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

ANS:

* Increase readability and maintainability. By breaking down a large program into smaller, more manageable functions, it becomes easier to read and understand the code. This makes it easier to find and fix bugs, and to make changes to the code without breaking other parts of the program.
* Promote code reuse. Once a function has been written and tested, it can be reused in other programs. This saves time and effort, and helps to ensure that the code is consistent across all of your programs.
* Improve performance. Functions can be optimized by the compiler to run more efficiently. This is because the compiler can analyze the function code and generate more efficient machine code.
* Reduce the risk of errors. By isolating complex code into functions, it becomes easier to test and debug the code. This helps to reduce the risk of errors in your programs.

Overall, functions are a powerful tool that can help you to write better, more reliable, and more efficient programs.

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

ANS: the function will not run until the function 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 a block of code that is given a name and can be executed repeatedly. A function call is the act of executing a function.

>To call a function, you use the function's name followed by parentheses

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. This is the scope that contains all of the variables that are defined at the top level of the program. Local scopes are created whenever a function is called. Each local scope contains all of the variables that are defined inside the function.

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

ANS: When a function call returns, the local variables in the function are destroyed. This means that the memory that was allocated for the variables is released, and the variables are no longer accessible.

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

ANS: In Python, a return value is the value that a function sends back to the caller when it completes its execution. It allows functions to provide output or results to the code that called them, which can then be stored in a variable or used for further processing

>A return statement is used to specify the return value in a function. When the function encounters a return statement, it immediately stops executing and returns the specified value.

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

ANS:   
If a function does not have a return statement, the return value of a call to that function is None.

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

ANS: To make a function variable refer to the global variable, you can use the global keyword. The global keyword tells the Python interpreter that the variable is defined in the global scope, so any changes to the variable will be reflected in the global scope.

10. What is the data type of None?

ANS: In Python, the data type of None is NoneType. None is a special object that represents the absence of a value. It is not the same as 0, False, or an empty string.

When a variable is assigned the value of None, it is said to be "None". This means that the variable does not currently contain any value.

11. What does the sentence import areallyourpetsnamederic do?

Ans:   
The sentence import areallyourpetsnamederic imports a module named areallyourpetsnamederic. A module is a collection of Python code that can be used in other Python programs.

The areallyourpetsnamederic module does not exist, so importing it will raise an error

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

ANS:   
If you had a bacon() feature in a spam module, you would call it spam.bacon() after importing spam.

For example, the following code imports the spam module and then calls the bacon() function:

import spam

bacon()

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

ANS: There are a few things you can do to save a program from crashing if it encounters an error.

1. Use exception handling. Exception handling is a way of gracefully handling errors in your program. When an error occurs, the program will jump to the exception handler, which will handle the error and allow the program to continue running.
2. Use try-catch blocks. Try-catch blocks are a way of specifying which parts of your program should be handled by exception handling. For example, the following code uses a try-catch block to handle the error that occurs when trying to divide by zero:

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

Ans:   
The try clause is used to execute a block of code that might raise an exception. If an exception is raised, the except clause will be executed.

The try and except clauses are used together to handle exceptions in Python. The try clause is used to specify the code that might raise an exception, and the except clause is used to specify the code that should be executed if an exception is raised.