PYTHON ASSIGNMENT 3

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

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

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

   The code in a function executes when the function is called, not when the function is defined.

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

   In Python, you define a function with the def keyword, and then write the function identifier (name) followed by parentheses and a colon. The next thing you have to do is make sure you indent with a tab or 4 spaces, and then specify what you want the function to do for you.

   ```
   def functionName():
       # What to make the function do
   ```

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

   Function in Python

   In Python, a function is a block of code that performs a specific task and may or may not return a value. A function is defined using the def keyword, followed by the function name and a set of parentheses that may include parameters. Here is an example of a function definition in Python:

   ```
   def greet(name):
       print("Hello, " + name)
   ```

   This function, called greet, takes a single parameter called name and prints a greeting message using the value of name.

   Function call in Python

   A function call is an expression that invokes a function. To call a function in Python, you simply write the function name followed by a set of parentheses that may include arguments. Here is an example of a function call in Python:

   ```
   greet("John")
   ```

   This function call invokes the greet function with the argument "John", which will cause it to print the message "Hello, John".

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

   A variable created inside a function belongs to the local scope of that function, and can only be used inside that function. A variable created in the main body of the Python code is a global variable and belongs to the global scope. Global variables are available from within any scope, global and local. There is one global scope, and a local scope is created whenever a function is called.

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

   When a function returns, the local scope is destroyed, and all the variables in it are forgotten.

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

   The Python return statement is a special statement that you can use inside a function or method to send the function's result back to the caller. A return statement consists of the return keyword followed by an optional return value. The return value of a Python function can be any Python object. Everything in Python is an object. So, your functions can return numeric values (int, float, and complex values), collections and sequences of objects (list, tuple, dictionary, or set objects), user-defined objects, classes, functions, and even modules or packages. Like any value, 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?**

   You can omit the return value of a function and use a bare return without a return value. You can also omit the entire return statement. In both cases, the return value will be None.

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

   Accessing global variables directly from inside a function is completely possible unless you have local variables with the same name in the containing function. Apart from that, if you need to change the variable's value, you can use the global keyword to declare that you want to refer to the global variable.

   ```
   global variable_0, variable_1, ..., variable_n
   ```

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

    The None keyword is used to define a null variable or an object. In Python, None keyword is an object, and it is a data type of the class NoneType . We can assign None to any variable, but you cannot create other NoneType objects.

11. What does the sentence import areallyourpetsnamederic do?

That import statement imports a module named areallyourpetsnamederic. (This isn't a real Python module, by the way.)

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

This function could be called with spam.bacon().

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

Place the line of code that might cause an error in a try clause.

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

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.