**Python Assignment - 3**

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

=> Functions provide several advantages in programming.

1. Code reusability:- Functions allow you to define a block of code that can be reused multiple times within a program or even in different programs.
2. Modularity:- Functions enable you to break down a complex problem into smaller, more manageable tasks.
3. Abstraction:- Functions allow you to abstract away implementation details.
4. Code maintenance:- Functions make code maintenance easier.
5. Code Readability:- Functions enhance code readability by providing meaningful names to blocks of code that describe their purpose or functionality.
6. Encapsulation:- Function allow you to encapsulate code and data within a self-contained unit.

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

=> The code in the function runs when the function is called.

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

=> ‘def’ statement is used to create a function.

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

=>

* Function: A function is a named block of code that performs a specific task or set of tasks. It encapsulates a sequence of statements and can have input parameters (optional) and return values (optional).
* Function Call: A function call is the actual execution of a function. It is the point in the program where the code within the function's body is invoked or executed.

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

=> In python there is one global scope and multiple local scopes.

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

=> When a function call returns in Python, the local variables within that function's scope cease to exist. They are destroyed, and their values are no longer accessible. This process is known as "variable scope and lifetime."

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

=>

* The concept of a return value in programming refers to the value that a function can provide as a result after its execution. When a function is called, it can perform some operations and optionally return a value back to the caller.
* It is not possible to have a return value directly within an expression.

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

=> If a function in Python does not have a return statement, the function will still return a value when called. However, the return value will be **None**.

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

=> In Python, if you want to make a function variable refer to a global variable, you can use the global keyword within the function. The **global** keyword allows you to explicitly declare that a variable within a function is referring to the **global** variable with the same name.

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

=> The data type of **None** is **NoneType.**

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

=> The sentence "import areallyourpetsnamederic" is not a valid Python import statement. It appears to be a made-up sentence that does not follow the standard syntax for importing modules in Python.

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

=> import spam

spam.bacon()

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

=> To prevent a program from crashing when it encounters an error, you can implement error handling techniques to catch and handle exceptions gracefully. This way, you can handle errors and take appropriate actions instead of the program abruptly terminating.

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

=>

* The purpose of the **try** clause in Python is to enclose a block of code that might raise an exception or error. The code within the **try** block is executed, and if an exception occurs during the execution, the flow of control is transferred to the corresponding **except** block.
* The purpose of the **except** clause is to specify how to handle exceptions that occur within the corresponding **try** block. The **except** block is responsible for catching and handling specific exceptions or a group of exceptions.