

## 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.

3.1

- 1) Perform common math and random operation
- 2) Work with the operating system and read the Python version

Algorithm:

- 1) import required modules: math, random, os, sys, statistics
- 2) math & random

Compute  $\sqrt{5}$  radians a random float in  $[0, 0, 1, 0]$ , a random integer in  $[2, 6]$ ,  $\pi$ ,  $\text{ceil}(2/3)$ ,  $\text{floor}(2/3)$ , factorial(5),  $\log_{10}(2)$  for  $a=100$ , and check NaN/Infinity.

3) OS & sys:

- Create C:\Python\lab if not present and point the current working directory
- Create c:\Python\lab if not present and change the current working directory to it.
- list all files/directories in the new current directory

# Program

import math

import random

import os

import sys

import statistics as stats

from pathlib import Path

Point("In - MATH & RANDOM -")

Point ("sqrt(5) = ", math.sqrt(5))

Point ("radians(30) = ", math.radians(30))

Point ("random() in [0,1] = ", random.random())

Point ("randint(2,6) = ", random.randint(2,6)) # inclusive

Point ("Pi = ", math.pi)

Point ("ceil(2.3) = ", math.ceil(2.3))

Point ("floor(2.3) = ", math.floor(2.3))

Point ("factorial(5) = ", math.factorial(5))

Point ("gcd(5,15) = ", math.gcd(5,15))

Point ("abs(-10) = ", abs(-10))

Point ("pow(3,5) = ", pow(3,5))

Point ("log base 3 of 2 = ", math.log(2,3))

a\_val = 100

Point ("log base 3 of 2 = ", math.log(2,3))

Point ("log10({a\_val}) = ", math.log(a\_val))

inf\_val = float('inf')

nan\_val = float('nan')

Point ("isinf(60) = {math.isinf(inf\_val)}, isnan(Nan) =

{math.isnan(nan\_val)}")

## Expected sample output

--- MATH & RANDOM

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

$$\text{random}(50) = 0.5235987755982988$$

$$\text{random}(7 \text{ in } [0,1]) = 0.37444887175646646$$

$$\text{randint}(2, 6) = 6$$

$$\pi = 3.141592653589793$$

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

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

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

$$\gcd(5, 15) = 5$$

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

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

$$\log_{\text{base } 3} \text{of } 2 = 0.6309297535714574$$

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

$$\text{isinf}(0) = \text{True}$$

$$\text{isinf}(NaN) = \text{True}$$

$$\text{isinf}(Inf) = \text{True}$$

$$\text{isinf}(-Inf) = \text{True}$$

$$\text{isinf}(NaN) = \text{False}$$

$$\text{isinf}(Inf) = \text{False}$$

$$\text{isinf}(-Inf) = \text{False}$$

$$\text{isinf}(NaN) = \text{False}$$

$$\text{isinf}(Inf) = \text{False}$$

$$\text{isinf}(-Inf) = \text{False}$$

Point ("In--OS & Sys--")

Path\_PythonLab = Path("C:/PythonLab")

Path\_PythonLab.mkdir(parents=True, exist\_ok=True)

Point(f"Created/ensured : {Path\_PythonLab}")

Point("Current working directory:", os.getcwd())

target\_dir = Path("C:/PythonLab")

target\_dir.mkdir(parents=True, exist\_ok=True)

os.chdir(target\_dir)

Point(f"Changed into : {target\_dir}")

Point("Directory contents:", os.listdir())

Point("Python Version:", sys.version)

Point("In--STATISTICS--")

data1 = [5, 6, 8, 10]

data2 = [2, 5, 3, 2, 8, 3, 9, 4, 2, 5, 6]

Point(f"mean ({data1}) = ", stats.mean(data1))

Point(f"median ({data1}) = ", stats.median(data1))

Point(f"mode ({data2}) = ", stats.mode(data2))

Point(f"stdev ({data2}) = ", stats.stdev(data2))

Output:

RESTART

C:\Users\Students\MAT2VC6833\AppData\Local\Programs\Python\Python  
11\lib\site-packages\casadi\my\_mod.py  
[5, 24, 13, 22, 29, 41, 38, 5), 4, 7, 34, 49, 14, 56, 37, 40, 15, 35, 17, 18, 33, 39, 3  
12, 12, 6,

(gust = glo\_tilde, gust = etilde) zibmr.zib\_tilde  
(zib\_tilde) zib.20

((gust - tildetilde) : otim\_bezier) #tildetilde  
(zibtilde, " : etilde) BezierCurve  
(molex, " : molex\_matt) #matt

(... - CIRCLE -- m) #m

[0, 8, 2, 2] = lotab

[2, 2, 8, 1, P, E, 8, 2, 2, 2] = satob

((lotab) marr.state) = (satob) #m

((satob) mibarr.state) = (satob) #m

((satob) gbarr.state) = (satob) #m

((satob) vobj2.state) = (satob) #m

3.2 Create a Python package named CardPack containing a module CardFun that imports the random module. Assign a range of card call a function from the module and display a random sample of cards.

### Algorithm:

1. Start
2. To Create a Package CardPack
3. To Create a module CardFun and import random Function
4. Assign a card range
5. Call a module function
6. Display the Sample Card
7. Stop

### Program:

```
Card Fun
import random
def func():
    cards = []
    for i in integers(1, 53):
        cards.append(i)
    shuffled_cards = random.sample(cards, k=52)
    print("In In", shuffled_cards, "In In")
```

### Mymod.py

```
import CardFun
CardFun.func()
```

## Output:

Addition: 15

Subtraction: 5

Multiplication: 50

Division: 2.0

## Algorithm:

1. Start



2. To create a middle carriage

3. To create a middle carriage and carry four and square numbers

4. Addify a carry and

5. Call a middle function

6. Divide all the square carriage

7. Stop

## Procedure:

Carry four

square + square

def func:

Carry = [ ]

for i in range(100):

Carry.append(i)

Bubble sort = square \* square \* square (Carry, 15, 2)

Sort((100, "Bubble sort", "square", "square"))

3.3

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

### Algorithm:

1. Define function for addition, subtraction, multiplication and division
2. Handle division by zero by raising an error if the divisor is zero.
3. import the module contain these function
4. initialize two numbers ( $a=10, b=5$ )
5. Print the result of all operation.

### 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 ValueError ("cannot divided 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))
```

3.4

You are working on a Python Project that requires you to perform various mathematical operation and geometric area calculation. To organize your code better, you decide to create a package named my Package which includes a sub package Pack1 and Pack2 with two module:

### Algorithm:

1. Create math function py module.
2. Create area function 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
```

Output:

Addition: 15

Subtraction: 5

Multiplication: 50

Division: 2

Circle area (radius=7): 153.93804002589985

Rectangle Area (5x10): 50

Triangle area (base=6, height=8): 24.0

C

but there are two ways to do it.

Program

3 Create the main.py file

import math functions

import area functions

# Using math functions

```
Point("Addition", mathfunctions.add(10,5))
```

```
Point("Subtraction", mathfunctions.subtract(10,5))
```

```
Point("Multiplication", mathfunctions.multiply(10,5))
```

```
Point("Division", mathfunctions.divide(10,5))
```

# Using area functions

```
Point("Circle Area (radius=7)", areafunctions.circle_area(7))
```

```
Point("Rectangle Area (5x10)", areafunctions.rectangle_area(5,10))
```

```
Point("Triangle Area(base = 6, height = 8)", areafunctions.triangle_area(6,8))
```

Result:

thus the program for importing Python modules and packages was successfully executed and the output was verified.

VEL TECH	
EX NO.	3
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	15
GIVEN WITH DATE	