



มหาวิทยาลัยขอนแก่น

วิทยา จริยา ปัญญา

KHON KAEN UNIVERSITY



# Standard C++ Strings and File I/O

Kornchawal Chaipah, PhD

Computer Engineering Department,  
Khon Kaen University



คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น  
FACULTY OF ENGINEERING KHON KAEN UNIVERSITY

# Agenda

- String
- String functions (methods)
- File Input / Output



# Standard C++ String

- Defined in <cstring>
- Can be treated as an array of characters
- Declared and Initialized in many ways

```
string str1;  
string str2 = "";  
string str3 = "New York";  
string str4(60, '*');  
string str5 = str4;  
string str6(str3, 4, 2);
```

Declared but not initialized  
= empty string



# Standard C++ String

- Defined in <string>
- Declared and Initialized in many ways

```
string str1;  
string str2 = "";  
string str3 = "New York";  
string str4(60, '*');  
string str5 = str4;  
string str6(str3, 4, 2);
```

Initialized to an empty  
string



# Standard C++ String

- Defined in <string>
- Declared and Initialized in many ways

```
string str1;  
string str2 = "";  
string str3 = "New York";  
string str4(60, '*');  
string str5 = str4;  
string str6(str3, 4, 2);
```

Initialized to "New York"



# Standard C++ String

- Defined in <string>
- Declared and Initialized in many ways

```
string str1;  
string str2 = "";  
string str3 = "New York";  
string str4(60, '*');  
string str5 = str4;  
string str6(str3, 4, 2);
```

Initialized to  
\*\*\*\*\*...\*  
└──────────┘  
60 of \*



# Standard C++ String

- Defined in <string>
- Declared and Initialized in many ways

```
string str1;  
string str2 = "";  
string str3 = "New York";  
string str4(60, '*');  
string str5 = str4;  
string str6(str3, 4, 2);
```

Same as str4



# Standard C++ String

- Defined in <string>
- Declared and Initialized in many ways

```
string str1;  
string str2 = "";  
string str3 = "New York";  
string str4(60, '*');  
string str5 = str4;  
string str6(str3, 4, 2);
```

- 1) Copy str3
- 2) Start from position 4 (position starts from 0 like an array of character)
- 3) Take 2 characters

"Yo"





# String Functions (Methods) 1/3

Function	Command	Description	Returned Value
Length	str.length()	Find the length of the string	unsigned int
Append	str.append( <i>string</i> )	Append <i>string</i> at the end of its current string	string
Getline	getline( <i>cin</i> , <i>str</i> )	Read the whole line of input (including space in between) via <i>cin</i> into <i>str</i>	The same as <i>cin</i>



# Function Usage Examples 1/3

Code:

```
string s = "ABCDEFGH";  
cout << s.length() << endl;
```

Output:

Code:

```
string s = "ABCDEFGH";  
string s1 = s.append("XYZ");  
cout << s1 << endl;  
string s2 = s + "HIJK";  
cout << s2 << endl;
```

Output:

Code:


```
string s1,s2,s3;  
cout << "Enter text: ";  
cin >> s1 >> s2;  
getline(cin, s3);  
cout << s1 << endl << s2 << endl  
    << s3 << endl;
```

Output:

```
Enter text: Hello World  
This is me.
```



# String Functions (Methods) 2/3

Function	Command	Details	Returned value
Substring	str.substr( <i>position</i> , <i>len</i> )	Copy <i>len</i> of characters of the string starting from <i>position</i> into another string	string
Find	str.find( <i>string</i> , <i>position</i> ) str.find( <i>string</i> )  Start finding at position 0	Search for <i>string</i> in the string str, starting at <i>position</i> . Then return the position of the first character of <i>string</i> . Return the length of the string if <i>string</i> not found.	unsigned int



# String Functions (Methods) 3/3

Function	Command	Details	Returned value
Erase	str.erase ( <i>position</i> , <i>len</i> )	Delete part of the string starting from <i>position</i> for <i>len</i> characters	string
Replace	str.replace ( <i>position</i> , <i>len</i> , <i>string</i> )	Replace part of the string with <i>string</i> starting from <i>position</i> , spanning <i>len</i> characters	string



