## Source files

1. The source file name consists of the case-sensitive name of the top-level class it contains plus the .java extension.

## Source file structure

1. A source file consists of, **in order**:

* License or copyright information, if present
* Package statement
* Import statements
* Exactly one top-level class

1. Each top-level class resides in a source file of its own.
2. Methods of a class that share the same name appear in a single contiguous group with no other members in between. The same applies to multiple constructors (which always have the same name). This rule applies even when modifiers such as static or private differ between the methods.

## Formatting

1. Braces are used with if, else, for, do and while statements, even when the body is empty or contains only a single statement.
2. Each statement is followed by a line break.
3. Java code has a column limit of 100 characters. A "character" means any Unicode code point.
4. After each comma that follows an enum constant, a line break is optional. Additional blank lines (usually just one) are also allowed.
5. Optional grouping parentheses are omitted only when author and reviewer agree that there is no reasonable chance the code will be misinterpreted without them, nor would they have made the code easier to read. It is not reasonable to assume that every reader has the entire Java operator precedence table memorized.
6. Every variable declaration (field or local) declares only one variable.
7. Local variables are **not** habitually declared at the start of their containing block or block-like construct. Instead, local variables are declared close to the point they are first used (within reason), to minimize their scope. Local variable declarations typically have initializers, or are initialized immediately after declaration.
8. No C-style array declarations
9. Any array initializer may optionally be formatted as if it were a "block-like construct."
10. Annotations applying to a class appear immediately after the documentation block, and each annotation is listed on a line of its own (that is, one annotation per line).
11. long-valued integer literals use an uppercase L suffix, never lowercase (to avoid confusion with the digit 1)

## Naming

1. Identifiers use only ASCII letters and digits
2. Package names use only lowercase letters and digits (no underscores). Consecutive words are simply concatenated together.
3. Class names are written in [UpperCamelCase](https://google.github.io/styleguide/javaguide.html#s5.3-camel-case).
4. Method names are written in lowerCamelCase
5. Constant names are written in UPPER\_SNAKE\_CASE
6. Non-constant field names use lowerCamelCase
7. Parameter names are written in lowerCamelCase
8. Local variables use lowerCamelCase

## Programming Practices

1. A method is marked with the @Override annotation whenever it is legal.
2. When a reference to a static class member must be qualified, it is qualified with that class's name, not with a reference or expression of that class's type.
3. Finalizers are not used

## Javadoc

1. Used basic formatting of Javadoc
2. At the minimum, Javadoc is present for every public class, and every public or protected member of such a class, with a few exceptions noted below.
3. Javadoc is optional for "simple, obvious" members like getFoo(), in cases where there really and truly is nothing else worthwhile to say but "Returns the foo”.
4. Javadoc is not always present on a method that overrides a supertype method.