Table of Contents (exploded view)

- The using declaration and class members (C++ only)
  - Overloading member functions from base and derived classes (C++ only)
  - Changing the access of a class member (C++ only)
- Multiple inheritance (C++ only)
  - Virtual base classes (C++ only)
  - Multiple access (C++ only)
  - Ambiguous base classes (C++ only)
- Virtual functions (C++ only)
  - Ambiguous virtual function calls (C++ only)
  - Virtual function access (C++ only)
- Abstract classes (C++ only)

- Special member functions (C++ only)
  - Overview of constructors and destructors (C++ only)
  - Constructors (C++ only)
    - Default constructors (C++ only)
    - Delegating constructors (C++11)
    - Constexpr constructors (C++11)
    - Explicit initialization with constructors (C++ only)
    - Initialization of base classes and members (C++ only)
    - Constructor execution order for class objects (C++ only)
  - Destructors (C++ only)
    - Pseudo-destructors (C++ only)
  - User-defined conversions (C++ only)
    - Conversion constructors (C++ only)
    - Explicit conversion constructors (C++ only)
    - Conversion functions (C++ only)
    - Explicit conversion operators (C++11)
  - Copy constructors (C++ only)
  - Copy assignment operators (C++ only)
- Templates (C++ only)
  - Template parameters (C++ only)
    - Type template parameters (C++ only)
    - Non-type template parameters (C++ only)
    - Template template parameters (C++ only)
    - Default arguments for template parameters (C++ only)
    - Naming template parameters as friends (C++11)
  - Template arguments (C++ only)
    - Template type arguments (C++ only)
    - Template non-type arguments (C++ only)
    - Template template arguments (C++ only)
  - Class templates (C++ only)
    - Class template declarations and definitions (C++ only)
    - Static data members and templates (C++ only)
    - Member functions of class templates (C++ only)
    - Friends and templates (C++ only)
  - Function templates (C++ only)
    - Template argument deduction (C++ only)
    - Overloading function templates (C++ only)
    - Partial ordering of function templates (C++ only)
  - Template instantiation (C++ only)
    - Explicit instantiation (C++ only)
    - Implicit instantiation (C++ only)
  - Template specialization (C++ only)
    - Explicit specialization (C++ only)
    - Partial specialization (C++ only)
  - Variadic templates (C++11)
  - Name binding and dependent names (C++ only)
  - The typename keyword (C++ only)
  - The template keyword as qualifier (C++ only)
- Exception handling (C++ only)
  - try blocks (C++ only)
    - Nested try blocks (C++ only)
  - catch blocks (C++ only)
    - Function try block handlers (C++ only)
    - Arguments of catch blocks (C++ only)
    - Matching between exceptions thrown and caught (C++ only)
    - Order of catching (C++ only)
  - throw expressions (C++ only)
    - Rethrowing an exception (C++ only)
  - Stack unwinding (C++ only)
  - Exception specifications (C++ only)
  - Special exception handling functions (C++ only)
    - The unexpected() function (C++ only)
    - The terminate() function (C++ only)
    - The set_unexpected() and set_terminate() functions (C++ only)
    - Example using the exception handling functions (C++ only)
- Preprocessor directives
  - Macro definition directives