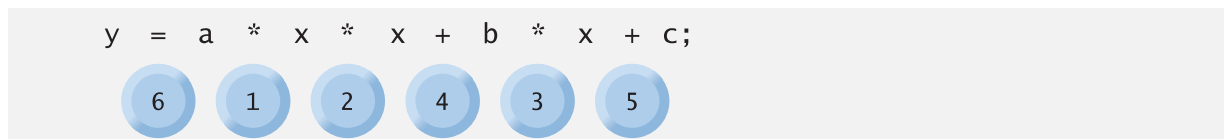


The circled numbers under the statement indicate the order in which C++ applies the operators. The multiplication, modulus and division are evaluated *first* in left-to-right order (i.e., they associate from left to right) because they have *higher precedence* than addition and subtraction. The addition and subtraction are applied next. These are also applied left to right. The assignment operator is applied *last* because its precedence is *lower* than that of any of the arithmetic operators.

Evaluation of a Second-Degree Polynomial

To develop a better understanding of the rules of operator precedence, consider the evaluation of a second-degree polynomial $y = ax^2 + bx + c$:



The circled numbers under the statement indicate the order in which C++ applies the operators. *There is no arithmetic operator for exponentiation in C++*, so we've represented x^2 as $x * x$. In Chapter 5, we'll discuss the standard library function `pow` ("power") that performs exponentiation.

Suppose variables `a`, `b`, `c` and `x` in the preceding second-degree polynomial are initialized as follows: `a = 2`, `b = 3`, `c = 7` and `x = 5`. Figure 2.11 illustrates the order in which the operators are applied and the final value of the expression.

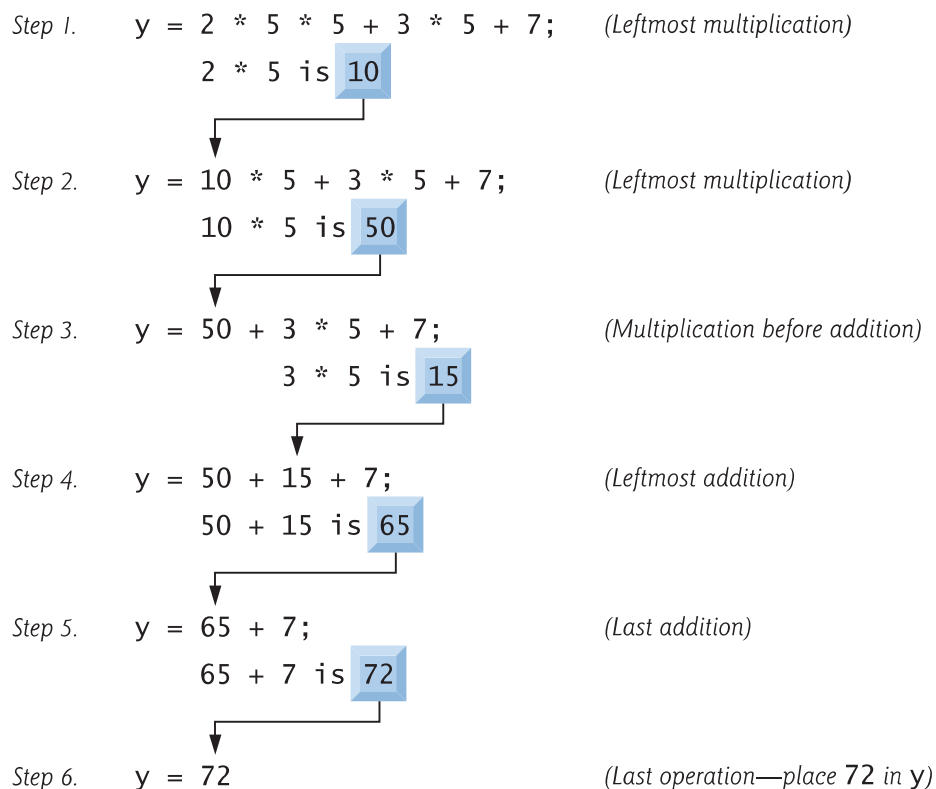


Fig. 2.11 | Order in which a second-degree polynomial is evaluated.