==================================================================================

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

* Functions make the code reusable
* Easy code maintenance
* Functions can be atomic – means, code may be for specific task only

==================================================================================

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

* Function code run when it is called only. Not when it is specified.

==================================================================================

3. What statement creates a function?

* Function is created using keyword “def”
* Common syntax:  
  def <function\_name>(<arguments?>) :  
   <function statements>

==================================================================================

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

* Function call is when the defined function is invoked
* Function is the declaring and define the function

==================================================================================

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

* There is only one global scope per program execution. It will remain until the program terminates
* There is only one local scope. It will remain until the function or that block executes.
* In general, there are four scopes: Local, Enclosing or non-local, Global, Built-in

==================================================================================

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

* When function call returns, local scope variables will get vanish.

==================================================================================

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

* Return value is any value that the function returns to the caller when the task is completed or as per function design
* Yes. Return value can be anything but the caller should be capable of receiving it.

==================================================================================

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

* **None** is the return value from function if there is not return value.

==================================================================================

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

* By using the **global** keyword, you can make the function variable refer to global variable

==================================================================================

10. What is the data type of None?

* **NoneType** is the data type of None

==================================================================================

11. What does the sentence import areallyourpetsnamederic do?

* This would import all the functions and classes available to the caller.

==================================================================================

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

* spam.py  
  def bacon():  
   pass
* test.py  
  import spam  
  // call the needed function using dot notation   
  spam.bacon()

==================================================================================

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

* Make use of Exception handling techniques to handle the error
* Keep the error suspect statements within **try** block and handle the errors if any in **except** block

==================================================================================

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

* **try**
  + error proving/suspecting statements were kept in try block
* **except**
  + used to manage or handle the errors occurred in try block
* Basically, try and except are used to manage the exceptions in turn ensures smooth the program execution

==================================================================================