Scala Style Guide

# Avoid Casts and Type Tests

* NEVER use ***isInstanceOf*** or ***asInstanceOf*** – there is always a better solution
* AVOID using casts – take a step back and think what you want to achieve/rethink the code

# Indentation

* The standard is to indent using 2 SPACES
* NO tabs for indentation

# Line Length and Whitespace

* Make sure the lines are not too long(hard to read)
* Introduce some local value bindings

# Use Local Values to simplify complex Expressions

* Store some arguments in a local value before passing them to the function, when a combined expression grows too big

# Choose meaningful Names for Methods and Values

* The names of methods, fields and values should be carefully chosen so that the source code is easy to understand
* A method name should make it clear what the method does

# Common Subexpressions

* AVOID unnecessary invocations of computation-intensive methods
* OPTIMISE by introducing a local value binding

# Don’t Copy-Paste Code

* Copy-pasting code is always a warning for bad style
* There are many disadvantages:
* The code is longer and it takes more time to understand
* If two parts are not identical, but similar, it is difficult to spot the differences
* Maintaining two copies and making sure that they remain synchronized is very error-prone
* The amount of work required to make changes to the code is multiplied
* You SHOULD factor out common parts into separate methods instead of copying code

# Scala doesn’t require Semicolons

* Semicolons are required only when writing multiple statements on the same line
* AVOID writing unnecessary semicolons

# Avoid using Return

* You often don’t need to use explicit returns because control structures such as if are expressions

# Avoid mutable local Variables

* Purely functional style doesn’t usually have side-effecting operations
* OPTIMISE by rewriting the code using mutable local variables to code with helper functions that take accumulators(NO ***var***)

# Eliminate redundant *IF* expressions

* AVOID returning the Boolean value of an expression using ***If-Else***