Return Type of Emplace Methods

This is the last topic of discussion for map enhancements.

Since C++11 most of the standard containers got `.emplace*` methods. With those methods, you could create a new object in place, without additional copying or temporary objects.

However, most of `.emplace*` methods didn't return any value - it was `void`. Since C++17 this has changed, and they now return the reference type of the inserted object.

For example:

```
// since C++11 and until C++17 for std::vector
template< class... Args >
  void emplace_back( Args&&... args );
// since C++17 for std::vector
template< class... Args >
  reference emplace_back( Args&&... args );
```

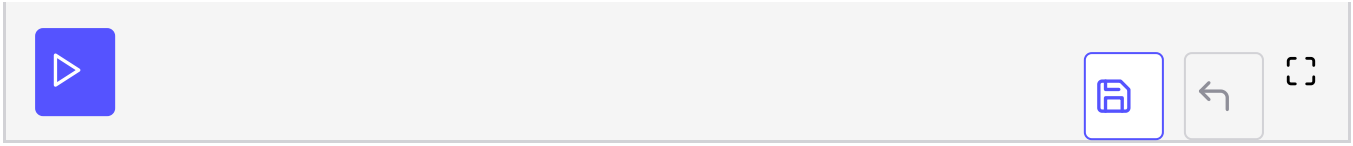
This modification should shorten the code that adds something to the container and then invokes some operation on that newly added object.

For example:

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;

int main() {
  std::vector<std::string> stringVector;
  // in C++11/14:
  stringVector.emplace_back("Hello");
  cout << "emplace_back -> " << stringVector[0] << endl;
  // emplace doesn't return anything, so back() needed
  stringVector.back().append(" World");
  cout << "append -> " << stringVector[0] << endl;
  // in C++17:
  stringVector.emplace_back("Hello").append(" World 2");
  cout << "C++17 -> " << stringVector[1] << endl;
```

```
}
```



> **Extra info:** See more information in the paper: P0084R2

That’s pretty much it for map and set enhancements. In the next lesson, we will be introduced to a new algorithm which is used in data sampling.