Coding Standard

Purpose	To guide implementation of Java programs
Program Headers	Begin all programs with a descriptive header.
Header Format	/** * **@author** Developer Name * **@version** x.x.x * **@description** brief description of what do the class? */
Listing Contents	Provide a summary of the listing contents
Contents Example	/** * **@attributes** * attribute1 * attribute2 * **@methods** * method 1 * method 2 */
Identifiers	Use descriptive names for all variable, function names, constants, and other identifiers. Avoid abbreviations or single-letter variables.
Identifier Example	**GOOD** **Attributes** CONSTANT_OF_ANY int linesCount; String username; **Variables** int i, j, k; **for loops** int count, size; **method variables** **Methods** getPersonCount (), calculateValue(); **BAD** **Attributes** int I, vf, LinesCount; String Str1; **Variables** Float: x4, j, ftave; int Count, SZ; **method variables** **Methods** GetPersonCount (), GetABC();
Comments	- Document the code so the reader can understand its operation. - Comments should explain both the purpose and behavior of the code. - Comment variable declarations to indicate their purpose.

Good Comment	**Line Comment**
	If(record_count > limit) //have all records been processed?
	Block Comment
	/**
	* this method make this task
	* **@param param1** description
	* **@param param2** description
	* **@return** description
	*/
	int method(param1, param2)

Bad Comment	**Line Comment**
	If(record_count > limit) // if greater than limit amount
	Block Comment
	/**
	* **@param param1**

	* **@param param2** * **@return** */ int method(param1, param2)
Major Sections	Precede major program sections by a block comment that describes the processing done in the next section.
Example	Comments for this part will be used for each method
Blank Spaces	- Write programs with sufficient spacing so they do not appear crowded. - Separate every program construct with at least one space.

Blank Spaces Example	**GOOD**
	amount = count * 25;
	BAD
	amount=count*25;
	amount = count * 25 ;

Indenting	- Indent each brace level from the preceding level. - Open and close braces should be on lines by themselves and aligned.

Indenting Example	**GOOD**
	for (Double dato : datos) {
	suma += dato;
	}
	BAD
	for (Double dato : datos) {suma += dato; }
	
	for (Double dato : datos)
	{
	suma += dato;
	}

Capitalization	- Capitalize all defines. - Lowercase all other identifiers and reserved words. - To make them readable, user messages may use mixed case.
Capitalization Examples	**Constants** CONSTANT_GOOD **Variables / attributes** personName, phoneNumberLarge; **Class** Statistics, Files, SoftwareTest
Class Structure	**Imports** **Header** **List Contents** **Class declaration** **Constants of class** **Attributes of class** **Constructors** **Public static methods** **Public methods** **Private methods**