1. In practice, what’s important is that a type that defines a **< operator** that “behaves normally” can be used as a key.
2. The default pair ctor **value initialized** the data members.
3. It is essential to remember that the **value\_type** of a map is a pair and that we can change the value but not the key member of that pair.
4. insert member of map return pair of iterator (**points to old value if failed** otherwise points to the new inserted value) and bool (indicating whether the element was inserted).
5. The associative containers supply an additional **erase** operation that takes a **key\_type** argument. This version removes all the elements, if any, with the given key and returns a count of how many elements were removed (0 if not found, 1 or more if found and removed).
6. Subscripting a **map** behaves quite differently from subscripting an array or vector: Using a key that is not already present add an element with that key to the map.
7. **Subscript** and **at** operations only for **map** and **unordered\_map** that are not const.
8. **find** return iterator points to the **first** capable item if found.
9. The iterator returned from **lower\_bound** may or may not refer to an element with the given key. If the key is not in the container, then **lower\_bound** refers to the first point at which this key can be inserted while preserving the element order within the container (i.e. the first element with a key larger than item to search or end()).
10. If the element is not in the multimap, then lower\_bound and upper\_bound will return equal iterators.