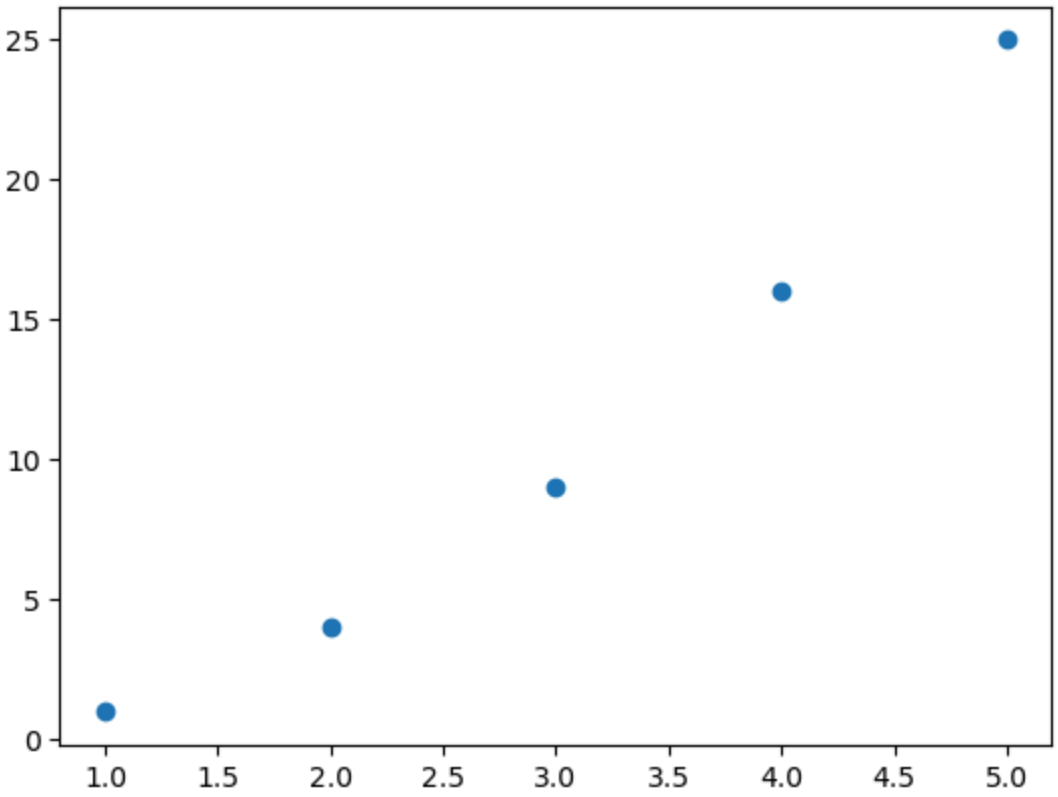
05) WAP to demonstrate the use of Bar chart.

