The New C Standard (Excerpted material)

An Economic and Cultural Commentary

Derek M. Jones
derek@knosof.co.uk

6.8.5.2 The do statement

do statement

The evaluation of the controlling expression takes place after each execution of the loop body. 1773

Commentary

The loop body of a **do** statement is always executed at least once.

Coding Guidelines

while statement

The benefits associated with having side effects in the controlling expression of a **while** statement are not applicable to a **do** statement (because the loop is always executed at least once). Given that the use of a **do** statement is relatively rare and that developers are likely to be familiar with the side effect idioms that occur in controlling expressions, no guideline recommendation is given here.

Example

macro function-like

One use of the **do** statement is to solve the dangling semicolon problem that can occur when a function-like macro replaces a function call. Bracketing a sequence of statement with braces creates a compound statement, which does not require a terminating semicolon. In most contexts a semicolon following a function-like macro invocation is a harmless null statement. However, as the following example shows, when it forms the first arm of an **if** statement that contains an **else** arm, the presence of a semicolon is a syntax violation. Enclosing a sequence of statements in the body of a **do** statement, whose controlling expression is false, avoids this problem.

```
#define STMT_SEQ(p)                  \
     do {                            \
        /* sequence of statements */ \
        }                            \
     while (0)

/* ... */

if (cond)
   STMT_SEQ(x);
else
   /* ... */
```

References