- The initializer for an enumerator
- The length of a bit field
- Nontype template parameters
- The initializer of a **constexpr** variable (see Section 2.1."**constexpr** Variables" on page 302)

Computing the value of expressions in these contexts requires that all ~~of their subexpressions~~ be known and evaluable at compile time, ~~except~~ those that are short-circuited by the logical *or* operator (`||`), the logical *and* operator (`&&`), and the *ternary* operator (`?:`):

```
constexpr int f(int x) { return x || (throw x, 1); }
constexpr int g(int x) { return x && (throw x, 1); }
constexpr int h(int x) { return x ? 1 : throw x; }

static_assert(f(true), "");    // OK, throw x is never evaluated.
static_assert(!g(false), "");  // OK,   "   "  "    "       "
static_assert(h(true), "");    // OK,   "   "  "    "       "
```

Note that the **controlling constant expression** for the preprocessor directives **#if** and **#elif**, while similar to general constant expressions, are computed before any functions — **constexpr** or not — are even parsed. Consequently, **constexpr** functions cannot be invoked as part of the controlling constant expression for preprocessor directives.

Second, the C++11 Standard identifies a clear set of operations that are not available for use in constant expressions and, therefore, cannot be relied upon for compile-time evaluation. Any operation that does the following is unavailable.

- Throws an exception
- ~~Invokes the~~ **new** ~~and~~ **delete** ~~operators~~
- ~~Invokes~~ a lambda ~~function~~
- Depends on runtime polymorphism, such as **dynamic_cast**, **typeid** on a polymorphic type, or invokes a virtual function, which cannot be **constexpr**
- Uses **reinterpret_cast**
- Modifies an object (increment, decrement, and assignment), including function parameters, member variables, and global variables
- Has *undefined behavior* such as integer overflow, dereferencing **nullptr**, or indexing outside the bounds of an array
- Invokes a non**constexpr** function or constructor, or a **constexpr** function whose definition has not yet been seen