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

**Answer:** Functions have many advantages as compared to normal code:  
\* Reusable : Can be called multiple times at different instances  
\* Understanding: Makes it easier for the developers to understand the code  
\* Maintain: Easy to maintain and update in real time as per the requirements without touching the main code  
\* Testing: Can be used to test a specific part of code independently

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

**Answer:** The code in a function will run when it’s called

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

**Answer:** def is used to create a function   
syntax: def function\_name(parameters):

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

**Answer:** Function: This is defined using ‘def’ which is a reusable code to perform a specific task  
Function call: This is calling a function inside a codebase or a program. When called we use parentheses ‘()’ along with the function name.

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

**Answer:** Python program have just one global scope whereas multiple local scopes. As, each creation of a function creates a new local scope. Local scopes exist only when a function is called.

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

**Answer:** When a function call returns, the variables inside it are deleted from the memory and the memory is made available for other purposes.

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

**Answer:** The return statement is used to specify the value to be returned from a function. When all the operations are executed in a function, it provides the result back from where the function was called.  
Yes, we can have a return value in an expression as we can return the value of a function directly without assigning the value to a variable again.  
Ex: def add(x,y):  
 return x+y  
result=add(5,5)/2  
print (result)

Output: 5.0

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

**Answer:** If a function does not have a return statement, it returns None  
Ex: def add(x,y):   
 x+y  
result=add(5,5)  
print (result)

Output: None

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

**Answer:** In order for a function to refer to the global variable, we use the ‘global’ keyword.   
Ex: a = 10  
def global\_variable\_check():  
 global a  
 a = a+ 5  
print(a)  
global\_variable\_check()

Output: 15  
even the value of a will be 15  
When global\_variable\_check() is called the 2nd time  
Output: 20

**10. What is the data type of None?**

**Answer:** data type of None is ‘NoneType’

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

**Answer:** The above code will show an error as  
“No module named ‘**areallyourpetsnamederic**’  
As, the import statement is used to import modules or packages which have predefined functions.

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

**Answer:** import spam

spam.bacon()

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

**Answer:** In order to save a program from crashing due to an error, we can use error handling technique. This can be done using try, except block.   
Example: ‘ZeroDivisionError’,’ValueError’,’TypeError’

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

**Answer:** The code which could create an exception is written within the "try" clause. It allows us to define a block of code where exceptions can occur.   
Whereas the ‘except’ clause is used to define what action needs to be taken when an exception occurs within the ‘try’ block.