

TABLE 10.10: *Casting numeric types*

Expression	Result
(int) f	Demote f to an int.
(double) f	Promote f to a double.
(long) f	Demote f to a long.
(float) l	Promote l to a float.
(int) i	No change
(double) (i+f)	First i is converted to a float, then the final result is converted to a double.

truncation. This means that the char value is always positive, but the short and byte equivalents may be negative. Table 10.11 lists some examples of what happens when you cast an int to a smaller type.

10.8.3 ~ Operator

The ~ operator is not represented in Table 10.6. The ~ operator takes an int or a long and inverts each bit. There is no instruction for this operator. Java compilers take note of the fact that, for a single bit x, computing ~x is equivalent to computing $x + -1$, where + is the exclusive-or operator. To invert all the bits in the number at once, the Java compiler uses the lxor or ixor instruction with the value consisting of 64 or 32 1's. In the two's complement notation used in the Java virtual machine, an integer consisting of all 1's is equal to -1. For example,

~x

TABLE 10.11: *Converting between int types*

Expression	Result
(short) 65555	-1
(char) 65535	65535
(byte) 65535	-1
(short) 160	160
(char) 160	160
(byte) 160	-2