**Coding Standard**

|  |  |
| --- | --- |
| Purpose | To guide the development of programs in Java |
| Program Headers | Import of external libraries |
| Header Format | Package NameOfPackage;  import NameOfLibraryPackage.NameOfLibrary ; |
| Listing Contents | Provide a summary of the listing contents. |
| Reuse Instructions | /\*  Objectives or returns  Param1: Definition  ParamN; Definition  \*/ |
| Reuse Example | /\*  Return the sum between param1 and param2  Param1: Numeric value between 0 and 9  Param 2: Numeric value between 0 and 9  \*/ |
| Identifiers | Use descriptive names for all variables less the counter to manage loops, function names, constants, and other identifiers. Avoid abbreviations or single letter variables. Use **PascalCase** in multiword names when is Public and use underscore before a name with **camelCase** when the statement is Private. |
| Identifier Example | Public int QuantityOfStudents; // Good  Private int QuantityOfStudents; // Bad  Private int \_quantityOfStudents; // Good  Public int \_summation; //Bad  Private int \_Summation; //Bad  Public int QttyStuds; // Bad |

(continued)**Coding Standard (continued)**

|  |  |
| --- | --- |
| Comments | Document only when the piece of code is complex and put this in the reuse section in a new line like a Note. |
| Good Comment | /\*  Note: This function include a validation when a Param is a negative value.  Return the sum between param1 and param2  Param1: Numeric value between 0 and 9  Param 2: Numeric value between 0 and 9  \*/ |
| Bad Comment | /\*  Note: This function is used to sum.  Return the sum between param1 and param2  Param1: Numeric value between 0 and 9  Param 2: Numeric value between 0 and 9  \*/ |
| Blank Spaces | * Separate every function with at least one space. * Leave a space between variables declaration and the code |
| Indenting | * Indent every level of brace from the previous one. * The content of braces must be more indent. |
| Indenting  Example | Pubic void SayHello(){  System.out.println(“Hello!”);  } |
| Capitalization | * Capitalize only Constants |
| Capitalization Example | Public/Private static final CONSTANT\_EXAMPLE = 5; |