In a dictionary, map and other associative containers, both elements together (key and value) are commonly called key-value pairs, or a pair for short.

>that does not work

I'm assuming you looked up KeyValueRef and tried to roll your own?

```
template <class K, class V>
struct KeyValueRef {
    const K& key;
    V&      value;
    KeyValueRef(const K& key, V& value) : key(key), value(value) {}
};
```

The reason your version doesn't work is because references *must* be initialized at time of creation, they cannot be nullptr and they cannot be changed to point to something else. That's one of the major differences between a reference and a pointer - so if you look at the code above you'll see that in the constructor those references are being initialized at key(key), value(value). That's why that version works and yours doesn't.