Effective and Modern C++ Programming

Lab 2 – Smart Pointers

In the zip file you can find use cases in cpp files with the expected outputs.
Write your solutions in appropriate header files and send them to Pegaz.

Exercise 1. Move semantics for a base class

File StaticContainer.h: Change the implementation of the class Container so that appropriate constructors and assignment operators are called ("copy" call "copy", "move" calls "move").

Exercise 2. Defaulted and Deleted constructors

File StaticContainer2.h: Change the implementation of the class Container so that Container can be moved but not copied.

Exercise 3. Unique Pointers

File DynamicContainer.h:
Change the implementation of the class Pointer so that it stores pointer to dynamically allocated Box in a `unique_ptr` not as a raw pointer.
Implement move semantics for Container.

Exercise 4. Shared Pointers

File String.h: Implement a class String that uses copy-on-change design pattern.
Use `shared_ptr` to store pointer to dynamically allocated object of MyString (or std::string).
See comments in the code for more details.