General Coding Standards
For
Software Development

REVISION RECORD					
REV.	**DATE**	**ORIGINATOR**	**REVIEWED**	**APPROVED**	**DESCRIPTION**
00	10.2.2014	M.Hamlan	M.Hamlan M.Varias G.Lopez		General Coding Standards

Table of Contents

1. Introduction

The main goal of this document is to set coding habits. These habits will be the standards of each program coded by an individual member or a team of programmers. The intent of these standards is to create a frequent programming style along with consistency.

Using these habits will not limit the freedom to practice new styles by each member as long as other members are informed. Thus, this document is adjustable and a revision table is included.

2. Purpose

Common standards will help each team member in any initiated project to work on other members codes. Debugging will be much easier, in addition to readability. Also, these guidelines will be helpful for any new members joining the project in different stages. These standards will reduce the time and effort to learn task due to code addressing by programmers other than the author. This will ensure that the project can be maintained and enhanced by any programmer in the absence of author.

3. Scope

The standards which are going to be implemented in this document are applicable for high level languages such as Java. As per the team agreement, Java is going to be the language for any project that the team is given in this group. These standards will be the base document. Each team member will make sure, to understand, agree and adapt these standards and implement them.

4. Terms used in this document

Any term or terminology is used in this document, will be defined in this section as follows:
A “unit” is a partial code of the main program.
A “control unit” is a unit that is used to control or make a logical decision about what other units will execute (i.e. if statements, for loops…).
A “Method” is a program unit with implement a process and may return a value.
An “identifier” is the term which refers to the name of any variable, constant or program unit.
A “module” is the combination of units working on same domain.
An “argument” is the data passed into a Method when it is called.

5. Naming convention

Constant variables will have all letters capitalized. If there are multiple words in the name they will be separated with an underscore (ex. CONSTANT_NAME). Class variables and method names will start with a lower case letter and if there are multiple words in ether, the following words will start with an upper case letter and no space in between(ex. variableName). Class name will have each word uppercase with no spaces (ex. ClassName). All variable names will have meaning to them with the exception of loop control variables.

6. Comments

As known, comments are used to give overview of code and to provide additional information that is not readily available or not obvious at first glance in any code. Both implantation and documentation comments will be used in codes. Therefore, Java Doc will be used as our commenting standard. This allows us to use a universally known method that is often supported with other features such as the ability to generate an API. We will also use comments not specific to Java Doc in order to make the code a little easier to read. We will put small comments around above or next to major control units and to signify the end of methods.

7. Other

For general read-ability, we will keep standards that do not change the actual code but how it is presented. When applicable, we will have our main method be the first method in our java files. For all uses of brackets they will have their own line with the exception of closing a Do/While statement. Units that are managed by the same control unit will be on the same indentation level which is one more that the control unit. We will have

an empty line between the end of one method and the comments of the next. If able we will not use brackets to encapsulate small control statements such as if or while statements, instead, we will just use indentation.