TASK-3 : Importing and creating Python modules and Packages in Python Program.

Aim: To implement and demonstrate the Process of importing built-in modules, creating user-defined modules and organizing code into Packages in Python, thereby Promoting code reusability, modularity and maintainability.

3.1

1) Perform common math and random operations.
2) work with the operating system (create/change directories, list contents) and read the Python version.
3) compute basic statistics (mean, median, mode, standard deviation)

Algorithm:

1) Import required modules: math, random, os, sys, statistics, Pathlib.
2) Math & random:
- compute sqrt(5), radians(30), a random float in [0.0, 1.0), a random integer in [3,6] (inclusive), π, ceil(2.3), floor(2.3), factorial(5), gcd(5,15), abs(-10), pow(3,5), log base 3 of 2, log 10(a) for a=100, and check NaN/Infinity.
3) os & sys:
- create C:\Python lab if not Present and Print the current working directory.
- create C:\Python slot S2L4 if not Present and change the current working directory to it.
- List all files/directories in the new current directory.
- Print Python interpreter version.
4) statistics:
- on lists: [5,6,8,10] and [2,5,3,2,8,3,9,4,2,5,6], compute mean, median, mode, stdev.
5) 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_Python lab = Path (r"c:\Python lab")
Path_Python lab.mkdir (Parents = True, exist_ok = True)
```

Expected sample output:

--- MATH & RANDOM ---
sqrt(5) = 2.23606797749979
radians(30) = 0.5235987755982988
random() in [0,1) = 0.3744488 7175646646 <- will vary
randint(2,6) = 6 <- inclusive; will vary
Pi = 3.141592653589793
ceil(2.3) = 3
floor(2.3) = 2
factorial(5) = 120
gcd(5,10) = 5
abs(-10) = 10
Pow(3,5) = 243
log base 3 of 2 = 0.6309297535714574
log 10(100) = 2.0
isinf(∞) = True, isnan(nan) = True
--- OS & SYS ---
created / ensured: C:\Python lab
current working directory: C:\... (your current path)
created/ensured & changed into: C:\Python slot 5214
Directory contents of C:\Python slot 5214: []

Python version: 3.x.x (... details ...)
--- STATISTICS ---
mean([5,6,8,10]) = 7.25
median([5,6,8,10]) = 7.0
mode([2,5,3,2,8,3,9,4,2,5,6]) = 2
stdev([2,5,3,2,8,3,9,4,2,5,6]) = 2.2715633383201093

```
Print(f"created / ensured: {path_Python lab}")
Print ("current working directory:", os.getcwd())
target_dir = Path (r"c :\Pythonslots2CL")
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-- STATISTICS ---")
data1 = [5, 6, 8, 10]
data2 = [2, 5, 3, 2, 8, 3, 9, 4, 2, 5, 6]
Print (f"mean ({data1}) = ", stats.mean (data1))
Print (f"median ({data1}) = ", stats.median (data1))
Print (f"mode ({data2}) = ", stats.mode (data2))
Print (f"stdev ({data2}) = ", stats.stdev (data2))
```

Result:

Hence, To implement and demonstrate the Process of importing built-in-modules using Python is successfully executed and output is displayed.

My.mod.py

```
import cardFun
cardFun.func()
```

Out Put:

RESTART:

C:\Users\student.MAT2VC6833\AppData\Local\Programs\Python\Python311\Lib\site-packages\cardPack\myMod.py

[5, 24, 13, 22, 20, 41, 38, 51, 4, 7, 34, 49, 14, 50, 37, 40, 15, 35, 17, 18, 33, 39, 36, 42, 12, 6, 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, 9]

3.2. 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:

step 1 : start
2. To create a Package card pack.
3. To create a module cardFUN and import random function
4. Assign a cards range.
5. call a module function.
6. Display the random sample cards
7. 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"]
```

Result:

Hence, creating a Python Package and cardpack is successfully completed.

Program: (my) math,

```
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==20.
raise value Error ("cannot divide by zero")
 return a/b
Import my math
 a=10
 b=5.
Print ("Addition:", mymath. add (a,b))
Print ("subtraction:", my math. subtract(a,b))
Print ("multiplication:", "mymath. multiply (a,b))
Print ("Division:", my math. divide (a,b))
```

output:

Addition : 15
subtraction : 5
Multiplication : 50
Division =2.0.

3.3 You are tasked with developing a modular calculator application in Python. The calculator should support basic arithmetic operations: addition, subtraction, multiplication, and division. Each operation should be implemented in a separate module Additionally, you should create a main program to handle user input, call the appropriate module and display the results.

Algorithm:

1. Define functions for addition, subtraction, Multiplication and division
2. Handle division by zero by raising an error if the divisor is zero.
3. Import the module (mymath) containing these functions.
4. Initialize two numbers (a=10, b=5)
5. Call each function using mymath.<function_name>(a,b).
6. Print the results of all operations.

Result:

Hence the modular calculator by using Python is successfully completed and output is displayed.

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
```

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 ("Divison:", 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 area (base=6, Height=8):", area functions. triangle_area
(6,8))
```

3.4: you are working on a Project that requires you to perform various mathematical operations and geometric area calculations. To organize your code better, you decide to create a package named my package which Includes sub demonstrate the use of the functions by Performing a few calculations and Printing the results.

Algorithm:

1. create math functions.py module:
2. create area functions.py module:
3. create main.py:
4. Print the output as expected.

Output:

Addition=15
subtraction=5
multiplication=50
Division=2.0
circle area = 153.93804002589985
rectangle area = 50
Triangle area = 24.0

VEL TECH - CSE	
EX NO.	3
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	25
TOTAL (20)	
SIGN WITH DATE	25/8/25

Result:

Thus the Program for Importing Python modules and Packages was successfully executed and the output was verified.