**SLF4J Logging Assignment Problems**

### Exercise 1: Basic Logging

**File:** 6. SL4J Logging exercises.pdf

**Problem:** Configure SLF4J with Logback and log a simple message at different levels.

**Code:**

import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class LoggingExample {  
 private static final Logger LOGGER = LoggerFactory.getLogger(LoggingExample.class);  
  
 public static void main(String[] args) {  
 LOGGER.info("This is an info message");  
 LOGGER.debug("This is a debug message");  
 LOGGER.warn("This is a warning message");  
 LOGGER.error("This is an error message");  
 }  
}

**Output:**

INFO - This is an info message  
DEBUG - This is a debug message  
WARN - This is a warning message  
ERROR - This is an error message

### Exercise 2: Logging in a Service Class

**File:** 6. SL4J Logging exercises.pdf

**Problem:** Use SLF4J in a service class and log method start and end.

**Code:**

import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class BankService {  
 private static final Logger LOGGER = LoggerFactory.getLogger(BankService.class);  
  
 public void transfer(String from, String to, double amount) {  
 LOGGER.info("Start transfer from {} to {}", from, to);  
 // Simulate transfer  
 LOGGER.debug("Transferring amount: {}", amount);  
 LOGGER.info("Transfer complete");  
 }  
  
 public static void main(String[] args) {  
 BankService service = new BankService();  
 service.transfer("Alice", "Bob", 500.0);  
 }  
}

**Output:**

INFO - Start transfer from Alice to Bob  
DEBUG - Transferring amount: 500.0  
INFO - Transfer complete

### Exercise 3: Log Exception

**File:** 6. SL4J Logging exercises.pdf

**Problem:** Log an exception with stack trace using SLF4J.

**Code:**

import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class ExceptionLogger {  
 private static final Logger LOGGER = LoggerFactory.getLogger(ExceptionLogger.class);  
  
 public static void main(String[] args) {  
 try {  
 int x = 5 / 0;  
 } catch (ArithmeticException e) {  
 LOGGER.error("An exception occurred", e);  
 }  
 }  
}

**Output:**

ERROR - An exception occurred  
java.lang.ArithmeticException: / by zero  
 at ExceptionLogger.main(ExceptionLogger.java:8)

### Exercise 4: Configure Logback XML

**File:** 6. SL4J Logging exercises.pdf

**Problem:** Create a logback.xml file to control logging output.

**File:** src/main/resources/logback.xml

<configuration>  
 <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">  
 <encoder>  
 <pattern>%d{HH:mm:ss.SSS} %-5level - %msg%n</pattern>  
 </encoder>  
 </appender>  
  
 <root level="debug">  
 <appender-ref ref="STDOUT" />  
 </root>  
</configuration>

**Effect:**

* All logs of level DEBUG and above are printed to console.
* Custom timestamped format used.