Clean Code

**Chapter 1 Clean Code**

The book is about good programming and focuses on code. The author examines code from various perspectives and by the end of the book, the reader will have a good understanding of code, including the difference between good and bad code, and how to write and improve code. The author refutes the idea that code will one day disappear and believes that the level of abstraction in programming languages may increase, but code will never be eliminated. This is because code represents the details of requirements that cannot be ignored, and even specifications written in higher level languages will still need to be detailed enough for a machine to understand and execute them. The author believes that code will always be necessary because it is the language in which requirements are expressed.

The author disagrees with the idea that good code doesn't matter, which is a premise made in the preface of Kent Beck's book "Implementation Patterns." The author gives examples of how bad code can impede progress and lead to failure, as seen in a company that went out of business due to messy code. The author reflects on how every programmer has written bad code and left it for later, which often results in never fixing it due to LeBlanc's law: "Later equals never." The author believes that good code is essential and has been proven through experience.

**Chapter 2 Meaningful Names**