Advanced Programming

String of Characters in C++

Dr. Zeyad Safaa Younus

Dr. Zeyad Safaa Younus Saffawi

String in C++

- In **C++**, there are **two main types of string representations**:
  1. **C-Style Character Strings**
  2. **C++ String Class (string)**
- Both are used to store and manipulate text, but they differ in simplicity, safety, and functionality.

Dr. Zeyad Safaa Younus

C-Style Character Strings (character array)

- A sequence of characters stored in contiguous memory locations and treated as a single unit.
- It is implemented as an **array of characters (char)** and is used to represent text data such as letters, digits, and special symbols.
- A string may contain:
  - Alphabetic characters (A–Z, a–z)
  - Digits (0–9)
  - Special characters (e.g., *, &, (, <, +, etc.)
- A **one-dimensional array** of characters **terminated** by a **null character '\0'.**
- This special character marks the **end of the string** in memory.

Dr. Zeyad Safaa Younus

Memory Representation

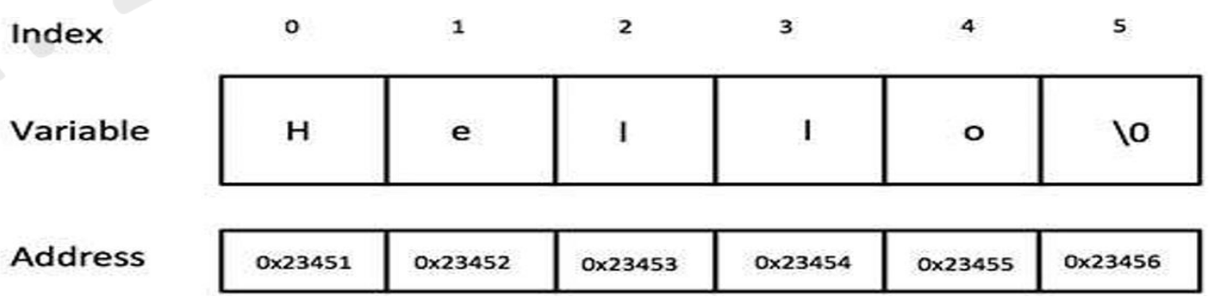
- A C-style string is internally stored as:
  - An array of characters terminated by a special character called the **null character ('\0')**.
  - The **null character** marks the **end of the string**.
- When declaring a character array to store a string:
  - The array size must be **one greater than the actual number of characters** in the string (The size must be **number of characters + 1**).
  - This extra space is required to store the null character ('\0') at the end.
- **Syntax:**

```
char    string_name[string_size];

char str[20];          // reserves space for 19 characters + '\0'
char s[11];          // can store 10 characters + '\0‘
```

- **string initialization**

```
char gr [6] = {'H','e','l','l','o','\0'};
```



Dr. Zeyad Safaa Younus

Memory Representation

- **Easier Initialization**
- Instead of writing '\0', it can write:

char gr [6] = "Hello";

- The compiler **automatically** adds **'\0'** at the end of the string when it initializes the array

Here:

- 5 letters
- 1 null character ('\0')
- Total size = 6

Dr. Zeyad Safaa Younus

Practical Example

Ex: Printing a C-Style String

```
#include <iostream>
using namespace std;
int main ()
{ char gr[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
cout << "Greeting message: "; cout << gr << endl;
 return 0;
 }
```

```
#include <iostream>
using namespace std;
int main()
{
    char gr[] = "Hello";
    cout << "Greeting message: ";
    cout << gr << endl;
    return 0;
}
```

Dr. Zeyad Safaa Younus

Common String Functions (#include <cstring>)

Function	Purpose
strcpy(s1, s2)	Copy string s2 into string s1
strcat(s1, s2)	Concatenates string s2 onto the end of string s1
strlen(s1)	Get string length s1
strcmp(s1, s2)	Compare two strings, returns 0 if s1 and s2 are the same (s1 = s2); less than 0 if s1<s2; greater than 0 if s1>s2.
strchr(s1, ch)	Find first occurrence of character. Returns a pointer to the first occurrence of character ch in string s1.
strstr(s1, s2)	Find substring. Returns a pointer to the first occurrence of string s2 in string s1.