In [13]:
```
x = [1,2,3,4,5]
y = [1,4,9,16,25]
plt.bar(x,y)
plt.show()
```



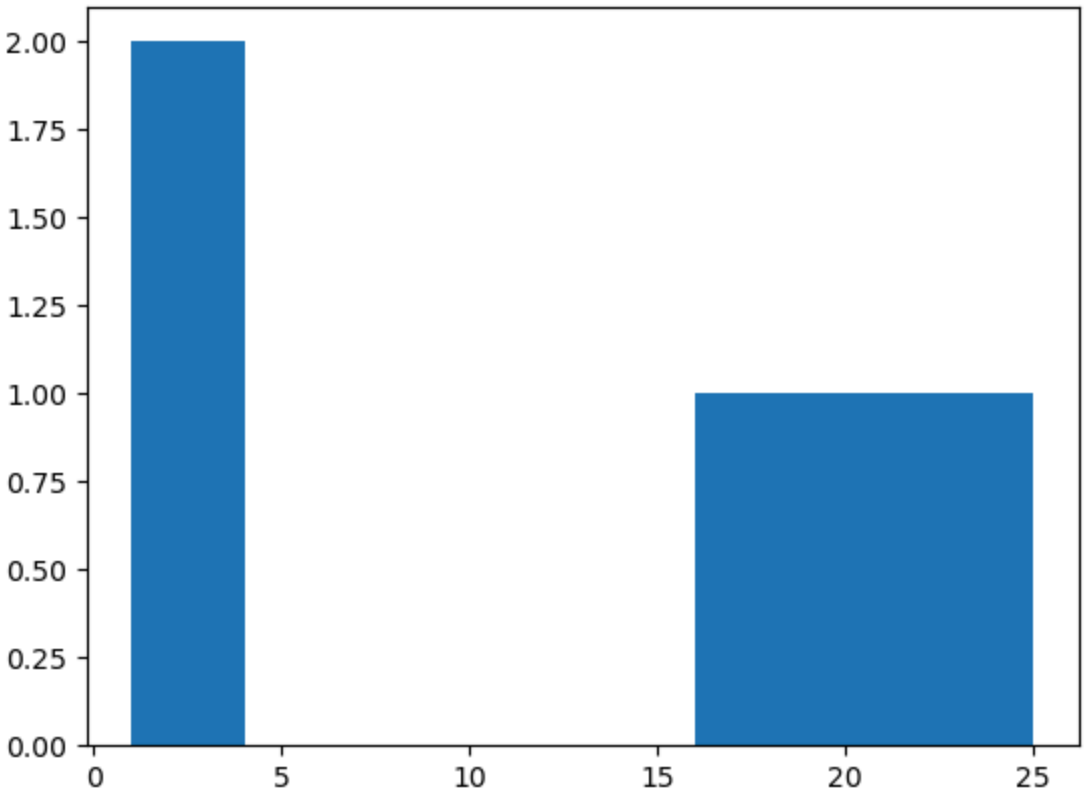
06) WAP to demonstrate the use of Scatter Plot.

In [14]:
```
x = [1,2,3,4,5]
y = [1,4,9,16,25]
plt.scatter(x,y)
plt.show()
```



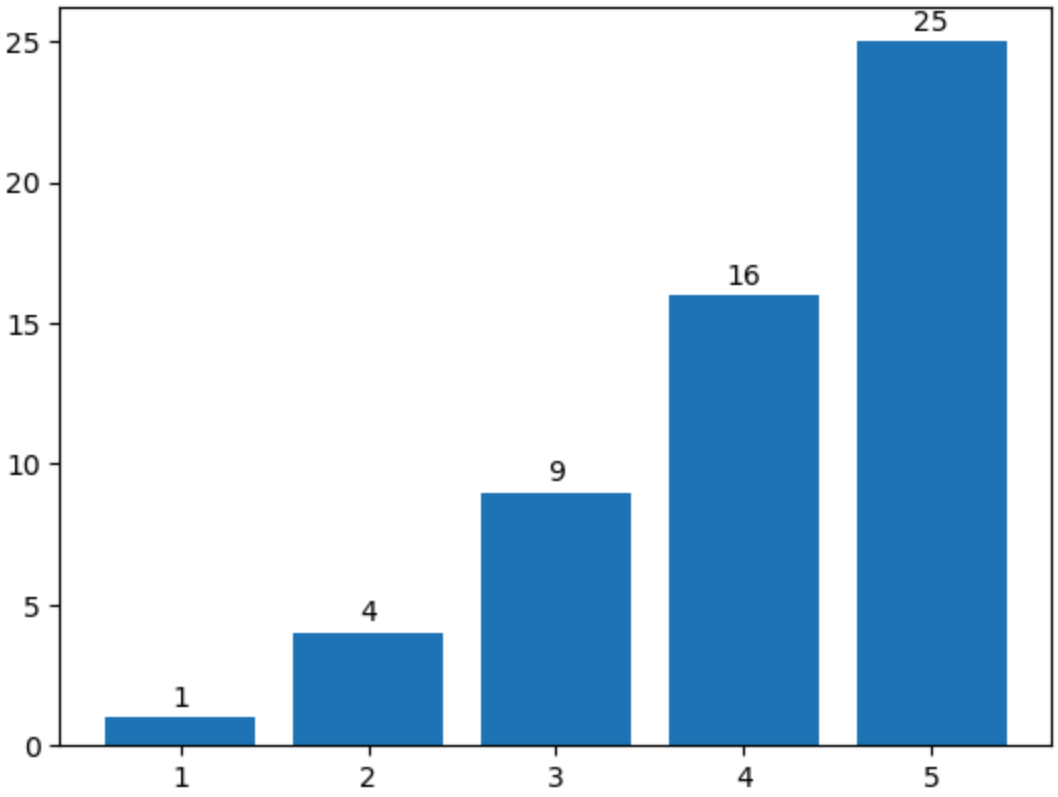
07) WAP to demonstrate the use of Histogram.

