Q1. Why are functions advantageous to have in your programs?

* Functions provide modularity, allowing you to break down your code into smaller, reusable chunks. This makes code easier to read, understand, and maintain.
* Functions promote code reusability, reducing duplication and improving efficiency.
* They help in organizing and structuring code, making it easier to debug and troubleshoot.
* Functions make it easier to collaborate on larger projects by dividing the work into manageable parts.

Q2. When does the code in a function run: when it's specified or when it's called?

* The code in a function runs when the function is called (invoked). Defining a function only specifies the behavior of the function, while calling it actually executes the code within it.

Q3. What statement creates a function?

* The def statement is used to create a function in Python. It is followed by the function name, parentheses for parameters (if any), a colon, and then the indented block of code that makes up the function's body.

Q4. What is the difference between a function and a function call?

* A function is a block of code that performs a specific task. It is defined using the def statement and contains a sequence of instructions.
* A function call is the act of invoking a function to execute its code. When you call a function, the program jumps to the function's definition, runs the code inside it, and then returns to where the function was called.

Q5. How many global scopes are there in a Python program? How many local scopes?

* A Python program has one global scope, which includes variables defined outside of any function.
* The number of local scopes depends on the number of functions defined in the program. Each function creates its own local scope.

Q6. What happens to variables in a local scope when the function call returns?

* When a function call returns, the local variables within that function's scope are destroyed (freed up). They are no longer accessible after the function call completes.

Q7. 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 gives back after it completes its execution. It's often used to pass data from a function back to the caller.
* Yes, it's possible to use a return value in an expression. For example, you can assign the return value of a function to a variable or use it directly in an operation.

Q8. 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, its return value is None. This is a special Python object that represents the absence of a value.

Q9. How do you make a function variable refer to the global variable?

* You can use the global keyword inside a function to indicate that a variable refers to a global variable rather than creating a new local variable with the same name.

For example:

x = 10

def modify\_global():

global x

x = 20

Q10. What is the data type of None?

* The data type of None is NoneType. It is a special value that represents the absence of a value or a null value.

Q11. What does the sentence import areallyourpetsnamederic do?

* The sentence import areallyourpetsnamederic attempts to import a module named areallyourpetsnamederic. If such a module exists, its contents become available for use. If it doesn't exist, it will result in an ImportError.

Q12. If you had a bacon() feature in a spam module, what would you call it after importing spam?

After importing the spam module, you can call the bacon() feature using the following syntax:

import spam

spam.bacon()

Q13. What can you do to save a programme from crashing if it encounters an error?

You can use error handling techniques like try and except blocks to catch and handle errors gracefully. This prevents the program from crashing and allows you to handle errors in a controlled manner.

Q14. What is the purpose of the try clause? What is the purpose of the except clause?

* The try clause is used to enclose code that might raise an exception. It's used to identify a section of code where an error might occur.
* The except clause is used to specify a block of code that should be executed if an exception of a specific type occurs within the try block. It provides a way to handle exceptions and prevent program crashes.