

C++ Strings

Data Processing Course,
I. Hrivnacova, IJCLab Orsay

- Using strings
- Manipulating strings as fundamental types (using operators)
- Manipulating strings as objects (using object-oriented syntax)
- Useful string operations

C++ Strings



- In C++, there are no built-in types for the use of strings.
- The `string` type is provided by the standard library
 - String classes enable you to use strings as normal types that cause no problems for the user
 - Different from “ordinary strings” of type `char*` or `const char*`
 - Copy, assignment and compare functions, as for fundamental types, are already available - no worry about memory corruption
- Defined in the header `<string>`
- Type: `string` = chain of characters of `char` type

```

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

int main() {
    // Create two strings
    string firstName = "bjarne";
    string lastName = "stroustrup";
    string name;

    // Manipulate strings
    firstName[0] = 'B';
    lastName[0] = 'S';

    // Chain strings
    name = firstName + " " + lastName;

    // Compare strings
    if ( name != "" ) {
        cout << name
             << " is founder of C++" << endl;
    }
}

```

Using Strings - I

- `firstName`, `lastName`
 - Variables of type `string` (in view of traditional programming)
 - Objects or instances of the *class* `string` in view of object-oriented programming)
- In object-oriented programming types are called *classes*
- The strings can be manipulated with operators as fundamental types

Using Strings - I

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

int main() {
    // Create two strings
    string firstName = "bjarne";
    string lastName = "stroustrup";
    string name;

    // Manipulate strings
    firstName[0] = 'B';
    lastName[0] = 'S';

    // Chain strings
    name = firstName + " " + lastName;

    // Compare strings
    if ( name != "" ) {
        cout << name
             << " is founder of C++" << endl;
    }
}
```

Program output:

```
> ./useStrings
```

```
Bjarne Stroustrup is founder of C++
```

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

```
int main() {
    // Create a string
    string name = "Agatha";
    // Append it with another string
    → name.append("Christie");

    // Find position of 'Christie'
    size_t idx = name.find("Christie");
    // Insert a space
    → name.insert(idx, " ");

    // Find position of the first 'i'
    size_t idxi = name.find("i");
    // Erase this character
    → name.erase(idxi, 1);

    // Print the result
    cout << name << endl;
}
```

Using Strings - II

- Object-oriented syntax for operations that are defined for string objects

objectName.function()

- This kind of function is called a *member function* in C++
- The effect of these functions depends on the implementation of these operations in the `string` class

- **size_t** – unsigned integer type used in standard library for array indexing and loop counting



Using Strings - II

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

```
int main() {
    // Create a string
    string name = "Agatha";
    // Append it with another string
    → name.append("Christie");

    // Find position of 'Christie'
    size_t idx = name.find("Christie");
    // Insert a space
    → name.insert(idx, " ");

    // Find position of the first 'i'
    size_t idxi = name.find("i");
    // Erase this character
    → name.erase(idxi, 1);

    // Print the result
    cout << name << endl;
}
```

Program output:
> ./useStrings2
Agatha Christie

C++ String and char Types



- Type: string = chain of characters of char type
- Note "a" represents a string, 'a' represent a character

```
// Loop over characters of a string
string name = "agatha";
for (size_t i = 0; i < name.length(); ++i) {
    char c = name[i];
    → cout << i << "th element of name is " << c << endl;
}
```

```
// Append a string to a string
name.append("Christi");
// Append a character ('e') to a string once
name.append(1, 'e');
// Append a character ('x') to a string 5x
→ name.append(5, 'x');
// Change the first character to upper case
name[0] = toupper(name[0]);
```

C++ String and char Types



- Type: string = chain of characters of char type
- Note "a" represents a string, 'a' represent a character

```
// Loop over characters of a string
```

```
string name = "agatha";
```

```
for (size_t i = 0; i < name.length(); ++i) {
```

```
    char c = name[i];
```

```
    cout << i << "th element of name is " << c << endl;
```

```
}
```

```
// Append a string to a string
```

```
name.append("Christi");
```

```
// Append a character ('e') to a string once
```

```
name.append(1, 'e');
```

```
// Append a character ('x') to a string 5x
```

```
name.append(5, 'x');
```

```
// Change the first character to upper case
```

```
name[0] = toupper(name[0]);
```

Program output:

```
./useStrings
```

```
0th element of name is a
```

```
1th element of name is g
```

```
2th element of name is a
```

```
3th element of name is t
```

```
4th element of name is h
```

```
5th element of name is a
```

```
AgathaChristiexxxxx
```


Useful String Operations

<code>=, assign()</code>	Assign new value
<code>swap()</code>	Swap values between two strings
<code>+=, append(), push_back()</code>	Append characters
<code>insert()</code>	Insert characters
<code>erase()</code>	Erase characters
<code>clear()</code>	Erase all characters (empty string)
<code>resize()</code>	Change number of characters (erase or add characters at the end)
<code>replace()</code>	Replace characters
<code>+</code>	Create sum string

Useful String Operations

<code>==, !=, <, <=, >, >=, compare()</code>	Compare two strings
<code>length(), size()</code>	Return number of characters
<code>empty()</code>	Test whether a string is empty
<code>[], at</code>	Access a single character
<code>>>, getline()</code>	Feed in string
<code><<</code>	Output string
<code>find functions</code>	Search for a part string or character
<code>substr()</code>	Issue a part string
<code>c_str()</code>	Use string as C-string
<code>copy()</code>	Copy a character into a character buffer

- An overview of string operations: see also C++ book, chapter 3.4.3, page 67