This is a coding book for labelling commenting guidelines.

This coding book includes two parts: ***an operating manual*** that explains what materials we have for coding, and what the steps are for labelling the materials; ***a CodesBook*** that explains what codes we have, and how the codes are defined.

### \*\*\*\*Operating manual:

**We have already prepared materials for coding in** guidelines\_thematic\_analysis\_results.csv, which contains commenting guidelines of 600 GitHub open source repositories that we retrieved.

In the sheet, each line represents a repository. Column b records links containing guidelines, and Column c contains the guideline rules that may be related to commenting practice (we retrieved from contributing.md, web pages, et al.).

**The steps of labeling guidelines are as follows.**

Step1: Get familiar with all the coding materials:

Read (skim over) all the commenting rules in Column c of guidelines\_thematic\_analysis\_results.csv;

Read the CodesBook explained below, which describes the codes we define and ready for labelling. In the CodesBook each line represents a code, “Definition” explains what the code is, and “codename” is the term used for labelling the materials.

\*Note there are different tiers of codes, each tier relies on its former tier, e.g., you need to have first, second tier codes, to get to the third tier codes. The first tier is “primary code”, the second tier is “secondary code”, et al.\*

Step2: Conducting coding:

For each line in [sampled\_repos - Google Sheets](https://docs.google.com/spreadsheets/d/1wac1FPLlVDTEHtYuB6GwM7NffSZspWF2of-TO99XfD8/edit#gid=1003037368), read commenting rules of each repository in column c to understand these rules, label the rules with matching codes (i.e., codenames defined in the following CodesBook), and leave the codes in a new blank column. Note that one repository may have multiple labels or no label at all.

### \*\*\*CodesBook:

**Codename = how to label each example in the sheets i.e.**

**repo … what->general\_rules**

| **Primary Codes** | **Codename** | **Definition** |
| --- | --- | --- |
| Where to comment | where | Which components in code files needs comments |
| What to comment | what | What should be written in comments |
| Other | other | Readability and maintenance of comments |

| **Secondary Codes** | **Codename** | **Definition** |
| --- | --- | --- |
| General rules | where->general\_rules | General rules that can be applied to all repositories |
| Project specific rules | where->project\_rules | Rules that related to some project specific components |
| Required content | what->required | What is required to be written in the comments |
| Forbidden content | what->forbidden | What is forbidden to be written in comments |
| Readability | other->readability | Rules that make comments easier to be read |
| Maintenance | other->maintenance | Comments should be updated timely with corresponding code changes |

| **Tertiary Codes** | **Codename** | **Definition** |
| --- | --- | --- |
| Explicit | where->general\_rules->explicit | Clear rules on which components need comments that have no misunderstanding, e.g., all public methods |
| Implicit | where->general\_rules->implicit | Vague rules on which components need comments that different developers may hold a different opinion, e.g., non-trivial methods |
| All android annotations | where->project\_rules->android\_javadoc | All android annotations must have Javadoc |
| All assertion methods | where->project\_rules->assertion\_javadoc | All assertion methods must have Javadoc |
| Functionality | what->required->functionality | What does the code do should be commented |
| Usage | what->required->usage | Information that how to call specific method should be commented, including information of usage, parameters, return value |
| Code example | what->required->code\_example | Related code examples should be commented |
| Limitation | what->required->limitation | The deficiency that other developers may not realize, e.g., if your class is thread safe and if your method relies on a specific version of the dependency |
| Author info | what->required->author\_info | author information should be commented, including @author tags |
| Creation date | what->required->creation\_date | When a method is created should be commented, including @since tags |
| Purpose | what->required->purpose | Why classes and methods are introduced and the meaning of classes should be commented |
| Relationship with other components | what->required->relationship | The relationship of other components should be commented, e.g., alternative for deprecated method |
| TODO comments | what->required->TODO | TODO comments should company with a specific date or a specific event or an issue |
| {@code} | what->required->@code | Use {@code} to wrap code statements or values such as null |
| @link | what->required->@link | Use the fully qualified name for {@link} elements |
| Test context | what->required->test | Preconditions, actions and expectations of the test |
| Author tag | what->forbidden->author\_tag | @author is forbidden in Javadoc |
| Since tag | what->forbidden->since\_tag | @since is forbidden in Javadoc |
| Commented code | what->forbidden->commented\_code | Commented code is forbidden to commit |
| TODO comments | what->forbidden->TODO | TODO comments are forbidden to commit |
| Space indent | other->readability->space\_indent | Rules that specify the indent in comments |
| Language style | other->readability->language | Use imperative style or third person descriptive verb in comments |
| The order of tags | other->readability->order\_of\_tags | The order of tags in Javadocs should be fixed |
| Use English | other->readability->english | Using English when writing comments |
| Updating comments | other->maintenance->update | Updating relevant comments in the same commit |

| **Quaternary Codes** | **Codename** | **Definition** |
| --- | --- | --- |
| All methods | where->general\_rules->explicit->all\_methods | All methods or all public methods should be commented |
| All classes, interfaces, enum | where->general\_rules->explicit->all\_classes | All classes, interfaces, enum should be commented |
| Test cases | where->general\_rules->explicit->test\_cases | All test cases should be commented |
| Member fields | where->general\_rules->explicit->member\_fields | All (public) member fields should be commented |
| No comments in overriding methods | where->general\_rules->explicit->none\_in\_overriding\_methods | Overriding methods should not be commented |
| No comments in test cases | where->general\_rules->explicit->none\_in\_test\_cases | Test cases should not be commented |
| Complex code | where->general\_rules->implicit->complex\_code | Complex code should be commented on. It can refer to method level or inline level |
| Nontrivial public method | where->general\_rules->implicit->nontrivial\_public\_method | Nontrivial methods should be commented on |