For the complex types, the corresponding real type is the original type without the word _Complex. (The corresponding real type for float _Complex is float, for example.)

We're now ready to discuss the general rules that govern type conversions involving complex types. I'll group them into three categories.

- *Complex to complex.* The first rule concerns conversions from one complex type to another, such as converting from float _Complex to double _Complex. In this situation, the real and imaginary parts are converted separately, using the rules for the corresponding real types (see Section 7.4). In our example, the real part of the float _Complex value would be converted to double, yielding the real part of the double _Complex value; the imaginary part would be converted to double in a similar fashion.

- *Real to complex.* When a value of a real type is converted to a complex type, the real part of the number is converted using the rules for converting from one real type to another. The imaginary part of the result is set to positive or unsigned zero.

- *Complex to real.* When a value of a complex type is converted to a real type, the imaginary part of the number is discarded; the real part is converted using the rules for converting from one real type to another.

One particular set of type conversions, known as the usual arithmetic conversions, are automatically applied to the operands of most binary operators. There are special rules for performing the usual arithmetic conversions when at least one of the two operands has a complex type:

1. If the corresponding real type of either operand is long double, convert the other operand so that its corresponding real type is long double.
2. Otherwise, if the corresponding real type of either operand is double, convert the other operand so that its corresponding real type is double.
3. Otherwise, one of the operands must have float as its corresponding real type. Convert the other operand so that its corresponding real type is also float.

A real operand still belongs to a real type after conversion, and a complex operand still belongs to a complex type.

Normally, the goal of the usual arithmetic conversions is to convert both operands to a common type. However, when a real operand is mixed with a complex operand, performing the usual arithmetic conversions causes the operands to have a common real type, but not necessarily the *same* type. For example, adding a float operand and a double _Complex operand causes the float operand to be converted to double rather than double _Complex. The type of the result will be the complex type whose corresponding real type matches the common real type. In our example, the type of the result will be double _Complex.