

Next: [Objective-C](#), Previous: [C Extensions](#), Up: [Top](#) [[Contents](#)][[Index](#)]

## 7 Extensions to the C++ Language

The GNU compiler provides these extensions to the C++ language (and you can also use most of the C language extensions in your C++ programs). If you want to write code that checks whether these features are available, you can test for the GNU compiler the same way as for C programs: check for a predefined macro `__GNUC__`. You can also use `__GNUG__` to test specifically for GNU C++ (see [Predefined Macros](#) in *The GNU C Preprocessor*).

• <a href="#">C++ Volatiles</a> :	What constitutes an access to a volatile object.
• <a href="#">Restricted Pointers</a> :	C99 restricted pointers and references.
• <a href="#">Vague Linkage</a> :	Where G++ puts inlines, vtables and such.
• <a href="#">C++ Interface</a> :	You can use a single C++ header file for both declarations and definitions.
• <a href="#">Template Instantiation</a> :	Methods for ensuring that exactly one copy of each needed template instantiation is emitted.
• <a href="#">Bound member functions</a> :	You can extract a function pointer to the method denoted by a ‘->’ or ‘.*’ expression.
• <a href="#">C++ Attributes</a> :	Variable, function, and type attributes for C++ only.
• <a href="#">Function Multiversioning</a> :	Declaring multiple function versions.
• <a href="#">Type Traits</a> :	Compiler support for type traits.
• <a href="#">C++ Concepts</a> :	Improved support for generic programming.
• <a href="#">Deprecated Features</a> :	Things will disappear from G++.
• <a href="#">Backwards Compatibility</a> :	Compatibilities with earlier definitions of C++.

Next: [Objective-C](#), Previous: [C Extensions](#), Up: [Top](#) [[Contents](#)][[Index](#)]