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

**Ans.** The first reason is reusability. Once a function is defined, it can be used over and over and over again. You can invoke the same function many times in your program, which saves you work. ... Another aspect of reusability is that a single function can be used in several different (and separate) programs

Functions reduce the need for duplicate code. This makes programs shorter, easier to read, and easier to update.

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

**Ans.** When it is called

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

**Ans.** The “def” keyword is a statement for defining a function in Python. You start a function with the def keyword, specify a name followed by a colon (:) sign. The “def” call creates the function object and assigns it to the name given. You can further re-assign the same function object to other names.

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

**Ans.** A function is procedure to achieve a particular result while function call is using this function to achieve that task

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

**Ans.** There are two types of variables: global variables and local variables. The scope of global variables is the entire program whereas the scope of local variable is limited to the function where it is defined.

A global variable is a variable that is accessible globally. A local variable is one that is only accessible to the current scope, such as temporary variables used in a single function definition

There's only one global Python scope per program execution. This scope remains in existence until the program terminates and all its names are forgotten

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

**Ans.** A local variable retains its value until the next time the function is called. A local variable becomes undefined after the function call completes. The local variable can be used outside the function any time after the function call completes.

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

**Ans.** A return statement is used to end the execution of the function call and “returns” the result (value of the expression following the return keyword) to the caller. The statements after the return statements are not executed. If the return statement is without any expression, then the special value None is returned.

Yes, it is possible that return values in expression.

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

**Ans.** None

**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.** NoneType

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

**Ans.**  Import function use for import module name areallyourpetsnamederic **(**areallyourpetsnamederic is not a real module in pyhton)

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

**Ans.** spam.bacon()

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

**Ans**. Place the line of code that might cause an error in a try clause.

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

**Ans**. The code that could potentially cause an error goes in the try clause. The code that executes if an error happens goes in the except clause.