Dr. Zeyad Safaa Younus

Example Using String Functions

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
   char str1[20] = "Hello";
   char str2[20] = "World";
   char str3[20];
   int len;
   strcpy(str3, str1);
   cout << "strcpy(str3, str1): " << str3 << endl;
   strcat(str1, str2);
   cout << "strcat(str1, str2): " << str1 << endl;
   len = strlen(str1);
   cout << "strlen(str1): " << len << endl;
   return 0;
}
```

Dr. Zeyad Safaa Younus

Write a C++ program to:

1.Read the first name and last name of a person.

2.Store the full name in a new string.

3.Print the full name.

4.Print the length of the first name and the last name using strlen().

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
   char fir[10], las[10], fuln[25];
   int len1, len2;
   cout << "Enter first name: ";
   cin.getline(fir, 10);
   cout << "Enter last name: ";
   cin.getline(las, 10);
   // Copy first name to full name
   strcpy(fuln, fir);
   // Add space between names
   strcat(fuln, " ");
   // Add last name
   strcat(fuln, las);
   cout << "Full name: ";
   cout << fuln << endl;
   // Calculate lengths
   len1 = strlen(fir);
   len2 = strlen(las);
   cout << "Length of first name: " << len1 << endl;
   cout << "Length of last name: " << len2 << endl;
   return 0;
}
```

Dr. Zeyad Safaa Younus

Write a C++ program to:
Read a line of characters and count the total number of:
Small letters
Capital letters
Digits
Space characters

```
#include <iostream>
using namespace std;
int main()
{
   char lin[100];
   int lc = 0, upc = 0, dig = 0, sp = 0;
   cout << "Enter a line: ";
   cin.getline(lin, 100);
   for(int i = 0; lin[i] != '\0'; i++)
   {
      if(lin[i] >= 'a' && lin[i] <= 'z')
         lc++;
      else if(lin[i] >= 'A' && lin[i] <= 'Z')
         upc++;
      else if(lin[i] >= '0' && lin[i] <= '9')
         dig++;
      else if(lin[i] == ' ')
         sp++;
   }
   cout << "\nSmall letters: " << lc << endl;
   cout << "Capital letters: " << upc << endl;
   cout << "Digits: " << dig << endl;
   cout << "Spaces: " << sp << endl;
   return 0;
}
```

Dr. Zeyad Safaa Younus

Write a C++ program to:
Read a string from the user.
Exchange (swap) the first character with the last character.
Print the modified string.

```
#include <iostream>
#include <cstring> // for strlen
using namespace std;
int main()
{
   char s[50];  // String array with enough space
   char temp;
   cout << "Enter the string: ";
   cin.getline(s, 50);
   // Swap first and last characters
   temp = s[0];
   s[0] = s[strlen(s) - 1];
   s[strlen(s) - 1] = temp;
   cout << "Modified string: " << s << endl;
   return 0;
}
```

Dr. Zeyad Safaa Younus

Write a C++ program to:
Read 5 names, each with a maximum of 10 characters.
Print all names that begin with the letter 'a' or ‘z'.

```
#include <iostream>
#include <cstring>                    // for string handling functions
using namespace std;
int main()
{   char names[5][11];                // 10 chars + 1 for null character
   int i;
   // Input 5 names
   for(i = 0; i < 5; i++)
   {
      cout << "Enter name " << i + 1 << ": ";
      cin.getline(names[i], 11);
   }
   cout << "\nNames beginning with 'a' or ‘z':\n";
   // Check first character of each name
   for(i = 0; i < 5; i++)
   {
      if(names[i][0] == 'a' || names[i][0] == ‘z')
      {
         cout << "Name " << i + 1 << ": ";
         cout << names[i] << endl;
      }
   }
   return 0;
}
```

Dr. Zeyad Safaa Younus

The C++ String Class

- The standard C++ library provides a string class type that supports all the operations, additionally much more functionality.
- Header file:

```
#include <string>
```

Advantages of string Class

- Automatic memory management
- Safer
- Easier concatenation
- Built-in functions
- No need for '\0'

