**C++11 Features:**

1. **Auto Keyword**:
   * Automatic type inference.
   * Example: **auto x = 5;** infers **x** as an **int**.
2. **Range-based for Loop**:
   * Simplifies iteration over elements in a container.
   * Example: **for (auto& element : container) { /\*...\*/ }**.
3. **Lambda Expressions**:
   * Inline anonymous functions.
   * Example: **auto func = [](int x, int y) { return x + y; };**.
4. **Smart Pointers**:
   * **std::unique\_ptr**, **std::shared\_ptr**, **std::weak\_ptr**.
   * Improved memory management.
5. **Move Semantics**:
   * Efficient transfer of resources.
   * Reduces unnecessary copying.
6. **Rvalue References**:
   * Distinguish between lvalues and rvalues.
   * Used in move semantics.
7. **nullptr Keyword**:
   * Null pointer literal.
   * Example: **int\* ptr = nullptr;**.
8. **Static Assertions**:
   * **static\_assert** for compile-time assertions.
   * Example: **static\_assert(sizeof(int) == 4, "int must be 4 bytes");**.
9. **New Standard Library Features**:
   * New containers, algorithms, and utilities.
   * Example: **std::unordered\_map**, **std::move**, **std::chrono**.

**C++14 Features (Building upon C++11):**

1. **Binary Literals**:
   * Literal integers can be expressed in binary.
   * Example: **int bin = 0b1010;**.
2. **Generic Lambdas**:
   * Lambda functions with auto-typed parameters.
   * Example: **auto sum = [](auto a, auto b) { return a + b; };**.
3. **Variable Templates**:
   * Templates that define variables.
   * Example: **template <typename T> constexpr T pi = T(3.1415926535897932385);**.
4. **Relaxed constexpr Functions**:
   * **constexpr** functions with non-constant expressions.
   * Example: **constexpr int fib(int n) { return n <= 1 ? n : fib(n - 1) + fib(n - 2); }**.
5. **Aggregate Member Initialization**:
   * Initializing aggregates (arrays, structs) with braces.
   * Example: **struct Point { int x, y; }; Point p = {1, 2};**.
6. **Digit Separators**:
   * Allows using single quotes as separators in numeric literals for readability.
   * Example: **int billion = 1'000'000'000;**.

**C++17 Features (Building upon C++14):**

1. **Structured Bindings**:
   * Destructuring of tuples and other structures.
   * Example: **auto [x, y] = std::make\_tuple(1, 2);**.
2. **if constexpr**:
   * Compile-time conditional statements based on template parameters.
   * Example: **template <typename T> void process(T value) { if constexpr(std::is\_integral<T>::value) { /\*...\*/ } else { /\*...\*/ } }**.
3. **Inline Variables**:
   * Definition of variables directly in the header file.
   * Example: **inline int value = 42;**.
4. **constexpr if**:
   * Conditional evaluation of code at compile time within a **constexpr** function or template.
   * Example: **template <typename T> void process(T value) { if constexpr(std::is\_integral<T>::value) { /\*...\*/ } else { /\*...\*/ } }**.
5. **std::optional**:
   * Represents an optional value.
   * Example: **std::optional<int> maybeValue;**.
6. **std::variant**:
   * Represents a type-safe union of types.
   * Example: **std::variant<int, double, std::string> var;**.
7. **std::filesystem**:
   * Classes and functions for working with file systems.
   * Example: **std::filesystem::path path = "/path/to/file.txt";**.
8. **New Standard Library Additions**:
   * Various new algorithms, utility functions, and improvements to existing library components.