In [18]:
```
x = [1,20,3,40,50]
y = [1,4,9,16,25]

plt.hist(x,y)
plt.show()
```



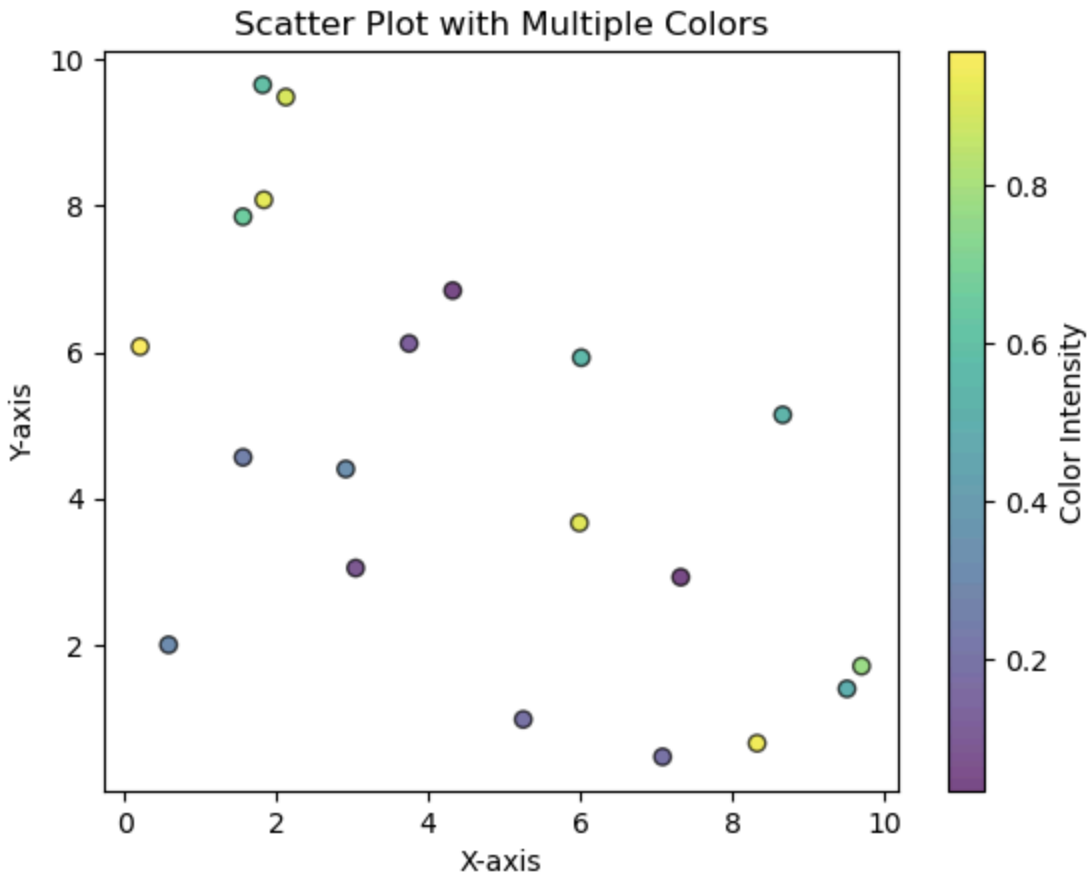
08) WAP to display the value of each bar in a bar chart using Matplotlib.

In [25]:
```
x = [1,2,3,4,5]
y = [1,4,9,16,25]
plt.bar(x,y)
for i, value in enumerate(y):
    plt.text(x[i], value + 0.2, str(value), ha='center', va='bottom')
plt.show()
```



