

ASSIGNMENT 3 : (202202519010054 MILAN PORIA S.)

'''1. Create a function calculate_area (length, width=5) that returns the area of a rectangle. If the width is not provided, it should default to 5.'''

```
def calculate_area(length,width=5):  
    return length * width  
  
print(f"Area of Rectangle with width: {calculate_area(10,10)}")  
print(f"Area of Rectangle with default width value : {calculate_area(10,5)}")
```

'''2. Write a function sum_all(*numbers) that takes any number of numerical arguments and returns their sum.

```
'''  
def sum_all(*numbers):  
    return sum(numbers)  
  
print(f"Sum : {sum_all(1,2,3,4)}")
```

''' 3. Write a function sum_all(*numbers) that takes any number of numerical arguments and returns their sum'''

```
def sum_all(*numbers):  
    return sum(numbers)  
  
print(f"Sum : {sum_all(1,2,3,4)}")  
'''
```

4. Create a module

operations.py with functions for arithmetic operations (add, subtract, multiply, divide). Import this module in another script and use its functions.

Take numbers

as user inputs and pass in module functions

```
'''
```

```
import operations
```

```
num1 = int(input('Enter a num1 : '))
```

```
num2 = int(input('Enter a num2 : '))
```

```
print(f"Add : {operations.add(num1,num2)} ")
```

```
print(f"Subtract : {operations.subtract(num1,num2)} ")
```

```
print(f"Multiply : {operations.multiply(num1,num2)} ")
```

```
print(f"Divide : {operations.divide(num1,num2)} ")
```

''' 5. Write a lambda function that takes two arguments and returns their product. Use this lambda function with the map() function to multiply elements of two lists together.'''

```
product = lambda x, y: x * y
```

```
list1 = [1, 2, 3, 4]
```

```
list2 = [5, 6, 7, 8]
```

```
result = list(map(product, list1, list2))
```

```
print(result)
```

```
'''
```

6. Write a Python function sum_and_product(numbers) that takes a list of numbers as input and returns a tuple containing two values: The sum of all the numbers in the list. The

product of all the numbers in the list.

```
'''
```

```
def sum_and_products(numbers):
```

```
    totalSum = sum(numbers)
```

```
    product = 1
```

```
    for num in numbers:
```

```
        product = product * num
```

```
    return (totalSum, product)
```

```
list = [1, 2, 3, 4]
```

```
print(f"The sum and product of the list {list} :- {sum_and_products(list)}")
```

```
'''
```

7. Write a function find_max(*args) that accepts any number of positional arguments and

returns the maximum value. Use argument unpacking to test your function with a list of numbers.

```
'''
```

```
def find_max(*args):
```

```
    return max(args)
```

```
list = [1,2,34,99]
print(f"Max number from {list} : {find_max(*list)}")
'''
```

8. Write a function `outer_function(text)` that defines a nested function `inner_function()` which prints the reversed version of `text`. The `outer_function` should call `inner_function` and return its result.

```
'''
```

```
def outer_function(text):
    def inner_function():
        return text[::-1]
    return inner_function()

print(outer_function('Hello'))
```

''' 9. Write a function `apply_operation(a, b, operation)` where `operation` is a function (like `add`, `subtract`, etc.). The `apply_operation` function should return the result of applying operation to `a` and `b`.'''

```
def apply_operation(a, b, operation):
    return operation(a, b)

print(apply_operation(5, 3, lambda x, y: x + y))
print(apply_operation(5, 3, lambda x, y: x * y))
```

```
def add(num1,num2):
    return num1+num2

def subtract(num1,num2):
    return num1-num2

def multiply(num1,num2):
    return num1*num2

def divide(num1,num2):
    return num1/num2

'''
```

10 Write a function `calculate_discount(price, discount=0.10, *, tax_rate)` that calculates the

final price after applying a discount and a tax. The tax rate is a required keyword argument.

```
'''
```

```
def calculate_discount(price, discount=0.10, *, tax_rate):  
    discounted_price = price * (1 - discount)  
    final_price = discounted_price * (1 + tax_rate)  
    return final_price
```

```
print(calculate_discount(100, tax_rate=0.07))
```

'''11. Write a Python script that uses the random package to simulate the rolling of two six-sided dice'''

```
import random
```

```
def roll_dice():  
    die1 = random.randint(1, 6)  
    die2 = random.randint(1, 6)  
    return die1, die2
```

```
dice1, dice2 = roll_dice()  
print(f"Die 1: {dice1}, Die 2: {dice2}, Total: {dice1 + dice2}")
```

''' 12. Write a program that generate two random numbers. If the total of the two dices is 7 or 11, print "You win!". Otherwise, print "Try again!".'''

```
import random
```

```
def roll_dice():  
    die1 = random.randint(1, 6)  
    die2 = random.randint(1, 6)  
    return die1 + die2
```

```
total = roll_dice()  
if total == 7 or total == 11:  
    print("You win!")  
else:  
    print("Try again!")
```

'''

13 Write a program for following. Create a python package "Shapes".
Create modules:

circle.py: This module should have a function area(radius) that calculates and returns the area of a circle given its radius.

rectangle.py: This module should have a function area(length, width) that calculates and

returns the area of a rectangle given its length and width.

Once the package is created, write a script that imports these modules and calculates:

The area of a circle with radius entered by user

The area of a rectangle with length and width entered by user

'''

Shapes/

circle.py

def area(radius):

return 3.14 * radius * radius

```
rectangle.py
```

```
def area(length, width):
```

```
    return length * width
```

```
def area(length, width):
```

```
return length * width
```

```
from Shapes import circle, rectangle
```

```
radius = float(input("Enter the radius of the circle: "))
```

```
length = float(input("Enter the length of the rectangle: "))
```

```
width = float(input("Enter the width of the rectangle: "))
```

```
circle_area = circle.area(radius)
```

```
rectangle_area = rectangle.area(length, width)
```

```
print(f"Area of the circle: {circle_area}")
```

```
print(f"Area of the rectangle: {rectangle_area}")
```

'''

14 . Create a package named geometry with two sub-packages:

- 2d_shapes: This sub-package should contain modules for calculating the perimeter of 2D

shapes. It should have:

-

circle.py: Function perimeter(radius) to calculate the perimeter of a circle.

-

square.py: Function perimeter(side) to calculate the perimeter of a square.

- 3d_shapes: This sub-package should contain modules for calculating the surface area of 3D

shapes. It should have:

-

sphere.py: Function surface_area(radius) to calculate the surface area of a sphere.

-

cube.py: Function surface_area(side) to calculate the surface area of a cube.

Write a script that imports these functions and calculates:

1. The perimeter of a circle with radius 4.

2. The surface area of a cube with side length 3

'''

geometry/
shapes_2d/

circle.py square.py

import math

def perimeter(radius):

return 2 * math.pi * radius

def perimeter(side):

return 4 * side

shapes_3d/

sphere.py cube.py

```
def surface_area(side):  
    return 6 * side**2  
  
import math  
  
def surface_area(radius):  
    return 4 * math.pi * radius**2
```

main.py

```
from geometry.shapes_2d.circle import perimeter as circle_perimeter  
from geometry.shapes_3d.cube import surface_area as cube_surface_area  
  
print(f"Perimeter of the circle: {circle_perimeter(4)}")  
print(f"Surface area of the cube: {cube_surface_area(3)}")
```