**Q1**

Functions allow you to break down your code into smaller, manageable pieces.

Once you've defined a function to perform a particular task, you can reuse it throughout your program or even in different projects.

Breaking code into functions with descriptive names can make your code more readable and self-documenting.

As your codebase grows, functions make it easier to manage complexity. By having distinct functions responsible for specific tasks

**Q2**

Code in function runs when it is called

**Q3**

The "def" statement stands for "define" and is followed by the function name, a pair of parentheses containing any function parameters, and a colon to indicate the start of the function's code block.

Ex

def sumTwoNo(a,b):

**Q4**

A function is a block of code that performs a specific task or a set of tasks.

Functions can be thought of as reusable blocks of code that can be executed whenever needed

A function call, also known as executing a function, is the act of actually using the function in your code. When you want to use a function to perform its defined task, you make a function call.

**Q5**

In a Python program, there is one global scope and potentially multiple local scopes. 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.

Local scopes are created within functions or code blocks. Variables defined within these local scopes are called local variables and are only accessible within the scope in which they are defined.

**Q6**

When a function call returns in Python, the local variables within that function's scope are destroyed. This means that the memory allocated for these local variables is released, and the variables are no longer accessible once the function's execution completes

**Q7**

The concept of a "return value" is used with functions in programming. When a function is executed it produce a value as a result of its computation. This value is known as the return value of the function

Yes, it is possible to have a return value in an expression.A function call that returns a value can be used directly within an expression

**Q8**

None.

In Python, None is a special constant that represents the absence of a value or a "null" value.

**Q9**

we can achieve this using the global keyword.

Ex-

var = "hello"

def modify\_global():

global var

var = "hii"

print(var)

modify\_global()

print(var)

output:

hello

hii

**Q10**

None type

**Q11**

The module name follows the "import" keyword, and it should be a valid module name that corresponds to an existing Python module.

"import areallyourpetsnamederic" doesn't have any functional meaning in standard programming contexts

**Q12**

Spam.bacon()

**Q13**

We can write Pass in except block (while using exception handling )

**Q14**

The try clause is used to enclose the code that you suspect might raise an exception. It's where you place the code that you want to protect from causing a program crash if an error occurs.

The except clause is used to specify how to handle specific types of exceptions that might be raised within the associated try block