

Task-3: Importing and creating python modules and packages in python program.

Aim: To implement and demonstrate the process of import built-in modules, creating user defined modules, organizing code into packages in python, thereby promoting code reusability, modularity & maintainability

3.1

1. perform common math and random operations
2. Work with operating system
3. compute basic statistics.

Algorithm:

- 1) Input required modules : math , random , os , sys.
- 2) Math & random :
  - compute  $\sqrt{5}$  , radians (30) , a random float in [0.0,1.0),
  - a random integer in [2,6]
- 3) os & sys :
  - create C:\pythonlab if not present print working directory
  - create C:\pythonlab\slot S2L4
  - print python interpreter version .
- 4) Statistics :
  - on list : [5,6,8,10] & {2,5,3,2,8,3,9,4,2,5,6} .
- 5) print neatly formatted results .

Program:

```
import math  
import random  
import os  
import sys  
import statistics as stats  
from pathlib import Path
```

output:

--- MATH & RANDOM ---

$$\sqrt{2} \approx 1.414213562373095050110872367944115226133834794482822218$$

$$\sin(30) \approx 0.5000000000000000$$

$$\text{random}() \text{ in } [0,1] \approx 0.37448871956 \leftarrow \text{well (very)} \quad \text{random}$$

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

$$\pi \approx 3.141592653589793$$

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

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

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

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

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

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

$$\log \text{ base } 3 \text{ of } 2 \approx 0.6309297535714529$$

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

$$\text{isint}(s) = \text{True}$$

--- OS & SYS ---

created /newred : c:\python\lab

went working directory : c:\l

Created /newred & changed into : c:\py\l

Python Version : 3.x.x.1

--- STATISTICS ---

$$\text{mean}([5, 6, 8, 10]) = 7.25$$

$$\text{median}([5, 6, 8, 10, 3]) = 7.0$$

$$\text{mode}([2, 5, 3, 2, 8, 3, 9, 4, 2, 5, 6]) = 2$$

$$\text{stddev}([2, 5, 3, 2, 8, 3, 9, 4, 2, 5, 6]) = 2.2700000000000002$$

```
print("In -- Math & Random -- ")
print("sqrt(5) = ", math.sqrt(5))
print("random(30) = ", math.random(30))
print("random() in (0,1) = ", random.random())
print("randint(2,6) = ", random.randint(2,6))
# include 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"isnan({inf_val}) = {math.isnan(inf_val)} ; isfinite({nan_val}) = {math.isfinite(nan_val)}")
print("In - OS & Sys -- ")
path_pythonlab = Path(r"C:\Pythonlab")
print(f"created / ensured : {path_pythonlab}")
print("current working directory : " + os.getcwd())
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("In - Statistics -- ")
del data1 = [5, 6, 8, 10]
del data2 = [2, 5, 3, 2, 8, 3, 9, 4, 2, 5, 6]
mean = data1.mean()
print(f"mean of data1 = {mean}")
```

3.2. Create a python package named cardpack containing a module cardfun that exports the random module.  
Assign a range of cards, call a function from the module and display a random sample of cards

Algorithm:

- 1, Step 1: Start
- 2, To create a package cardpack
- 3, To create a module cardfun and import random functions
- 4, Assign a cards range.
- 5, call a module function.
- 6, Display the random sample cards
- 7, Stop

Program:

```
cardFun  
import random  
def fun1():  
    cards = []  
    for i in range(1, 53):  
        cards.append(i)  
    shuffled_cards = random.sample(cards, k=52)  
    print("n", shuffled_cards, "n")
```

MyMod.py

```
import cardFun  
cardFun.fun1()
```

Output:

Restart:

c:\users\student\MAT286833\appdata\local  
programs\python\python 3.11\lib\site-packages  
cardpack\MyMod.py

[5, 24, 13, 22, 20, 41, 38, 51, 4, 7, 34, 49, 14, 50, 37,  
40, 11, 31, 46, 28, 21, 82, 8, 25, 30, 23, 26, 10, 43,  
47, 3, 44, 52, 1, 45, 9].

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 .

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 containing these functions .
4. Initialize two numbers ( $a=10, b=5$ ) .
5. Call each function using mymath .
6. print the results of the 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  
  
def divide(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.subtract(a,b))  
print ("Multiplication : ", mymath.multiply(a,b))  
print ("Division : ", mymath.divide(a,b))

output:  
addition: 15  
subtraction: 5  
multiplication: 50  
division: 20

3.4. You are working on a python project that require you to perform various mathematical operations & 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 function by performing a few calculations and printing the results.

Algorithm:

1. Create mathfunctions .py module:
2. Create areafunctions .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 module

```
import math  
def circle_area(radius):  
    return math.pi * radius * radius
```



def rectangle - area (length , width) ;

returns length \* width

def triangle - area (base , height) :

returns 0.5 \* base \* height

3. create the main .py file

import mathfunctions.

import areafunctions

# using mathfunctions

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.

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)	
SIGNATURE DATE	15

*(Signature)*  
Result : Thus, the program for importing python modules & packages was successfully executed & the output was verified.