

§6 String

ENGG1111

Computer Programming and Applications

Dirk Schnieders

Outline

- Introduction
- Char Array vs String
- Subscript Operator
- String Concatenation
- I/O with String
- Functions

Introduction

- C++ provides a string type for easy string manipulations
- To use the string, simply include the header file <string> using the include directive
 - Note that #include <string> appears in iostream and it is not necessary to include string again if you already include iostream
- Unlike char array, we can assign text values to a string variable after its declaration

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string name = "Dirk";
    cout << name << endl;
}
```

Char Array vs String

- We can assign a char array to a string
- Simply use the assignment operator to assign the text represented by the char array to the string
- We can assign a string to a string
- Use the assignment operator to assign the value of a string variable to another string variable

```
#include <iostream>
using namespace std;
int main() {
    string name1, name2, name3;
    name1 = "Dirk";
    char a[] = "George";
    name2 = a;
    name3 = name1;
    cout << name1 << endl;
    cout << name2 << endl;
    cout << name3 << endl;
}
```

Subscript Operator

- Each individual character in a string can be accessed using its index with the subscript operator []

```
#include <iostream>
using namespace std;
int main() {
    string name = "Dirk";
    cout << name[0];
    cout << name[1];
    cout << name[2];
    cout << name[3] << endl;
}
```

String Concatenation

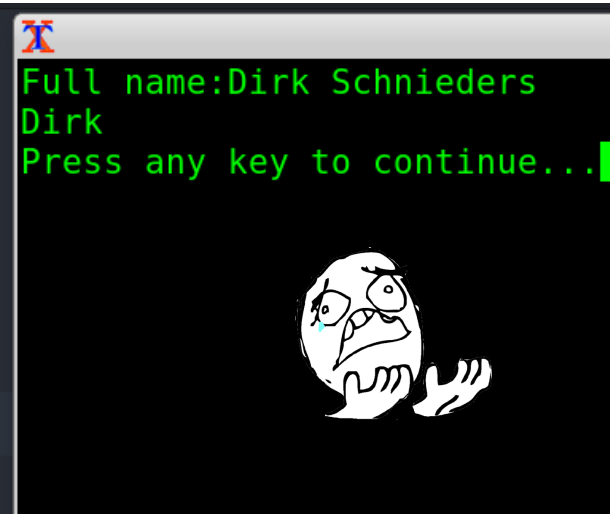
- You can create a new string by concatenating two strings

```
#include <iostream>
using namespace std;
int main() {
    string a = "a";
    string b = "b";
    string result = a+b;
    cout << result << endl;
}
```

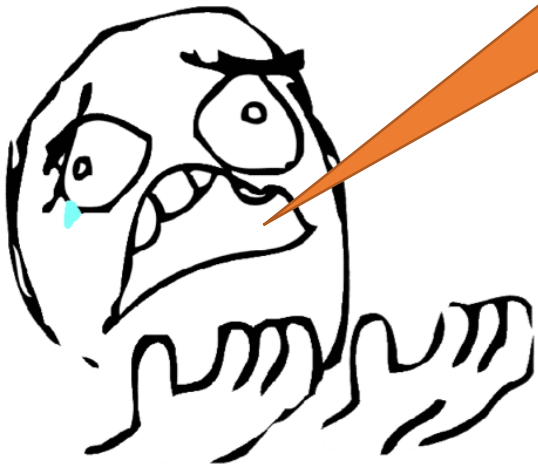
I/O with String

- Use cout to output the content of a string
- Use cin to read user input into a string
- Points to note
 - The extraction operator >> ignores initial whitespace and stops reading when it encounters more whitespace

```
#include <iostream>
using namespace std;
int main() {
    cout << "Full name:";
    string name;
    cin >> name;
    cout << name << endl;
}
```



How to cin a string
with spaces?



```
#include <iostream>
using namespace std;
int main() {
    cout << "Full name:";
    string name;
    getline(cin, name);
    cout << name << endl;
}
```

A terminal window with a dark background and green text. It shows the output of the C++ program. The first line is "Full name:Dirk Schnieders", the second line is "Dirk Schnieders", and the third line is "Press any key to continue..." followed by a green cursor block.

```
Full name:Dirk Schnieders
Dirk Schnieders
Press any key to continue...
```


Functions

- string has a number of functions which facilitate string manipulation
- Some useful member functions include
 - length - returns length of the string
 - empty - returns true if empty string
 - substr - returns a substring
 - find - finds the first occurrence
 - rfind - finds the last occurrence
 - insert - inserts content
 - erase - erases characters
 - replace - replaces part of the string

string::length

- Returns the number of characters in the string

length()

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      cout << "Full name:";
5      string name;
6      getline(cin, name);
7      cout << name.length() << endl;
8  }
```

string::empty

- Returns true if the string is empty, false otherwise

empty()

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      string name = "";
5      while (name.empty()) {
6          cout << "Full name:";
7          getline(cin, name);
8      }
9      cout << name << endl;
10 }
```

string::substr

- Returns a string whose content is a substring of the current string starting at the character position pos and having a length of n characters

substr(pos, n)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T	h	i	s		i	s		a		s	t	r	i	n	g	.

substr(pos)

```
#include <iostream>
using namespace std;
int main() {
    string s1 = "This is a string.";
    cout << s1.substr(0, 10) + s1.substr(12) << endl;
}
```

string::find

- Searches the current string for the content specified in str, and returns the position of the first occurrence

find(str)

- When pos is specified the search only includes characters at or after position pos, ignoring any possible occurrences in previous locations

find(str, pos)

- If there is no occurrence of str, -1 will be returned

string::find - Example

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
M	y		n	a	m	e		i	s		D	i	r	k

```
#include <iostream>
using namespace std;
int main() {
    string s = "My name is Dirk";
    string s2 = "Dirk";
    int pos = s.find(s2);
    cout << pos << endl;
}
```

string::rfind

- Searches the current string for the content specified in str, and returns the position of the last occurrence

rfind(str)

- If there is no occurrence of str, -1 will be returned

string::rfind - Example

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
M	y		n	a	m	e		i	s		D	i	r	k

```
#include <iostream>
using namespace std;
int main() {
    string s = "My name is Dirk";
    cout << s.find("i") << endl;
    cout << s.rfind("i") << endl;
}
```


string::insert

- Inserts the content specified in str at position pos of the current string and returns the resultant string

insert(pos,str)

```
#include <iostream>
using namespace std;
int main() {
    string name;
    cout << "Name: ";
    cin >> name;
    name.insert(name.length(), ".txt");
    cout << "filename: " << name << endl;
}
```

string::erase

- Erases n characters starting at position pos from the current string and returns the resultant string

erase(pos,n)

```
#include <iostream>
using namespace std;
int main() {
    string s = "Dirk is unhappy.";
    s.erase(8,2);
    cout << s << endl;
}
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
D	i	r	k		i	s		u	n	h	a	p	p	y	.

string::replace

- Replaces n characters starting at position pos from the current string by the content specified in str and returns the resultant string

replace(pos, n, str)

```
#include <iostream>
using namespace std;
int main() {
    string s = "Dirk is unhappy.";
    s.replace(8,7, "very happy");
    cout << s << endl;
}
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
D	i	r	k		i	s		u	n	h	a	p	p	y	.