

TASK-3 :- importing and creating python modules and packages in python program
2018/25

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.

Algorithm:-

1) import required modules : math, random, os, sys, statistics, pathlib.

2) math & random:

- compute square(5), radians(80), a random float in [0.0, 1.0], a random integer in [2, 6], π , ceil(2.3), floor(2.3), factorial(5), gcd(5, 15) abs(-10), pow(3.5), log base 3 of 8,

3) os & sys :

- Create C:\PythonLab if not present and print the current working directory

- Create C:\Python\ if not present and change the current working

- List all files 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, 8, 5, 6] , compute mean, median,

expected sample output :-

-- MATH & RANDOM --

$$\text{sqrt}(5) = 2.23606797749979$$

$$\text{radians}(30) = 0.5235987755982988$$

random() in $[0, 1] = 0.3744488717564664$

$$\text{radian}(2/6) = \pi$$

$$\pi = 3.141592653589793$$

$$\text{ceil}(2.3) = 3$$

$$\text{floor}(2.3) = 2$$

$$\text{factorial}(5) = 120$$

$$\text{gcd}(5, 15) = 5$$

$$\text{abs}(-10) = 10$$

$$\text{pow}(3, 5) = 243$$

$$\log_{\text{base } 3.0} 2 = 0.630929753571956$$

$$\log_{10}(100) = 2.0$$

$$\text{isinf}(\infty) = \text{True}, \text{isnan}(\text{NaN}) = \text{True}$$

-- OS & SYS --

created / ensured : C:\Python1ab

current working directory : C:\YourComputerName

5) Print neatly formatted results.

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.radians(30))
print("radians(30) =", math.radians(30))
print("random() = ", random.random())
print("randint(2,6) = ", random.randint(2,6))
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("log base 3 OF 2 = ", math.log(2,3))
a_val = 100
```

created/ensured & charged into: C:\pythonSLOTs2L4

directory contents of C:\pythonSLOTs2L4:

Python version: 3.7.3 (-- 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

stddev([2, 5, 3, 2, 8, 3, 9, 4, 2, 5, 6]) = 2.271563338320109

```
Print(F"LOG10({a_val}) = ", math.log10(a_val))  
inf_val = float('inf')  
nan_val = float('nan')  
Print(F" is inf(∞) = {math.isinf(inf_val)}\n(NAN) = {math.isnan(nan_val)}")
```

```
Print("Ih---os sys :-")  
path_pythonlab = path(r'c:\pythonlab')  
path_pythonlab.mkdir(parents=True, exist_ok=True)
```

```
Print(F"Created/ensured: {path_pythonlab}\nCurrent working directory:", os.getcwd())  
os.chdir(target_dir)  
Print(F"changed into : {target_dir}")  
Print("directory contents:", os.listdir())  
Print("Python version:", sys.version)
```

```
Print("in -> STATISTICS -")  
data1 = [5, 6, 8, 9, 0]  
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"std dev({data2}) = ", stats.stdev(data2))

Q.2 :- Create a python package named cardPack containing a module cardfun that imports the random module
Aim :- To 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 a module card fun and import random function

Step 4 :- Assign a cards range

Step 5 :- Call a module function

Step 6 :- Display the random sample card

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=5)

def gcd(a, b):

if b == 0:

return a

else:

return gcd(b, a % b)

print("GCD of 10 and 20 is", gcd(10, 20))

print("GCD of 15 and 25 is", gcd(15, 25))

print("GCD of 30 and 45 is", gcd(30, 45))

OUTPUT:-

RESTART:

```
C:\Users\student.MAT\RC68331\APP Data\Local\Programs\Python\Python 3.11\lib\site-packages\cardpack\MyMod.py
```

```
[5, 24, 13, 22, 20, 41, 38, 51, 4, 7, 34, 89, 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]
```

```
print("In\n", shuffled_cards, "\nIn")
```

mymod.py

```
import cardfun  
cardfun.funC()
```



Result:- Thus, the program python package
~~has been verified and executed successfully~~

3.3 You are tasked with developing a modular calculator application in Python

Aim :- You are tasked with developing a modular calculator applications in Python.

Algorithm :-

- 1) Define functions for addition, sub, mult, division
- 2) handle division by zero by raising an error if division 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 for all operations

Program : (my)math :-

```
def add(a,b):  
    return a+b
```

```
def subtract(a,b):  
    return a-b
```

```
def multiply(a,b):  
    return a*b
```

Digitized by srujanika@gmail.com

Digitized by srujanika@gmail.com

Digitized by srujanika@gmail.com

Output:-

Addition:

Subtraction:

Multiplication:

Division:

```
def devide(a,b):  
    if b==0:  
        raise ValueError("cannot divide by zero")  
    return a/b  
  
import mymath  
a=10  
b=5  
print("Addition:", mymath.add(a,b))  
print("Subtraction:", mymath.sub(a,b))  
print("Multiplication:", mymath.multiply(a,b))  
print("Division:", mymath.divide(a,b))
```

Result:- Thus, the program's development
development had been executed and
verified successfully

8.4 You are working on Python project that requires you to perform various mathematical operations

Aim :- You are working on a Python project that requires you to perform various mathematical operations and geometric area calculation.

Algorithm :-

- 1) Create mathfunctions.py module.
- 2) Create areafunctions.py module.
- 3) Create main.py.
- 4) Print the output as expect.

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.
```

Operations

Addition: 15

Subtraction: 5

Multiplication: 50

Division: 20

Circle Area $\pi r^2 = \pi \times 7^2 = 153.93804002589985$

Rectangle Area ($a \times b$): 50

Triangle Area (base=6, height=8) = 24.0

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 mathfunctions
import areafunctions
# Using math functions
print("Addition:", mathfunctions.add(10, 5))
print("Subtraction:", mathfunctions.sub(10, 5))
print("Multiplication:", mathfunctions.mult(10, 5))
print("Division:", mathfunctions.divide(10, 5))

# Using area functions
print("circle Area (radius=7):", areafunctions.circle_area(7))
print("rectangle Area (5x10):", areafunctions.rectangle_area(5, 10))
print("triangle Area (base=6, height=8):", areafunctions.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 | 15 |

✓

Result:

:- Thus, the program for importing python modules and packages was successfully executed and the output was verified.