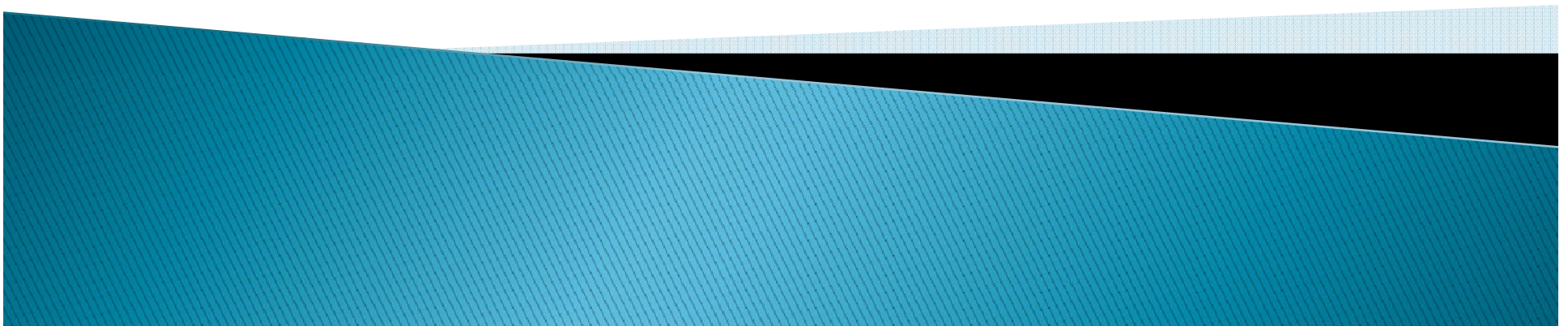


CPP– part 3 – File IO

Minh Nguyen
Computer Science 210



Input/Output with files

- ▶ It's important and has many applications:
 - Copy files.
 - Generate many files from a list of contacts.
 - Search, replace strings in text files.
- ▶ C++ provides the following classes to perform output and input of characters to/from files:
 - **ofstream**: Stream class to write on files
 - **ifstream**: Stream class to read from files
 - **fstream**: Stream class to both read and write from/to files.



Basic file operations: write

```
// basic file operations
#include <iostream>
#include <fstream>
using namespace std;
int main () {
    ofstream myfile;
    myfile.open ("example.txt");
    myfile << "Writing this to a file.\n";
    myfile.close();
    return 0;
}
```

[file example.txt]
Writing this to a file.

For absolute path:
C:/example.txt

This code creates a file called example.txt and inserts a sentence into it in the same way we are used to do with cout, but using the file stream myfile instead.

Basic file operations: read

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main () {
    string line;
    ifstream myfile ("example.txt");
    if (myfile.is_open())
    {
        while ( myfile.good() )
        {
            getline (myfile,line);
            cout << line << endl;
        }
        myfile.close();
    }
    else cout << "Unable to open file ";
    return 0;
}
```

Print from [example.txt] :

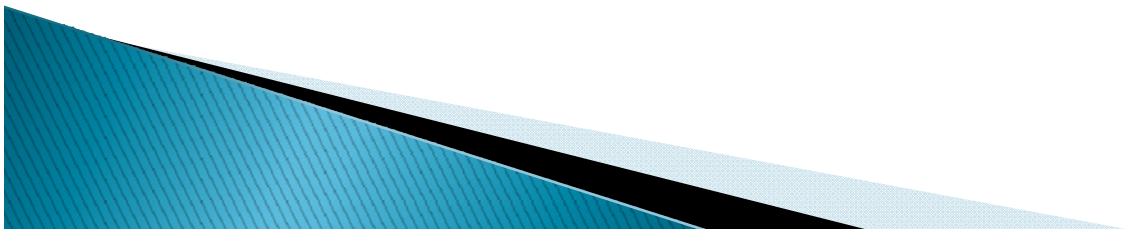
This is a line.
This is another line.

For absolute path:
C:/example.txt

This last example reads a text file and prints out its content on the screen. Notice how we have used a new member function, called `good()` that returns true in the case that the stream is ready for input/output operations. We have created a while loop that finishes when indeed `myfile.good()` is no longer true, which will happen either if the end of the file has been reached or if some other error occurred.

Exercises – use ‘nz.txt’

1. Copy file “nz.txt” to “nz_copy.txt”.
2. Copy file “nz.txt” to “nz_upper_case.txt”.
Which turn all characters in “nz.txt” to be upper case.
3. Count how many repetition of each character.
 - A:3
 - B:5
 - C:6
 - ...



Exercise answers 1

```
cout << "Copy file to nz_copy.txt: \n===== \n";
ofstream myfile2;
myfile2.open("nz_copy.txt");
ifstream myfile ("nz.txt");
cout << "Print the whole file: \n===== \n";
if (myfile.is_open())
{
while ( myfile.good() )
{
getline (myfile,line);
myfile2 << line << "\n";
}
myfile.close();
}
else cout << "Unable to open file";
myfile2.close();
cout << "Successfully copy nz_copy.txt\n";
```

Exercise answers 2

```
cout << "Upper case nz.txt: \n===== \n";
ifstream myfile;
myfile.open("nz.txt");
ofstream myfile2;
myfile2.open("nz_upper_case.txt");

if (myfile.is_open())
{
    while ( myfile.good() )
    {
        getline (myfile,line);
        int count = 0;
        while(1){
            if(line[count] >= 'a')
                line[count]=line[count] - ('a'-'A');

            if(line[count] == 0) break;
            count++;
        }
        myfile2 << line << "\n";
    }
    myfile.close();
}
else cout << "Unable to open file";
myfile2.close();
```

Exercise answers 3

```
cout << "Count characters in nz.txt: \n===== \n";
ifstream myfile;
myfile.open("nz.txt");
int characters[26];
for(int i = 0; i < 26; i++){
    characters[i]=0;
}
if (myfile.is_open())
{
    while ( myfile.good() )
    {
        getline (myfile,line);
        int count = 0;
        while(1){
            if(line[count] >= 'a') line[count] -= ('a'-'A');
            if(line[count] >= 'A'){
                characters[line[count]-'A']++;
            }
            if(line[count] == 0) break;
            count++;
        }
    }
    myfile.close();
}
else cout << "Unable to open file";
for(int i = 0; i < 26; i++){
    cout << (char)('A'+i) << ":" << characters[i] << "\n";
}
```