

# Week 3: Coding Standards

Tags: W3 coding standard

## IMPORTANT: update on coding standard

Note in particular:

- The opening `/**` on a separate line
- **Write the first sentence as a short summary of the method**, as Javadoc automatically places it in the method summary table (and index).
  - In method header comments, the first sentence should start in the form `Returns ...`, `Sends ...`, `Adds ...` (not `Return` or `Returning` etc.)
- Subsequent `*` is aligned with the first one
- Space after each `*`
- Empty line between description and parameter section
- Punctuation behind each parameter description
- No blank line between the documentation block and the method/class
- `@return` can be omitted if the method does not return anything
- When writing Javadocs for overridden methods, the `@inheritDoc` tag can be used to reduce duplication.

New update: please take note!

Javadoc of class members can be specified on a single line as follows:

## Q&As

**Q: (Line 7) Must we give an access modifier?**

```
List<String> pastDescription = new ArrayList<>(); // a list of past descriptions
```

**A:** The coding standard does not require it. In any case, not having an access modifier is a legit choice in Java (it is called *private package* access). So, if that is the appropriate access level, it is fine to not have an access modifier.

**Q: (Line 7) Is having comment at the end of the line ok?**

```
List<String> pastDescription = new ArrayList<>(); // a list of past descriptions
```

**A:** It is fine to have a comment at the end of the line. But when the comment is put above a statement, it should be indented to match the statement. The coding standard disallows comments with mismatching indentation, but does not mention/disallow end-of-line comment.

**Q: (Line 22) Can we omit javadoc for overridden methods?**

```
@Override  
public String toString() { return descriptionPrefix + description; }
```

**A:** No. The ‘*must have x*’ rule applies for all cases except for those cases that are explicitly mentioned as ‘*can omit for cases...*’ Java has a way to specify that a method inherits the comment from the parent class (`@inheritDoc`), which means there is no need to duplicate the comment in all child classes. (More info here: [#44 \(https://github.com/nus-cs2103-AY2223S1/forum/issues/44#issuecomment-1221465365\)](https://github.com/nus-cs2103-AY2223S1/forum/issues/44#issuecomment-1221465365))

## Noteworthy

- Some IDEs will use wildcard import if there are more than a certain number of classes from the same package. This is against our coding standard.
- Although coding standard doesn't specifically say parameter names should not be single letters, it mentions that scratch variables can be single letters. It is OK if someone interprets this as other names should not be single letters.

```
public Task(String d) {  
    ...  
}
```

- The coding standard says 'this.' should not be used unless necessary. But that is an advanced rule (optional to follow). Only basic and intermediate level guidelines are applicable to the module.

```
this.important = true
```

- The style in DESCRIPTION\_PREFIX (i.e. all capitals, separated by underscores) is sometimes called *screaming snake* style