I have learned the difference between move operations and copy operations. Move operations transfer all the data from one pointer to another, and the original data is removed from the initial location. While copy operations copy the data from one pointer to another, and the original data remain in the initial location without any changes.

I have learned the reason for using move operations and for using copy operations. Move operations avoid unnecessary duplicates, and are useful when we only want to keep one version of the data. While copy operations are usually used when we want to backup something. By coping, we are able to modify the copied content, and keep the original data untouched and safe.

I have learned to use dynamically allocated array to store addressed to objects. The array is declared as a pointer. The array size is not fixed at compile time, but is chosen at runtime. The declared type does not mean the type of the array elements. Instead, it means the type of the object to which each of the elements hold the address.

I have the Text class included the move constructor and move operator. In w2 module, the statement “d = std::move(a);” moves the content from a to d, and removes the content in a after the transfer is done using move constructor. The same procedure happens in “a = std::move(d);” except that “a” is not newly created and the statement is using move assignment.

I have learned to use objects to std::chrono::steady\_clock::time\_point and its member function ::now() to record time point. I have also learned to use objects to ::duration class to store a time period between 2 time points.

I have learned that std::string is a class. I can create objects to it to store C string and make use of the class’s functionality to perform operations on the stored C strings conveniently.