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

Ans:

Functions are advantageous in programs because they promote modularity, code reusability, abstraction, code readability, code organization, and simplify complex long codes.

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

Ans:

The code in a function runs when it is called.

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

Ans:

The "def" statement creates a function in Python.

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

Ans:

A function is the definition of a code block, while a function call is the act of running that code block to perform its intended operations

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

Ans:

There is only one global scope in a Python program.

The number of local scopes is not fixed or predetermined. It depends on the structure of the program and how many times functions are called or blocks of code are executed.

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

Ans:

When a function call returns in Python, the variables in its local scope are destroyed and 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 refers to the value that a function can send back to the caller. When a function is executed, it can perform operations and optionally provide a result back to the code that called the function. It can be used in an expression to perform further calculations.

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

Ans:

The return value of a call to a function that does not have a return statement is **None**.

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

Ans:

To make a function variable refer to a global variable in Python, use the **global** keyword before the variable assignment inside the function.

A screenshot of a computer

Description automatically generated

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

Ans:

The data type of **None** in Python is **NoneType**.

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

Ans:

The sentence import areallyourpetsnamederic does not do anything as Python does not have any such existing module or library.

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

Ans:

We can call the **bacon()** function using **spam.bacon()** after importing the **spam** module.

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

Ans:

To save a program from crashing when encountering an error, you can use exception handling with the **try-except** block to catch and handle the error

Eg:

try:

result = 10 / 0

except ZeroDivisionError:

print("Error: Division by zero occurred.")

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

Ans:

The purpose of the **try** clause is to enclose code that might raise an exception, and the purpose of the **except** clause is to handle specific exceptions that occur within the **try** block.

­­