Task - 3.1

Date : 20/8/25

Aim: To implement and demonstrate the process of importing built-in modules, creating user-defined modules, and organizing code into packages code into packages in python, thereby promoting code reusability, modularity, and maintainability.

Algorithm:

→ Import required modules: math, random, os, sys, statistics, pathlib.

→ Math & random:

- compute sqrt(5), radians(30), a random float in [0.0, 1.0), a random integer in [2, 6] (inclusive), ceil(2.3), floor(2.3), factorial(5), gcd(5, 15), abs(-10), pow(3, 5), log base 3 of 2, log10(a) for a=100, and check if NaN/Infinity.

→ os & sys:

- create c:\pythonlab if not present and print the current working directory.
- create c:\pythonslot214 if not present and change the current working directory to it.

→ statistics:

- on List1: [5, 6, 8, 10] and [2, 5, 3, 2, 8, 3, 9, 4, 2, 5, 6], compute mean, median, mode, stdev.

→ print neatly formatted results.

Program:

```
import math
import random
import os
import sys
import statistics as stats
from pathlib import Path

print ("\n---MATH & RANDOM---")
print ("sqrt(5)=", math.sqrt(5))
print ("radians(30)=", math.radians(30))
print ("random()in[0,1) =", random.random())
print ("randint(2,6)=", random.randint(2,6)) #inclusive
print ("pi=", math.pi)
print ("ceil(2.3)=", math.ceil(2.3))
print ("floor(2.3)=", math.floor(2.3))
print ("factorial(5)=", math.factorial(5))
print ("gcd(5,15)=", math.gcd(5,15))
print ("abs(-10)=", abs(-10))
print ("pow(3,5)=", pow(3,5))
print ("log base 3 of 2 =", math.log(2,3))
a_val = 100
print (f'log10({a_val}) =", math.log10(a_val))
inf_val = float('inf')
nan_val = float('nan')
print(f"isinf(∞) = {math.isinf(inf_val)}, isnan(NaN)=
                {math.isnan(nan_val)}")
print ("\n--- OS & SYS---")
path = pythonlab = Path(r"C:\pythonlab")
path.pythonlab.mkdir(parents=True, exist_ok=True)
print(f"created/ensured: {path.pythonlab}")
```

```
print ("current working directory:" os.getcwd())
target_dir = path (r"C:\python lot 9214")
target_dir.mkdir (parents = True, exist_ok = True)
os.chdir (target_dir)
print (f"changed into: {target_dir}")
Print ("Directory contents:", os.listdir())
print ("Python version:", sys.version)
Print ("\n --- STASTISTICS ---")
data 1 = [5,6,8,10]
data 2 = [2,5,3,2,8,3,9,4,2,5,6]
print (f'mean({data1}) =", stats.mean (data 1))
Print(f'median({data1}) =", stak.median (data 1))
print (f'mode ({data2}) = ", stak.mode[data2])
print (f'stdev [{data 2}) = ", stak.stdev (data 2)
```

output:-

```
= = = MATH & RANDOM ---
sqrt(5) -2.2360678774999.79
radius (30) = 0.5235987755982988
return ( ) in [0,1] = 0.37444887175646646
rand. int [2,6] = 6
Pi = 3.141592623589793
ceil (2-3) = 3
floor (2-3) = 2
factorial (5) = 120
gcd (5,15) = 5
abs (-10) = 10
```

pow (3.5) = 24.3

log base 3 of 2 = 0.6309297535714574

log 10(100) = 2.0

isinf (2) = True, isnan(NaN) = True

---- OS & SYS ---

Created/ensured :(:\Python lab)

current working directory :(-__ (your current path)

current /ensured & changed into: (:\Python slot S2L4)

Directory contents of (:\Python slot S2L4:[]

Python ~~contents~~ Of
version = 3.xx (--- details..)

--- STATISTICS ---

mean ([5,6,8,10]) = 7.25

median ([5,6,7,10] = 7.0

mode ([2,5,3,2,8,3,9,2,5,6]) = 2

stdev ([2,5,3,2,8, 3,9,4,5,6])

= 2.27156333 8320093

Task No-3.2

Date:

Aim:- To create a python package named cardpack containing a module cardfun that imports the random module. Assign a range of cards, call a function from the module, and display a random sample of cards.

