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

Functions are advantageous in Python for the following reasons:

1. Modularity:

Functions allow you to break your code into smaller, reusable, and self-contained blocks. This promotes modular programming, making your code easier to understand and maintain.

1. Reusability:

Once defined, functions can be called multiple times from different parts of your program, reducing code duplication and making it more efficient.

1. Abstraction:

Functions hide the implementation details, allowing you to focus on the high-level logic. This makes your code more readable and easier to work with.

1. Testing and Debugging:

Functions enable easier testing and debugging of specific parts of your code, as you can isolate and analyse individual functions.

1. Collaboration:

Functions facilitate collaboration among developers, as they can work on different functions independently, and the functions can be integrated into the main program.

1. Code Maintenance:

Functions make it simpler to update and maintain your code since changes are isolated within the function, reducing the risk of unintentional side effects on other parts of the program.

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

* The code within a function is executed when the function is called, not when it's defined or specified.

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

* you can create a function using the “def” statement followed by a function name.

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

* A function is a reusable block of code that performs a specific task when it's defined and can be called later in your program.
* A function call is the act of invoking or using a function, triggering its execution to perform the task it was designed for.

In short, a function is the code itself, while a function call is when you use that code to perform a specific action.

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

* There is typically only one global scope. The global scope is the highest level of scope in Python.
* a local scope is created whenever a function is called. Each function call establishes its own local scope, which is separate from the global scope. This means that the number of local scopes in a Python program depends on how many functions are called within the program.

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

* Variables in a local scope are destroyed (they go out of scope) when the function call returns. When a function is called, a local scope is created for that function, and any
* variables defined within that function are only accessible within that local scope. When the function call returns (i.e., when the function execution is complete), the local scope is destroyed, and the variables defined within it cease to exist.

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

* A return value is the value that a function provides as output when it is called and executed
* Yes, we can have a return value in an expression defined in a single line or "short" expression.

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

* if a function does not have a return statement, it will implicitly return None. None is a special value in Python that represents the absence of a value or a null value.

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

* To make a function variable refer to a global variable, you need to use the global keyword. This tells Python that the variable you're working with inside the function should be treated as a global variable.

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

* The data type of ‘None’ for a function is simply the ‘NoneType’. None is a special singleton object of the ‘NoneType’ that represents the absence of a value or a null value. When a function doesn't explicitly return a value, it effectively returns None.

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

* The sentence "import areallyourpetsnamederic" does not have any specific meaning or functionality in standard programming or natural language. It appears to be a combination of words that doesn't form a valid statement or command in most programming languages or communication contexts.

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

If you have a bacon() function defined in a module named spam and you want to call it after importing the spam module in Python, you would call it like this:

import spam

spam.bacon()

This assumes that the bacon() function is defined within the spam module.

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

* The primary mechanism for handling errors is through the use of exception handling.

Try-Except Blocks:

Wrap the code that might raise an exception in a try-except block. If an exception occurs within the try block, the program will not crash, and you can handle the exception gracefully.

try:

# Code that might raise an exception

except ExceptionType as e:

# Handle the exception

Replace ExceptionType with the specific exception you want to catch, or use Exception for a general catch-all.

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

* The try clause is used to implement exception handling. It is part of a try-except block, which allows you to gracefully handle exceptions or errors that may occur during the execution of your code. The purpose of the try clause is to enclose a block of code where you anticipate that an exception might occur.
* The except clause is used in exception handling to specify a block of code that should be executed when an exception is raised in the corresponding try block. The purpose of the except clause is to handle exceptions gracefully, preventing your program from crashing and allowing you to take appropriate actions when an unexpected error occurs.