Coding Standards

A coding standards document's purpose is to make sure that all code is designed, written and laid out the same to make it easier for a developer to switch from one person work to another without the needed change of mentality to read someone else's style.

Naming Conventions
We have named our class with UpperCamelCase, methods with lowerCamelCase, package names are lowercase, constants are named UpperCase with an underscore. All the local variables and parameters are named with lowerCamelCase.

Indentation
To have a proper indentation structure, line wrapping and horizontal whitespaces are done according to the formatter available in the Eclipse tool. All the if, for and while loops are indented by giving single tab space based on the surrounding code.

Layout
We wanted to keep the layout simple and as a result, we have kept the braces on the same line as the opening method line.

Exception Handling
We have used try and catch block to handle the exceptions and in case of an exception, we have outputted meaningful sentences which can be identified by any programmer and help in fixing the same.

Commenting
We have standards to dictate grammatical correctness, and that comments begin on the line before, not on the same line, this increases readability.

A brief description is stated at the beginning of every class or method.
@param - is used to define the parameters used in the class or method.
@return - is used to return the value from a method.

We have written in-code comment lines using “//” and comment blocks using “/**”.

Exposure
We have implemented our code by providing the lowest level of access to all the member variables **private** and all the classes and methods are provided with **public** access.

Javadoc comments have been used extensively on our complete project by using special tags on classes, methods and member variables.