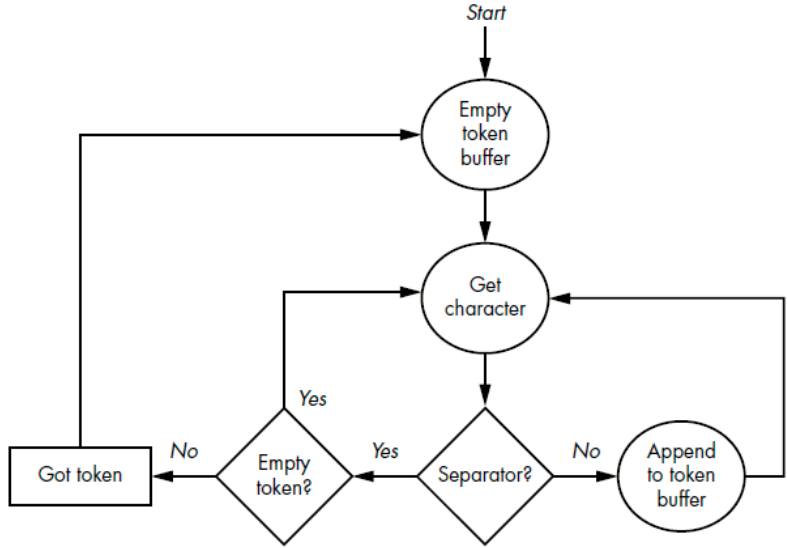


Understand Computers—Craft Better Code

by Jonathan E. Steinhart

errata updated to print 3

| Page | Error | Correction | Print corrected | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|---|---|-----------------|------|------|-----|------|------|-----|------|------|-----|------|----|----|-----|----|----|---|----|---|---|-----|----|---|--|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|----|----|-----|----|----|---|----|----|---|-----|----|---|---------|
| 13 | Figure replacement | <div><div><div><div>+1</div><div>10</div><div>10</div><div>10</div><div>01</div></div><div><div>-1</div><div>1</div><div>1</div><div>1</div><div>1</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div><div>0</div></div></div></div> <div>Figure 1-12: Finding -1</div> | Print 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 22 | <table><tr><th>Dec</th><th>Hex</th><th>Char</th><th>Dec</th><th>Hex</th><th>Char</th><th>Dec</th><th>Hex</th><th>Char</th><th>Dec</th><th>Hex</th><th>Char</th></tr><tr><td>16</td><td>10</td><td>DLE</td><td>48</td><td>30</td><td>0</td><td>80</td><td>5</td><td>P</td><td>112</td><td>70</td><td>p</td></tr></table> | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | 16 | 10 | DLE | 48 | 30 | 0 | 80 | 5 | P | 112 | 70 | p | <table><tr><th>Dec</th><th>Hex</th><th>Char</th><th>Dec</th><th>Hex</th><th>Char</th><th>Dec</th><th>Hex</th><th>Char</th><th>Dec</th><th>Hex</th><th>Char</th></tr><tr><td>16</td><td>10</td><td>DLE</td><td>48</td><td>30</td><td>0</td><td>80</td><td>50</td><td>P</td><td>112</td><td>70</td><td>p</td></tr></table> | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | 16 | 10 | DLE | 48 | 30 | 0 | 80 | 50 | P | 112 | 70 | p | Print 2 |
| Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16 | 10 | DLE | 48 | 30 | 0 | 80 | 5 | P | 112 | 70 | p | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16 | 10 | DLE | 48 | 30 | 0 | 80 | 50 | P | 112 | 70 | p | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | Figure replacement | <div><div><div><div><div>1514131211109876543210</div><div>0000000000000001</div></div><div><div>1514131211109876543210</div><div>00100000000001</div></div></div><div><div>Unicode A (0x0041)</div><div>UTF-8 A (0x41)</div></div><div><div><div>1514131211109876543210</div><div>0000000111000000</div></div><div><div>1514131211109876543210</div><div>11000000000000</div></div></div><div><div>Unicode π (0x03C0)</div><div>UTF-8 π (0xCF 0x80)</div></div><div><div><div>1514131211109876543210</div><div>0010001100100011</div></div><div><div>1514131211109876543210</div><div>11010000000011</div></div></div><div><div>Unicode ♣ (0x2663)</div><div>UTF-8 ♣ (0xE2 0x99 0xA3)</div></div></div></div> | Print 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Page | Error | Correction | Print corrected |
|------|---|--|-----------------|
| 158 | One of Metcalfe's big innovations was <i>random back-off-and-retry</i> . | Metcalfe used <i>random back-off-and-retry</i> , an innovation pioneered by ALOHAnet, a packetswitched radio network developed at the University of Hawaii. | Print 3 |
| 177 | And the concept of a <i>user</i> appeared so that machines could tell what belonged to who . | And the concept of a <i>user</i> appeared so that machines could tell what belonged to whom . | Print 3 |
| 221 | Figure replacement |  <pre> graph TD Start([Start]) --> EmptyTokenBuffer((Empty token buffer)) EmptyTokenBuffer --> GetCharacter((Get character)) GetCharacter --> Separator{Separator?} Separator -- Yes --> EmptyToken{Empty token?} Separator -- No --> AppendBuffer((Append to token buffer)) AppendBuffer --> GetCharacter EmptyToken -- Yes --> GotToken[Got token] EmptyToken -- No --> EmptyTokenBuffer GotToken --> EmptyTokenBuffer </pre> <p>Figure 8-1: Simple lexical analysis</p> | Print 3 |