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

Ans -With the help of functions, **we can avoid rewriting the same logic or code again and again in a program**. In a single Program, we can call Python functions anywhere and also call multiple times. We can track a large Python program easily when it is divided into multiple functions.

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

Ans-The code in a function executes **when the function is called**, not when the function is defined

3.What statement creates a function?

Ans- you define a function with the **def keyword**, then write the function identifier (name) followed by parentheses and a colon. The next thing you have to do is make sure you indent with a tab or 4 spaces, and then specify what you want the function to do for you.

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

Ans- A function is a block of code that does a particular operation and returns a result. It usually accepts inputs as parameters and returns a result. The parameters are not mandatory. A function call is the code used to pass control to a function.

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

Ans-At any given time during execution, you'll have at most four active Python scopes—local, enclosing, global, and built-in—depending on where you are in the code. On the other hand, you'll always have **at least two active scopes**, which are the global and built-in scopes.

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

Ans-When the execution of the function terminates (returns), **the local variables are destroyed**.

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

Ans-A return is a value that a function returns to the calling script or function when it completes its task. A return value can be any one of the four variable types: handle, integer, object, or string. The type of value your function returns depends largely on the task it performs.

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

Ans-If no return statement appears in a function definition, control automatically returns to the calling function after the last statement of the called function is executed. In this case, the return value of the called function is **undefined**.

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

Ans-Normally, when you create a variable inside a function, that variable is local, and can only be used inside that function. To create a global variable inside a function, you can **use the global keyword**.

10. What is the data type of None?

Ans-The None keyword is used to define a null value, or no value at all. None is not the same as 0, False, or an empty string. **None is a data type of its own (NoneType)** and only None can be None.

11. What does the sentence import areallyourpetsnamederic do?

Ans- That import statement **imports a module named areallyourpetsnamederic**.

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

Ans-This function can be called with **spam.** **bacon()**.

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

Ans- Prevents program from crashing if an error occurs. If an error occurs in a program, we don't want the program to unexpectedly crash on the user. Instead, error handling can be used to notify the user of why the error occurred and gracefully exit the process that caused the error.

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

Ans-The try block lets you test a block of code for errors. The except block **lets you handle the error**.