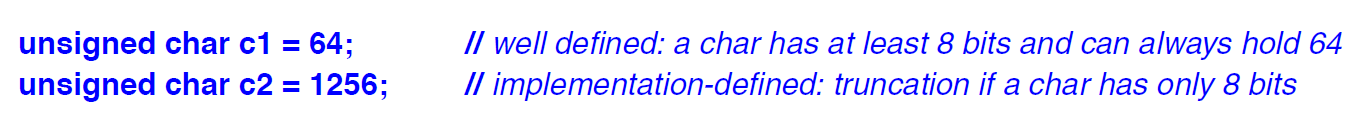
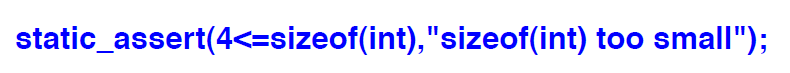
* Many important things are deemed as implementation-defined by the standard. For example,

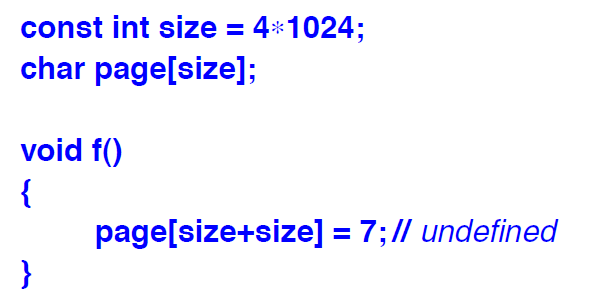


If the char has only 8 bits, then 1256 will be converted to 232. (No idea how?)

* Many assumptions about implementation-defined features can be checked by stating them as static assertions. For example,

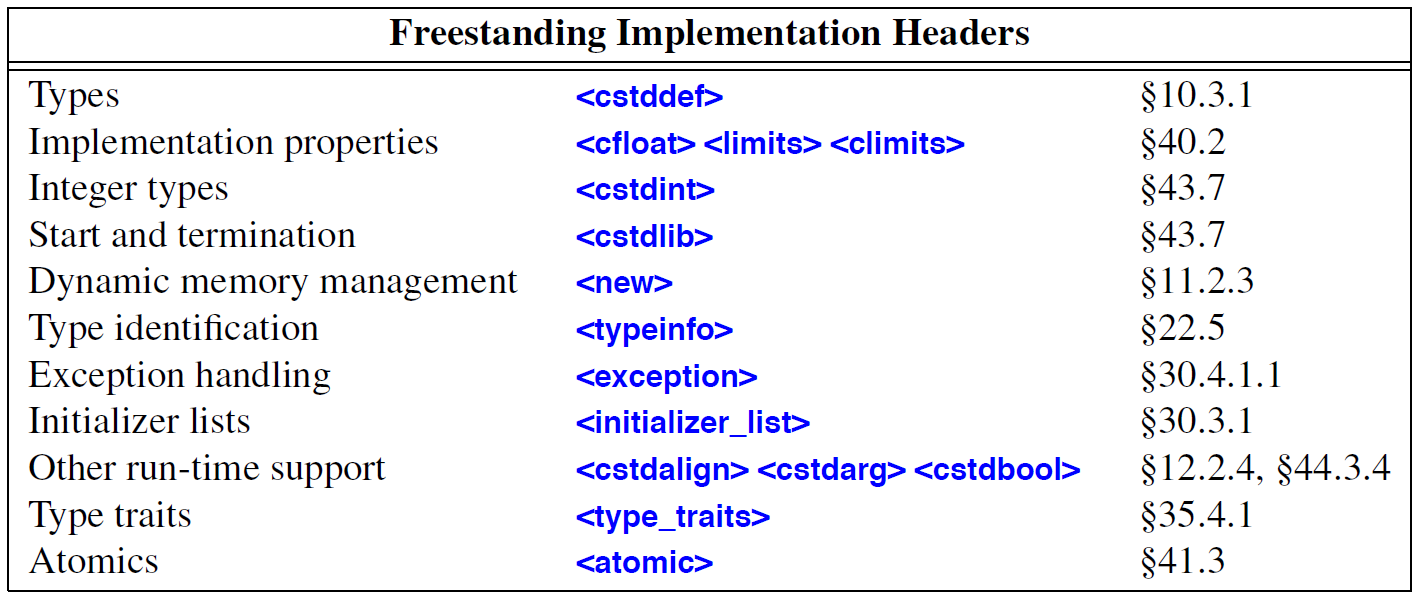


* A construct is deemed undefined by the standard if no reasonable behaviour is required by the implementation. For example,

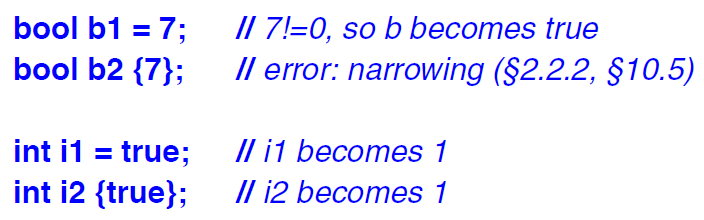


Plausible outcomes of this code fragment include overwriting unrelated data and triggering a hardware error/exception.

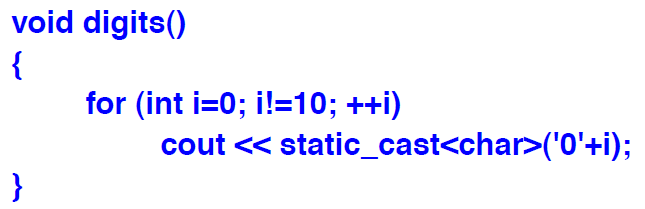
An implementation is not required to choose among plausible outcomes.



* C++ has a set of fundamental types –
* Boolean type (bool)
* Character types (char, wchar\_t )
* Integer types (int, long long)
* Floating point types (double, long double)
* Type to signify absence of information (void)
* Other types, constructed from the fundamental types, using declarator operators –
* Pointer types (int\*)
* Array types (char[])
* Reference types (double&, vector<int>&&)
* User defined additional types –
* Data structures and classes
* Enumeration types for representing specific sets of values (enum, enum class)
* **Integral types:** Boolean, character, integer types.
* **Arithmetic types:** Integral and floating-point types.
* **User-defined types:** Must be defined by users rather than being available for use without previous declaration, e.g. enums and classes.
* **Built-in types:** Fundamental types, pointers and references.
* **Booleans:** By definition, true has the value 1 when converted to integer, false has the value 0. Conversely, integers can be converted to bool values: non-zero integers convert to true and zero integers convert to false.



* **Character types:** C++ offers a variety of character types –
* *char*: Default character type. Usually 8 bits.
* *signed* *char*: Capable of holding both positive and negative values.
* *unsigned* *char*: A char that is guaranteed to be unsigned.
* *wchar\_t:* Can hold characters of a larger character set such as Unicode. Its size is implementation dependent.
* *char16\_t*: Holds 16-bit character sets, such as UTF-16.
* *char32\_t*: Holds 32-bit character sets, such as UTF-32.
* The following is an example of the use of *static\_cast –*



* The above is a program to print the 10 integer numbers, i.e. 0, 1, 2 and so on.
* By leaving out the *static\_cast*, the output will be 48, 49, 50 and so on.