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

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).

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() !=

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

In Python, the assert statement is used to continue the execute if the given condition evaluates to True. If the assert condition evaluates to False, then it raises the AssertionError exception with the specified error message.

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

Debugging, in computer programming and engineering, is a multistep process that involves identifying a problem, isolating the source of the problem, and then either correcting the problem or determining a way to work around it. The final step of debugging is to test the correction or workaround and make sure it works.

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?

6. What are the five levels of logging?

Logging levels explained. The most common logging levels include FATAL, ERROR, WARN, INFO, DEBUG, TRACE, ALL, and OFF. Some of them are important, others less important, while others are meta-considerations.

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

logger = logging. getLogger('my-logger') logger. propagate = False # now if you use logger it will not log to console. This will prevent logging from being send to the upper logger that includes the console logging.

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

One of the advantages of using the logging module to track our codes is the ability to format the messages based on our needs. For example, in my code, I would like to log the date and time with appropriate messages.

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

If the current line contains a function call, Step Over runs the code and then suspends execution at the first line of code after the called function returns. Step Out continues running code and suspends execution when the current function returns. The debugger skips through the current function

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

Continue execution, only stop when a breakpoint is encountered. Set the next line that will be executed. Only available in the bottom-most frame. This lets you jump back and execute code again, or jump forward to skip code that you don't want to run.

In software development, a breakpoint is an intentional stopping or pausing place in a program, put in place for debugging purposes. It is also sometimes simply referred to as a pause.In software development, a breakpoint is an intentional stopping or pausing place in a program, put in place for debugging purposes. It is also sometimes simply referred to as a pause.In software development, a breakpoint is an intentional stopping or pausing place in a program, put in place for debugging purposes. It is also sometimes simply referred to as a pause.