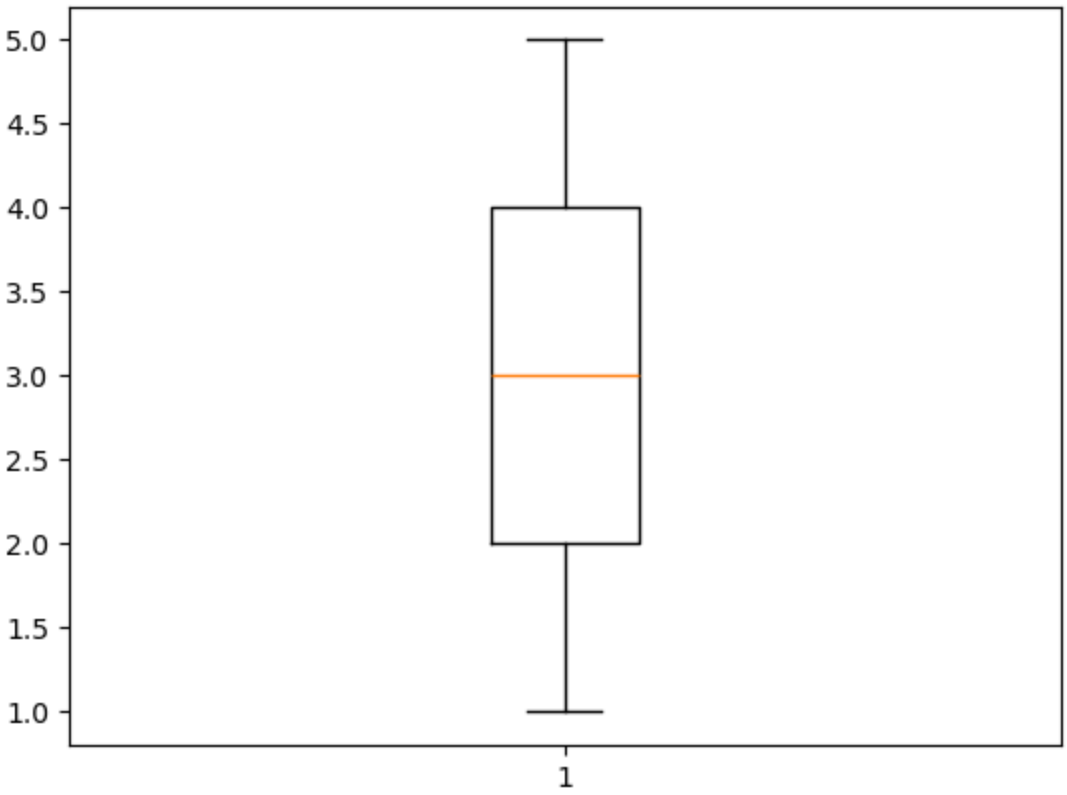
09) WAP create a Scatter Plot with several colors in Matplotlib?

In [30]:
```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(42)
x = np.random.rand(20) * 10
y = np.random.rand(20) * 10
colors = np.random.rand(20)

plt.scatter(x, y, c=colors, cmap='viridis', alpha=0.7, edgecolors='black')


plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Scatter Plot with Multiple Colors")

plt.colorbar(label="Color Intensity")

plt.show()
```



10) WAP to create a Box Plot.

In [34]:
```
x = [1,2,3,4,5]
plt.boxplot(x,horizontal=TRUE)
plt.show()
```



Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294 - Jay Vegad

Lab - 13

OOP

01) Write a Program to create a class by name Students, and initialize attributes like name, age, and grade while creating an object.

In [1]:
```
class Student:
    def __init__(self, name, age):
        print(f'Name: {name}, Age: {age}')

s1 = Student('KAMLESH', 13)
```

```
Name: KAMLESH, Age: 13
```

02) Create a class named Bank_Account with Account_No, User_Name, Email,Account_Type and Account_Balance data members. Also create a method GetAccountDetails() and DisplayAccountDetails(). Create main method to demonstrate the Bank_Account class.

In [9]:
```
class Bank_Account:

    def GetAccountDetails (self, acc_no, user_name, email, acc_type, balance
        self.acc_no = acc_no
        self.user_name = user_name
```

```
        self.email = email
        self.acc_type = acc_type
        self.balance = balance

    def DisplayAccountDetails(self):
        print(self.acc_no)
        print(self.user_name)
        print(self.email)
        print(self.acc_type)
        print(self.balance)

a1 = Bank_Account()
a1.GetAccountDetails(1,2,3,4,5)
a1.DisplayAccountDetails()
```

```
1
2
3
4
5
```

03) WAP to create Circle class with area and perimeter function to find area and perimeter of circle.

In [11]:
```
class circle:
    def __init__(self, r):
        self.r = r

    def calcPerimeter(self):
        print(2*3.14*self.r)

    def calcArea(self):
        print(3.14*(self.r ** 2))

c1 = circle(10)
c1.calcPerimeter()
c1.calcArea()
```

```
62.800000000000004
314.0
```

04) Create a class for employees that includes attributes such as name, age, salary, and methods to update and display employee information.

