1. **Constexpr** imposes a top level const on the object it defines
2. The keyword *typedef* may appear as part of the base type of a declaration(§ 2.3, p50)
3. Because a declaration can involve only a single base type, the initializers for all the variables in the declaration must have types that are consistent with each other.(using **auto type specifier**)
4. **auto** ordinarily **ignores top-level consts**. As usual in initializations, low-level consts such as when an initializer is a pointer to int. If you want the deduced type to have a top-level const, we must say so explicitly.
5. **decltype** returns the type of its operand,. The compiler analyzes the expression to determine it’s type **but does not evaluate the expression**.
6. **decltype** is the *only* context in which a variable defined as a reference is not treated as a synonym for the object to which it refers.
7. In-class initializers are restricted as to the form we can use: They must either be enclosed inside curly braces or follow an = sign. We may not specify an in-class initializer inside parentheses