

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
Department of Electrical & Computer Engineering

ECE 150 *Fundamentals of Programming*

# C-style strings

Prof. Hiren Patel, Ph.D.  
Prof. Werner Diel, Ph.D.  
Douglas Wilhelm Harder, M.Math. 1994

© 2018 by Douglas Wilhelm Harder and Hiren Patel. All rights reserved.

CC BY NC SA

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
Department of Electrical & Computer Engineering

C-style strings 2

## Outline

- In this lesson, we will:
  - Describe what a string is
  - Look at how to store a string in a character array
  - See some of the limitations of the choice for C-style strings
  - Look at two functions that work with C-style strings

CC BY NC SA

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
Department of Electrical & Computer Engineering

C-style strings 3

## Strings

- A string is a sequence of characters meant to represent an expression in a language
  - All entries of a string come from a fixed alphabet
  - It makes sense that this could be stored as an array
- If the phrase is in English, we can use a character array
  - Creating strings from other languages requires Unicode
  - New-line and tab characters ('`\n`' and '`\t`') could be used for some basic formatting
- Additionally, we may want to make changes to the string:
  - Adding a few characters
  - Delete a few characters
  - Replace a character

CC BY NC SA

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
Department of Electrical & Computer Engineering

C-style strings 4

## Strings

- What strategy would you use to store such a string?
  - We're definitely going to use an array of characters

```
int main() {
    // Initialize the first eight characters with
    // "Good day"
    // - all other characters are set to '\0'
    // that is, the null character or 0b00000000
    char phrase[100]{'G', 'o', 'o', 'd', ' ', ' ', 'd', 'a', 'y'};

    // Use this character array

    return 0;
}
```

- How do we determine the end of the string?

CC BY NC SA

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
FACULTY OF INFORMATION AND COMMUNICATIONS

C-style strings 5

## Strings

- Question: How do you record how long the actual string is?
  - That is, which characters actually make up the string in question
- One solution: Have a second local variable:

```
int main() {
    char phrase[100]{'G', 'o', 'o', 'd', ' ', 'd', 'a', 'y'};
    unsigned int phrase_length(8);

    // Use this character array

    return 0;
}
```

- Problem: Now you must always pass around two variables



UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
FACULTY OF INFORMATION AND COMMUNICATIONS

C-style strings 7

## C-style strings

- A C-style string does exactly that
  - The null character '\0' is used to designate the end of the string
  - The null character is not part of the string
    - It is used to mark the end of the string
  - All other characters after the null character are ignored
  - If a character array does not have a null character, it is not a string
- When calculating the *length* of a string, the null character is not counted
  - This string has a length of eight
 

```
char phrase[100]{'G', 'o', 'o', 'd', ' ', 'd', 'a', 'y'};
```
  - The capacity of the character array is 100

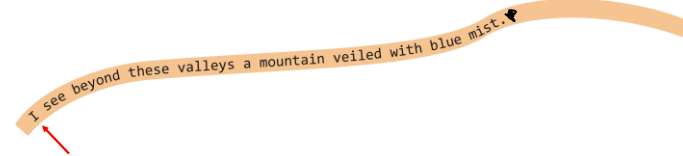


UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
FACULTY OF INFORMATION AND COMMUNICATIONS

C-style strings 6

## Strings

- Suppose you can write a message to a friend on a tape
  - You can indicate where the message starts
  - Question: does it end with the '.', or does it continue?



- A reasonable solution is to use something not normally used to flag the end of the message
  - For example, a character not expected to appear in a string



UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
FACULTY OF INFORMATION AND COMMUNICATIONS

C-style strings 8

## C-style strings

- Fortunately, the default character is the null character:

```
int main() {
    // This represents the empty string "" as the first
    // character is the null character
    char phrase[100]{};

    // Use this character array

    return 0;
}
```

'\0'	'\0'	'\0'	'\0'	'\0'	'\0'	'\0'	'\0'	'\0'	'\0'	'\0'	'\0'	...
------	------	------	------	------	------	------	------	------	------	------	------	-----



UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
FACULTY OF INFORMATION TECHNOLOGY

