Glossary

class type – a kind of user-defined type (UDT) whose definition is introduced using a class key (one of **`class`**, **`struct`**, or **`union`**). **`alignas`** (168), Generalized PODs '11 (405), **`friend`** '11 (1031)

closure – a (1) closure object or (2) closure type. Local Types '11 (87), Lambdas (578), *Generic* Lambdas (982), Lambda Captures (986), **`auto`** Return (1197)

closure object – a callable object created via a lambda expression. Lambdas (578), *Generic* Lambdas (968), **`auto`** Return (1197)

closure type – the (unnamed) type of a closure object produced by a lambda expression. Lambdas (578), *Generic* Lambdas (968)

code bloat – excessive object code as might result from (1) the inlining of a large function body invoked from numerous call sites or (2) many similar but not identical instantiations of a function template, say, when employing perfect forwarding of string literals (see Section 2.1. "Forwarding References" on page 377). **`extern template`** (353), **`friend`** '11 (1054)

code elision – a compiler optimization whereby code that provably can never execute is simply omitted from the generated object code. **`noexcept`** Specifier (1136)

code motion – a general term for compiler optimizations that reorder provably independent evaluations to potentially improve performance, based on the capabilities of hardware on which the code will execute. **`noexcept`** Specifier (1136)

code point – a single character (e.g., glyph, control character, or modifier) in a character set. Unicode comprises 1,114,112 code points. ASCII comprises 128 code points. Unicode Literals (129)

code unit – the smallest subdivision of a code point for a specific encoding. In Unicode, for example, the UTF-8 encoding uses 8-bit code units (one to four code units to represent each code point), and the UTF-32 encoding uses 32-bit code units (one per code point). Generalized PODs '11 (476)

cold path – a segment of generated object code that is executed only rarely or in exceptional cases (e.g., code within a **`catch`** block); such exceptional code is sometimes relegated to physically separate locations in memory so as to eliminate *any* added runtime cost associated with its existence (e.g., zero-cost-exception model). **`noexcept`** Specifier (1103)

Collatz conjecture – one that states that for all positive integers N, the Collatz sequence is finite. Note that the length, however, varies widely with successive values N.

Collatz function – one that, given an integer N, returns $N/2$ if N is even and $3N + 1$ if N is odd; see also Collatz conjecture. **`constexpr`** Variables (313)

Collatz length – the cardinality of the shortest Collatz sequence starting with N that contains 1; note that the length varies widely with successive values of N. **`constexpr`** Variables (313)

Collatz sequence – one in which each successive element is obtained by starting with N and repeatedly applying the Collatz function to obtain the next element in the sequence. **`constexpr`** Variables (313)

comma operator – (1) the built-in sequencing operator that evaluates and discards the result of its left-hand side expression and then unconditionally evaluates its right-hand side expression, which becomes the result of evaluating the operator or (2) an *overloaded* operator, having the name **`operator,`**, that unconditionally evaluates both of its arguments and returns a result as determined by its user-provided definition. **`constexpr`** Functions (268)

common initial member sequence (CIMS) – the longest *initial* sequence (in declaration order) of nonstatic data members and bit fields within a class type that is the same between two standard-layout types. Generalized PODs '11 (406)