1. Create an assert statement that throws an AssertionError if the variable spam is a negative integer.

Ans:

spam = -1

assert spam > 0, ‘only positive numbers are allowed’

2. Write an assert statement that triggers an AssertionError if the variables eggs and bacon contain strings that are the same as each other, even if their cases are different (that is, 'hello' and 'hello' are considered the same, and 'goodbye' and 'GOODbye' are also considered the same).

Ans:

Eggs = 'goodbye'

Bacon = 'Goodibye'

assert Eggs.lower() == Bacon.lower()

3. Create an assert statement that throws an AssertionError every time.

Ans:

a = 1

b = 2

assert a == b

4. What are the two lines that must be present in your software in order to call logging.debug()?

Ans:

import logging  
logging.basicConfig(level=logging.DEBUG, format=' %(asctime)s -%(levelname)s - %(message)s')

5. What are the two lines that your program must have in order to have logging.debug() send a logging message to a file named programLog.txt?

Ans:

import logging

logging.basicConfig(filename='programLog.txt', level=logging.DEBUG,

format=' %(asctime)s - %(levelname)s - %(message)s')

6. What are the five levels of logging?

Ans:

DEBUG, INFO, WARNING, ERROR, and CRITICAL

7. What line of code would you add to your software to disable all logging messages?

Ans:

Ans:

logging.disable(logging.CRITICAL)

8.Why is using logging messages better than using print() to display the same message?

Ans:

We can disable logging messages without removing the logging function calls. We can also selectively disable lower-level logging messages. We can create custom logging messages. Logging messages provides a timestamp.

9. What are the differences between the Step Over, Step In, and Step Out buttons in the debugger?

Ans: Step Into causes the debugger to go into the next function call and break there. Step Over causes the debugger to execute the next function and break afterwards. Step Out causes the debugger to finish the current function and break after it.

10.After you click Continue, when will the debugger stop ?

Ans:  Continue allows the program to move on to the next steps or we have to manually stop to end execution.

11. What is the concept of a breakpoint?

Ans: A breakpoint is an intentional stopping or pausing place in a program for debugging purposes