**Example Java Coding Standard**

|  |  |
| --- | --- |
| Purpose | To guide implementation of Java programs |
| Classes Headers | Begin all programs with a descriptive header. |
| Header Format | / \*\*class begin  \*##class name  \* Program Assignment: the program number  \* Description: a short description of the program  \*/  //class end |
| Reuse Instructions | * Describe how the program is used: declaration format, parameter values, types, and formats. * Provide warnings of illegal values, overflow conditions, or other conditions that could potentially result in improper operation. |
| Reuse Instruction Example | / \* method begin  \* #method name  \* @param student  \* @return  \*/  //method end |
| Identifiers | Use descriptive names for all variable, function names, constants, and other identifiers. Avoid abbreviations or single-letter variables. |
| Identifier Example | int student = 0; // This is GOOD  float x4, j, ftave; // This is BAD |
| Comments | * Document the code so the reader can understand its operation. * Comments should explain both the purpose and behavior of the code. * Comment variable declarations to indicate their purpose. |
| Good Comment | if (record\_count > limit) //have all records been processed? |
| Bad Comment | if (record\_count > limit) //check if record count exceeds limit |
| Blank Spaces | * Write programs with sufficient spacing so they do not appear crowded. * Separate every program construct with at least one space. |
| Indenting | * Indent each brace level from the preceding level. * Open and close braces should be on lines by themselves and aligned. |
| Indenting Example | while (true){  system.out.println(“Hello World!”);  } |
| Capitalization | * Capitalize all finals. * Use Camel-case formula to name all other identifiers. * To make them readable, user messages may use mixed case. |
| Capitalization Examples | final int DEFAULT\_NUMBER\_OF\_STUDENTS = 15；  int classSize = DEFAULT\_NUMBER\_OF\_STUDENTS； |