GROUP 6: Lamport

Team Members:

Mehmet Efe Caylan

Emil Rahn-Siegel
Will Crawford
Ivan Bega
Orhan Aydin

StaticString

Class Description:

A string type that enables users to define the static length of a string which provides a faster manipulation for strings without missing the base operators of a std::string type. This class aims to provide higher performance for string operations and manipulations.

Similar Classes:

Std::array
Std::string_view
std::all_of
std::atomic

Key Functions:

Better templated class,
findAll: returns indexes of all occurrences,
rFind: returns index but reversed search,
Replace: templated replace of given string, character,string_view,
Replace_if (Takes a lambda for the condition): replaces based on the given conditional lambda,
$\gg$, $\ll$
Iterators,
Insert: inserts strings, characters, string_views,
Erase: removes a given pattern for all occurrences by default if not specified any number,
Remove: removes a range of characters from the string,
Swap: swaps the values of two StaticString objects can take a lambda as condition,

Trim: removes the trailing white space,
Foreach: A lambda as a parameter conducts lambda operation for each character in the string,
Transform: As a parameter takes a lambda that takes a char and returns a modified char,
Count: Returns count of each character in the string (used for application),
Split: Takes a delimiter to return a list of Strings split by delimiter,
Compare: Takes a custom lambda function for two staticString objects.

Error Conditions:

User Errors:
Out of range: catching invalid attempts to access or modify.
Invalid size: requesting invalid size for a substring.
Invalid argument (for comparison or copy): trying to modify a string in a way that will exceeds the limit.

Programmer Error:
Out of range: trying to access or modify the static string, even if the value is valid, and in between the static size, they might be empty.

Recoverable Error:
Invalid size: requesting invalid size for a copy or other operations.
Out of memory: requesting a size larger than memory, which would make the static string slower.

Expected Challenges:

We expect to face challenges on the usage of lambda functions as a condition or return for some of the functions, since we will rely on users to make sure the right values are provided. We also think that we might have a slight challenge on templates since some of the functions have lambda functions as parameters and might require additional templates.

Other Class Projects:

Group 7, AuditedArray
Group 10, MemoryFunction

StringSet

Class Description:

Manage string collections based on StaticString class, which handles various set operations such as union, subtract, get strings with specific conditions, rules(maybe use regular expressions to define the rules), intersection, differences and similarities between two set or more.

A templated string set that lets a user choose either a static string or std::string for the set operations.

Similar Classes:

Std::set,unordered_set c++ classes similar to that.

Key Functions:

Search:
Insert multiple strings at the same time
Check if one set is a subset of another
SubstringFilter(std::string) - build a lambda and call Filter()
SizeFilter(int)
Union_with(StringSet, function): union based on function
Symmetric_difference: returns elements not common for both side
To_vector:
sort(Function): sorts the sets based on comparator and returns a set
Search: uses wildcard and returns all the possible values Ex: "a?c*"
Count_occurance: return occurrence of a substring
random_sample(int n): returns number of samples randomly from set which fit to condition
Statistic: returns statistic of set, number of string, avg length, longest, shortest strings.

Error Conditions:

User errors:
Search - passing invalid pattern
Random_sample - negative value

Recoverable errors:
Unable to allocate more memory when working with large string sets

Programmer errors:
SizeFilter - negative size
Passing invalid functions, lambda expressions, comparators

Expected Challenges:

We expect to have challenges related mostly with the invalid input from a programmer or user, as well as special-case strings like "", \0, std::empty. Also, for functions like SubstringFilter we must decide if passing an empty string argument should erase a set or do nothing.

Other Class Projects:

Project 1, Random
Project 2, Audited Vector
Project 4, Index Set