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

ANSWER:

Functions in programming are advantageous for the following key reasons:

Code Reusability: They allow code to be reused, minimizing repetition.

Code Organization: They provide a way to structure code into logical and manageable sections.

Code Maintenance: Changes needed can be implemented to a specific function without affecting other parts of the program, simplifying updates and bug fixes.

Code Sharing: Functions can be shared across multiple programs, enhancing modularity and code reuse.

Abstraction: They encapsulate complex operations, hiding the details and reducing complexity.

Testing: Functions make it easier to conduct unit testing and ensure each part of the program works correctly.

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

ANSWER:

The code in a function runs when the function is called, not when the function is defined.

3. What statement creates a function?

ANSWER:

def

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

ANSWER:

A function is a reusable piece of code that performs a specific task. In Python, a function is defined using the def keyword followed by the function's name and any parameters it takes. The body of the function contains the code that will be executed when the function is called. Simply defining a function does not cause it to run.

However, a function call is a command that executes a function that has been previously defined. When the function is called, the code within the function's body is executed.

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

ANSWER:

A Python program has one global scope and as many local scopes as the number of function calls. Each function call creates its own local scope, separate from the global scope and any other local scopes.

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

ANSWER:

Local variables only exist during the execution of the function they are defined in. Once the function call returns, the local variables are cleared out

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

ANSWER:

A return value is the value that a function gives back to the part of the program that called it. This is accomplished using the return statement in Python. Once a return statement is executed in a function, the function stops executing and passes the return value back to the caller.

Yes, it is possible to have a return value in an expression

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

ANSWER:

If a function does not have a return statement, the default return value of a call to that function is None in Python.

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

ANSWER:

you can use the *global* keyword before the variable

example:

```
x = 10  # This is a global variable

def func():

    global x  # This refers to the global variable x

    x = 20  # This modifies the global variable

func()

print(x)  # This will print: 20
```

10. What is the data type of None?

ANSWER:

None is a special constant in Python that represents the absence of a value or a null value. It is an object of its own datatype - the NoneType.

11. What does the sentence import areallyourpetsnamederic do?

ANSWER:

The sentence *import areallyourpetsnamederic* in Python will import a module named *areallyourpetsnamederic*.

for this *import areallyourpetsnamederic* to work, there needs to be a module with the filename areallyourpetsnamederic.py If there isn't a module with that name, you'll get a ModuleNotFoundError.

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

ANSWER:

```
import spam

spam.bacon()
```

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

ANSWER:

These error handling techniques can help you prevent your program from crashing when it encounters an error and allow your program to handle the error gracefully and continue its execution.

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

ANSWER:

the purpose of the try clause is to contain code that might raise an error or exception, and the purpose of the except clause is to specify how to handle the error or exception.