

SNO	DATE	IDX	PROGRAMS	SIGNATURE
5		5.1	Create a Demo function in Python	
		5.2	Create a function with variable length of arguments	
		5.3	Return multiple values from a function	
		5.4	Create a function with a default argument	
		5.5	Create an inner function to calculate the addition	
		5.6	recursive function to calculate the sum of numbers	
		5.7	Assign a different name to function and call it through the new name	
		5.8	Keyword Arguments	
		5.9	Variable-length arguments	

1. Write a program to create a function that takes two arguments, name and age, and print their value

```
def Demo(name,age):
    print(" My Name is : ", name)
    print(" My age is : ",age)
```

```
n=input("Enter the Name : ");
a=int(input("Enter the Age : "))
Demo(n,a)
```

OUTPUT :

```
C:\PPLAB\Week5>py W5_PRG1.py
Enter the Name : Hemalatha
Enter the Age : 50
My Name is : Hemalatha
My age is : 50
```

2. Write a program to create function to accept a variable length of arguments and print the sum and average of their values.

```
def percentage(*subjects):
    sum = 0
    for i in subjects:
        sum = sum + i
    avg = sum / len(subjects)
    print('Sum of ',len(subjects),' values =', sum)
    print('Average =', avg)
```

```
percentage(10,20,30)
percentage(10,20,30,40,50)
```

OUTPUT:

```
C:\PPLAB\Week5>py W5_PRG2.py
Sum of 3 values = 60
Average = 20.0
Sum of 5 values = 150
Average = 30.0
```

3. Write a program to create function calculation() such that it can accept two variables and calculate arithmetic operations and return the results in a single return call.

```
def calculation(a, b):
```

```
    s=a+b
    d=a-b
    m=a*b
    q=a//b
    r=a%b
    p=a**b
    return s,d,m,q,r,p
```

```
a=int(input(" Enter first value : "))
b=int(input(" Enter Second Value : "))
res = calculation(a, b)
print("sum,difference,product,quotient,remainder,power values are ")
print(res)
```

OUTPUT:

```
C:\PPLAB\Week5>py W5_PRG3.py
Enter first value : 10
Enter Second Value : 3
sum,difference,product,quotient,remainder,power values are
(13, 7, 30, 3, 1, 1000)
```

4. Write a program to create a function `show_employee()` using the following conditions.
- It should accept the employee's name and salary and display both.
 - If the salary is missing in the function call then assign default value 9000 to salary

```
def show_employee(name, salary=9000):  
    print("Name:",name,"salary:",salary)
```

```
show_employee("Hema", 12000)  
show_employee("Latha")
```

OUTPUT:

```
C:\PPLAB\Week5>  
C:\PPLAB\Week5>py W5_PRG4.py  
Name: Hema      salary: 12000  
Name: Latha     salary: 9000
```

5. Create an inner function to calculate the addition in the following way
- Create an outer function that will accept two parameters, a and b
 - Create an inner function inside an outer function that will calculate the addition of a and b
 - At last, an outer function will add 5 to addition and return it.

```
def outer_fun(a, b):  
    def addition(a, b):  
        return a + b  
    add = addition(a, b)  
    print("sum=",add)  
    return add + 5
```

```
a,b=map(int, input(" Enter any two numbers : ").split())  
result = outer_fun(a, b)  
print(" sum + 5 = ",result)
```

OUTPUT:

```
C:\PPLAB\Week5>py W5_PRG5.py
Enter any two numbers : 10 5
sum= 15
sum + 5 = 20
```

6. create a recursive function to calculate the sum of first n natural numbers

```
def Sum(num):
    if num:
        return num + Sum(num-1)
    else:
        return 0

n=int(input(" Enter the n value : "))
res = Sum(n)
print(" Sum of first ",n," natural numbers = ",res)
```

OUTPUT :

```
C:\PPLAB\Week5>py W5_PRG6.py
Enter the n value : 5
Sum of first 5 natural numbers = 15
```

7. Write a program to Assign a different name to function and call it through the new name

- Below is the function
`display_student(name, age)`
 - Assign a new name
`show_student(name, age)`
- to it and call it using the new name.**

```
def display_student(name, age):  
    print("Name : ",name,"\t Age : ",age)  
  
n,a=input("Enter your name, age : ").split()  
print(" Display name and age with display_student() function")  
display_student(n, a)  
show_student = display_student  
print(" Display name and age with show_student() function")  
show_student(n,a)
```

OUTPUT:

```
C:\PPLAB\Week5>py W5_PRG7.py  
Enter your name, age : HemaLatha 50  
Display name and age with display_student() function  
Name : HemaLatha      Age : 50  
Display name and age with show_student() function  
Name : HemaLatha      Age : 50
```

8. Let's define a function student() with four arguments name, age, grade, and school. In this function, grade and school are default arguments with default values.

- If you call a function without school and grade arguments, then the default values of grade and school are used.
- The age and name parameters do not have default values and are required (mandatory) during a function call.

```
def student(name, age, branch="ECE", college="LBRCE"):  
    print('Student Details:', "\n Name : ",name,"\n Age : ",age, "\n Branch : ",branch, "\n College : ",college)  
  
student(name='Hema', age=50)
```

```
student(age=50,branch="IT",name="Latha")
student(college="ABC",branch="CSE",name="Jay",age=18)
```

OUTPUT:

```
C:\PPLAB\Week5>py W5_PRG8.py
Student Details:
Name : Hema
Age : 50
Branch : ECE
College : LBRCE
Student Details:
Name : Latha
Age : 50
Branch : IT
College : LBRCE
Student Details:
Name : Jay
Age : 18
Branch : CSE
College : ABC
```

9. Determine the number of subjects at runtime, write a function that could calculate the average of all subjects no matter how many there are.

```
def percentage(**kwargs):
    sum = 0
    for sub in kwargs:
        sub_name = sub
        sub_marks = kwargs[sub]
        sum=sum+kwargs[sub]
        print(sub_name, "=", sub_marks)
    avg=sum/len(kwargs)
    print("Average Marks = %.2f"%avg)
percentage(math=56, english=61, science=73)
```

OUTPUT:

```
C:\PPLAB\Week5>py W5_PRG9.py
math = 56
english = 61
science = 73
Average Marks = 63.33
```