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

- After defining the variable spam…
- assert spam>=0

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).

- After defining the variables eggs and bacon
- assert eggs.lower() != bacon.lower()

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

- assert False

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

- import logging
- logging.basicConfig(filename = 'somefilename.log', level = logging.DEBUG, format = "%(name)s %(asctime)s %(message)s"
- The arguments can vary according to the requirements, but we need to import the logging module and set the basic configuration

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
- logging.basicConfig(filename = 'programLog.txt', level = logging.DEBUG, format = "%(asctime)s %(message)s"
- There can be some more arguments according to the context and requirements and the format can also be different.

6. What are the five levels of logging?

- Critical - 50, Error - 40, Warning - 30, Info - 20, Debug - 10

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

- logging.disable(logging.CRITICAL)

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

- Logging messages provide more information like custom formatted timestamps with less code
- You can save the logging messages in a file for future reference
- You can differentiate users/computers in the logging messages

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

- Step Over: Step over action will step over the given line of code, if the line of code is a function, the function will be executed and the result will be returned without debugging each line.
- Step In: If the step-in action contains a function, it will enter the called function and continue to debug each line.
- Step Out: Step out action will take in the debugger that returns to the line where the current function is called.

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

- When it reaches a breakpoint or by the end of the code

11. What is the concept of a breakpoint?

- Breakpoint is function and when it is called inside a program and when the program is executed, and when it hits the break point, it pauses immediately there and you can print the current values of whatever variables you want that are involved in the program.
- Breakpoint is a function, so it must be called
- Breakpoint only works in python 3.7 and newer versions
- In the older versions of python, you can use pdb.set_trace(), after importing pdb.