20/8/25

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

Aim:-

To implement and demonstrate importing built in modules creating user defined modules and organizing code into Packages in Python reusability, modularity.

3.1

1. Perform common math and random operations.
2. work with operating system and read the Python version.
3. compute basic statistics.

Algorithm:-

1. Import required modules: math, random, os, statistics.
2. math & random:
   → compute sqrt(5), radians(30), a random float in [0.0, 1.0), a random [2,6], π, ceil(2.3), floor(2.8), factorial(5).
3. os & sys:
   → create c:\Python lab if not Present and Print the current working directory.
   → List all files / directories in new current directory.
   → Print Python interPreter version.
4. statistics:
   on lists: [5,6,8,10] and [25,32,8,32,46,28,6] compute
5. Print neatly formatted results.

output:-

```
--- math & random ---
sqrt(5) = 2.23606797749979
radians(30) = 0.5235987755982988
random() in (0,1) = 0.37446887175646 (will vary)
Pi = 3.141592653589793
ceil(2.3) = 3.
floor(2.3) = 2.
factorial(5) = 120
gcd(5,15) = 5
abs(-10) = 10
pow(3,5) = 243.
log base 3 of 2 = 0.6309297535714574
log10(100) = 2.0
isinf(∞) = true  isnan(nan) = true
--- os & sys ---
```

Statistics.

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

median([5,6,8,10]) = 7.0

mode([2,2,3,3,4,2,5]) = 2.

stdev([2,3,2]) = 2.27156.

Program:-

```
import math
import random
import os.
import sys
import statistics as stats
from pathlib import Path.
Print ("In -- 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)) #inclu
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 30f 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 (
```

```
={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:\Pythonslots24")
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))
```

Output:-

Restart:

C:\Users\student.MAT2VC-6833\AppData\Local\Programs\Python\Python37-32\lib\site-packages

{5, 24, 13, 22, 20, 41, 48, 38, 51, 4, 7, 34, 44, 16, 50, 31, 40, 15, 35, 17, 18, 33, 39, 36, 42, 12, 6, 1, 9, 46, 29, 2, 27}

3.2

create a Python Package named cardPack containing a module cardfun that imports the random module

Algorithm:-

step 1: start.
step 2: To create a Package cardPack.
step 3: To create random function.
step 4: Assign a cards range.
step 5: call a module function.
step 6: Display the random sample cards.
step 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")
```

mymod.py.

```
import cardFun
card Fun.func()
```

3.3

You are tasked with developing a modular calculator application in Python. The calculator should support basic arithmetic operations: addition, subtraction.

Algorithm:-

1. Define functions for addition, subtraction, multiplication
2. Handle division by zero by raising an error if the divisor is zero.
3. Initialize two numbers (a=10, b=5)
4. Call each function using mymath.<function_name>(a,b)
5. 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
```

Output:

Addition: 15

subtraction: 5

multiplication: 50.

Division: 2.0.

```
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:-

Addition: 15

Subtraction: 5

multiplication: 50

division: 2.0.

Circle Area (radius=7): 153.9380

Rectangle Area(5x10): 50.

Triangle Area(base: 6 height=8): 24.0.

3.4

You are working on a Python Project that requires you to Perform various mathematical operations and geometric area calculations.

Algorithm:-

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

Program:-

1. create the mathfunctions.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 areafunctions.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:", mathfunctions.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 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)	15
SIGN WITH DATE	1/3/24

Result:

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