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

try:

spam = int(input('enter int : '))

assert spam > 0, 'Assertain error happened'

except Exception as e:

print(e)

**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).**

try:

eggs = input('enter eggs : ')

bacon = input('enter bacon : ')

assert (eggs.lower()!=bacon.lower()), 'Assertain error happened'

except Exception as e:

print(e)

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

try:

a = input('enter something : ')

assert False,'Assertain error'

except Exception as e:

print(e)

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

import logging

logging.BasicConfig(filename=’xyz.log’, 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?**

Error, warning, info, debug, critical

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

Logging.disable()

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

Print msg will not be useful as in logging from anywhere and from anytime one can see status of running application as why error happened or one can debug it.

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

Step over will skip present line and move to the next line without executing current line of code. Step into will go inside code and show what is executing inside code. Step out will make step out side to the caller method

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

Until reaching next breakpoint

**11. What is the concept of a breakpoint?**

Breakpoint is point where execution of program stops and user can debug code at that location.