Java Program Style Guide

This guide summarizes the conventions associated with good programming style. All programming assignments should adhere to these guidelines.

Identifier: Naming

1. Use descriptive names, avoiding single character names.
2. Make identifiers easy to read (e.g., use currentValue instead of curval).
3. Keep identifiers to a reasonably short length.
4. Use the underscore character to separate words in a constant.
5. Use UPPERCASE for constant identifiers.

White Space

1. Indent all statements within a code block an equal amount (2 to 5 spaces).
2. If the loop body, `if` statement, or `else` clause is a single statement, embed that statement in a code block (using curly braces).
3. Put the opening left brace ({) starting a new code block on a new line immediately below the start of the previous line. Align the closing right brace (}) with the opening brace and on a line of its own.
4. Use ctrl–shift–f to reformat the java file
5. Separate code blocks with blank lines.
6. Lines should be no longer than a maximum of 80 characters. Press "Enter" at the end of a line to force a new line; don't use tab or spaces to move the cursor down.

Messages & Prompts

1. Always inform the user of the program title or purpose.
2. Always inform the user of the purpose and type of input required.
3. Do not condescend.
4. Do not attempt to be humorous.
5. Be informative, but succinct.
6. Define specific input options in prompts when appropriate.
7. Include one or two spaces at the end of a prompt, then leave the cursor on the same line as the prompt (for visibility).
8. Messages and prompts should be nicely formatted.

Output

1. Label all output clearly and well formatted.
2. Provide information to the user in a consistent manner.
3. End the last line of output (`println`).

Header & Block Comments

1. Every source file should contain a header comment neatly specifying the program name, the assignment, its purpose, and the author and his/her user ID.
2. Each block comment should have a distinct delimiter on the top and bottom, such as the following:

```
/*******************************************************************************
*  Filename:   delinquencies.java
*  Assignment: lab 10, exercise #2
*  Author:     N. Van Cleave (nkvancleave)
*  Purpose:    Print a report of delinquent customers based on a data file
*              specified by the user and an overdue bill of more than
*              five days.
*              Each line of the file must conform to the following format:
*                  Field 1: integer, customer ID number
*                  Field 2: string, 35 characters, customer name
*                  Field 3: string, 50 characters, customer address
*                  Field 4: float, customer charge amount
*                  Field 5: integer/integer/integer, month, day, and
*                                year of last bill sent
*              Fields are comma separated.
*******************************************************************************/
```

3. All comments should be meaningful, describing the logic or reason for the code block, and readable.
4. Line comments should be used to explain a block of code – it isn't necessary to comment every single line. Rather, they should describe a logical block of code small enough to be easily understood and modifiable.