**Java Coding/Standards & Project Expectations**

Start with readme.txt documentation for project work.

Explain the complete life cycle of the project in detail.

Project structure & package naming should be uniform.

No wildcard imports.

Braces are used even when the body is empty or contains a single statement.

2 spaces indentation.

Avoid the use of + or += to concatenate strings.

Use java standards designed for that purposes such as String.format, StringBuilder, etc.

You can declare all the variables as private.

To access the object outside the class, you can then use the GET and SET method.

Avoid Using Inner Classes.

Column limit can be 80 or 120 characters.

No multiple return statements.

Overuse of arrays in place of collections.

Too much or no whitespace.

Creating Class/Methods - should follow the Single Responsibility Principle of SOLID Principles.

Always Access specifiers.

Prefer returning Empty Collections instead of Null.

Avoid unnecessary Objects.One of the most expensive operations (in terms of Memory Utilization) in Java is Object Creation.

Thus it is recommended that Objects should only be created or initialized if necessary.

Always release database connections when querying is complete.

Try to use Finally block as often possible.

Release instances stored in Static Tables.

Keep classes small.

Name methods carefully.

Use a consistent style.

Use built-in logging.

Packages are all lower case.

Coding Formatting/Indentation should be proper format.

Make proper comments( Javadoc is Must )

You need a static/finally block for connection/streaming.

Java Exception Propagation - Hierarchical should be captured properly and if needed, re-throw into appropriate level.

Naming Conventions. (Keep the coding standards high)

Project Build Tool : Maven is Mandatory.

Version control System like GIT/SVN/Perforce/Clearcase, Git is Mandatory other optionals.

Unit testing - e.g. Junit(Unit testing Mocking libraries like Mockito)