C-style strings 9

## C-style strings

G	o	d	d	y	\0	y	\0	\0	\0	...
---	---	---	---	---	----	---	----	----	----	-----

- You can now manipulate strings:

```
int main() {
    // Initialize the first eight characters with
    // "Good day"
    // - all other characters are set to '\0'
    // that is, the null character or 0b00000000
    char phrase[100]{'G', 'o', 'o', 'd', ' ', 'd', 'a', 'y'};
    std::cout << phrase << std::endl;

    phrase[1] = '\0';
    phrase[2] = 'd';
    phrase[3] = 'a';
    phrase[4] = 'y';
    phrase[5] = '!';
    phrase[6] = '\0';

    std::cout << phrase << std::endl;

    return 0;
}
```

Output:  
Good day  
G'day!

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
FACULTY OF INFORMATION TECHNOLOGY

C-style strings 10

## Warnings

- It is important to remember that a space ' ' is different from the null character '\0'
- If you print a character array that does not have a null character, the print mechanism will just keep printing the subsequent bytes until it finds one byte that is all zeros

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
FACULTY OF INFORMATION TECHNOLOGY

C-style strings 11

## Odd behavior: Example 1

- Explain the output of this program:

```
#include <iostream>

// Function declarations
int main();

// Function definitions
int main() {
    char phrase_1[7]{' ', 't', 'h', 'e', 'r', 'e'};
    char phrase_2[5]{'H', 'e', 'l', 'l', 'o'};

    std::cout << phrase_2 << std::endl;

    return 0;
}
```

Output:  
Hello there

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
FACULTY OF INFORMATION TECHNOLOGY

C-style strings 12

## Odd behavior: Example 2

- Even better, explain this program:

```
#include <iostream>

// Function declarations
int main();

// Function definitions
int main() {
    int data[3]{ 1831291168, 1651458592, 33 };
    // int data[3]{541665133, 541224802, 553648128};

    char phrase[6]{'H', 'e', 'l', 'l', 'o', ','};

    std::cout << phrase << std::endl;

    return 0;
}
```

Output:  
Hello, I'm Bob!

Alternative output:  
Hello,m'I boB

UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

C-style strings 13

## Length of a string

- We can author a function to calculate the length of a string:

```
unsigned int string_length( char str[] );

unsigned int string_length( char str[] ) {
    for ( unsigned int k{0}; true; ++k ) {
        if ( str[k] == '\0' ) {
            return k;
        }
    }
    assert( false );
}
```



UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

C-style strings 14

## Comparing two strings

- We can compare if two strings are equal:

```
bool string_compare( char str0[], char str1[] );

bool string_compare( char str0[], char str1[] ) {
    for ( unsigned int k{0}; true; ++k ) {
        if ( str0[k] != str1[k] ) {
            return false;
        } else if ( str0[k] == '\0' ) {
            return true;
        }
    }
    assert( false );
}
```



UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

C-style strings 15

## Mistakes with C-style strings

- What is wrong with this program?

```
#include <iostream>

// Function declarations
int main();

// Function definitions
int main() {
    char string[3];
    string[0] = 'H';
    string[1] = 'i';
    string[2] = '!';
    string[3] = '\0';

    std::cout << string << std::endl;

    return 0;
}
```



UNIVERSITY OF WATERLOO  
FACULTY OF ENGINEERING  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

C-style strings 16

## Summary

- Following this lesson, you now
  - Understand the difference between a character array and a C-style string
    - A C-style string is a character array with the string being all characters up to but not including the first null character
  - Are aware that you must be careful that the capacity of the array is at least one greater than the length of the string being stored
  - Have looked at a number of functions that look at C-style strings
  - Later, once we see classes, we will investigate the C++ `std::string`





## References

- [1] Wikipedia:  
[https://en.wikipedia.org/wiki/Kahlil\\_Gibran](https://en.wikipedia.org/wiki/Kahlil_Gibran)



## Disclaimer

These slides are provided for the ECE 150 *Fundamentals of Programming* course taught at the University of Waterloo. The material in it reflects the authors' best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.



## Colophon

These slides were prepared using the Georgia typeface. Mathematical equations use Times New Roman, and source code is presented using Consolas.

The photographs of lilacs in bloom appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens on May 27, 2018 by Douglas Wilhelm Harder. Please see

<https://www.rbg.ca/>

for more information.

