https://www.eskimo.com/~scs/cclass/int/sx4cb.html

The rules, then (which you can also find on page 44 of K&R2, or in section 6.2.1 of the newer ANSI/ISO C Standard) are approximately as follows:

1. First, in most circumstances, values of type `char` and `short int` are converted to `int` right off the bat.
2. If an operation involves two operands, and one of them is of type `long double`, the other one is converted to `long double`.
3. If an operation involves two operands, and one of them is of type `double`, the other one is converted to `double`.
4. If an operation involves two operands, and one of them is of type `float`, the other one is converted to `float`.
5. If an operation involves two operands, and one of them is of type `long int`, the other one is converted to `long int`.
6. If an operation involves both signed and unsigned integers, the situation is a bit more complicated. If the unsigned operand is smaller (perhaps we're operating on `unsigned int` and `long int`), such that the larger, signed type could represent all values of the smaller, unsigned type, then the unsigned value is converted to the larger, signed type, and the result has the larger, signed type. Otherwise (that is, if the signed type can *not* represent all values of the unsigned type), both values are converted to a common unsigned type, and the result has that unsigned type.
7. Finally, when a value is assigned to a variable using the assignment operator, it is automatically converted to the type of the variable if (a) both the value and the variable have arithmetic type (that is, integer or floating point), or (b) both the value and the variable are pointers, and one or the other of them is of type `void *`.