CC-112 Programming Fundamentals

Introduction to C Programming - II

Nazar Khan
Department of Computer Science
University of the Punjab

Arithmetic Operators

C operation	Arithmetic operator	Algebraic expression	C expression
Addition	+	$f + 7$	`f + 7`
Subtraction	-	$p - c$	`p - c`
Multiplication	*	bm	`b * m`
Division	/	x / y or $\frac{x}{y}$ or $x \div y$	`x / y`
Remainder	%	$r \bmod s$	`r % s`

Note that *integer division* produces an integer result. So 17/5 produces 3 but 17.0/5 produces 3.4.

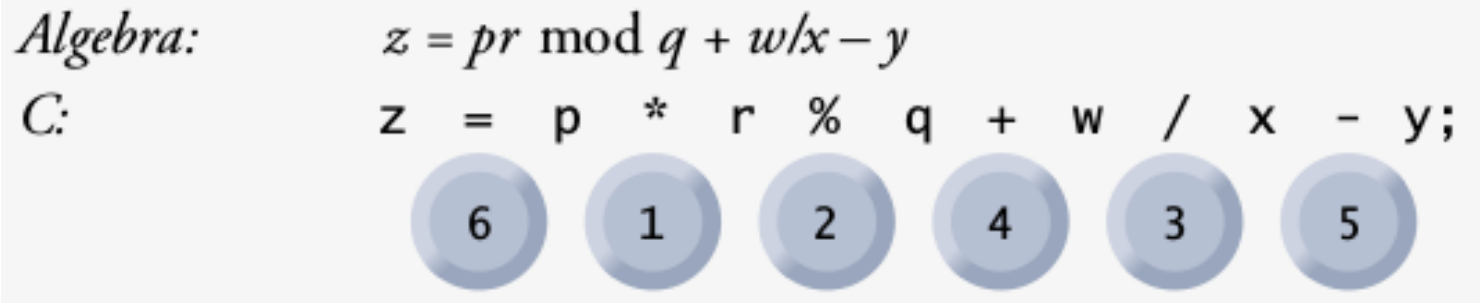
Rules of Operator Precedence

1. Operators in expressions contained within pairs of parentheses are evaluated first. Parentheses are said to be at the *highest level of precedence*. In cases of nested parentheses, such as `( ( a + b ) + c )` the operators in the innermost pair of parentheses are applied first.
2. Multiplication, division and remainder operations are applied next. If an expression contains several multiplication, division and remainder operations, evaluation proceeds from left to right. Multiplication, division and remainder are said to be on the same level of precedence.
3. Addition and subtraction operations are evaluated next. If an expression contains several addition and subtraction operations, evaluation proceeds from left to right. Addition and subtraction also have the same level of precedence, which is lower than the precedence of the multiplication, division and remainder operations.
4. The assignment operator (=) is evaluated last.

Rules of Operator Precedence

Operator(s)	Operation(s)	Order of evaluation (precedence)
( )	Parentheses	Evaluated first. If the parentheses are nested, the expression in the *innermost* pair is evaluated first. If there are several pairs of parentheses "on the same level" (i.e., not nested), they're evaluated left to right.
* / %	Multiplication Division Remainder	Evaluated second. If there are several, they're evaluated left to right.
+ -	Addition Subtraction	Evaluated third. If there are several, they're evaluated left to right.
=	Assignment	Evaluated last.

Rules of Operator Precedence



- What will be the sequence of evaluation in the following expression?

```
y=a*x*x+b*x+c;
```

- What will be the sequence of evaluation in the following expression?

```
y=(a*x*x)+(b*x)+c;
```

- Therefore, just as in algebra, use parantheses in your code for clarity.

Relational Operators

Algebraic equality or relational operator	C equality or relational operator	Example of C condition	Meaning of C condition
Relational operators			
>	>	x > y	x is greater than y
<	<	x < y	x is less than y
≥	>=	x >= y	x is greater than or equal to y
≤	<=	x <= y	x is less than or equal to y
Equality operators			
=	==	x == y	x is equal to y
≠	!=	x != y	x is not equal to y

Relational Operators

```c
// Using if statements, relational
// operators, and equality operators.
#include <stdio.h>

// function main begins program execution
int main( void )
{
   printf( "Enter two integers, and I will tell you\n" );
   printf( "the relationships they satisfy: " );
   int num1; // first number to be read from user
   int num2; // second number to be read from user
   scanf( "%d %d", &num1, &num2 ); // read two integers

   if ( num1 == num2 ) {
      printf( "%d is equal to %d\n", num1, num2 );
   } // end if

   if ( num1 != num2 ) {
      printf( "%d is not equal to %d\n", num1, num2 );
   } // end if

   if ( num1 < num2 ) {
      printf( "%d is less than %d\n", num1, num2 );
   } // end if

   if ( num1 > num2 ) {
      printf( "%d is greater than %d\n", num1, num2 );
   } // end if
```

Relational Operators

```
  if ( num1 <= num2 ) {
    printf( "%d is less than or equal to %d\n", num1, num2 );
  } // end if

  if ( num1 >= num2 ) {
    printf( "%d is greater than or equal to %d\n", num1, num2 );
  } // end if
} // end function main
```

Precedences

Operators	Associativity
()	left to right
* / %	left to right
+ -	left to right
< <= > >=	left to right
== !=	left to right
=	right to left

Precedence and associativity of arithmetic and relational operators.

The result of the expression

$$3*6+4*4/2<=25-6*4<2$$

is 1. Can you see how?

C Keywords

Keywords

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

Keywords added in C99 standard

_Bool _Complex _Imaginary inline restrict

Keywords added in C11 standard

_Alignas _Alignof _Atomic _Generic _Noreturn _Static_assert _Thread_local

C keywords are reserved. You cannot use these as variable names.