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

A. 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?

A. The code in a function executives when the function is called, not when the function is defined.

3. What statement creates a function?

A. The def statement defines (that is, creates) a function.

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

A. A function consists of the def statement and the code in its def clause.  
  
A function call is what moves the program execution into the function, and the function call evaluates to the function's return value

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

A. There is one global scope, and a local scope is created whenever a function is called.

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. Otherwise, the next time you were to run the program, the names would remember their values from the previous run.

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.

**Variables that are defined inside a function body have a local scope, and those defined outside have a global scope**. This means that local variables can be accessed only inside the function in which they are declared, whereas global variables can be accessed throughout the program body by all functions

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

A. When a function returns, the local scope is destroyed, and all the variables in it are forgotten.

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

A. A retrun value is the alue that a function call evaluates to. Like any value, a return value can be used as part of an expression.

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

A. If there is no retrun statement for a function, its return value is None.

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

A. A global statement will force a variable in a function to refer to the global variable.

10. What is the data type of None?

A. The data type of None is NoneType

11. What does the sentence import areallyourpetsnamederic do?

A. That import statement **imports a module named areallyourpetsnamederic**. (This isn't a real Python module, by the way.)

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

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

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

A. When it encounters an error, the control is passed to the except block, skipping the code in between. As seen in the above code, we have moved our code inside a try and except statement. **Try running the program** and it should throw an error message instead of crashing the program

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

A. The try block lets you **test a block of code for errors**.

The except block **lets you handle the error**.