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

Ans. assert spam<0, “Negative number is not 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 = ‘hello’

bacon = ‘hello’

assert eggs.lower() != bacon.lowe(), “eggs and bacon are not allowed to have the same string.”

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

Ans. assert False, “Error”

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

Ans. The first line is importing the logging module and the second one is calling the basicConfig method in order to call logging.debug().

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. The first line is importing the logging module and the second one is calling the basicConfig method in order to call logging.debug().

6. What are the five levels of logging?

Ans. Debug, Info, Warning, Error, Critical.

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

Ans. set the logging level greater than critical which is highest, this will not log any messages.

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

Ans. The logging package has a lot of useful features:

- Easy to see where and when (even what line no.) a logging call is being made from.
- You can log to files, sockets, pretty much anything, all at the same time.
- You can differentiate your logging based on severity.

Print doesn't have any of these.

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

Ans. step over will go on each line of our program, execute it and stop.

step in will go on each line of our code but it also go into the execution on in-built function

step out just come out of the execution of code.

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

Ans. debugger stops at the next breakpoint.

11. What is the concept of a breakpoint?

Ans. breakpoint is the line in our program where our execution of code stops and runs after we allow to run.