# COMP3512 - Lab Exercise 8 (Dec 4 - Dec 8, 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 Storage that has a large array of datatype T. You will need to implement proper Copy Constructor, Move Constructor, Destructor and move assignment operator.

## 1. Project Setup

1. Open Lab8.sln in Visual Studio 2017
2. Add Storage.h file to your project. (refer to Lab 1 if you don't know how)
3. Add the following content in the header file.

|  |
| --- |
| #pragma once  namespace lab8  {  template<typename T>  class Storage  {  public:  Storage(int length);  Storage(const Storage& other);  Storage(Storage&& other);  Storage& operator=(Storage&& other);  const T\* GetData() const;  int GetLength() const;  virtual ~Storage();  private:  T\* mData;    // other private variables/methods here  };  // Put your implementations here  } |

1. 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 Storage class

* The constructor takes integer length to allocate an array of T.
* The copy constructor is just a regular copy constructor. Make sure you deep copy any variables in the original.
* The move constructor takes in rvalue reference as its parameter. The temporary object referenced by rvalue reference must be transferred (the temp object’s memory should be transferred over. Value types should be set back to initial values).
* The assignment operator (=) takes in rvalue reference as its right hand side operand. The temporary object referenced by rvalue reference should be transferred as well.
* GetData returns a pointer to the first element of mData.
* Getlength returns the length of the array of T.
* Destructor is just a regular destructor that frees up the memory when the Storage object is destroyed.

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

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

|  |
| --- |
| #include "Storage.h"  int main()  {  // using move constructor  lab8::Storage<int> storage1(10000);  lab8::Storage<int> storage1Copy(std::move(storage1)); // storage1's memory should be nulled/zeroed  // using assignment operator  lab8::Storage<char> storage2(20000);  lab8::Storage<char> storage2Copy = std::move(storage2); // storage2's memory should be nulled/zeroed  // using regular copy constructor  lab8::Storage<double> storage3(10000000);  lab8::Storage<double> storage3Copy(storage3); // storage3 still has its own data  return 0;  } |

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

You know the drill :)

# 