1.Why are functions advantageous to have in your programs?  
functions are block of code that will be used repeatedly.

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

3. What statement creates a function?  
def func1():  
print(“I am function”)

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

5. How many global scopes are there in a Python program? How many local scopes?  
Only one global scope for program executed  
local scopes – unlimited / infinity

6. What happens to variables in a local scope when the function call returns?  
variables of local scope are gone as soon as function call returns

7. What is the concept of a return value? Is it possible to have a return value in an expression?  
return value carries the function’s outcome in a return object.  
return value can be used in an expression

8. If a function does not have a return statement, what is the return value of a call to that function?  
None

9. How do you make a function variable refer to the global variable?  
global variables are available and can be accessed in local scopes as is.  
x = 10  
def func1():  
print(x) # x can be accessed in local scope

10. What is the data type of None? NoneType

11. What does the sentence import areallyourpetsnamederic do?  
ModuleNotFoundError : No module named 'areallyourpetsnamederic'

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

13. What can you do to save a programme from crashing if it encounters an error?  
Put the code in try: Except block

14. What is the purpose of the try clause? What is the purpose of the except clause?  
try – put error prone code  
except – handle the error caused in associated try block