Algorithm:-

-> Start
-> To create a package cardpack
-> To create a module cardfun and import random functio
-> Assign a cards range.
-> Call a module function.
-> Display the random sample cards s
-> Stop.

Program:

Cardfun

```
import random
def func():
 cards = [ ]
for i in range (1,53):
 cards . append (i)
shuffled — cards = random.sample (cards, k = 52)
  print ("\n\n", shuffled — cards, "\n\n")
```

Mymod.py

```
  import cardfun
   cardfun.func()
```

output :

[5, 24, 13, 22, 20, 41, 38, 51, 4, 7, 34, 49, 14, 50, 37, 40, 6,
35, 17, 18, 33, 39, 36, 42, 12, 16, 19, 48, 29, 2, 27, 11, 31,
46, 28, 21, 32, 8, 25, 30, 23, 26, 10, 43, 47, 3, 44, 52, 1, 45,

Task NO-3.3

Date: 20/8/25

Aim: To develop a modular calculator application in python.

~~Task 1~~-

Algorithm:-

-> Define functions for addition, subtraction, multiplication and division.
-> Handle division by zero by raising an error if the divisor is zero.
-> Import the module (mymath) containing these functions.
-> Initialize two numbers (a=10, b=5).
-> call each function using mymath.<function-name>(a,b).
-> print the results of all operations.

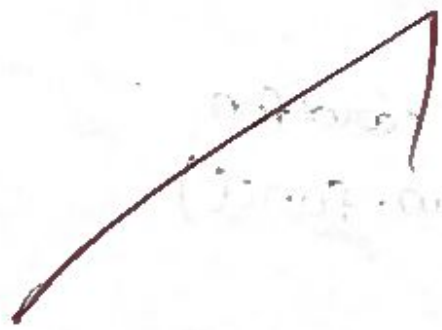
Program:

```
def add(a,b):
    return a+b
def subtract(a,b):
    return a-b
def multiply(a,b):
    return a*b
def divide(a,b):
    if b==0:
        raise value error("cannot divide by zero")
    return a/b
import mymath
a=10
b=5
print("addition",mymath.add(a,b))
print("subtraction:",mymath.subtract(a,b))
print("multiplication:",mymath.multiply(a,b))
print("Division:", mymath.divide(a,b))
```

output:

= = = = = RESTART :C =

Addition : 15
Subtraction: 5
multiplication: 50
Division : 2.0



Task NO- 3.4

Date :- 30/8/25

Aim :-

To perform various mathematical operations and geometric area calculations for python project.

Algorithm :-

-> create math-functions.py module:
-> create area-functions.py module:
-> create main.py:
-> print the output as expected.

Program :-

1. create the math-functions.py module

```
def add(a,b):
    return a+b
def subtract(a,b):
    return a-b
def multiply(a,b):
    return a*b
def divide(a,b):
    if b==0:
    return "Error! Division by zero!"
    return a/b
```

2. Create the area-functions.py module

```
import math
def circle_area(radius):
return math.pi*radius*radius
def rectangle_area(length,width):
    return length*width
def triangle_area(base,height):
    return 0.5*base*height.
```

output :-

= = > ==> RESTART : C :

Addition : 15
Subtraction : 5
multiplication : 50
Division : 2 : 0
Circle Area (radius = 7) : 153 9380400258 9985
Rectangle area (5 x 10) : 50
Triangle Area (base=6, height=8) : 24.0

3. Create the main.py file

```
import math_functions
import area_functions
#using math functions
Print("Addition:", math_functions.add(10,5))
print("Subtraction:", math_functions.subtract(10,5))
Print("Multiplication:", math_functions.multiply(10,5))
Print("Division", math functions.divide(10,5))
#using area functions
print("circle area(radius=7):", area functions.
                                circle_area(7))
print("Rectangle area(5x10):", area functions.rectangle
                                - area(5,10))
print("Triangle (base=6, height=8):", area functions.
                                triangle_area(6,8))
```

EX NO.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	3
VIVA VOCE (3)	3
RECORD (4)	4
TOTAL (15)	
SIGN WITH DATE	

Result: Thus, the Program for importing Python modules and Package was Successfully executed and the output was verified.