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

```
assert spam >= 0, 'The spam variable 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).

```
assert eggs.lower() != bacon.lower(), 'The eggs and bacon variables are the same!' or assert eggs.upper() != bacon.upper(), 'The eggs and bacon variables are the same!'
```

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

```
assert False, 'This assertion always triggers.'
```

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

```
import logging
logging.basicConfig(level=logging.DEBUG, format=' %(asctime)s -%(levelname)s - %(message)s')
```

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 - %(levelname)s - %(message)s')
```

6. What are the five levels of logging?

```
DEBUG, INFO, WARNING, ERROR, and CRITICAL
```

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?

```
You can disable logging messages without removing the logging function calls. You can selectively disable lower-level logging messages. You can create logging messages. Logging messages provides a timestamp.
```

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

STEP INTO:Step into is used for debugging the test steps line by line.Step into enables to get inside the procedure and debugs the procedure steps line by line when it is called.

STEP OVER: Step over will be enabled ,only after the debugging is started with Step into /Run to Step. It is used for debugging the test steps line by line ,but when the procedure is called, the entire procedure call will be executed as a single step. The cursor never gets inside the procedure.

STEP OUT:Step out will enable , only after the debugging cursor gets inside a procedure .The cursor will get inside a procedure only when the test is debugging with Step Into.

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

The debugger stops when a breakpoint is encountered .

11. What is the concept of a breakpoint?

A breakpoint is a setting on a line of code that causes the debugger to pause when the program execution reaches the line.