The goal of the coding or programming activity is to implement the design in the best possible manner. The coding activity affects both testing and maintenance profoundly. As we saw earlier, the time spent in coding is a small percentage of the total software cost, while testing and maintenance consume the major portion. Thus, it should be clear that the goal during coding should not be just to reduce the implementation cost, but help reduce the cost of later phases. During coding, it should be kept in mind that the programs should not be constructed so that they are easy to write, but in a manner that they are easy to read and understand. A program is read a lot more often and by a lot more people during the later phases.

Having readability and understandability as a clear objective of the coding activity can itself help in achieving it. A famous experiment by Weinberg showed that if programmers are specified a clear objective for the program, they usually satisfy it [82]. In the experiment, five different teams were given the same problem for which they had to develop programs. However, each of the teams was specified a different objective, which it had to satisfy. The different objectives given were: minimize the effort required to complete the program, minimize the number of statements, minimize the memory required, maximize the program clarity, and maximize the output clarity. It was found that in most cases each team did the best for the objective that was specified to it. The rank of the different teams for the different objectives is shown in Table 7.1.

The experiment clearly shows that if objectives are clear, programmers tend to achieve that objective. Hence, if readability is an objective of the coding activity, then it is likely that programmers will develop easily understandable
Table 7.1: The Weinberg experiment.

<table>
<thead>
<tr>
<th></th>
<th>Resulting Rank (1 = Best)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>O1</td>
</tr>
<tr>
<td>Minimize effort to complete (O1)</td>
<td>1</td>
</tr>
<tr>
<td>Minimize number of statements (O2)</td>
<td>2–3</td>
</tr>
<tr>
<td>Minimize memory required (O3)</td>
<td>5</td>
</tr>
<tr>
<td>Maximize program clarity (O4)</td>
<td>4</td>
</tr>
<tr>
<td>Maximize output clarity (O5)</td>
<td>2–3</td>
</tr>
</tbody>
</table>

programs. It also shows that if the focus is on minimizing coding effort, program clarity takes a big hit. For our purposes, ease of understanding and modification are the basic goals of the programming activity.

In this chapter we will discuss:

- Some principles like structured programming, information hiding, use of coding standards, which can help develop more readable programs.
- Some programmer-level processes like incremental development and test-driven development, for efficiently developing high-quality code.
- How to manage evolving code by using proper source code control and refactoring.
- How to unit test modules using unit testing frameworks.
- A structured code inspection process that can be used effectively to improve the quality of the code.

### 7.1 Programming Principles and Guidelines

The main task before a programmer is to write readable code with few bugs in it. An additional goal is to write code quickly. Writing solid code is a skill that can only be acquired by practice. However, based on experience, some general rules and guidelines can be given for the programmer. Good programming (producing correct and simple programs) is a practice independent of the target programming language, although well-structured programming languages make the programmer’s job simpler. In this section, we will discuss some concepts and practices that can help a programmer write higher-quality code that is also easier to understand.