# Function Usage Examples 2/3

Code:

```
string s = "ABCDEFGHJKLM";  
string s1 = s.substr(5, 3);  
cout << s1 << endl;
```

Output:

Code:

```
string s7 = "Mississippi River  
Basin";  
cout << s7.find("si",0) << endl;  
cout << s7.find("si") << endl;  
cout << s7.find("si",5) << endl;  
cout << s7.find("so",5) << endl;
```

Output:

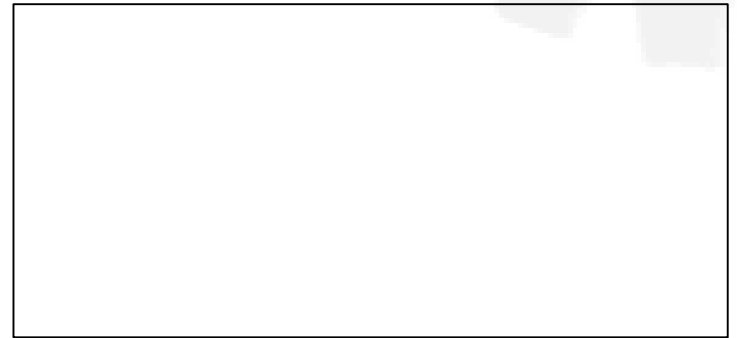


# Function Usage Examples 3/3

Code:

```
string s6 = "ABCDEFGHGIJK";  
s6.erase(4, 2);  
cout << s6 << endl;  
s6.replace(5, 2, "xyz");  
cout << s6 << endl;
```

Output:



# Basic Operators

Operator	Details	Return value
<code>==, !=</code> <code>&lt;, ≤</code> <code>&gt;, ≥</code>	Compare strings	boolean
<code>+</code>	Add additional <i>string</i> at the end of its current string (similar to the <i>append</i> function)	string



# Comparing Strings

Code:

```
string s1 = "Somchai";  
string s2 = "Somjai";  
if (s1 < s2){  
    cout << s1 << " comes first."  
        << endl;  
}
```

Output:

Code:

```
string s4 = "I love you.";  
cin >> s3;  
while (s4 == s3)  
{  
    cout << "I love you too!"  
        << endl;  
    cin >> s3;  
}
```

Output:



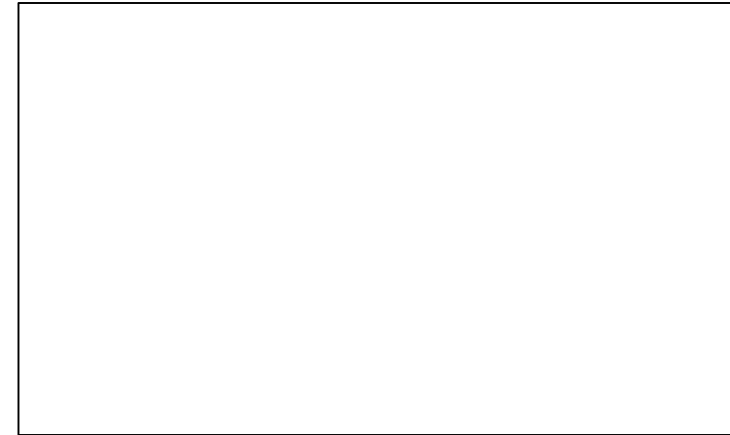


# String Concatenation

Code:

```
string s = "ABCDEFGH";  
string s5 = s + "HIJK";  
cout << s5 << endl;  
  
string s2 = "XYZ";  
s2 += s5;  
cout << s2 << endl;
```

Output:

