# COMP3512 - Lab Exercise 6 (Nov 6-10, 2017)

This is an exercise that you need to do on a computer. You'll need to commit and push your code to your GitLab repo, and submit for automated marking via Slack.

For this exercise, you will need to write a C++ class FixedVector using Class Templates that will be used to perform various vector operations.

## 1. Project Setup

1. Open Lab6.sln in Visual Studio 2017
2. Add FixedVector.h file to your project. (refer to Lab 1 if you don't know how)
3. Add the following content in the header file. (Please type, not copy and paste)

#pragma once

namespace lab6

{

template<class T, size\_t N>

class FixedVector

{

public:

FixedVector();

~FixedVector();

const T& Get(unsigned int index) const;

T& operator[](unsigned int index);

int GetIndex(const T& t) const;

size\_t GetSize() const;

size\_t GetCapacity() const;

bool Add(const T& t);

bool Remove(const T& t);

private:

// private variables here

};

}

1. Add FixedVector.cpp file to your project.
2. Note that the solution will not build because some methods are not returning the right types. You will have to implement them to be able to build the solution.

### Expected Behavior of FixedVector class

* FixedVector class must be able to store different data types (int, float, double, bool, char, etc).
* When being instantiated, the capacity of FixedVector is set via template parameter, not via constructor, and should not change during its lifecycle.
* Get Method takes index as a parameter and returns an element on that index. You can assume that the index will always be within range (will not be above the size of the FixedVector)

ex>

FixedVector v = [1, 2, 3, 4]

v.Get(0) returns 1

* Subscript operator ([]) has exactly the same behaviour as Get method.

ex>

FixedVector v = [1, 2, 3, 4]

V[0] returns 1

* GetIndex finds a given element, and returns the index of the first occurrence of the element. If the data is not found in the FixedVector, return -1.
* GetSize should return the current number of elements in FixedVector.
* GetCapacity should return the total number of elements (capacity) that FixedVector can store.
* Add method adds a new element at the end of the FixedVector. If it’s successfully added, return true. Else false. (ie> if the capacity of the FixedVector is full, Add should return false.)
* Remove method removes the first occurrence of a given element from the FixedVector. After removal the order of remaining elements should be kept and there should be no empty slots between remaining elements. If the element is not found, return false. Else return true.

## 2. Implement All Class Functions Introduced in Step 1

* You can test your code using something like the following code in main.cpp

#include "FixedVector.h"

#include <iostream>

int main()

{

lab6::FixedVector<int, 10> numbers;

numbers.Add(1);

numbers.Add(2);

std::cout << numbers[0] << std::endl;

return 0;

}

## 3. Commit, Push and Ask for a Build

You know the drill :)

# 