Clearness

Interviewers make their hiring decisions based on candidates' code. Code clearness has an important impact on the decision. As shown in Figure 5-1, handwriting, layout, and naming are three factors of code clearness important in interviews.

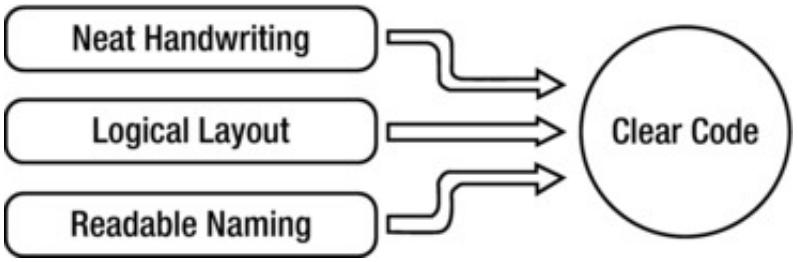

Figure 5-1. *Three factors of clear code: handwriting, layout, and naming*

First, clear code is written neatly. Engineers write code with keyboards for their daily development, but they have to write code on paper or whiteboards in most interviews. It is a good idea to practice writing code on paper during interview preparation and, if you have a whiteboard, a little practice can make a big difference. Many candidates' handwriting is illegible when they feel rushed, especially when they find problems in their own code and have to make some changes. The number of lines of code for interview problems is usually less than 50, so it does not take too much time to write. In other words, candidates should take their time, within reason, while writing code. The key factor is to form clear solutions and express them with neat and readable code.

Second, clear layout shows logical structure. Programmers usually write code in an integrated development environment, such as Visual Studio or Eclipse, which facilitates layout greatly. Candidates should pay attention to layout when such tools are unavailable in interviews. Candidates shouldn't write too large on a whiteboard or they may not have room to finish their code and then have to worry about cramming it in somewhere, interrupting their thinking about the code itself. Similarly, indentation level increases when there are many loops and conditional checks. Complex code without clear layout makes an interviewer confused if indentation does not show the logical structures, or curly brackets do not appear in pairs.

Last but not least, names in clear code are easy to read and understand. Many novices like the shortest names. They name variables as `i`, `j`, `k`, and name functions as `f`, `g`, `h`. Since such names do not show the meaning and purpose of variables and functions, they are very difficult to understand when a function gets long. It is recommended that you use word concatenations for names. For example, if a function takes a root node of a binary tree as a parameter, the parameter can be declared as `BinaryTreeNode* pRoot` in C/C++. Readable variables and function names make it simpler for interviewers to understand candidates' code. Otherwise, interviewers have to guess. For instance, is a variable `m` the maximum or minimum in an array?

Tip In order to improve code readability, it is a good practice to name variables and functions (methods) with English words or word concatenations.
