1. assert spam >= 10, 'The spam variable is less than 10.'

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, 'This assertion always triggers.'

4. The 2 lines are:

import logging

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

5. The 2 lines are:

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

6. 5 levels are : critical log, warning log, info log, debug log, error log

7. logging.disable(logging.CRITICAL)

8. It is because logging gives us a flexibility to understand that what problem may have occurred within our code, why taking more time or why it may have crashed. The main purpose for using logging is to filter out the exact timestamp and duration and what is the cause of error or exception occurred. Out of a large bundle of code, we can’t identify error/exception using print(), so logging conveniently provides the same.

9. The Step button will move the debugger into a function call. The Over button will quickly execute the function call without stepping into it. The Out button will quickly execute the rest of the code until it steps out of the function it currently is in.

10. After Continue execution, debugger will only stop when a breakpoint is encountered.

11.  breakpoint is an intentional stopping or pausing place in a program, put in place for debugging purposes. It is also sometimes simply referred to as a pause.