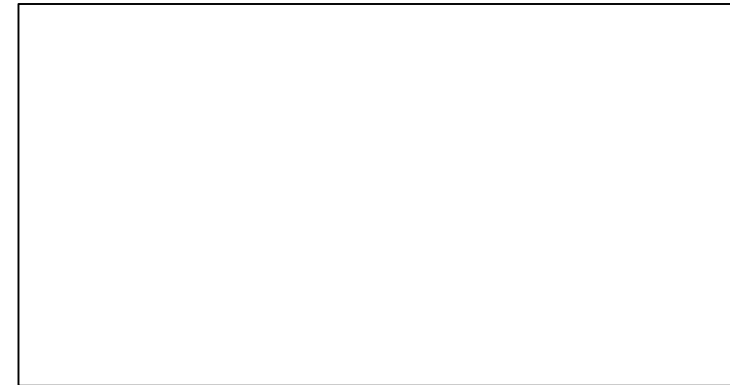


# String as an array of characters

Code:

```
string s = "ABCDEFGH";  
char c = s[4];  
cout << c << endl;  
  
s[2] = "*";  
cout << s << endl;
```

Output:



# String Example

Code:

```
int main(){
    string S1 = "Sawasdee Ja";
    string S2 = "Somsri Konkeng";
    string S3 = "He is a student of KhonKaen University";
    string S4;
    cout << S3.length() << endl;
    S4 = S2 + S3.substr(2, 13);
    cout << S4 << endl;
    cout << S3.find("en", 0) << endl;
    S4 = S1.replace(9, 2, S2.erase(7,7));
    cout << S4 << endl;
    cout << (S2 < "Somsak Kondee" << endl;
    return 0;
}
```

Output:



