LAB MANUAL 7

# # Python built-in Functions  
*'''for i in range(1, 10):  
 print(i, end=' ')  
# Output 1 2 3 4 5 6 7 8 9'''*# function  
'''def message():  
 print("Welcome to PYTHON MODULE")  
 for i in range(1, 10):  
 print(i, end=' ')  
  
  
# call function using its name  
message()  
message()  
message()  
message()'''  
  
#Creating a function with parameters  
  
# function  
'''def course\_func(name, course\_name):  
 print("Hello", name, "Welcome to PYnative")  
 print("Your course name is", course\_name)  
  
# call function  
course\_func('John', 'Python')'''  
  
#Creating a function with parameters and return value  
  
# function  
'''def calculator(a, b):  
 add = a + b  
 print(a, b)  
 # return the addition  
 return add  
  
# call function  
# take return value in variable  
returnfromadd = calculator(22, 66)  
  
print("Addition :", returnfromadd)  
calculator(22, 66)'''  
  
#Calling a function  
  
# function  
'''def even\_odd(n):  
 # check numne ris even or odd  
 if n % 2 == 0:  
 print('Even number')  
 else:  
 print('Odd Number')  
  
# calling function by its name  
even\_odd(45)  
  
even\_odd(90)  
  
even\_odd(99)'''  
  
#  
  
# import randint function  
'''from random import randint  
  
# call randint function to get random number  
print(randint(10, 20))'''  
  
#single line docstring  
'''def factorial(x):  
 """This function returns the factorial of a given number."""  
 return x  
  
# access doc string  
  
print(factorial.\_\_doc\_\_)'''  
  
  
# Multi-line Docstring  
'''def any\_fun(parameter1):  
 """  
 Description of function  
  
 Arguments:  
 parameter1(int):Description of parameter1  
  
 Returns:  
 int value  
 """  
  
  
print(any\_fun.\_\_doc\_\_)'''  
  
# Return outcome of the function  
  
'''def is\_odd(list1):  
 odd\_num = []  
 for n in list1:  
 if n % 2 == 1:  
 odd\_num.append(n)  
 # return a list  
 return odd\_num  
  
# Pass list to the function  
odd\_num = is\_odd([2, 3, 42, 51, 62, 70, 5, 9, 5, 7,8, 9, 10, 14])  
print("Even numbers are:", odd\_num)'''  
  
  
# Multiple return values  
'''def arithmetic(num1, num2):  
 add = num1 + num2  
 sub = num1 - num2  
 multiply = num1 \* num2  
 division = num1 / num2  
 # return four values  
 return add, sub, multiply, division  
  
  
# read four return values in four variables  
a, b, c, d = arithmetic(10, 2)  
  
  
print("Addition: ", a)  
print("Subtraction: ", b)  
print("Multiplication: ", c)  
print("Division: ", d)'''  
  
# The pass statement for function  
'''def addition(num1, num2):  
 # Implementation of addition function in comming release  
 # Pass statement  
 pass'''  
  
  
# Varibale scope and areas  
  
'''global\_lang = 'DataScience'  
  
def var\_scope\_test():  
 local\_lang = 'Python'  
 a = "hello inside the function"  
 print(local\_lang)  
 print(a)  
  
var\_scope\_test()  
# Output 'Python'  
  
# outside of function  
print(global\_lang)'''  
# Output 'DataScience'''  
  
# NameError: name 'local\_lang' is not defined  
  
  
'''def function1():  
 # local variable  
 loc\_var = 888  
 print("Value is :", loc\_var)  
  
def function2():  
  
 print("Value is :", loc\_var)  
  
function1()  
function2()'''  
  
# Global Variable use  
'''global\_var = 999  
  
def function1():  
 print("Value in 1nd function :", global\_var)  
  
def function2():  
 print("Value in 2nd function :", global\_var)  
  
function1()  
function2()'''  
  
  
# Global variable  
'''global\_var = 5  
  
def function1():  
 print("Value in 1st function :", global\_var)  
  
def function2():  
 # Modify global variable  
 # function will treat it as a local variable  
 global\_var = 555  
 print("Value in 2nd function :", global\_var)  
  
def function3():  
 print("Value in 3rd function :", global\_var)  
  
function1()  
function2()  
function3()'''  
  
#As you can see, function2() treated global\_var as a new  
# variable (local variable). To solve such issues or access/modify  
# global variables inside a function, we use the global keyword.  
  
# Global variable  
x = 5  
  
# defining 1st function  
def function1():  
 print("Value in 1st function :", x)  
  
# defining 2nd function  
def function2():  
 # Modify global variable using global keyword  
 global x  
 x = 555  
 print("Value in 2nd function :", x)  
  
# defining 3rd function  
def function3():  
 print("Value in 3rd function :", x)  
  
function1()  
function2()  
function3()