In [ ]:
```
class employee:
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary

    def updateUser(self,name,age,salary):
        self.name = name
        self.age = age
```

```
        self.salary = salary

    def displayUser(self):
        print(f'{self.name} {self.age} {self.salary}')

e1 = employee(1,2,3)
e1.updateUser(4,5,6)
e1.displayUser()
```

```
1 2 3
```

05) Create a bank account class with methods to deposit, withdraw, and check balance.

In [2]:

```
class bank:
    def __init__(self,cb):
        self.cb = cb

    def deposit(self, b):
        self.cb += b

    def withdraw(self, b):
        if self.cb < b:
            print("Not sufficiant balance")
        else:
            self.cb -= b

    def display(self):
        print("Current Balance: ", self.cb)

b1 = bank(10000)
b1.deposit(10000)
b1.withdraw(5000)
b1.display()
```

```
Current Balance:  15000
```

06) Create a class for managing inventory that includes attributes such as item name, price, quantity, and methods to add, remove, and update items.

In [19]:

```
class inv:
    def add(self,name, price, qnt):
        self.name = name
        self.price = price
        self.qnt = qnt
        print(f'item added {self.name}')

    def update(self,name, price, qnt):
        self.name = name
        self.price = price
        self.qnt = qnt
        print(f'item updated {self.name}')
```

```
    def remove(self):
        self.name = ""
        self.price = ""
        self.qnt = ""
        print(f'item removed {self.name}')

c1 = inv()
c1.add(1,2,3)
c1.update(4,5,6)
c1.remove()
```

```
item added 1
item updated 4
item removed
```

07) Create a Class with instance attributes of your choice.

In [3]:
```
class NoNamedClass:
    def __init__ (self,name):
        self.name = name

m1 = NoNamedClass("Free Fire")
m2 = NoNamedClass("Pubg")

print(m1.name)
print(m2.name)
```

```
Free Fire
Pubg
```

08) Create one class student_kit

Within the student_kit class create one class attribute principal name (Mr ABC)

Create one attendance method and take input as number of days.

While creating student take input their name .

Create one certificate for each student by taking input of number of days present in class.

In [5]:
```
class student_kit:
    prn = "Narendra Modi"
    def __init__(self, stn):
        self.stn = stn

    def atndnc(self, prdays):
        self.prdays = prdays

    def cirty(self):
        print(f' {self.stn} {self.prdays}')
```

```
s1 = student_kit("Vahh")
s1.atndnc(123)
s1.cirty()
```

```
Vahh 123
```

09) Define Time class with hour and minute as data member. Also define addition method to add two time objects.

In [31]:
```
class samay:
    def __init__(self, hr, m):
        self.hr = hr
        self.m = m

    def addTimes(self, t1, t2):
        self.m = t1.m + t2.m
        self.hr = t1.hr + t2.hr
        if self.m >= 60:
            self.hr += 1
            self.m -= 60

        print(f'{self.hr}:{self.m}')

s1 = samay(0,0)
t1 = samay(1,60)
t2 = samay(2,30)
s1.addTimes(t1,t2)
```

```
4:30
```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

23010101294 - Jay Vegad

Lab - 13

Continued..

10) Calculate area of a ractangle using object as an argument to a method.

In [3]:
```
class Calculator:
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def findArea(self):
        area = self.length * self.width
        print(area)

c = Calculator(10,20)
c.findArea()
```

```
200
```

11) Calculate the area of a square.

Include a Constructor, a method to calculate area named area() and a method named output() that prints the output and is invoked by area().

In [25]:
```
class CalcArea:
    area = 0
    def __init__(self, length):
        self.length = length

    def area(self):
        area = self.length * self.length
        self.output(area)

    def output(self,a):
        print(f"Output of Squre is {a}")

c = CalcArea(10)
c.area()
```

```
Output of Squre is 100
```

12) Calculate the area of a rectangle.

Include a Constructor, a method to calculate area named area() and a method named output() that prints the output and is invoked by area().

Also define a class method that compares the two sides of reactangle. An object is instantiated only if the two sides are different; otherwise a message should be displayed : THIS IS SQUARE.

In [43]:
```
class Calulatearea:
    length = 0
    width = 0
    def __init__(self,length, width):
        if length == width:
            print("Given value is about squre")
            return
        self.length = length
        self.width = width



    def area(self):
        area = self.length * self.width
        self.output(area)

    def output(self,a):
        print(f"Area of rect is {a}")


c = Calulatearea(10,20)
c.area()
```

```
Output of Squre is 200
```

13) Define a class Square having a private attribute "side".

