Mini-project: C++ Standard Template Library (STL)

1. Project Goal:

In this mini-project, you will practice using containers from the C++ Standard Template Library (STL), such as std::vector, std::map, and std::string. Efficient use of the STL can help improve code readability and performance.

2. Containers to Practice with:

1. std::vector

- std::vector is a dynamic array that can change size as elements are added or removed. It provides random access to elements, similar to arrays, but with dynamic resizing capabilities.
- Common operations:
  - push_back(): Adds an element to the end of the vector.
  - at(): Accesses an element at a specific index with bounds checking.
  - size(): Returns the number of elements in the vector.
  - begin()/end(): Provides iterators to the beginning and end of the vector.

2. std::map

- std::map is an associative container that stores key-value pairs in sorted order by key. Each key is unique, and the map automatically maintains the order.
- Common operations:
  - insert(): Inserts a key-value pair.
  - operator[]: Accesses or modifies the value associated with a key.
  - find(): Searches for a key and returns an iterator to it.
  - erase(): Removes a key-value pair by key.

3. std::string

- std::string is the standard class for handling sequences of characters. It provides dynamic string manipulation, unlike C-style character arrays.
- Common operations:
  - append(): Appends another string or character to the string.
  - substr(): Extracts a substring starting from a specific position.
  - find(): Searches for a substring within the string.
  - length(): Returns the number of characters in the string.

3. Example Project Structure and Steps:

Step 1: Implement a class that uses std::vector to store a list of integers. Add methods to insert values, retrieve values, and calculate the sum of the elements.

Step 2: Use std::map to store settings (e.g., configuration key-value pairs) in the class. Add methods to add, retrieve, and remove settings.

Step 3: Use std::string to store and manipulate text. Add methods to concatenate strings, search for substrings, and retrieve string lengths.

Step 4: Create a main() function to demonstrate how these containers are used in practice. Populate the vector, map, and string, and print the results to the console.