1)A)It helps to divide the large programs into small groups so that we can read the code, and debug the program faster and better.

—>Python Functions stop us from writing the same logic various times. We can bind the logic in one def and then call the same over and over.

—>Many persons can work on the same program by assigning different methods to each of them.

—>It encourages us to call the same method with different inputs over multiple times.

2)A)When a function is "called" the program "leaves" the current section of code and begins to execute the first line inside the function. Thus the function "flow of control" is:

—>The program comes to a line of code containing a "function call".

—>The program enters the function (starts at the first line in the function code).

—>All instructions inside of the function are executed from top to bottom.

—>The program leaves the function and goes back to where it started from.

—>Any data computed and RETURNED by the function is used in place of the function in the original line of code.

3)A)A function is a block of instructions that performs an action and, once defined, can be reused. Functions make code more modular, allowing you to use the same code over and over again.

—>Python has a number of built-in functions that you may be familiar with, including:

—>print() which will print an object to the terminal

—>int() which will convert a string or number data type to an integer data type

—>len() which returns the length of an object

—>Function names include parentheses and may include parameters.

4)A)A function is a block of code that does a particular operation and returns a result. It usually accepts inputs as parameters and returns a result. The parameters are not mandatory.

—>E.g:

Function add(a,b)

return a+ b

A function call is the code used to pass control to a function.

—>E.g:

b = add(5,6)

Now b will have the value 11.

5)A)variable is a label or a name given to a certain location in memory. This location holds the value you want your program to remember, for use later on. What's great in Python is that you do not have to explicitly state what the type of variable you want to define is - it can be of any type (string, integer, float, etc.). To create a new variable in Python, you simply use the assignment operator (=, a single equals sign) and assign the desired value to it.

Python has some rules that you must follow when creating a variable...

It may only contain letters (uppercase or lowercase), numbers or the underscore character \_.

It may not start with a number.

It may not be a keyword (you will learn about them later on).

6)A)A local variable retains its value until the next time the function is called

—>A local variable becomes undefined after the function call completes

—>The local variable can be used outside the function any time after the function call completes.

7)A)return statement is used to end the execution of the function call and “returns” the result (value of the expression following the return keyword) to the caller. The statements after the return statements are not executed. If the return statement is without any expression, then the special value None is returned.

Syntax:

def fun():

statements

.

.

return [expression]

8)A)In Python, we can return multiple values from a function. Following are different ways.

Using Object: This is similar to C/C++ and Java, we can create a class (in C, struct) to hold multiple values and return an object of the class.

Using Tuple: A Tuple is a comma separated sequence of items. It is created with or without (). Tuples are immutable. See this for details of tuple.

9)A)Global Variables

Variables that are created outside of a function (as in all of the examples above) are known as global variables.

Global variables can be used by everyone, both inside of functions and outside.

Local variables are those which are initialized inside a function and belongs only to that particular function. It cannot be accessed anywhere outside the function

10)A)The None keyword is used to define a null value, or no value at all.

None is not the same as 0, False, or an empty string. None is a data type of its own (NoneType) and only None can be None.

None is used to define a null value. It is not the same as an empty string, False, or a zero. It is a data type of the class NoneType object.

Assigning a value of None to a variable is one way to reset it to its original, empty state.

Example 1:

print(type(None))

Output

<class 'NoneType'>

11)A)It will import the libaray name areallyourpetsnamederic

12)A)This function can be called with spam.bacon().

13)A)Try running the program and it should throw an error message instead of crashing the program. The reason for the exception is also returned as an exception message. In the above code, we have handled the file reading inside an IOError exception handler

14)A)In the try clause, all statements are executed until an exception is encountered. except is used to catch and handle the exception(s) that are encountered in the try clause. else lets you code sections that should run only when no exceptions are encountered in the try clause.