

Gebze Technical University
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2019
Homework # 6
Inheritance, Templates, STL
Due date
Dec 24th 2019

In this homework, you will write a templated class hierarchy for a simple container class hierarchy.

The class **GTUContainer** is an abstract class with the following pure virtual member functions.

<u>empty</u>	Test whether container is empty
<u>size</u>	Return container size
<u>max_size</u>	Return maximum size
<u>insert</u>	Insert element, throws exception <code>std::bad_parameter</code> if there is a problem with insertion
<u>erase</u>	Erase element
<u>clear</u>	Clear all content
<u>begin</u>	Return iterator to beginning
<u>end</u>	Return iterator to end

The class **GTUSet<T>** derives from the base class and implements all of the functions appropriately for a set class.

The class **GTUVVector<T>** derives from the base class and implements all of the functions appropriately for a vector class. This class will also overload the [] operator. You may write other helper classes to make your work easier.

All classes will keep their data using dynamic memory techniques with `shared_ptr` STL pointers. Do not use regular pointers or STL container classes.

The classes **GTUIIterator** and **GTUIIteratorConst** implement iterator operators such as *, ->, ++, --, =, and ==.

You will also implement the following global functions from the STL library which will accept GTUIterator parameters instead of regular STL iterators

- **`find`**
- **`find_if`**
- **`for_each`**

Write your driver program to test the all the classes and all of their functions. Do not forget to test the global functions with all concrete classes.

Notes

- Use separate header and implementation files for each class.
- Use name spaces.
- Do not forget to test the thrown exceptions.