

Naming Conventions

Classes and Interfaces :

- Class names should be nouns, in mixed case with the first letter of each internal word capitalised. Interfaces name should also be capitalised just like class names.
- Use whole words and must avoid acronyms and abbreviations.

Methods :

- Methods should be verbs, in mixed case with the first letter lowercase and with the first letter of each internal word capitalised.

Variables : Variable names should be short yet meaningful.

- Should not start with underscore('_') or dollar sign '\$' characters.
- Should be mnemonic i.e, designed to indicate to the casual observer the intent of its use.
- One-character variable names should be avoided except for temporary variables.
- Common names for temporary variables are i, j, k, m, and n for integers; c, d, and e for characters.

Constant variables:

- Should be all uppercase with words separated by underscores (" _").
- There are various constants used in predefined classes like Float, Long, String etc.

Packages:

- The prefix of a unique package name is always written in all-lowercase ASCII letters and should be one of the top-level domain names, like com, edu, gov, mil, net, org.
- Subsequent components of the package name vary according to an organisation's own internal naming conventions.

2.12 The terms get/set must be used where an attribute is accessed directly.

```
employee.getName();
```

```
employee.setName(name);
```

```
matrix.getElement(2, 4);
```

```
matrix.setElement(2, 4, value);
```

Common practice in the Java community and the convention used by Oracle for the Java core packages.

2.13 is prefix should be used for boolean variables and methods.

`isSet, isVisible, isFinished, isFound, isOpen`

This is the naming convention for boolean methods and variables used by Oracle for the Java core packages.

Using the is prefix solves a common problem of choosing bad boolean names like status or flag. `isStatus` or `isFlag` simply doesn't fit, and the programmer is forced to choose more meaningful names.

Setter methods for boolean variables must have set prefix as in:

`void setFound(boolean isFound);`

There are a few alternatives to the is prefix that fits better in some situations. These are has, can and should prefixes:

`boolean hasLicense();`
`boolean canEvaluate();`
`boolean shouldAbort = false;`

File name:

The source file name consists of the case-sensitive name of the top-level class it contains (of which there is exactly one), plus the .java extension

Whitespace characters:

Aside from the line terminator sequence, the ASCII horizontal space character (0x20) is the only whitespace character that appears anywhere in a source file. This implies that:

- All other whitespace characters in string and character literals are escaped.
- Tab characters are not used for indentation.