

**Username:** Pralay Patoria **Book:** Coding Interviews: Questions, Analysis & Solutions. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

---

## Summary

This chapter has discussed how to improve code quality in interviews with clearness, completeness, and robustness.

Candidates write code on paper or whiteboards in most interviews. It improves code readability if code is written neatly with a clear layout and indentation, and variables and functions are named reasonably.

It is a good practice to figure out all possible inputs before writing code. Code is complete only when it fulfills functional requirements as well as when it handles boundary values and errors appropriately.

Defensive programming is a good habit to improve code robustness. Candidates should check input validity at the entry of each function and handle invalid inputs gracefully.