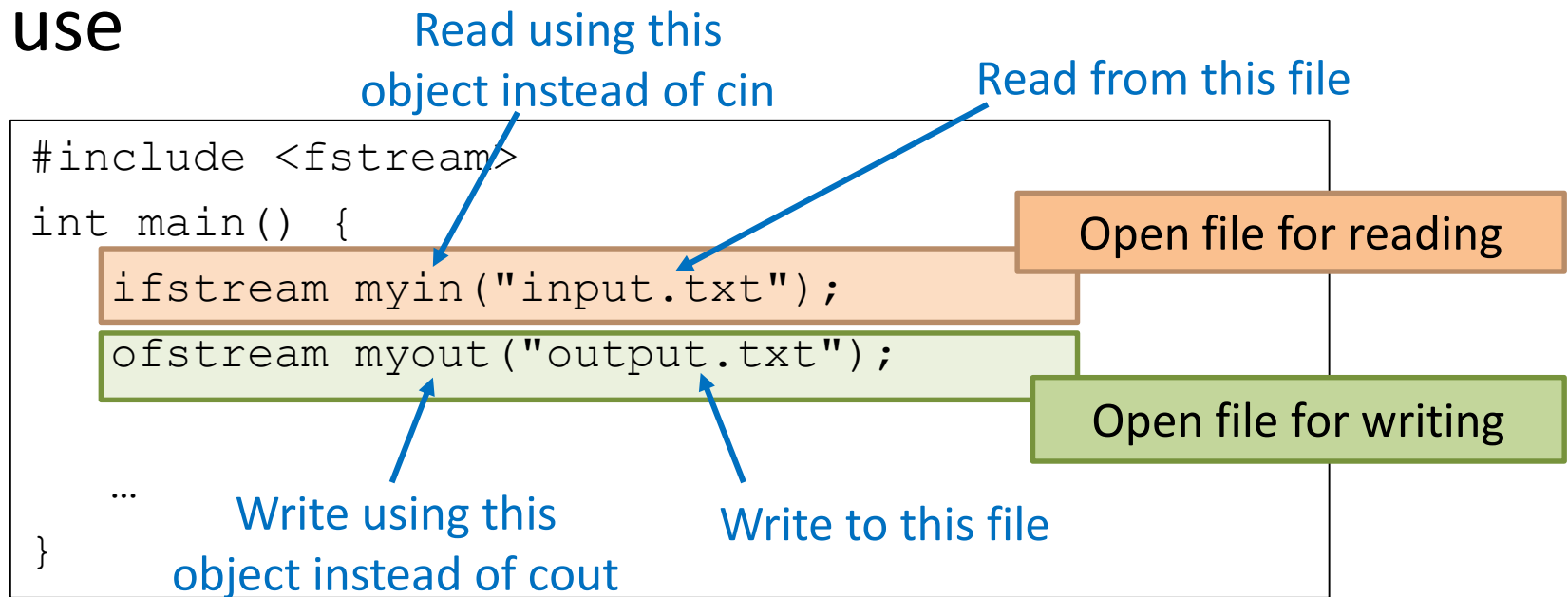
# File Processing

- Reading from files and writing to files are very similar to reading from / writing to console (cin, cout)
  - Same kind of stream objects
- An input file is managed by *ifstream*
  - Same way as *istream* manages *cin*
- An output file is managed by *ofstream*
  - Same way as *ostream* manages *cout*



# Using File Streams

- `#include <fstream>`
- Declare `ifstream` / `ofstream` explicitly before use



# Output File Stream

## Output to Console

```
#include <iostream>

int main()
{
    string str = "my text";
    cout << str;
    return 0;
}
```

## Output to the file named "myfile.dat"

```
#include <fstream>

int main()
{
    string str = "my text";
    ofstream fout("myfile.dat");
    fout << str;
    return 0;
}
```



# Input File Stream

## Output to Console

```
#include <iostream>
```

```
int main()
{
    int value = 0;
    cin >> value;
    return 0;
}
```

## Read from the file named "number.dat"

```
#include <fstream>
```

```
int main()
{
    int value = 0;
    ifstream fin("number.dat");
    fin >> value;
    return 0;
}
```



# Reading until the end of stream

- The expression *cin* has a returned value as a Boolean indicating whether the input is read successfully

```
int main()
{
    int value;
    while(cin >> value){
        cout << "value = "<< value << endl;
    }
    return 0;
}
```

- Same as `getline(cin,s)` and the input file stream reader!!





# Reading until the end of stream

Code:

```
#include <fstream>
int main()
{
    ifstream fin("number.dat");
    int value = 0;
    int sum = 0;

    while (fin >> value)
    {
        sum += value;
    }
    cout << "sum = " << sum << endl;
    return 0;
}
```

Read from a file

This will return false  
while there is no more  
data to read (after 36)

Write to screen

number.dat

5 6 7 33 4 0 19 1 4 9 24 36

Console Output:

You can read from a file or console,  
then write to a file or console



# Read/Write Odd/Even Values

Code:

```
#include <fstream.h>
int main()
{
    int value = 0;
    ifstream fin("number.dat");
    ofstream fout_odd("odd.dat");
    ofstream fout_even("even.dat");
    while (fin >> value)
    {
        if(value%2){
            fout_odd << value;
        }
        else {
            fout_even << value;
        }
    }
    cout << "Done!" << endl;
    return 0;
}
```

number.dat

12 11 20 15 35 10 69 71 23 80

odd.dat

even.dat

Console Output:



# Take Home Messages

- String declaration and assignments
- String functions (methods)
- String operations
  - String comparisons
  - String as an array of characters
- Reading from a file vs. reading from keyboard
  - Read until the end of file
- Writing to a file vs. writing to a console



# References

- คณาจารย์คณะวิศวกรรมศาสตร์. (n.d.). เอกสารชุดนำเสนอภาพและบรรยายวิชาการเขียนโปรแกรม (ส่วนกลาง)
- คณาจารย์คณะวิศวกรรมศาสตร์. (n.d.). เอกสารประกอบการสอนวิชาการเขียนโปรแกรม (ส่วนกลาง)
- รศ. วิโรจน์ ทวีปวรเดช. (2554). การเขียนโปรแกรมคอมพิวเตอร์ **Computer Programming**. พิมพ์ครั้งที่ 2 โรงพิมพ์มหาวิทยาลัยขอนแก่น
- Cplusplus.com. (n.d.). **C++ Documentation**. สืบค้นเมื่อ 18 กุมภาพันธ์ 2555, <http://www.cplusplus.com/>
- Cplusplus.com. (n.d.). **Library Reference**. สืบค้นเมื่อ 18 กุมภาพันธ์ 2555, <http://www.cplusplus.com/>
- ISO/IEC 14882 Programming Language – C++

