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

Ans. **Modularity:** Functions allow you to break down a program into smaller, manageable pieces. Each function can perform a specific task, making the overall program easier to understand, debug, and maintain.

**Reuse:** Once a function is defined, it can be reused multiple times within the program or even in different programs. This promotes code reusability and avoids redundancy.

**Abstraction:** Functions abstract away implementation details, allowing you to focus on what each function does rather than how it does it. This improves code readability and makes it easier to collaborate with other developers.

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

Ans. When you define a function in your code, you're essentially telling the program what actions to take when that function is called, but those actions aren't executed until the function is actually invoked elsewhere in the program.

3. What statement creates a function?

Ans. In Python, the def keyword is used to define a function.

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

Ans. The difference between a function and a function call lies in their roles and actions within a program:

Function:

* A function is a block of code that performs a specific task or set of tasks.
* It typically has a name, parameters (optional), and a body of code.
* Functions are defined using a function declaration or definition syntax in the programming language. For example, in Python, functions are defined using the def keyword, while in JavaScript, the function keyword is used.

Function Call:

* A function call is an instruction in a program that tells the program to execute a specific function.
* When a function is called, the program transfers control to the function, executes the code inside it, and then returns control to the point in the program immediately after the function call.
* A function call typically includes the function name followed by parentheses, optionally containing arguments or parameters that are passed to the function.
* Function calls are how you use the defined functions in your program to perform specific tasks.

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

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

Global Scope:

* There is only one global scope per Python program.
* The global scope is the outermost scope, and it exists throughout the entire program.
* Variables defined outside of any function or class, as well as those explicitly declared with the global keyword within a function, belong to the global scope.
* Variables in the global scope are accessible from anywhere within the program.

Local Scopes:

* Each function call creates its own local scope.
* Local scopes are temporary and exist only during the execution of the function.
* Variables defined within a function are local to that function and are accessible only within the function's scope.
* When a function is called, its local variables are created, and when the function completes execution, its local variables are destroyed.

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

Ans. When a function call returns in Python, the local variables within the function scope are destroyed. This process is known as variable deallocation or garbage collection. Here's what happens:

* Variable Deletion: When the function call returns, Python deallocates (destroys) the local variables that were created within the function scope.
* Memory Reclamation: The memory space allocated to the local variables is freed up, allowing it to be reused by other parts of the program.
* Scope Cleanup: The local scope associated with the function call is removed, and any variables defined within that scope are no longer accessible.

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

Ans. The concept of a return value is fundamental in programming. When a function is called, it may perform some computation or task and then optionally return a value to the caller. The return value is the data or result that the function provides back to the part of the program that called it.

Return value in an expression is very common to use the return value of a function directly within expressions to perform further computations or operations.

Here's an example:

def add(x, y):

return x + y

result = add(3, 4) \* 2

print(result)

# Output: 14

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

Ans. If a function does not have a return statement, the return value of a call to that function will be None. In Python, None is a special built-in constant that represents the absence of a value or a null value. When a function does not explicitly return a value using the return statement, Python implicitly returns None at the end of the function execution.

Here's an example:

def greet(name):

print("Hello,", name)

result = greet("Alice")

print(result)

# Output: None

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

Ans. In Python, if you want to make a function variable refer to a global variable, you can use the global keyword within the function to explicitly declare the variable as global. This tells Python to use the global variable with the specified name rather than creating a new local variable within the function.

10. What is the data type of None?

Ans. In Python, None is a built-in constant that represents the absence of a value or a null value. It is a singleton object of the NoneType class. Therefore, the data type of None is NoneType

11. What does the sentence import areallyourpetsnamederic do?

Ans. The sentence "import areallyourpetsnamederic" in Python is a valid import statement, assuming there exists a Python module named areallyourpetsnamederic.

However, in terms of practicality, it's highly unlikely that such a module exists in the standard Python library or in any commonly used third-party libraries. The statement would typically raise an ImportError if the module is not found.

Python's import statement is used to import modules, which are Python files containing definitions and statements. When you import a module, Python searches for it in the list of directories defined in the sys.path variable. If the module is found, its contents become available in your current Python session, allowing you to use the functions, classes, and variables defined in the module. If the module is not found, Python raises an ImportError.

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

Ans. After importing the spam module in Python, if you want to access the bacon() function from the spam module, you would call it using dot notation.

Example:

import spam  
  
spam.bacon()

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

Ans. Specific Exception Handling: Catch specific exceptions to handle them differently. This allows for more targeted error handling.

try:

# Code that may raise an exception

result = int("abc") # This will raise a ValueError

except ValueError:

# Handle the specific exception

print("Error: Unable to convert to integer")

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

Ans.

Try Clause:

* The try clause is used to enclose the code that might raise an exception.
* The purpose of the try clause is to execute the code block within it. If an exception occurs within this block, Python immediately stops executing the code in the try block and jumps to the corresponding except block.
* The try clause ensures that if an exception occurs, the program does not crash immediately but instead gracefully handles the error.

Except Clause:

* The except clause is used to specify the actions to be taken if a specific exception occurs within the corresponding try block.
* You can have one or more except clauses to handle different types of exceptions or handle exceptions in different ways.
* The purpose of the except clause is to catch and handle exceptions raised by the code in the try block. It allows you to define how your program should respond to specific types of errors.