Std::map:

1. Inserting values into the map:
2. Insert vs Emplace
3. Operator [] behaves differently then other two.
4. Iterating over the map
5. 3rd template argument
6. Different constructors / copy constructors
7. Implementation – red black tree
8. Sorts every element. Important to remember!

Std::multimap:

1. Same as map but with multiple same keys allowed.

Std::set:

1. Difference between std::set and std::map.

Std::unordered\_map:

1. Implementation – hash table.
2. Hash table overview – hash function, dealing with hash collisions.
3. Std::unordered\_map interface. Many functions the same as in std::map.
4. Std::unordered\_map advanced functions.

Std::array:

1. More friendly syntax then C-style array.
2. Easier to read code.
3. Less chance for bugs.

Std::deque:

1. Implementation – double ended queue.
2. Interface.

Std::priority\_queue:

1. Implementation – priority queue based on heap.
2. Example of inner implementation (what is under the hood).
3. Interface.
4. Workaround for inserting values without braced initialization { }
5. Constructing out of std::vector.

Std::queue:

1. Implementation – queue.
2. Interface.
3. Different constructor.

Std::stack:

1. Implementation – stack.
2. Interface.