FAST National University of Computer and Emerging Sciences Peshawar

OOP Lab # 2.2

DEPARTMENT OF COMPUTER SCIENCE

C++ Programming

Computer Instructor: Muhammad Abdullah Orakzai

Prepared By: Muhammad Abdullah Orakzai (Lab Instructor CS)

Contents

C++ Strings

Strings are used for storing text.
A string variable contains a collection of characters surrounded by double quotes:

Example
Create a variable of type string and assign it a value:

```
string greeting = "Hello";
```

C++ Strings…

To use strings, you must include an additional header file in the source code, the <string> library:

Example

```
// Include the string library
#include <string>

// Create a string variable
string greeting = "Hello";
```

C++ Strings…

```
#include <iostream>
#include <string>
using namespace std;
int main() {
  string greeting = "Hello";
  cout << greeting;
  return 0;
}
```

Output: Hello

String Concatenation

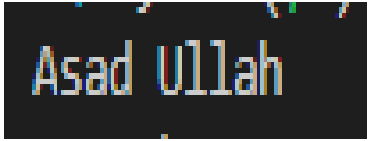
The + operator can be used between strings to add them together to make a new string. This is called **concatenation**:

Example

```
string firstName = "John ";
string lastName = "Doe";
string fullName = firstName + lastName;
cout << fullName;
```

String Concatenation…

```
#include<iostream>

using namespace std;

int main()
{

    string firstName="Asad ";
    string lastName="Ullah";

    string fullName =firstName+lastName;
    cout<<fullName;

}
```



String Concatenation…

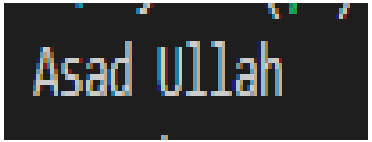
In the example above, we added a space after firstName to create a space between John and Doe on output. However, you could also add a space with quotes (" " or ' '):

Example

```
string firstName = "John";
string lastName = "Doe";
string fullName = firstName + " " + lastName;
cout << fullName;
```

String Concatenation...

```
#include<iostream>

using namespace std;

int main()
{

    string firstName="Asad";
    string lastName="Ullah";

    string fullName =firstName+" "+lastName;
    cout<<fullName;

}
```



Append

- A string in C++ is actually an object, which contain functions that can perform certain operations on strings. For example, you can also concatenate strings with the append() function:

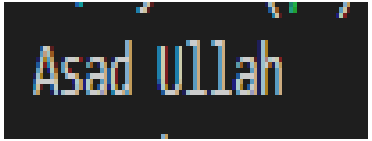
Example

```
string firstName = "John ";
string lastName = "Doe";
string fullName = firstName.append(lastName);
cout << fullName;
```

- It is up to you whether you want to use + or append(). The major difference between the two, is that the append() function is much faster.
- However, for testing and such, it might be easier to just use +.

Append

```
#include<iostream>

using namespace std;

int main()
{

    string firstName="Asad ";
    string lastName="Ullah";

    string fullName =firstName.append(lastName);
    cout<<fullName;

}
```



Adding Numbers and Strings

WARNING!

C++ uses the + operator for both **addition** and **concatenation**.

Numbers are added. Strings are concatenated.

If you add two numbers, the result will be a number:

Example

```
int x = 10;
int y = 20;
int z = x + y;      // z will be 30 (an integer)
```

Adding Numbers and Strings...

If you add two strings, the result will be a string concatenation:

Example

```
string x = "10";
string y = "20";
string z = x + y;   // z will be 1020 (a string)
```

Adding Numbers and Strings...

If you try to add a number to a string, an error occurs:

Example

```
string x = "10";
int y = 20;
string z = x + y;
```

C++ String Length

To get the length of a string, use the length() function:

Example

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

int main() {
  string txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  cout << "The length of the txt string is: " << txt.length();
  return 0;
}
```

The length of the txt string is: 26

C++ String Length…

Tip: You might see some C++ programs that use the size() function to get the length of a string. This is just an alias of length(). It is completely up to you if you want to use length() or size():

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

int main() {
  string txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  cout << "The length of the txt string is: " << txt.size();
  return 0;
}
```

```
The length of the txt string is: 26
```

Access Strings

You can access the characters in a string by referring to its index number inside square brackets [].
This example prints the **first character** in **myString**:

Example

```
string myString = "Hello";
cout << myString[0];
// Outputs H
```

Note: String indexes start with 0:

[0] is the first character.

[1] is the second character, etc.

Changing String Characters

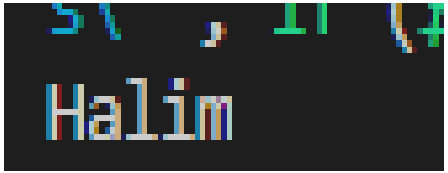
To change the value of a specific character in a string, refer to the index number, and use single quotes:

Example

```
string myString = "Hello";
myString[0] = 'J';
cout << myString;


// Outputs Jello instead of Hello
```

Changing String Characters...

```
#include<iostream>

using namespace std;

int main()
{

    string myString="Aalim";

    myString[0]='H';

    cout<<myString;
    return 0;

}
```



User Input String

It is possible to use the extraction operator >> on cin to display a string entered by a user:

Example

```
string firstName;
cout << "Type your first name: ";
cin >> firstName; // get user input from the keyboard
cout << "Your name is: " << firstName;

// Type your first name: John
// Your name is: John
```

User Input String…

However, cin considers a space (whitespace, tabs, etc) as a terminating character, which means that it can only display a single word (even if you type many words):

Example

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

int main()
{
    string fullName;
    cout<<"Enter full name: ";
    cin>>fullName;
    cout<<"Your name is: "<<fullName;
    return 0;
}
```

```
Enter full name: Muhammad Uzair
Your name is: Muhammad
```

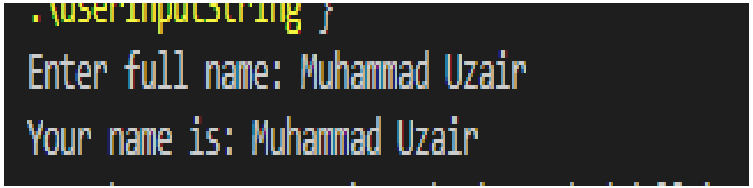
User Input String…

From the example above, you would expect the program to print "John Doe", but it only prints "John".

That's why, when working with strings, we often use the getline() function to read a line of text. It takes cin as the first parameter, and the string variable as second:

User Input String...

```
#include<iostream>

using namespace std;

int main()
{

    string fullName;
    cout<<"Enter full name: ";
    getline(cin, fullName);
    cout<<"Your name is: "<<fullName;
    return 0;
}
```



References

- https://beginnersbook.com/2017/08/cpp-data-types/
- https://www.geeksforgeeks.org/c-data-types/
- http://www.cplusplus.com/doc/tutorial/basic_io/
- https://www.geeksforgeeks.org/basic-input-output-c/
- https://www.w3schools.com/cpp/default.asp
- https://www.javatpoint.com/cpp-tutorial

THANK YOU