**Assignment\_3**

1. **Why are functions advantageous to have in your programs?**

* Functions saves out time by using it as many times we wish to use throughout the programming by just calling it in one line of code with appropriate argument if needed.
* It makes our code a bit simple.
* It makes our code small in size.

1. **When does the code in a function run: when it’s specified or when it’s called?**

The code in a function run when it’s called.

1. **What statement creates a function?**

We use statement “def func\_name:” to create a function.

1. **What is the difference between a function and a function call?**

Function can be inbuilt function such as print, range etc or user defined function that user defines in his/her code itself or save it to same directory where he/she will be using that function in other code saved in same directory. We use statement “def func\_name:” to create a function.

Function is called when we need that function in our coding as many times we want. When it is called interpreter look for function in the code or directory and execute it and whatever value is returned is assigned to a variable.

1. **How many global scopes are there in a Python program? How many local scopes?**

The part of a program where a variable is accessible is called its scope. There are four major types of variable scope and is the basis for the LEGB rule. The letters in the acronym LEGB stand for Local, Enclosing, Global, and Built-in scopes.

**Local Scope:** Whenever you define a variable within a function, its scope lies ONLY within the function. It is accessible from the point at which it is defined until the end of the function and exists for as long as the function is executing (Source). Which means its value cannot be changed or even accessed from outside the function.

**Global scope:** The names that you define in this scope are available to all your code.

The names in your programs will have the scope of the block of code in which you define them. When you can access the value of a given name from someplace in your code, you’ll say that the name is in scope. If you can’t access the name, then you’ll say that the name is out of scope.

1. **What happens to variables in a local scope when the function call returns?**

Function uses variable in local scope to return a value when function is called. Local variable defined inside function can’t be accessed or modified from outside the function.

1. **What is the concept of a return value? Is it possible to have a return value in an expression?**

we use return in function where we don’t want value immediately after execution. A return statement is used to end the execution of the function call and “returns” the result.

No, it is not possible to have a return value in an expression.

1. **If a function does not have a return statement, what is the return value of a call to that function?**

If a function does not have a return statement, then return value of a call to that function is “<class 'NoneType'>”.

1. **How do you make a function variable refer to the global variable?**

Normally, when you create a variable inside a function, that variable is local, and can only be used inside that function. To create a global variable inside a function, you can use the global keyword.

1. **What is the data type of None?**

Data Type of None is “<class 'NoneType'>”

1. **What does the sentence import areallyourpetsnamederic do?**

Since, In the directory in which I am writing this code, this ‘**areallyourpetsnamederic**’ module is not available so it will through an error (ModuleNotFoundError).

1. **If you had a bacon () feature in a spam module, what would you call it after importing spam?**

If I have a bacon () feature in a spam module, I would call feature from spam module like this “spam.bacan()”. We can assign this to any variable.

1. **What can you do to save a programme from crashing if it encounters an error?**

We use exception handling concept to save a programme form crashing if it encounters an error. We do have five keywords in exception handling **try, except, else, raise, and finally**. By using these keywords in coding, we can save our programming form crashing.

1. **What is the purpose of the try clause? What is the purpose of the except clause?**

In try clause we write our main part of code. If every thing in the try clause runs fine then interpreter jumps the exception clause execution. If some kind of error is found then interpreter run exception clause. In general, in exception block, programmer writes code for reasons for error occurred.

**The End**