RIT | Golisano College of Computing and Information Sciences
School of Information

ISTE-722 Database Connectivity and Access
Practice Exercise 5 - Exception Handling

Assignment Purpose: Practice implementing exception handling.

In class we discussed the importance of trapping all exceptions to prevent any structural information from being exposed. This exercise will require you to handle problems such as malformed queries.

1. Create a data layer object class named "DLException" – a custom exception class that uses supplied error message(s) and optional information to log errors to a file.
    a. Do not provide a default constructor, as DLException requires information to log
    b. Provide a constructor that accepts a single parameter of type Exception
    c. Provide a constructor that accepts a parameter of type Exception and additional string values. You may do this with *one* of the following or another creative means:
        i. Using the programming language's optional parameters feature (String… values)
        ii. A subclass of Map, or single 2D arraylist/list
        iii. Two parallel arraylists/lists
    d. Provide a method named "writeLog" that writes out all available information, including a timestamp, to a text log file. DLException constructors generally call the writeLog() method. Create the log file in the current directory as the code. Do not hard code a directory path for the file.
2. Modify your classes to individually catch **all** possible exceptions and throw a DLException with an appropriate message and information available regarding the failure. Available information is logged, for you or someone else to later understand what caused the problem, debug it, modify the code so it doesn't happen again.
    a. Set message to DLException's to some innocuous string indicating failure. Something like but different than: "Unable to complete operation."
3. Write a main program to test connection and queries. The following produces 2 log entries.
    a. *Connection failure*: Test with a bad database, username, or password. DLException use is expected to write the log and report a message to the user
    b. *Connection success*: Successful connection tested. Dropbox with this good connection.
    c. *Fetch failure test*: query to the fetch() that adds a wrong projection column name, (eg: BadTestField) or a wrong WHERE field name of the equipment table in order to force an error and log file creation using the same fetch(). After logging, remove the error.
    d. *Fetch success test*: a good fetch() query to the equipment table that prints the requested information. This proves the connection and a good query to the equipment table will work

Dropbox is to include all code needed to successfully run this assignment, and the log file containing at least the two errors shown above. Make sure the log file ends with the two above errors being logged.