**ASSIGNMENT 3 SOLUTION**

1. **A function** **is basically a block of statements that performs a particular task. Suppose a task needs to be performed continuously on many data at different points of time, like one at the beginning of the program and one at the end of the program, so instead of writing the same piece of code twice, a person can simply write it in a function and call it twice. And after the execution of any function block, the control always comes back to the *main()* function.**

**Advantages:-**

1. **Enhancing the readability of a program by breaking the code into smaller functions, which keeps the program organized, easy to understand, and makes it reusable.**
2. **Allowing the divide and conquer strategy to be used for the development of programs.**
3. **Helping avoid duplication of effort and code in programs.**

**[2]  when we called the function then the code in a function will run.**

**[3]  by the help of def keyword  we can create our function.**

**e.g;   def fact():   # fact() is the function name.**

**return 1**

**[4]  A function is a block of code that does a particular operation and returns a result. It usually accepts inputs as parameters and returns a result. The parameters are not mandatory.**

**By the help of function calling we can reuse the function multiple times for the same task in anywhere of the program.**

**e.g;   def fact():   # fact() function will return 1**

**return 1**

**fact()    # here, I’m calling the function**

**Output:- 1**

**[5]   Local scope refers to variables inside a function that can only be accessed within that function. Control blocks like if, while, and for do not create a new local scope. Variables inside them belong to the enclosing function.**

def local\_scope():   
    if True:   
        x = 'x'   
    print(f'{x} is local')  # outputs `x is local`   
print(f'{x} is local')  # raises NameError: name 'x' is not defined

**Global scope refers to variables declared outside functions that can be accessed throughout the program. Technically, Python does not have program-level scope, only module-level scope. Global variables can be accessed from local scope, but to modify them from local scope, you need to use the global keyword.**

X = 'X' 

def global\_scope():   
    global X   
    print(f'{X} is global')   
    X += 'Y'   
global\_scope()  # outputs 'X is global'   
global\_scope()  # outputs 'XY is global'

**[6]  A variable if defined inside  a function then that variable scope will be local and after calling that function that local scope variable will also run but if that the same variable is also defined outside the function then during the function calling, value will be taken inside of the function  after function calling means outside the function global variable value will be counted.**

**num = 20    # global variable**

**def** **val\_calc():**

**num = 40    # local variable**

**print(num)     à output for this line will be 40.**

**val\_calc()**

**print(num)   à output for this line will be 20.**

**[7]  A** **return statement is used to end the execution of the function call and “returns” the result (value of the expression following the return keyword) to the caller. The statements after the return statements are not executed. If the return statement is without any expression, then the special value None is returned. A return** **statement is overall used to invoke a function so that the passed statements can be executed.**

**Normally, return statement returns any expression value but if we want to return any expression then it is not possible to return that but if we return that expression inside the single or doube quotes then Expression can be returned.**

**[8]  None will be return if we did’t pass return statement in function.**

**def** **val\_calc():**

**num = 40**

**print(num)**

**print(val\_calc())  # since I’m using return statement so this function call will also print None.**

**Output:-    40**

**None    à because I did’t pass return statement during the function definition.**

**[9]  by using global keyword we can make function variable to global variable.**

**Code:-** def     val\_calc():

   global num

    num = 40

    print(num)

val\_calc()

print(num)

**[10]  datatype of None is NoneType.**

**[11]  import areallyourpetsnamederic  à by using this we are importing areallyourpetsnamederic   package or module in my current python or .py file.**

**[12]    import spam  # spam module imported**

**spam.bacon()  # now, bacon() feature will be called.**

**[13]    by using try/Except block we can save our program to crash If it encounters error.**

**[14]    try block used to test a block of code for errors and except block handle the error.**

**try:**

**print(x)**

**except:**

**print("An exception occurred")**

**since x is not defined that’s why try block will throw error but except block will handle this error and it will print the An exception occurred.**