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

**Answer:** The following are the advantages,

1. *Code reusability* – a set of statements can be written once and used multiple times by just calling the function.
2. *Reduce Duplication* – since the code can be reused, thus it avoids duplicating the same code at various instances.
3. *Abstraction* – The information or the functionality inside the function can be hidden thus providing abstraction and the user can just call the function to perform the operation.
4. *Code Readability* – By breaking complex code into smaller functions, it improves the code readability and thus making the code easy to understand.

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:** When it is called.

--------------------------------------------------------------------------------------------------------------------------------------

3. What statement creates a function?

**Answer:** The keyword **def** is used to create a function, followed by the function name, then parenthesis and a colon.

**For Example: def function\_name():**

**//statements**

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:** A function is a set of codes that are instructed to perform a particular operation and it may or may not return a value as output. There are built-in functions as well as user defined functions.

For Example (user defined),

**def sum (num1, num2):**

**result = num1+num2**

**return result**

A function call is a statement that is used to invoke a function that has been defined already (with or without arguments based on the signature of the function)

For example,

Value = sum(7, 11)

The above statement sum(7,11) calls the ‘sum’ function and the result is stored in the variable ‘Value’

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:** There is one global scope and there can be any number of local scopes in python program.

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:** Variables in the local scope exists only when the function is executing and when the function ends, the local variables are destroyed and cannot be accessible outside the function.

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:** Return value is the value that is generated as an output to be returned after the function is executed.

Yes, it is possible to have return value in an expression.

For example,

**name = “Pradeep”**

**value = len(name) + 3**

**//in the above expression len(name) returns 7 and 3 is added and finally 10 is returned to the variable ‘value’**

--------------------------------------------------------------------------------------------------------------------------------------

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’*

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:** If the global variable is not modified, then it can be used directly.

For example,

gb\_variable = 7

def fun ():

lc\_variable = gb\_variable

print (f"Local variable = {lc\_variable}")

fun ()

>>> Local variable = 7

But, if the global variable must be modified, we need to use ‘global’ keyword.

For example,

gb\_variable = 7

def fun ():

global gb\_variable

gb\_variable = gb\_variable + 3

fun ()

print (f"Global variable = {gb\_variable}")

>>> Global variable = 10

--------------------------------------------------------------------------------------------------------------------------------------

10. What is the data type of None?

**Answer:** NoneType

--------------------------------------------------------------------------------------------------------------------------------------

11. What does the sentence import areallyourpetsnamederic do?

**Answer:** The above statement imports a module named areallyourpetsnamederic into our pyhton program

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:** If the spam module does not have any alias, the function can be called with the module name itself as spam.bacon(), else let say if the spam module has an alias as ‘sp’, then the function can be called as sp.bacon()

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:** Exception handling can be done to save the programme from unexpected crashing. Exception handling has several parts as follows:

try – contains the code that may cause unexpected crashing.

except – contains handling of exception code, executed if try block fails.

else – optional block that contains code to be executed only if there are no exceptions.

finally – optional block that contains code that will be executed always.

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:** Purpose of try clause – is to execute a set of codes that may or may not raise an exception or unexpected programme crashing or behaviour.

Purpose of except clause – is to handle these unexpected behaviours by using Exception classes. Expected or known exceptions can be handled inside respective classes explicitly and the unknown exceptions can be handled inside the class called BaseException which is the base class for all exceptions.

--------------------------------------------------------------------------------------------------------------------------------------