**Logging Best Practices**

Well designed and descriptive log messages are essential in **trouble-shooting** problems.

* Does the log entry contain the following information:
  + Date Time (military time)
  + Log Level
  + Program/Class Name
  + Thread Id
  + Message
* Are there appropriate logging messages for the following locations?
  + called services
  + FATAL or ERROR conditions that would require immediate action
* Have appropriate levels been assigned to all log statements?
* Is TRACE level used? It is not widely supported.
* Is the category name either the class name or the relevant special purpose category specified in the design document?
* Are message contents descriptive? Can you understand what it means without further research?
* Do the critical areas of code where errors are likely to occur have their entries and exits logged?
* Do every method entry and exit logged? This results in logs that are too cumbersome.
* Has the state of relevant objects been logged when a logging event occurs?
* Are objects logging state via a toString() method, rather than via manually concatenated strings?
* Are all caught exceptions logged?
* Are any caught exceptions swollen?
* Does the root cause exception is specified as a cause of the new exception when an exception is caught and another exception thrown?
* Can you find the log message in the source code?
* Is the code vulnerable for logging injections?
* Are the log files configured to be rolled?
* Is there sensitive information in the logs?