CSE 3150 — C++ Essentials — Yufeng Wu — Fall 2024
Programming Assignment 8: Object orientation. Due: 11:59 pm @ 11/1/2024.

In this assignment, you will implement several design patterns that were taught in the class. This assignment is about the document editor as taught in the lecture.

In this assignment, you are going to implement several classes of a text document editor. The key for this assignment is practicing object oriented programming with C++. In particular, you will implement two design patterns.

1. Model-view-controller (MVC)
2. Command

More specifically, you will need to support the following functionalities. You will be given a written document class (called ECTextDocument). What you need to do is implementing document editing operations in the document controller class (called ECTextDocumentCtrl).

1. InsertTextAt: Insert a segment of characters at a given position.
2. RemoveTextAt: Remove a segment of characters of certain length at a given position.
3. CapTextAt: Capitalize a segment of characters of certain length at a given position.
4. LowerTextAt: Lowercase a segment of characters of certain length at a given position.
5. Undo/redo of any operations you have done before. Note: your code must allow multiple undo/redo operations whenever possible. I believe you have used Undo/Redo in a word processor like MS Word. You want your code to be able to do just that (well, you won't see it visually since we don't have a UI; but you get the idea..).

You are given a set of files as starter code (along with a simple test code). Take a closer look at the provided files (the header files and the test code) to get a better idea about what you need to do. To compile the starter code, type:

g++ ECTextDocument.cpp ECCommand.cpp ECTextDocumentTest.cpp -o editor

Here is an example.

1. Invoke InsertTextAt at position 0, text="abcde". The current text is "abcde".
2. Invoke InsertTextAt at position 2, text="fgh". The current text is "abfghcde".
3. Invoke RemoveTextAt at position 3, length to remove is 1. The current text is "abfhcde".
4. Invoke CapTextAt at position 2, length to cap is 3. The current text is "abFHCde".
5. Invoke LowerTextAt at position 3, length to lower is 1. The current text is "abFhCde".