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

**Solution:** Functions are advantageous to have in your programs because they allow you to reuse code and make your code more modular. This makes it easier to write, read, and maintain your code.

By using functions, you can break your code into smaller, self-contained pieces that each perform a specific task. This can make your code easier to understand and debug, because you can focus on one function at a time. Additionally, if you need to make a change to your code, you can often make the change in one function and have it propagate throughout your program, rather than having to search through all of your code to make the same change in multiple places

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

**Solution:** When you define a function, you are creating a block of code that has a name and can be executed by calling the function by its name. The function definition does not execute the code inside the function. Instead, it creates the function object that can be called later.

3. What statement creates a function?

**Solution:** In Python, the def statement is used to create a function.

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

**Solution:** A function is a block of code that is defined and given a name. It can accept input in the form of arguments, and it can return output in the form of a return value.

A function call is an expression that executes a function. When you call a function, you provide the function's name and any required arguments, and the function's code is executed.

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

**Solution:** In a Python program, there is only one global scope, but there can be multiple local scopes.

The global scope is the top-level scope in a Python program. It includes all variables that are defined at the top level of the program, as well as all functions and classes that are defined at the top level. These variables, functions, and classes are all available throughout the program, and they can be accessed from any function or class defined in the global scope.

Local scopes, on the other hand, are scopes that are defined inside a function or class. Each time a function or class is called or instantiated, a new local scope is created. Local scopes can access variables from the global scope, but they cannot modify them. Any variables that are defined within a local scope are only available within that scope, and they are not accessible from outside the function or class.

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

**Solution:** When a function call returns, the local scope that was created for that function call is destroyed, and any variables that were defined within that local scope are no longer accessible.

In Python, a new local scope is created every time a function is called. This local scope contains any variables that are defined within the function, as well as any arguments that are passed to the function. These variables are only accessible within the function, and they are not available outside of the function.

When the function returns, the local scope is destroyed, and any variables that were defined within that local scope are no longer accessible. This means that if you try to access a local variable after the function has returned, you will get an error.

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

**Solution:** A return value is the value that a function or method returns to the caller when it has completed its execution. In most programming languages, functions and methods can return a single value or no value at all.

An expression is a piece of code that produces a value. Some expressions return a value, while others do not. For example, in the C programming language, the expression x + y returns the sum of x and y, while the expression x = y does not return a value.

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

**Solution**: If a function does not have a return statement, or if the return statement is reached by falling through the end of the function, the return value of the function is undefined. In most programming languages, this means that the value of the function's return variable is not guaranteed to be valid or predictable, and using it in any way can lead to unpredictable behavior.

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

**Solution:** To make a function variable refer to a global variable, you can use the global keyword in the function. The global keyword is used to indicate that a variable is a global variable, rather than a local variable that is defined within the function.

10. What is the data type of None?

**Solution:** In Python, None is a special constant 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?

**Solution:** The sentence import areallyourpetsnamederic is a Python import statement that attempts to import the areallyourpetsnamederic module.

In Python, the import statement is used to import modules, which are packages of Python code that can be reused in other Python programs. The import statement allows you to use the code in the imported module by referring to the module's name.

If a module with the name areallyourpetsnamederic exists in the Python path, the import statement will succeed, and you will be able to use the code in the module. If the module does not exist, the import statement will raise an ImportError exception.

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

**Solution:**

import spam

spam.bacon()

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

**Solution**: There are several ways you can prevent a program from crashing when it encounters an error:

Use exception handling: Exception handling is a mechanism for handling runtime errors in a program. In most programming languages, you can use try-except blocks to catch exceptions that might be thrown by a piece of code, and take appropriate action to handle the exception. This can help to prevent the program from crashing when an error occurs.

Validate user input: User input is a common source of errors in programs. To prevent errors caused by invalid user input, you can validate the input before using it in your program. For example, you can check that a user-entered number is within a certain range, or that a string contains only valid characters.

Test your code: Testing your code thoroughly can help to identify and fix errors before they cause problems in the program. You can use unit tests, integration tests, and other testing techniques to ensure that your code is correct and robust.

Use debugging tools: Most programming environments include debugging tools that can help you find and fix errors in your code. These tools allow you to set breakpoints, inspect variables, and step through your code to find the cause of an error.

Use logging: Logging is a way to record messages about the execution of a program. By adding log statements to your code, you can track the flow of execution and identify where an error occurred. This can help you to understand the cause of an error and fix it.

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

**Solution:**The try clause is used to specify a block of code that should be executed to try to accomplish a task. The except clause is used to specify a block of code that should be executed if an exception is raised during the execution of the try block.