

# Homework 5 – Maps

C++ //

## Description

For this assignment we will use the AVL tree we created to create a map of key – value pairs.

## Specifications

You will need to complete the following:

1. Write an implementation of the map class.
2. Use the avl\_tree we created in class as a starting point.
3. The node struct will now have a key (this is the unique value in the map) and the value which may not be unique in the map. Remember trees do not allow duplicate values. In this case, a duplicate value is a duplicate key only.
4. You will need to add / change the following:
  - a. Iterators both const and non-const iterators. The iterator should start with the min key (begin) and go to max (end).
  - b. The struct will need to templated values a K key and a V value.
  - c. The struct will also need a pointer to parent so you can iterate over the map.
  - d. The remove method of the avl\_tree will remain. You will pass a key and it will remove it and its associated value from the map.
  - e. The contains method will be replaced with two methods.
    - i. contains\_key(K key) which return true if the key exists in the map, otherwise false.
    - ii. contains\_value(V value) which returns true if the value exists in the map, otherwise false.
    - iii. insert(K key, V value) if the key is already in the map, then replace its values with the one given.
  - f. Add a method get(K key) which returns the value associated with the key. If the key does not exists in the tree, then throw an exception – length\_error would be fine in the case (included in <stdexcept>)
  - g. Override the subscript [] operator so that it can be used to change a value or get a value. Something like map\_idenfier[key\_value] = value or V value = map\_idenfier[key\_value];
5. In the main method, instantiate your map class and fill it with 50 random integer and string values (map<int, std::string> students; .... then write a loop that iterates over the map both forwards and backwards. Test all the methods (including the subscript operator) to show they work as expected. When you pass the map to the stream insertion operator, it should display the key value pairs inorder of the keys (low to high).

## Documentation

You will create a document (.docx, .rtf, .pdf) which contains the following:

- Your name and assignment.
- A screenshot of your code output.
- Explain how the AVL tree keeps the map balanced.
- Why is it important keep the data structure balanced?

## What to Submit

You need to submit your C++ code files along with your document. Make sure your document is in the correct format and all your files include your name and assignment. **ZIP** your C++ code, but **DO NOT** zip your document file.