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

**assert spam < 0, “The spam variable can’t be 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).

**assert eggs.lower() != bacon.lower(), “Variables are the same!”**

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

**assert False, “this assert statement throws an AssertionError every time”**

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

**import logging**

**FORMAT = ’%(asctime)s %(name)s %(levelname)s %(message)s’**

**logging.basicConfig(level = logging.DEBUG, format= FORMAT)**

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?

**import logging**

**FORMAT = ’%(asctime)s %(name)s %(levelname)s %(message)s’**

**logging.basicConfig(level = logging.DEBUG, format= FORMAT, file=’programLog.txt’)**

6. What are the five levels of logging?

**INFO, DEBUG, WARNING, CRITICAL, ERROR**

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

**logging.disable(level=CRITICAL)**

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

**Using logging messages is better, because it has more functionality than print. It records all the events as well as other information like module name and line number.**

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

**Step Over advances the debugger over a line and if a line contains a function, it will execute it and return the result without debugging each line of a function. Step Into will enter the function and will do line-by-line debugging. Step Out will execute all remaining code in a function without going line-by-line and will exit it.**

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

**The debugger stops at the next breakpoint or the end of the program.**

11. What is the concept of a breakpoint?

**Breakpoint makes the debugger to stop a program at the line where it placed. It is used for debugging purposes to see where possible error occurs in the code.**