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

integer.

Answer:

assert(spam>=0, ‘The variable spam is a negative’)

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).

Answer:

assert(eggs.lower()!=bacon.lower(), ‘The eggs and bacon variable are the same’)

assert(eggs.upper()!=bacon.upper(), ‘The eggs and bacon variable are the same’)

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

Answer:

assert(False, ‘This assert statement always throws an AssertionError’)

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

Answer:

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?

Answer:

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?

Answer: The five levels of logging are: INFO, DEBUG, WARNING, ERROR and CRITICAL.

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

Answer: logging.disable(logging.CRITICAL)

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

Answer: Logging messages is always better than using print() to display the same message because we can disable the logging message without removing the logging function calls.

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

Answer:

Step Over Button: It will quickly execute the function call without stepping into it.

Step In Button: It will move the debugger into the function call.

Step Out Button: It will quickly execute the rest of the code until it steps out of the function it currently is in.

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

Answer: When we click continue, the debugger will stop when it reaches the end of the line.

11. What is the concept of a breakpoint?

Answer: It is a setting on a line of code that causes the debugger to pause when the program execution reaches the line.