Gebze Technical University
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2017
Homework # 6
Inheritance, Templates, STL
Due date
Dec 13th 2017

In this homework, you will write a templated class hierarchy for sets and maps.

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

empty	
	Test whether container is empty
size	
	Return container size
max_size	
	Return maximum size
insert	
	Insert element, throws exception std::bad_pafram if the element is already in the set
erase	
	Erase element
clear	
	Clear all content
find	
	Get iterator to element
count	
	Count elements with a specific value
begin	
	Return iterator to beginning
end	
	Return iterator to end

The class GTUSet<T> derives from the base class and implements all of the functions appropriately. It will keep its data using dynamic memory techniques with shared_ptr STL pointers. Do not use regular pointers or STL container classes.

The class GTUMap<K, V> derives from GTUSet<std::pair<K, V> > and implements the following extra function

operator[]

Access element

```
V& operator[] (const K& k);
```

If *k* matches the key of an element in the set, the function returns a reference to its mapped value.

The class GTUIterator implements iterator operators such as *, ->, ++, --, =, and ==.

You will also write the following global function

```
template <class T>
std::shared_ptr<GTUSetBase<T> > setIntersection<T> (const GTUSetBase<T>&, const GTUSetBase<T>&);
```

The returned set is the intersection of the two sets.

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

Notes

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