**Python Assignment - 3**

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

**Ans:** Functions reduce the need for duplicate code. This makes programs shorter, easier to read, and easier to update.

With the help of functions, we can avoid rewriting the same logic or code again and again in a program. In a single Program, we can call Python functions anywhere and also call multiple times.

2. When does the code in a function run: when it's specified or when it's called?

**Ans:** The code inside a function is not executed when the function is defined. The code inside a function is executed when the function is invoked. It is common to use the term "call a function" instead of "invoke a function".

3. What statement creates a function?

**Ans:** In Python, you define a function with the def keyword, then write the function identifier (name) followed by parentheses and a colon.

4. What is the difference between a function and a function call?

**Ans:** A function consists of the def statement and the code in its def clause.

A function call is what moves the program execution into the function, and the function call evaluates to the function’s return value.

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

**Ans:** There's only one global Python scope per program execution. This scope remains in existence until the program terminates and all its names are forgotten. Otherwise, the next time you were to run the program, the names would remember their values from the previous run.

 Local scope is created whenever a function is called.

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

**Ans:** When the execution of the function terminates (returns), the local variables are destroyed. Codelens helps you visualize this because the local variables disappear after the function returns.

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

**Ans:** A return is a value that a function returns to the calling script or function when it completes its task. A return value can be any one of the four variable types: handle, integer, object, or string. The type of value your function returns depends largely on the task it performs. A return value can be used as part of an expression.

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

**Ans:** If there is no return statement for a function, its return value is None.

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

**Ans:** A global statement will force a variable in a function to refer to the global variable.

10. What is the data type of None?

**Ans:** The data type of None is NoneType.

11. What does the sentence import areallyourpetsnamederic do?

**Ans:** That import statement imports a module named areallyourpetsnamederic. (But a module of this name doesnt exists in Python)

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

**Ans:** This function can be called with spam.bacon().

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

**Ans:** We can use try except clause to save a program from crashing.

14. What is the purpose of the try clause? What is the purpose of the except clause?

**Ans:** The code that could potentially cause an error goes in the try clause.

The code that executes if an error happens goes in the except clause.