Floating Point: Recap

Format

The IEEE floating point format consists of three fields (from right to left, or from least significant to most significant): fraction, exponent, and sign. For the two most common formats, these fields are packed in either a 32-bit (single-precision, "float") or 64-bit (double-precision, "double") words.

	Single-precision	**Double-precision**
Fraction width (*f*)	23 bits (bits 0-22)	52 bits (bits 0-51)
Exponent width (*e*)	8 bits (bits 23-30)	11 bits (bits 52-62)
Sign width (*s*)	1 bit (bit 31)	1 bit (bit 63)
Bias	127 (2^8-1)	1023 ($2^{11}-1$)

The interpretation of a given floating point representation depends on the value of the exponent field (*e*) (the table below assumes the single-precision floating point format; for double-precision 255 would become 2047, the rest stays the same).

Value of *e*	**Representation**	**Effective exponent (*E*)**	**Effective fraction (*M*)**	**Value**
$e = 0$	Denormalized	$E = 1 - \text{bias}$	$M = 0.f$	Value = $2^E \times M$
$0 < e < 255$	Normalized	$E = e - \text{bias}$	$M = 1.f$ (implied 1)	Value = $2^E \times M$
$e = 255$	Special values	Not applicable	Not applicable	Value = ∞ or NaN

Special values include ∞ and NaN. The latter stands for Not a Number and is used to encode results that cannot be given as a real number or infinity (e.g., 0/0, ∞ − ∞, or square root of −1).

Rounding

The IEEE floating point format specifies four rounding modes. The default mode is Round-to-even.

Mode	**$1.40**	**$1.60**	**$1.50**	**$2.50**	**$-1.50**
Round-to-even	$1	$2	$2	$2	$−2
Round-toward-zero	$1	$1	$1	$2	$−1
Round-down	$1	$1	$1	$2	$−2
Round-up	$2	$2	$2	$3	$−1

Floating Point in C

All versions of C provide two different floating-point data types: **float** and **double**. On machines that support IEEE floating point, these data types correspond to single- and double-precision floating point.

There is no standard method in C to change the rounding mode or to get the special values such as −0, +∞, −∞, or *NaN*.