# Coding Standards

## Best Practices:

* BE OBJECTIVE
* Use try catch everywhere where it is needed or it would be appropriate
* In logging strings use variables stored in one place
* When using operations with Strings, be careful with case sensitivity - use lower case for functional Strings
* Methods and packages begin with lower case letter, classes with capital
* Interface implementation named: interfaceImpl (class and package)
* UML creating (flip chart first, later digital form)
* Code written in American English
* For new sprint create separate branch in git and merge only after local version has been tested and is running
* Override methods toString(), hashCode(),equals() in each class.
* Use code formatting(ctrl + shift + o, ctrl + shift + f) before every commit
* Define non-parametric constructor for each class

## Code Comment:

* Use comments for each method, but please no comments for comments
* Always comment purpose, function and usage, not type, name or parameters
* Try to write comments at first and then write a code
* Delete all old comments
* Comments written in American English
* Comments for methods are in interface and interfaceImpl

## Try To Avoid:

* Comments for comments
* Not needed getter, setter or empty constructor
  + If they are needed, put their purpose to comment
* Complicated and long if statements
  + Better to use nested statements
* Recurring code
  + Make a method for it
* Typo in method name
* Ifs where switch can be used

## Naming Convention:

* Logical methods naming and try to avoid types in names
* generate everything that IDE can generate