compatible types. C99 also requires that either both structures have the same tag or neither has a tag.

trailing comma in enumerations   In C99, the last constant in an enumeration may be followed by a comma.

17   Advanced Uses of Pointers

restricted pointers   C99 has a new keyword, `restrict`, that can appear in the declaration of a pointer.

flexible array members   C99 allows the last member of a structure to be an array of unspecified length.

18   Declarations

block scopes for selection and iteration statements   In C99, selection statements (`if` and `switch`) and iteration statements (`while`, `do`, and `for`)—along with the "inner" statements that they control—are considered to be blocks.

array, structure, and union initializers   In C89, a brace-enclosed initializer for an array, structure, or union must contain only constant expressions. In C99, this restriction applies only if the variable has static storage duration.

inline functions   C99 allows functions to be declared `inline`.

21   The Standard Library

`<stdbool.h>` header   The `<stdbool.h>` header, which defines the `bool`, `true`, and `false` macros, is new in C99.

22   Input/Output

...printf conversion specifications   The conversion specifications for the *...*`printf` functions have undergone a number of changes in C99, with new length modifiers, new conversion specifiers, the ability to write infinity and NaN, and support for wide characters. Also, the `%le`, `%lE`, `%lf`, `%lg`, and `%lG` conversions are legal in C99; they caused undefined behavior in C89.

...scanf conversion specifications   In C99, the conversion specifications for the *...*`scanf` functions have new length modifiers, new conversion specifiers, the ability to read infinity and NaN, and support for wide characters.

snprintf function   C99 adds the `snprintf` function to the `<stdio.h>` header.

23   Library Support for Numbers and Character Data

additional macros in `<float.h>` header   C99 adds the `DECIMAL_DIG` and `FLT_EVAL_METHOD` macros to the `<float.h>` header.