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

Ans:

assert spam >= 0,“spam must be a non-negative integer”

1. 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:

assert eggs.lower() != bacon.lower(), eggs and bacon cannot be the same, even if their cases are different”

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

Ans:

assert False, "This assert statement always throws an AssertionError"

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

Ans:

import logging and logging.basicConfig(level=logging.DEBUG) are the two lines that must be present in your software in order to call logging.debug()

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

1. What are the five levels of logging?

Ans:

* Critical Logging
* Error Logging
* Warning Logging
* Info Logging
* Debug Logging

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

Ans:

logging.disable(logging.CRITICAL) is used to disable all logging messages.

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

Ans:

Using logging messages is better than using print() to display the same message because it provides more flexibility and control over the messages are displayed.

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

Ans:

The differences between the Step Over, Step In, and Step Out buttons in the debugger is,

* The Step Over button executes the current line of code and moves to the next line.
* The Step In button moves to the next line, but steps into a function if the current line contains a function call.
* The Step Out button continues executing code until it reaches the end of the current function and then returns to the line that called the function.

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

Ans:

After clicking Continue, the debugger will stop when it reaches a breakpoint or when the program terminates.

1. What is the concept of a breakpoint?

Ans:

A breakpoint is a point in the code where the debugger will pause the execution to allow inspect the state of the program. You can set a breakpoint at a specific line of code, and when the program reaches that line, the debugger will pause execution.