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

Answer: Functions in a program are advantageous by following reasons :

1. Consider a code will repeat number of times in a program, it is easy to write a function once and call functions wherever needed. This avoids writing code multiple times and also shortens the length of code.
2. Shortening the length of code makes code reading easy.
3. If there are changes in code required, it becomes easy to change code at a single place if used function, than to change it at multiple locations. This makes code manipulation easy during requirements and also saves time.
4. Even dividing a considerably long length into small sub sections becomes easy with use of functions.
5.

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

Answer: Code in a function runs when it is called and not when it is defined.

Code:

```
def  print_name():

   print(“name”)

print_name()  #function print_name is called here and code in function will execute after this call.
```

3. What statement creates a function?

Answer : Use of word def(definition) creates a function.

Code:

```
def  print_name():  #print_name function is created.

   print(“name”)

print_name()
```

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

Answer: Function is a block if which code is written for a particular required output, it is initiated or defined by ‘def’ statement.

Function call is when function is called during a program execution. Once a function call is done, program execution shifts itself to a function called, and code inside a function is executed.

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

Answer : Scope is the boundness or area in code upto to which a variable can be used. There is only one global scope in Python Program. Local scopes are scopes of variable which restricts itself in a particular loop, function ,etc.

Code :

```
a=2 # global scope, can be used anywhere in the program.

for i in range(2):

   b=2 #variable with local scope, restricted to this for loop.

  print(a+b)
```

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

Answer : Local scope in a function are restricted upto that function execution itself, once function call returns , use of variables is complete and variables gets forgotten.

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

Answer : Return value is like result or final evaluated output provided by function after execution. This value can be used in an expression in the code.

Code :

```
def  return_int():

   return 2 #return value form function

a=return_int()

sum= 5+ a #returned value used in expression

print(sum)
```

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, its return value of a call to that function is None.

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

Answer: Use of keyword 'global' before a variable in a function extends its scope to whole program like and as a global variable.

```
a=2 # global scope, can be used anywhere in the program.

def global_var():

  global string   #string variable now can be used throughout the program.

 String='globalvariable'
```

10. What is the data type of None?

Answer: Data type of None is NoneType.

11. What does the sentence import areallyourpetsnamederic do?

Answer: This sentence will import module areallyourpetsnamederic .

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

Answer: bacon() feature in a spam module, can be called after importing spam lke :

1. from spam import bacon as b #any name can be used instead of b also.
2. spam.bacon()

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

Answer: Use of a proper Exception Handling for the block of code that seems problem causing can be done to avoid programme from crashing if it encounters an error.

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

Answer: Block of code that seems to create a problem or error must be written in try clause. Except clause comes in picture if code in try blocks generates an error, so code in except clause is executed.