Date:-20/8/25 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 userdefined 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 directions, 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 [2, 6] (inclusive), π, ceil (2.3), floor(2.3), factorial(5), gcd(5,15), abs(-10), power(3,5), log base 3 of 2, log 10(a) for a=100, and check NaN/Infini
3) OS & sys:
   ⇒ create C:\pythonlab if not present and print the current working directory.
   ⇒ create C:\Pythonslots24 if not present and change the current working directory to it.

⇒ List all fiels/directions 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 & RANDAM —") print("sqrt(5)=",
math.sqrt(5)) print ("radians(30) =", math.radians
                                    (30))
print("random() in [0,1)=", random.random())
print("random(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))
```

Expected sample output :-

— MATH & RANDOM —
sqrt(5) = 2.23606797749979 radians(30) =
0.5235987755982988 random() in (0,1) =
0.37448871756146646 <- will vary randint(2,10)
<- inclusive ; will vary
PI = 3.141592653589793 ceil(2.3) = 3
factorial(5) = 120 gcd(5,15) = 5
abs(-10) = 10 pow(3,5) = 243 log base 3 of 2 =
0.6309297535714574
log 10(100) = 2.0 isinf(∞) = True,
is nan(NAN) = True
— OS & SYS —
created/ensured : C:\Pythonlab
current working directory : C:\ -- (your current Path)
created/ensured & changed into : C:\Python-slot SQL4
Directory contents of C:\Pythonslot SQL4 : []
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("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:\pythonslot-sql4")
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"meadian({data1})=", stats.
median(data1)) print(f"mode({data})=",
```

```
stats.mode(data2))
print(f"stdev({data2}) =", stats.stdev(data2))
```

Result:- Thus, the python program implement and modules packages are executed successfully.

output :

RESTART:

C:\useas\student.MAT2VC6833\AppData\Local\Programs\Python311\Lib\site-packages\caadpack\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 Aim:- create a python package named cardpack containing a module cardfun that imports the random module. Assign of cards, call a function from the module, and display a random sample of cards.

Algorithm:-

Step1:- start
Step2:- To create a package cardpack
Step3:- To create a module cardfun and import random function.
Step4:- Assign a card range.
Step5:- call a module function.
Step6:- Display the random sample cards
Step7:- 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()
```

Result:- Thus, the python program of package and card pack module is executed successfully.

33 Aim:- you are tasked with developing a modular calculator application in python. The calculator should support basic arithmetic operations: addition, substraction, 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, substraction, multiplication and division.
2. Handle division by zero by raising an error if the divisor is zero.
3. Import the module 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.

program

(my)math

def add(a,b): return a+b def substract(a,b): return a-b def

output:-

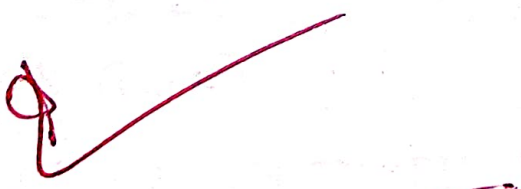
Addition :- 15

subtraction : 5

multiplication : 50

DIVISION : 2.0

```
multiply (a,b):
return a*b def divide (a,b): If b==0;
raise valueError ("cannot divide by zero")
return a/b
import mymath a=10 b=5
Point ("Addition:", my math.add (a,b))
Print("subtraction:", my math. subtract (a,b))
Print ("multiplication:" my math. multiply (a,b))
Print ("Division:", mymath.divide (a,b))
```



Result:- Thus, the python program is arithamatic operators are executed successfully

3.4 Aim:- you are working on a python 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 mypackage which includes sub packages pack1 and pack2 with two modules mathfunctions and areafunctions demonstrate the use of the functions by performing a few calculations and printing the results.

Algorithm:-

1. Create mathfunction.py module:
2. Create areafunction.py module:
3. Create main.py:
4. Print the output as expected.

Program:-

1. Create the mathfunction.py module

```
def add(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):
```

Output :-

Addition : 15
subtraction : 5
multiplication :- 50
Division : 2.0
Circle Area (radius=7) : 153.93804002589985
Rectangle Area (5x10) = 50
Triangle Area (base=6, height=8) :- 24.0

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 mathfunctions
import areafunctions
# using math functions
print("Addition:", mathfunctions.add(10,5)) print
("subtraction:", mathfunctions.subtract(10,5))
print("Multiplication:", mathfunctions.multiply
(10,5))
print("Division:", mathfunctions.divide(10,5))
# using area functions.
print("circle Area (radius=7):", areafunctions.cir
—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))
```

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

Result:- Thus, the program for importing py modules and packages was sucessfully execu and output was verified.