**COURSE OUTCOME 3(CO3):**

1. Work with built-in packages

import statistics

# Calculate average values

print("Mean : ",statistics.mean([1, 3, 5, 7, 9, 11, 13]))

print("Mean : ",statistics.mean([1, 3, 5, 7, 9, 11]))

print("Mean : ",statistics.mean([-11, 5.5, -3.4, 7.1, -9, 22]))

print("===============================")

# Calculate middle values

print("Median : ",statistics.median([1, 3, 5, 7, 9, 11, 13]))

print("Median : ",statistics.median([1, 3, 5, 7, 9, 11]))

print("Median : ",statistics.median([-11, 5.5, -3.4, 7.1, -9, 22]))

print("===============================")

# Calculate the mode

print("Mode :",statistics.mode([1, 3, 3, 3, 5, 7, 9, 11]))

print("Mode :",statistics.mode([1, 1, 3, -5, 7, -9, 11]))

print("Mode :",statistics.mode(['red', 'green', 'blue', 'red']))

print("===============================")

# Calculate the variance from a sample of data

print("Varience :",([1, 3, 5, 7, 9, 11]))

print("Varience :",statistics.variance([2, 2.5, 1.25, 3.1, 1.75, 2.8]))

print("Varience :",statistics.variance([-11, 5.5, -3.4, 7.1]))

print("Varience :",statistics.variance([1, 30, 50, 100]))

print("===============================")

# Calculate harmonic mean

print("Hermonic mean",statistics.harmonic\_mean([40, 60, 80]))

print("Hermonic mean",statistics.harmonic\_mean([10, 30, 50, 70, 90]))

print("-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-")

import random

print(random.random())

print("===============================")

mylist = ["apple", "banana", "cherry"]

random.shuffle(mylist)

print(mylist)

print("===============================")

random.seed(10)

print(random.random())

print("===============================")

mylist = ["apple", "banana", "cherry"]

print(random.choice(mylist))

print("===============================")

print(random.randrange(3, 9))

**OUTPUT**

Mean : 7

Mean : 6

Mean : 1.8666666666666667

===============================

Median : 7

Median : 6.0

Median : 1.05

===============================

Mode : 3

Mode : 1

Mode : red

===============================

Varience : [1, 3, 5, 7, 9, 11]

Varience : 0.4796666666666667

Varience : 70.80333333333334

Varience : 1736.9166666666667

===============================

Hermonic mean 55.38461538461538

Hermonic mean 27.97513321492007

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

0.9903035965488661

===============================

['cherry', 'banana', 'apple']

===============================

0.5714025946899135

===============================

banana

===============================

6

2. Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import \* statements)

**rectangle.py**

def rectangle(l,b):

print("Area of rectangle :",l\*b)

print("Perimeter of rectangle :",2\*(l+b))

**circle.py**

def circle(r):

def circle(r):

print("Area of Circle :",3.14\*r\*r)

print("Perimeter of rectangle :",2\*3.14\*r)

**appackage.py**

from Graphics import circle

from Graphics import rectangle

l = int(input("Enter length of rectangle :"))

b = int(input("Enter breadth of rectangle :"))

rectangle.rectangle(l,b)

r = int(input("Enter Radius of circle :"))

circle.circle(r)

Perimeter of \*r)

**OUTPUT**

Enter length of rectangle :5

Enter breadth of rectangle :3

Area of rectangle : 15

Perimeter of rectangle : 16

Enter Radius of circle :7

Area of Circle : 153.86

Perimeter of circle : 43.96