

# Appendix 2

## Precedence and Associativity Rules

PRECEDENCE	ASSOCIATIVITY
From highest at top to lowest at bottom. Operators in the same group have equal precedence.	
Dot operator, array indexing, and method invocation: <code>.</code> , <code>[ ]</code> , <code>()</code>	Left to right
<code>++</code> (postfix, as in <code>x++</code> ), <code>--</code> (postfix)	Right to left
The unary operators: <code>+</code> , <code>-</code> , <code>++</code> (prefix, as in <code>++x</code> ), <code>--</code> (prefix), <code>!</code> , <code>~</code> (bitwise complement) <sup>a</sup>	Right to left
<code>new</code> and type casts ( <i>Type</i> )	Right to left
The binary operators <code>*</code> , <code>/</code> , <code>%</code>	Left to right
The binary operators <code>+</code> , <code>-</code>	Left to right
The binary operators <code>&lt;&lt;</code> , <code>&gt;&gt;</code> , <code>&gt;&gt;&gt;</code> (shift operators) <sup>a</sup>	Left to right
The binary operators <code>&lt;</code> , <code>&gt;</code> , <code>&lt;=</code> , <code>&gt;=</code> , <code>instanceof</code>	Left to right
The binary operators <code>==</code> , <code>!=</code>	Left to right
The binary operator <code>&amp;</code>	Left to right
The binary operator <code>^</code> (exclusive or) <sup>a</sup>	Left to right
The binary operator <code> </code>	Left to right
The binary operator <code>&amp;&amp;</code>	Left to right
The binary operator <code>  </code>	Left to right
The ternary operator (conditional operator) <code>?:</code>	Right to left
The assignment operators <code>=</code> , <code>*=</code> , <code>/=</code> , <code>%=</code> , <code>+=</code> , <code>-=</code> , <code>&amp;=</code> , <code> =</code> , <code>^=</code> , <code>&lt;&lt;=</code> , <code>&gt;&gt;=</code> , <code>&gt;&gt;&gt;=</code>	Right to left

<sup>a</sup> Not discussed in this book.