

```
package edu.osu.cse421;

import java.util.Arrays;
import java.util.logging.Logger;
import java.util.logging.Handler;
import java.util.logging.Level;

/**
 * Utility class for diagnosing Logging configuration. The class provides
 * methods for displaying information relevant to the behavior of a logger.
 *
 * @author paolo
 */
public final class Loggers {

    /**
     * Private constructor to prevent this class from being instantiated.
     */
    private Loggers() {
    }

    /**
     * Calculates the levels of all the handlers in an array of handlers.
     *
     * @param h
     *         an array of handlers
     * @return x[] such that x.length = h.length for all i in 0 <= i < h.length,
     *         x[i] = level of h[i]
     */
    private static Level[] handlerLevels(final Handler[] h) {
        Level[] hlevels = new Level[h.length];
        for (int i = 0; i < h.length; i++) {
            hlevels[i] = h[i].getLevel();
        }
        return hlevels;
    }

    /**
     * Displays relevant information for a single logger.
     *
     * @param logger the logger to be displayed
     */
    public static void showLoggerInfo(final Logger logger) {
        if (logger.getParent() == null) {
            System.out.println("Root Logger");
        } else {
            System.out.println("Child Logger (Parent is \""
                + logger.getParent().getName() + "\"");
        }
        System.out.println("  Name:      " + "\"" + logger.getName() + "\"");
        System.out.println("  Level:    " + logger.getLevel());
        System.out.println("  Handlers: "
            + Arrays.toString(logger.getHandlers()));
        System.out.println("  Handler levels: "
            + Arrays.toString(handlerLevels(logger.getHandlers())));
    }

    /**
     * Displays relevant information for the hierarchy of loggers
     * above (by the parent relationship) of a single logger.
     *
     * @param logger the leaf of the path of loggers to be displayed
     */

```

```
public static void showLoggerTreeInfo(final Logger logger) {  
    if (logger.getParent() == null) {  
        showLoggerInfo(logger);  
    } else {  
        showLoggerTreeInfo(logger.getParent());  
        showLoggerInfo(logger);  
    }  
}
```