## Naming

**Package names**

Package names are all lowercase, with consecutive words simply concatenated together (no underscores). For example, com.example.deepspace, not com.example.deepSpace or com.example.deep\_space.

**Class names**

Class names are written in UpperCamelCase.

Test classes are named starting with the name of the class they are testing, and ending with Test. For example, HashTest or HashIntegrationTest.

**Method names**

Method names are written in lowerCamelCase.

Method names are typically verbs or verb phrases. For example, sendMessage or stop.

**Constant names**

Constant names use CONSTANT\_CASE: all uppercase letters, with words separated by underscores.

**Non-constant field names**

Non-constant field names (static or otherwise) are written in lowerCamelCase.

These names are typically nouns or noun phrases. For example, computedValues or index.

**Parameter names**

Parameter names are written in lowerCamelCase.

One-character parameter names should be avoided.

**Local variable names**

Local variable names are written in lowerCamelCase.

One-character names should be avoided, except for looping variables.

**Variable declarations**

**One variable per declaration**

Every variable declaration (field or local) declares only one variable: declarations such as int a, b; are not used.

**No C-style array declarations**

The square brackets form a part of the type, not the variable: String[] args, not String args[].

**Programming practices**

**Overloads: never split**

When a class has multiple constructors, or multiple methods with the same name, these appear sequentially, with no intervening members.

**Empty blocks**

An empty block or block-like construct may be closed immediately after it is opened, with no characters or line break in between ({}).

Example:

void doNothing() {}

### Braces

Braces are used with if, else, for, do and while statements, even when the body is empty or contains only a single statement.

**Switch statements**

**Fall-through: commented**

This special comment is not required in the last statement group of the switch block. Example:

switch (input) {

case 1:

case 2:

prepareOneOrTwo();

// fall through

case 3:

handleOneTwoOrThree();

break;

default:

handleLargeNumber(input);

}

##### 

##### **The default case is present**

Each switch statement includes a default statement group, even if it contains no code.

**@Override: always used**

If you override some method write like this:

@Override public void method() {}

**Do not ignore exeption**

If you do this write a comment. Example:

try {

int i = Integer.parseInt(response);

return handleNumericResponse(i);

} catch (NumberFormatException ok) {

// it's not numeric; that's fine, just continue

}

return handleTextResponse(response);

### 

**Static members: qualified using class**

When a reference to a static class member must be qualified, it is qualified with that class's name, not with a reference of that class's type.

Foo aFoo = ...;

Foo.aStaticMethod(); // good

aFoo.aStaticMethod(); // bad