APPENDIX F

Bit Operations

To write programs at the machine-level, often you need to deal with binary numbers directly and perform operations at the bit-level. Java provides the bitwise operators and shift operators defined in Table F.1.

The bit operators apply only to integer types (`byte`, `short`, `int`, and `long`). A character involved in a bit operation is converted to an integer. All bitwise operators can form bitwise assignment operators, such as `=`, `|=`, `<<=`, `>>=`, and `>>>=`.

TABLE F.1

Operator	*Name*	*Example (using bytes in the example)*	*Description*
`&`	Bitwise AND	10101110 **&** 10010010 yields 10000010	The AND of two corresponding bits yields a 1 if both bits are 1.
`|`	Bitwise inclusive OR	10101110 **|** 10010010 yields 10111110	The OR of two corresponding bits yields a 1 if either bit is 1.
`^`	Bitwise exclusive OR	10101110 **^** 10010010 yields 00111100	The XOR of two corresponding bits yields a 1 only if two bits are different.
`~`	One's complement	**~**10101110 yields 01010001	The operator toggles each bit from 0 to 1 and from 1 to 0
`<<`	Left shift	10101110 **<<** 2 yields 10111000	Shift bits in the first operand left by the number of bits specified in the second operand, filling with 0s on the right.
`>>`	Right shift with sign extension	10101110 **>>** 2 yields 11101011 00101110 **>>** 2 yields 00001011	Shift bit in the first operand right by the number of bits specified in the second operand, filling with the highest (sign) bit on the left.
`>>>`	Right shift with zero extension	10101110 **>>>** 2 yields 00101011 00101110 **>>>** 2 yields 00001011	Shift bit in the first operand right by the number of bits specified in the second operand, filling with 0s on the left.