

### Task-3

Importing and creating Python modules and package 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 coding reusability, modularity and maintainability.

3.1

1. Perform common math and random operations.
2. work with the operating system.
3. computer basic statistics (mean, median, mode standard deviation)

#### Algorithm:

1. Import required modules, math, random as sys, statistics, Pathlib:  
compute  $\sin(5)$ ,  $\text{radians}(30)$  a random float in  $[0,0,1]$ ,  
a random integer in  $(2,6)$  (inclusive)  $\lceil \text{ceil}(2.3) \rceil$ ,  $\lfloor \text{floor}(2.3) \rfloor$ , factorial(5), gcd(5,5), abs(-10), pow(3,5), log base 3 of 2.  
~~log 10(a)~~ for  $a=100$  check  $\text{NaN}$  ( $\infty$ ).
2. Math & Random:

Expected sample output:

- MATH & RANDOM -

sqrt(s) = 223606.799749979 random(30).

0.5235987755982988 random(1) in (0,1)

0.3744887175646646 ← will vary depending (2,6) =

← inclusive will vary

$\pi = 3.141592653589793$  cell (2,3) = 3

floor (2,3) = 2 factors of (5) = 120 gcd (8,15) = 5

abs(-10) = 10 Pow (3,5) = 243 log base 3 of 22

0.63092975357147574

$\log_{10}(100) = 2.0$  isinf(∞) = True

is nan (None) = False

is -os & sys -

Create (ensured :c:\Python\lab

current working directory :c:\ - (your current path).

ensured & changed into :c:\Python  
slot SLC4

Directory contents of c:\Python\slot SLC4:[ ]

Python version :3x.x (- details)

SATISTICS-mean [5, 6, 8, 10] = 7.25

### 3. os & sys.

- create c:\Pythonlab if not present and print the current working directory.
- create c:\Pythonlab\src if not present and change the current working directory to it.
- list all files / directories 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 medium mode stdDev .

### 5. Print neatly formatted results.

#### Program

```
import math  
import random  
import os  
import sys  
import statistics as sttats  
from Pathlib import Path  
  
print("In.. MATH & RANDOM--")  
print("sqrt(5) = ", math.sqrt(5))  
print("radians(30) = " + str(math.radians(30)))  
print("random() in (0,1) = " + str(random.random()))  
print("random(2,6) = " + str(random.randint(2,6)) + "# include.
```

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

mode  $((2, 5, 3, 2, 8, 3, 9, 4, 2, 5, 6]) = 2$

stden  $((2, 5, 3, 2, 8, 3, 9, 4, 2, 5, 6]) =$

2.271563338201693.

Printf ("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" log 10 ({a-val}) = " math.log 10 (a-val))

inf-val = float ('inf')

nan-val = float ('nan')

Print (f" isinf (inf) = {math.isinf (inf-val)} {isinf (NaN)} = {math.isnan (nan-val)}")

Print f"\n ... os & sys ...")

Path - Pythonlab = Path (r"c:\Pythonlab")

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

Printf (f" created : {Path - Pythonlab}")

Printf ("current working directory : " os.getcwd ())

target - dir = Path (r"c:\Pythonlab\src")

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

os.chdir (target - dir)

Printf ("changed into : {target - dir}")

Print ("Directory contents : " os.listdir ())

Print ("Python version : " sys.version).

```
Print("In... STATISTICS...")  
data1=[5,6,8,10]  
data2=[2,5,3,2,8,3,9,4,2,5,6]  
Print("mean ({data1}) = " stats.mean (data1))  
Print("median ({data1}) = " stats.median (data1))  
Print("mode ({data2}) = " stats.mode (data2))  
Print("stdev ({data2}) = " stats.stdev (data2))
```

Result: To implement and demonstrate the process of Porting built in modules is verified successfully.

3.2

Create a Python package named cardpack containing a module cardfun that imports the random module. Assign of cards function from module and display a random sample of cards.

Aim: To create package named cardpack containing a module cardfun that imports the random module assign of cards function from module

Algorithm:

1. Start
2. To create a package cardpack
3. To create a module cardfun and import random function
4. Assign a cards range
5. Call a module function
6. Display the random sample 8 cards.
7. Stop

Program

card fun  
import random  
~~def func():~~  
~~cards = []~~

output

RESTART

c:\users\student.MA72VC6833(APP Data\local  
Programs\Python 3.1\lib\site-packages\  
comback\mymod.py.

(5, 24, 13, 22, 10, 41, 38, 51, 4, 7, 34, 49, 14, 50, 37)  
40, 15, 35, 17, 18, 33, 39, 36, 42, 12, 6, 9, 46, 48  
29, 2, 27, 11, 31, 46, 28, 21, 32, 8, 15, 30, 23, 26,  
10, 43, 47, 44, 3, 52, 1, 4, 5, 9).

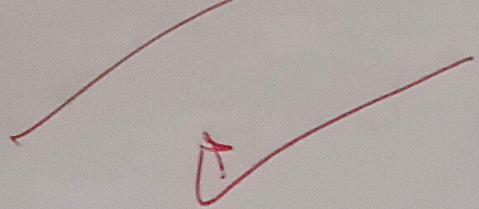
The command has function of ~~return~~  
and return a list of values to print

```
for i in range(1,53):
    cards.append(i)
shuffled_cards = random.sample(cards, k=52)
print("In", shuffled_cards "In")
```

my module

```
import cardfun
cardfun.fun()
```

Result: To create a Python package named  
cardpack containing a module cardfun that  
imports & verified successfully



33

you are tasked with developing a modular calculator application in Python the calculator should support basic arithmetic operations addition subtraction multiplication and division each operation should be implemented in a separate module additionally you should create main program to handle user input call the appropriate module and display the result

Aim: To create a main programme to handle user input call appropriate module and display results.

Algorithm:

1. Define functions for addition subtraction multiplication and division.
2. Handle division by zero by raising an error if division is zero
3. Import the module (mymath) containing these function
4. Initialize two number ( $a = 10, b = 5$ )
5. Call each function using mymath <function-name>(a)
6. Print the result of all operations

outPut

Addition : 15

Subtraction : 5

multiplication : 50

Division : 20.

↳ more space will be shown on the  
left side for addition or subtraction  
operations between numbers

### 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 divide by zero")  
    return a/b
```

```
import math
```

```
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)).
```

Result: Thus to create a main Program to handle user input call appropriate module

B verified successfully

34

You are working on a Python Project that require you to perform various mathematical operations and geometric area calculation To organize your code better you decide to two create a package named mypackage which include sub package pack-1 and pack-2 with to numbers math function and area function Demonstrate the use of the function by performing a few calculation and printing the result.

Aim: To working on a Python Project that required you to perform various mathematical operation and geometric to two create a package.

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 mathfunctions.py module.

def add(a,b):

    return a+b

def subtract(a,b)

    return a-b

outline

Addition : 15

Subtraction : 5

multiplication : 50

Division : 1.0

circle area (radius : 7) : 153. 93804002589985

Rectangle area ( $5 \times 10$ ) = 50

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

```
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):
```

```
    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:", math.functions.add(10,5))
```

```
print("Subtraction:", math.functions.subtract(10,5))
```

```
print("Multiplication:", math.function.multiply(10,5))
```

```
print("Division:", math.function.division(10,5))
```

~~# include using area function~~

~~printf("circle Area (radius:=7):", area.function.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 (3)	3
VIVA VOCE (3)	3
RECORD (4)	4
TOTAL (15)	15
SIGN WITH DATE	✓

Result: thus the Program for importing Python modules and package was successfully executed and the output was verified.