Dr. Zeyad Safaa Younus

Example Using string

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
   string str1 = "Hello";
   string str2 = "World";
   string str3;
   int len;
   str3 = str1;
   cout << "str3: " << str3 << endl;
   str3 = str1 + str2;
   cout << "str1 + str2: " << str3 << endl;
   len = str3.size();
   cout << "str3.size(): " << len << endl;
   return 0;
}
```

Write a C++ program that:

- Creates two strings: "Hello" and "World".
- Copies the first string into another variable and prints it.
- Combines the two strings together and prints the result.
- Prints the length of the final string.

Dr. Zeyad Safaa Younus

Important String Class Functions

Function	Description
.size()	Returns length of string (Number of characters)
.length()	Returns length of string. It is an alias for the .size() method, and both functions return the exact same value.
.append()	Add text (characters or string at the end of an existing string)
.substr()	It is used to extract a substring from the given string
.find()	It is used to search inside string and find the first occurrence of a substring in the given string
.replace()	It is used to replace part of string with different text.

Dr. Zeyad Safaa Younus

Example:

```
#include <iostream>
#include <string>  // for string class
using namespace std;
int main() {
    string str1 = "Hello";
    string str2 = "World";
    string str3;
    cout << "Length of str1 (size): " << str1.size() << endl;
    cout << "Length of str1 (length): " << str1.length() << endl;
    str3 = str1;       // copy str1 to str3
    str3.append(" ");  // add space
    str3.append(str2);  // append str2
    cout << "After append: " << str3 << endl;
     return 0;
}
```

Write a C++ program that:

- Creates two strings "Hello" and "World".
- Prints the length of the first string using two different functions.
- Copies the first string into another string.
- Adds a space and then adds the second string to it using append.
- Prints the final result.

Dr. Zeyad Safaa Younus

Strings Input in C++

- **Reading Strings**
- In C++, **C-style strings** (character arrays) can be read using several built-in functions. The key functions are cin.getline().

Syntax :

cin.getline(arg1, arg2);

- arg1 → array name (where the string will be stored)
- arg2 → size of the array (maximum number of characters including the null character \0)

Example 1:

```
#include <iostream>
using namespace std;
int main() {
   char str[20];
   cout << "Enter a string: ";
   cin.getline(str, 20);  // read a full line
   cout << "You entered: " << str << endl;
   return 0;
}
```

Notes:

- cin.getline() reads the entire line, including spaces.
- It stops reading when either the newline character is found or the size limit is reached.

Dr Zeyad Safaa Younus

Strings Output in C++

Printing Strings

1. **cout** in C++:

```
cout << str;
```

2. **puts()** in C++ with <cstdio>:

```
puts(string_name);          // prints string and adds a newline
```

Example:

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    char s[5];
    cin.getline(s, 5);          // ex: input: soft
    puts(s);                    //       output: soft
    return 0;
}
```

Dr. Zeyad Safaa Younus

Convert Input Characters to Uppercase

```
#include <iostream>
#include <cstdio>
using namespace std;
int main() {
   char s[10];
   cin.getline(s, 10);
for(int i = 0; s[i] != '\0'; i++)
{
      // if character is lowercase, convert it to uppercase
      if(s[i] >= 'a' && s[i] <= 'z')
{
         s[i] = s[i] - 32;            // convert to uppercase
      }
   }
   puts(s);                          // print string
   return 0;
}
```

Dr. Zeyad Safaa Younus

Convert uppercase letters to lowercase and lowercase letters to uppercase.

```
#include <iostream>
using namespace std;
int main()
{
    char s[100];
     cout << "Enter a string: ";
    cin.getline(s, 100);
for (int i = 0; s[i] != '\0'; i++)
{
        if (s[i] >= 'A' && s[i] <= 'Z')
        s[i] = s[i] + 32;
         else if (s[i] >= 'a' && s[i] <= 'z')
         s[i] = s[i] - 32;
}
     cout << "Converted string: " << s << endl;
    return 0;
}
```

Dr. Zeyad Safaa Younus

References

- Gaddis, T. (2014). *Starting out with C++: From control structures through objects* (8th ed.). Pearson.
- Soulié, J. (2007, April 24). C++ language tutorial. cplusplus.com.
- Tutorials Point. (n.d.). *Learn C++ programming language*. Tutorials Point.

Dr. Zeyad Safaa Younus