CS 686-02 - Lab 1 Spring, 2018

Lab 1 - Logger

The goal of this assignment is to practice the object-oriented portions of python.

Background

In some cases, you may have to write a subsystem to log the activities of a software system. For example java.util.logging is a java package for logging. You are provided with a partial implementation for a simple logger. Your task is to complete the implementation and ensure it runs and produces the required output.

Requirements

Perform the following steps:

1. Get the files in the CS 686-02 repository on github at https://github.com/dbrizan/cs686-2018-01; ideally, you should fork this to your own github repository.
2. Complete the implementation by adding two files: file_logger.py and stdout_logger.py.
    a. The stdout_logger.py instance should send output to the console.
    b. The file_logger.py instance should send output to a file. By default, this file should be named `file_log.txt`, but your implementation must allow for the user to provide another file name when file_logger is instantiated.
3. Check the correctness of your implementation. The output should look like the following:

```
0: Starting logger (type = stdout) at log level 0.
0: Ending logger.
0: Starting logger (type = stdout) at log level 1.
1: Important message.
0: Ending logger.
0: Starting logger (type = stdout) at log level 2.
1: Important message.
2: Less important message.
0: Ending logger.
0: Starting logger (type = stdout) at log level 3.
1: Important message.
2: Less important message.
3: Not important message.
0: Ending logger.
```

Submission

Submit your source code or link to your github repository on Canvas.

Grading

Your grade for this assignment will be based equally on the logger implementations as follows:

- 50% = Implementation works correctly.
- 75% = Implementation works but contains minor errors (eg. the log level does not work correctly).
- 50% = Implementation works but contains major errors (eg. does not log correctly.
- 0% = Implementation not attempted or not submitted on time.