Compile-Time Accessible Variables

A variable or **variable template** of **literal type** can be declared to be **constexpr**, ensuring it is initialized and can be used at compile time.

Description

Variables of *all* built-in types and certain user-defined types, collectively known as **literal types**, can be declared **constexpr**, allowing them to be initialized at compile-time and subsequently used in **constant expressions**:

```
          int i0 = 5;              // i0 is not a compile-time constant.
    const int i1 = 5;              // i1 is a compile-time constant.
constexpr int i2 = 5;              // i2 " "   "     "     "

          double d0 = 5.0;         // d0 is not a compile-time constant.
    const double d1 = 5.0;         // d1 " " "  "     "     "
constexpr double d2 = 5.0;         // d2 is a compile-time constant.

          const char* s1 = "help"; // s1 is not a compile-time constant.
constexpr const char* s2 = "help"; // s2 is a compile-time constant.
```

Although **const** variables of integral types having preceding initialization with a **constant expression** can be used within constant expressions (e.g., as the first argument to **static_assert**, as the size of an array, or as a non-type template parameter), such is not the case for any other type:

```
static_assert(i0 == 5, "");        // Error, i0 is not a compile-time constant.
static_assert(i1 == 5, "");        // OK, const is "magical" for integers ~~(only)~~.
static_assert(i2 == 5, "");        // OK

static_assert(d1 == 5, "");        // Error, d1 is not a compile-time constant.
static_assert(d2 == 5, "");        // OK

static_assert(s1[1] == 'e', "");   // Error, s1 is not a compile-time constant.
static_assert(s2[1] == 'e', "");   // OK

int a1[s1[1]];                    // Error, s1 is not a compile-time constant.
int a2[s2[1]];                    // OK, a C-style array of 101 (e) integers.

std::array<int, s1[1]> sa1;       // Error, s1 is not a compile-time constant.
std::array<int, s2[1]> sa2;       // OK, an std::array of 101 (e) integers.
```

Prior to C++11, the types of variables usable in a **constant expression** were quite limited: