

---

# Table of Contents

<b>Also by Pete Goodliffe</b> .....	<b>xi</b>
<b>Introduction</b> .....	<b>xiii</b>
<b>1. Care About the Code</b> .....	<b>1</b>
<i>Adopting the correct approach and attitude to code</i>	
<hr/>	
<b>Part I. you.write(code);</b>	
<b>2. Keeping Up Appearances</b> .....	<b>7</b>
<i>Code presentation: layout and naming</i>	
<b>3. Write Less Code!</b> .....	<b>17</b>
<i>Avoiding unnecessary lines of code</i>	
<b>4. Improve Code by Removing It</b> .....	<b>29</b>
<i>Identifying and removing dead code</i>	
<b>5. The Ghost of a Codebase Past</b> .....	<b>37</b>
<i>Learning from the code you wrote in the past</i>	
<b>6. Navigating a Route</b> .....	<b>45</b>
<i>How to start working with unfamiliar code</i>	
<b>7. Wallowing in Filth</b> .....	<b>55</b>
<i>Dealing with unpleasant, messy code</i>	
<b>8. Don't Ignore That Error!</b> .....	<b>63</b>
<i>Healthy attitudes for error handling</i>	

<b>9. Expect the Unexpected.</b> .....	69
<i>Writing robust code that considers all possibilities</i>	
<b>10. Bug Hunting.</b> .....	75
<i>How to find and fix bugs</i>	
<b>11. Testing Times.</b> .....	87
<i>Developer testing: unit, integration, and system tests</i>	
<b>12. Coping with Complexity.</b> .....	103
<i>Designing code well, to avoid unnecessary complexity</i>	
<b>13. A Tale of Two Systems.</b> .....	113
<i>The consequences of good and bad design</i>	

---

## **Part II. Practice Makes Perfect**

<b>14. Software Development Is.</b> .....	131
<i>What is this software stuff?</i>	
<b>15. Playing by the Rules.</b> .....	141
<i>Inventing rules that define your development team</i>	
<b>16. Keep It Simple.</b> .....	145
<i>Striving for simplicity in our software</i>	
<b>17. Use Your Brain.</b> .....	153
<i>Programmers are allowed and encouraged to use their brain; don't be stupid!</i>	
<b>18. Nothing Is Set in Stone.</b> .....	157
<i>No code is sacred, everything changes</i>	
<b>19. A Case for Code Reuse.</b> .....	165
<i>The healthy way to reuse code</i>	
<b>20. Effective Version Control.</b> .....	171
<i>Using version control well</i>	

<b>21. Getting One Past the Goalpost.</b> .....	<b>183</b>
<i>Working effectively with the QA team</i>	
<b>22. The Curious Case of the Frozen Code.</b> .....	<b>195</b>
<i>Code freeze: what it is, and whether it is necessary</i>	
<b>23. Please Release Me.</b> .....	<b>203</b>
<i>Making software releases</i>	

---

### **Part III. Getting Personal**

<b>24. Live to Love to Learn.</b> .....	<b>215</b>
<i>How to learn effectively</i>	
<b>25. Test-Driven Developers.</b> .....	<b>225</b>
<i>Driving as an analogy to programming: how do we learn and pass the test?</i>	
<b>26. Relish the Challenge.</b> .....	<b>231</b>
<i>How to find the right challenges to stay motivated and keep your skills sharp</i>	
<b>27. Avoid Stagnation.</b> .....	<b>237</b>
<i>Preventing your programming skills from going stale</i>	
<b>28. The Ethical Programmer.</b> .....	<b>243</b>
<i>Ethical issues in the developer's life</i>	
<b>29. A Love for Languages.</b> .....	<b>253</b>
<i>Learning many programming languages and loving the ones you use</i>	
<b>30. Posturing Programmers.</b> .....	<b>261</b>
<i>Improving the programmer's health: posture, eye strain, and keeping your spirits up</i>	

---

### **Part IV. Getting Things Done**

<b>31. Smarter, Not Harder.</b> .....	<b>273</b>
<i>Working effectively: avoiding unnecessary work and solving the right problems</i>	
<b>32. It's Done When It's Done.</b> .....	<b>283</b>
<i>Defining your programming tasks and knowing exactly when you're done</i>	

<b>33. This Time I've Got It.....</b>	<b>291</b>
<i>Avoiding a narrow focus: find the best way to solve a problem</i>	
<hr/>	
<b>Part V. The People Pursuit</b>	
<b>34. People Power.....</b>	<b>299</b>
<i>How to position yourself alongside excellent programmers, and how to work well in a team</i>	
<b>35. It's the Thought That Accounts.....</b>	<b>305</b>
<i>Accountability: how it improves you and your work</i>	
<b>36. Speak Up!.....</b>	<b>313</b>
<i>Communication skills for the software developer</i>	
<b>37. Many-festos.....</b>	<b>323</b>
<i>Software manifestos: what and why?</i>	
<b>38. An Ode to Code.....</b>	<b>329</b>
<i>A cautionary tale of software mismanagement</i>	
<b>Epilogue.....</b>	<b>333</b>
<b>Index.....</b>	<b>337</b>