1 – spam = 0

**assert** spam < 0

2 –

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- assert False

4 –

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

5 –

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

6 –

DEBUG, INFO, WARNING, ERROR, and CRITICAL

7 - logging.disable(logging.CRITICAL)

8 - 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 –

Step Into will cause the debugger to go into the next function call and break there. Step Over will tell the debugger to execute the next function and break afterwards. Step Out **will tell the debugger to finish the current function and break after it.**

**10 – It** will stop your program at whatever line it has just executed

11 - a breakpoint is **an intentional stopping or pausing place in a program, put in place for debugging purposes**