National Research University Higher School of Economics
Faculty of Computer Science
Bachelor's Program "HSE University and University of London Double Degree Program in Data Science and Business Analytics"

Introduction to Programming

Workshop #11

Wed 17.02.2021

Julio Carrasquel

Today's Outline

- Exercises
- Maps
- Sets
- Misc

Exercise

Let’s divide us in two groups. Both groups use the textfile: *titanic.csv*

Group 1

- Count the number of passengers per class
- Count the number of survivors per class.

Group 2

- Count the number of male and female passengers.
- Count the number of male and female survivors.

C++ STL containers

Container class templates

Sequence containers:

array C++11	Array class (class template)
vector	Vector (class template)
deque	Double ended queue (class template)
forward_list C++11	Forward list (class template)
list	List (class template)

Container adaptors:

stack	LIFO stack (class template)
queue	FIFO queue (class template)
priority_queue	Priority queue (class template)

Associative containers:

set	Set (class template)
multiset	Multiple-key set (class template)
map	Map (class template)
multimap	Multiple-key map (class template)

Unordered associative containers:

unordered_set C++11	Unordered Set (class template)
unordered_multiset C++11	Unordered Multiset (class template)
unordered_map C++11	Unordered Map (class template)
unordered_multimap C++11	Unordered Multimap (class template)

https://www.cplusplus.com/reference/stl/

Maps

Maps are *associative containers* that store *key-value elements*.

- *Dynamic* as a vector – no size needed to define before.
- *Keys* work as indexes of values (basic arrays only allow integers).
- Only allows one element with a specific *key*!

  For example, if `map` = [(‘a’ , 1)], then (‘a’, 2) cannot be inserted in `map`.

Sets

Sets are containers that store *unique elements* following a specific order.

- Sets are guaranteed to remain in a specific ordering.
- This is a difference with *vectors*. Elements in the vector are where you put them.
- As the *internal data structure*, sets are implemented with *“Red-Black-trees”*, exactly like *maps*. But, sets are just one element, whereas maps are <key,value> elements.
- Like maps, they do not allow repetition of elements!

Misc

- Iterator types
- Use of aliases