Implement get_side and set_side methods to accees the private attribute from outside of the class.

In [47]:
```
class Simpleclass:
    __side = 0

    def __init__(self,side):
        self.__side = side

    def get_side(self):
        print(self.__side)

    def set_side(self, newSide):
        self.__side = newSide
        print(self.__side)

s = Simpleclass(10)
s.get_side()
s.set_side(20)
```

```
10
20
```

14) Create a class Profit that has a method named getProfit that accepts profit from the user.

Create a class Loss that has a method named getLoss that accepts loss from the user.

Create a class BalanceSheet that inherits from both classes Profit and Loss and calculates the balanace. It has two methods getBalance() and printBalance().

In [51]:
```
class Profit:
    def get_profit(self, profit):
        self.profit = profit

class Loss:
    def get_loss(self,loss):
        self.loss = loss

class BalanceSheet(Profit, Loss):
    def getBalance(self):
        balance = self.profit - self.loss
        return balance

    def printBalance(self):
        print(self.getBalance())
```

```
blc = BalanceSheet()
blc.get_profit(1000)
blc.get_loss(200)
blc.printBalance()
```

```
800
```

15) WAP to demonstrate all types of inheritance.

In [53]:
```
# Python program to demonstrate
# hybrid inheritance


class School:
    def func1(self):
        print("This function is in school.")


class Student1(School):
    def func2(self):
        print("This function is in student 1. ")


class Student2(School):
    def func3(self):
        print("This function is in student 2.")


class Student3(Student1, School):
    def func4(self):
        print("This function is in student 3.")


# Driver's code
object = Student3()
object.func1()
object.func2()


# ---------------------------------------------------------
# Python program to demonstrate
# Hierarchical inheritance


# Base class
class Parent:
    def func1(self):
        print("This function is in parent class.")

# Derived class1


class Child1(Parent):
    def func2(self):
```

```
        print("This function is in child 1.")

# Derivied class2


class Child2(Parent):
    def func3(self):
        print("This function is in child 2.")


# Driver's code
object1 = Child1()
object2 = Child2()
object1.func1()
object1.func2()
object2.func1()
object2.func3()

class Grandfather:

    def __init__(self, grandfathername):
        self.grandfathername = grandfathername


# ---------------------------------------------------------------------


class Father(Grandfather):
    def __init__(self, fathername, grandfathername):
        self.fathername = fathername

        # invoking constructor of Grandfather class
        Grandfather.__init__(self, grandfathername)

# Derived class


class Son(Father):
    def __init__(self, sonname, fathername, grandfathername):
        self.sonname = sonname

        # invoking constructor of Father class
        Father.__init__(self, fathername, grandfathername)

    def print_name(self):
        print('Grandfather name :', self.grandfathername)
        print("Father name :", self.fathername)
        print("Son name :", self.sonname)


#  Driver code
s1 = Son('Prince', 'Rampal', 'Lal mani')
print(s1.grandfathername)
s1.print_name()
```

```
This function is in school.
This function is in student 1.
This function is in parent class.
This function is in child 1.
This function is in parent class.
This function is in child 2.
Lal mani
Grandfather name : Lal mani
Father name : Rampal
Son name : Prince
```

16) Create a Person class with a constructor that takes two arguments name and age.

Create a child class Employee that inherits from Person and adds a new attribute salary.

Override the **init** method in Employee to call the parent class's **init** method using the super() and then initialize the salary attribute.

In [ ]:
```python
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

class Employee(Person):
    def __init__(self, name, age, salary):
        super().__init__(name, age)
        self.salary = salary
```

17) Create a Shape class with a draw method that is not implemented.

Create three child classes Rectangle, Circle, and Triangle that implement the draw method with their respective drawing behaviors.

Create a list of Shape objects that includes one instance of each child class, and then iterate through the list and call the draw method on each object.

In [57]:
```python
class Shape:
    def draw(self):
        return

class Rect(Shape):
    def draw(self):
        print("Rect")
```

```
class Cir(Shape):
    def draw(self):
        print("Cir")

class Tri(Shape):
    def draw(self):
        print("Tri")

rect = Rect()
cir = Cir()
tri = Tri()

li = [rect, cir, tri]

for i in li:
    i.draw()
```

```
Rect
Cir
Tri
```

In [ ]: