# **String Operations Program**

## **Overview**

This C++ program provides a comprehensive set of string manipulation and processing operations. It allows users to perform various tasks such as changing the case of characters, counting specific types of characters, trimming whitespace, reversing words, and more. The program is interactive, guiding users through input and operation selection, and displaying the results.

## **Enums**

### **enOperationName**

Defines the various operations that can be performed on a string.

* eFirstLetterOfEachWord
* eLowerAllString
* eUpperAllString
* eInvertAllStringCase
* eUpperFirstLetterOfEachWord
* eLowerFirstLetterOfEachWord
* eCountTypeOfDigitInString
* eCountLetterFrequancyInString
* ePrintEachWordInString
* eCountLetter
* eIsVowel
* eCountWords
* eTrimLeft
* eTrimRight
* eTrim
* eReverceWordsInString
* eReplaceWordInString
* ePunctuationsRemoved
* eConvertNumberToText

### **enWhatToCount**

Defines the types of characters to count within a string.

* All
* CapitalLetters
* SmallLetters
* PunctualSymbols
* Spaces

## **Structs**

### **stOperationInfo**

Holds the information related to a string operation.

* std::string Text: The input text.
* enOperationName OperationName: The chosen operation.
* std::string result: The result of the operation.

## **Functions**

### **void ShowMainMenuOptions()**

Displays the available string operations to the user.

### **std::string ReadInput()**

Prompts the user to enter a text string and reads it.

### **enOperationName GetUserChoice()**

Prompts the user to choose an operation and validates the input.

### **std::string FirstLetterOfEachWord(std::string str)**

Extracts and returns the first letter of each word in the input string.

### **std::string LowerFirstLetterOfEachWord(std::string str)**

Converts the first letter of each word in the input string to lowercase.

### **std::string UpperFirstLetterOfEachWord(std::string str)**

Converts the first letter of each word in the input string to uppercase.

### **char InvertCharacterCase(char Character)**

Inverts the case of a single character.

### **std::string InvertAllStringCase(std::string str)**

Inverts the case of all characters in the input string.

### **short CountTypeOfDigitInString(std::string str, enWhatToCount WhatToCount)**

Counts characters in the input string based on the specified type (e.g., all characters, capital letters, etc.).

### **enWhatToCount WhatToCount()**

Prompts the user to choose the type of characters to count and validates the input.

### **std::string CountTypeOfDigitInStringScreen(std::string str)**

Prompts the user to choose the type of characters to count and returns the count as a string.

### **std::string LoweAllString(std::string str)**

Converts all characters in the input string to lowercase.

### **std::string UpperAllString(std::string str)**

Converts all characters in the input string to uppercase.

### **char ReadCharacter()**

Prompts the user to enter a single character and validates the input.

### **short CountLetterFrequancyInString(std::string str, char Character)**

Counts the frequency of a specific character in the input string.

### **std::string CountLetterFrequancyInStringScreen(std::string str)**

Prompts the user to enter a character and returns its frequency in the input string as a string.

### **std::string EachWordInString(std::string str)**

Extracts and returns each word in the input string.

### **std::string CountLetter(std::string S1, char Letter, bool MatchCase = true)**

Counts occurrences of a specific letter in the input string, optionally case-sensitive.

### **bool IsMatchCase()**

Prompts the user to specify if the letter count should be case-sensitive.

### **std::string CountLetterScreen(std::string str)**

Prompts the user to enter a character and returns the count of the character in the input string, considering case sensitivity.

### **bool IsVowel(char Letter)**

Checks if a given character is a vowel.

### **bool IsVowelScreen()**

Prompts the user to enter a character and checks if it is a vowel.

### **std::string CountWords(std::string S1)**

Counts the number of words in the input string.

### **std::string TrimLeft(std::string str)**

Removes leading whitespace from the input string.

### **std::string TrimRight(std::string str)**

Removes trailing whitespace from the input string.

### **std::string Trim(std::string str)**

Removes both leading and trailing whitespace from the input string.

### **std::string ReverceWordsInString(std::string S1)**

Reverses the order of words in the input string.

### **std::string ReplaceWordInString(std::string str, std::string StrToReplace, std::string StrReplaceBy)**

Replaces occurrences of a word in the input string with another word.

### **std::string ReplaceWordInStringScreen(std::string str)**

Prompts the user to enter the word to be replaced and the replacement word, and returns the modified string.

### **std::string RemovePunctuations(std::string S1)**

Removes all punctuation characters from the input string.

### **std::string RemovePunctuationsScreen(std::string str)**

Prompts the user and removes all punctuation characters from the input string.

### **std::string ReadNumber()**

Prompts the user to enter a number and reads it as a string.

### **std::string ConvertNumberToText(int Number)**

Converts a numeric value to its English text representation.

### **std::string ConvertNumberToTextScreen()**

Prompts the user to enter a number and returns its English text representation.

### **std::string ChooseOperationToPerform(stOperationInfo & OperationInfo)**

Executes the operation based on user choice and returns the result.

### **void Start()**

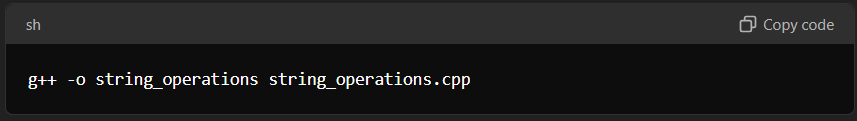
Handles the overall flow, from showing menu options to performing the chosen operation and displaying the result.

### **int main()**

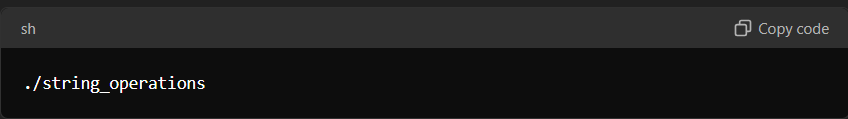
Entry point of the program, calling Start() and pausing execution.

## **Usage**

1. **Compile the Program:**



2. **Run the Program**:

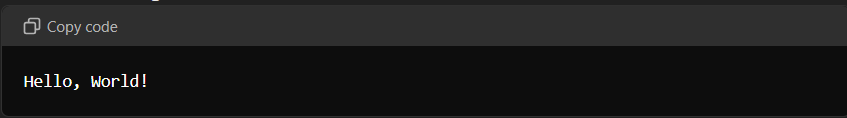


1. **Follow the On-Screen Prompts:**
   * Enter a text string when prompted.
   * Choose a string operation from the displayed menu.
   * View the result of the operation.

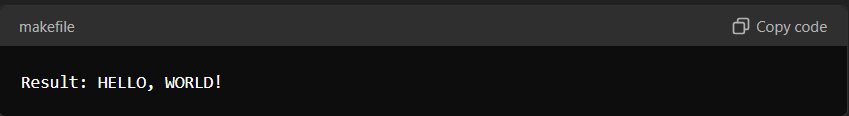
## **Example**

Here's a step-by-step example of using the program:

1. **Run the program.**
2. **Enter a text string:**

****

**Choose an operation, e.g., convert to uppercase (option 3).**

**View the result:  
**

**This program provides a versatile tool for various string manipulations, useful in many contexts, from simple text processing to more complex data handling.**