Wording for [[fallthrough]] attribute.

Document No.: P0188R1
Revises: P0188R0
Project: Programming Language C++
Audience: Core
Author: Andrew Tomazos <andrewtomazos@gmail.com>
Date: 2016-02-29

Summary

Wording for the [[fallthrough]] attribute described in the previous revision is proposed for application to the C++17 working draft.

Wording

Edit [stmt.expr] 6.2p1:

An expression statement with the expression missing is called a null statement.

Add new section to [dcl.attr] 7.6:

7.6.6 Fallthrough attribute

1. The attribute-token fallthrough may be applied to a null statement (6.2); such a statement is a fallthrough statement. The attribute-token fallthrough shall appear at most once in each attribute-list and no attribute-argument-clause shall be present. A fallthrough statement may only appear within an enclosing switch statement (6.4.2). The next statement that would be executed after a fallthrough statement shall be a labeled statement whose label is a case label or default label for the same switch statement. The program is ill-formed if there is no such statement.

2. [Note: The use of a fallthrough statement is intended to suppress a warning that an implementation might otherwise issue for a case or default label that is reachable from another case or default label along some path of execution.]
Implementations are encouraged to issue a warning if a fallthrough statement is not dynamically reachable. -- end note

3. [Example:

```c
void f(int n) {
    void g(), h(), i();
    switch (n) {
        case 1:
        case 2:
            g();
            [[fallthrough]];
        case 3: // warning on fallthrough discouraged
            h();
        case 4: // implementation may warn on fallthrough
            i();
            [[fallthrough]]; // ill-formed
    }
}

-- end example]