

Remember "Reviewing C++" has many more examples and topics.

Tutorial 10 **Character Sequences**

Basics

In C++, the string class is provided to help us manipulate strings and characters. A string can also be thought of as a sequence of characters. In C++, a character is denoted in between single quotes while a string is denoted in between double quotes.

Since we now know about arrays (tutorial 7), we can think of a string as an array of characters. Every character sequence in C++ is terminated by the null character, '\0' It is a back-slash and the number zero, not the letter o. This tells the program that a specific sequence is finished.

Below are some ways to declare a character array:

Example 1:

```
char arr[20];
```

Will declare an array of type char that will have a capacity of 20 individual characters.

Example 2:

```
char arr[] = "Hello";
```

Will declare an array of type character with a capacity to be determined by C++ (thats why there are empty brackets).

Example 3:

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

Will manually declare an array of characters that are terminated by the null character.

Let's observe what is going on here. Example 1 is the most common way of declaring a character sequence. It will reserve a space 20 characters long in memory, just like an array.

Example 2 will, internally, break up the string into individual characters. The length of the array will be 6 because of the null character and the initial 5 characters. The indices of the array will go from 0 to 5.

Example 3 will manually declare the array into broken up pieces. The same applies for the indices as example 2.

Remember "Reviewing C++" has many more examples and topics.

Example 1: Sample program

```
#include <iostream>
#include <string>
using namespace std;

int main(){

    char welcome[] = "Welcome to the program!\n";
    char message[] = "Enter your name: ";
    char name[100];

    cout << welcome << message;
    cin.getline(name,100);

    string yourName = name;

    cout << "Thank you " << yourName << endl;

    return 0;
}
```

Let's observe the above program. At first, we simply declare the arrays we are using. Here, we are using an array called `welcome[]` and an array called `message`. The `name` array will be used to store the name entered by the user.

When it is time to enter the user's name, we use the `getline()` function from the `cin` stream. We give it the name of the character array first, here *name*, followed by the capacity of the array, here 100.

The final part is a conversion from the character array `name` to a string called *yourName*. This is perfectly fine in C++ as each string can be thought of as a sequence of characters.

The very end prints out a thank you message and ends the program.