Cheat Sheet

C For Dummies

From C For Dummies, 2nd Edition by Dan Gookin

The C programming language is fast and versatile. You can use just 32 keywords and some fairly intuitive symbols to do comparisons and conversions. Then you get to numeric data and math symbols, which are pretty much as you expect as well.

C Language Comparison Symbols

If you're writing programs in C, you need to use comparison symbols. The symbols C uses, their meanings, and examples are shown in the following table:

Symbol	Meaning or Pronunciation	"True" Comparison Examples
<	Less than	`1 < 5` `8 < 9`
==	Equal to	`5 == 5` `0 == 0`
>	Greater than	`8 > 5` `10 > 0`
<=	Less than or equal to	`4 <= 5` `8 <= 8`
>=	Greater than or equal to	`9 >= 5` `2 >= 2`
!=	Not equal to	`1 != 0` `4 != 3.99`

C Language Comparisons and Their Opposites

If you're programming in C — or any other language — you need to use If/Else statements. The comparison symbols you need if you're working with C and the Else statements they generate are shown in the following table:

If Comparison	Else Statement Executed By This Condition
<	>= (Greater than or equal to)
==	!= (Not equal to)
>	<= (Less than or equal to)
<=	> (Greater than)
>=	< (Less than)
!=	== (Equal to)

C Language Conversion Characters

When programming in C, you use conversion characters — the percent sign and a letter, for the most part — as placeholders for variables you want to display. The following table shows the conversion characters and what they display:

Conversion Character	Displays Argument (Variable's Contents) As
%c	Single character
%d	Signed decimal integer (int)
%e	Signed floating-point value in E notation
%f	Signed floating-point value (float)
%g	Signed value in %e or %f format, whichever is shorter
%i	Signed decimal integer (int)
%o	Unsigned octal (base 8) integer (int)
%s	String of text

%u	Unsigned decimal integer (int)
%x	Unsigned hexadecimal (base 16) integer (int)
%%	(percent character)

C Language Escape Sequences

Programming in C is fast — all you have to do is type a short sequence of keystrokes — generally just two — to get a tab, a new line, a question mark, and more. The following table shows the sequences you need to accomplish a variety of tasks:

Sequence	Represents
\a	The speaker beeping
\b	Backspace (move the cursor back, no erase)
\f	Form feed (eject printer page; ankh character on the screen)
\n	Newline, like pressing the Enter key
\r	Carriage return (moves the cursor to the beginning of the line)
\t	Tab
\v	Vertical tab (moves the cursor down a line)
\\	The backslash character
\’	The apostrophe
\”	The double-quote character
\?	The question mark
\0	The “null” byte (backslash-zero)
\x*nnn*	A character value in hexadecimal (base 16)
\X*nnn*	A character value in hexadecimal (base 16)

C Language Keywords

The C programming language has just 32 keywords for you to build robust programs. With only 32 keywords, they all fit nicely into a short table. Use them wisely and well.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	static	void
default	goto	sizeof	volatile
do	if	signed	While

C Language Numeric Data Types

When programming with C, keywords and variables go together like the 4th of July and fireworks, although with a bit less drama. The following table shows C keywords, their variable types, and their ranges:

Keyword	*Variable Type*	*Range*
`char`	Character (or string)	-128 to 127
`int`	Integer	-32,768 to 32,767
`short` `short int`	Short integer	-32,768 to 32,767
`long`	Long integer	-2,147,483,648 to 2,147,483,647
`unsigned char`	Unsigned character	0 to 255
`unsigned int`	Unsigned integer	0 to 65,535
`unsigned short`	Unsigned short integer	0 to 65,535
`unsigned long`	Unsigned long integer	0 to 4,294,967,295
`float`	Single-precision floating point (accurate to 7 digits)	$\pm 3.4 \times 10^{-38}$ to $\pm 3.4 \times 10^{38}$
`double`	Double-precision floating point (accurate to 15 digits)	$\pm 1.7 \times 10^{-308}$ to $\pm 1.7 \times 10^{308}$

C Language Mathematical Symbols

Programming math functions with C is fairly straightforward: a plus sign works like any sixth-grader knows it should and does addition. The mathematical symbols and the function they serve in C are shown in the following table:

Operator or Symbol	*What You Expected*	*As Pronounced By Sixth-Graders*	*Task*
+	+	"Plus"	Addition
-	-	"Minus"	Subtraction
*	x	"Times"	Multiplication
/	÷	"Divided by"	Division
%		??	Modulus