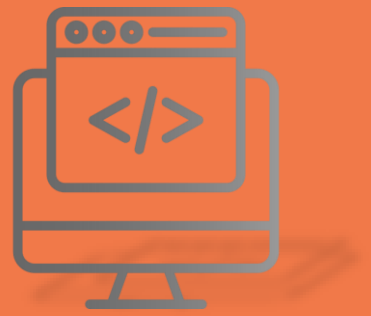
Tishk International University
Faculty of Applied Science
Information Technology Department

File Stream



Programming II

Lecture 5

Spring 2026

Course Code: IT118

Grade 1

Islam Abdulazeez
islam.abdulaziz@tiu.edu.iq

June 7, 2026

Outlines

- ✓ Files and Streams
- ✓ File Stream Objects
- ✓ Opening and Closing Files
- ✓ File Open Modes
- ✓ Reading from Files
- ✓ Writing to Files
- ✓ File Processing Applications

Learning Outcomes

- **At the end of today's session, you will be able to:**

  - ✓ Define files and streams in C++.
  - ✓ Explain the use of file stream classes.
  - ✓ Open and close files correctly.
  - ✓ Read and write data using file streams.
  - ✓ Develop simple programs for file processing.

File

- A file is a set of data stored on a computer, often on a disk drive
- Programs can read from, write to files

❑ Used in many applications:

  - Word processing: Microsoft Word, Google Docs, or Pages
  - Databases: MySQL, Microsoft Access, SQL Sever, ..
  - Spreadsheets: Microsoft Excel, Google Sheets, or Numbers
  - Compilers: read files, process the code, and generate an executable file

File Name

- File name can be a full pathname to file:

  c:\data\student.txt

  tells compiler exactly where to look .

- File name can also be simple name:

  student.txt

  this must be in the same directory as the program executable, or in the compiler's default directory

Stream

- **Stream:** A transfer of information in the form of a sequence of bytes
- **Input stream:** (i.e.: keyboard, file “ disk drive”, network connection) to program (main memory)
- **Output stream:** (i.e.: screen, printer, file “ disk drive”, network connection)

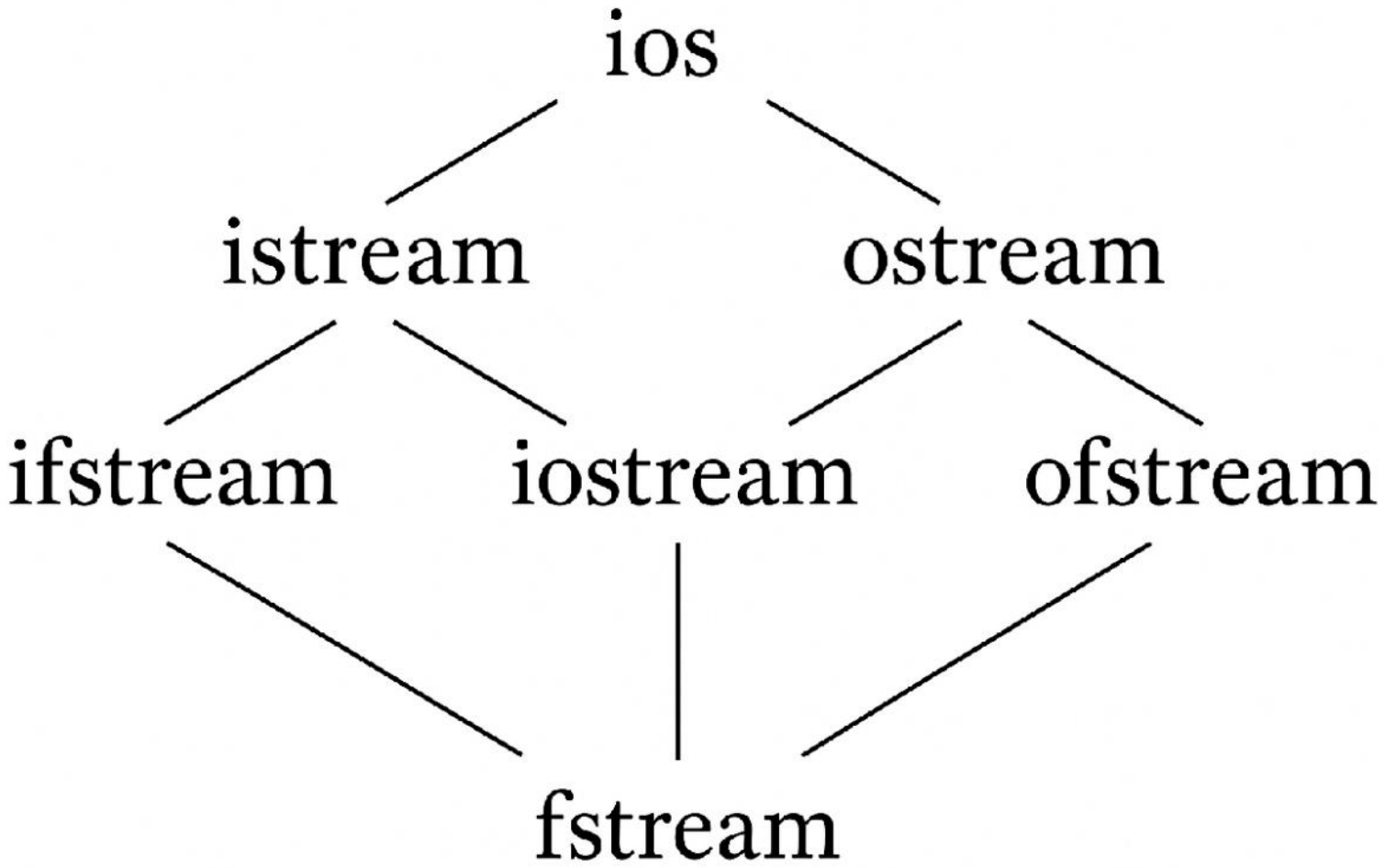
Stream in C++

- Streams in C++ provide a convenient way to perform input and output operations with various devices, such as files, standard input/output, and strings.
  - Input is reading data from a file or user input
  - Output is writing data to a file or the console.
- The <iostream> header provides the necessary classes and functions for stream operations.
- It includes two main stream objects: cin and cout.
  - **cin** is the standard input stream, which is used for reading input from the user via the keyboard.
  - **cout** is the standard output stream, which is used for displaying output to the user on the console.

Stream in C++

- C++ streams are divided into two categories: input streams and output streams.
- An **istream** object named **cin** connects program and keyboard
- An **ostream** object named **cout** connects the program and the screen

Stream in C++



Stream in C++

- Must first connect file to stream object
- For reading from a file (input): Use an **ifstream** object
- For writing to a file (output): Use an **ofstream** object.
  - Classes ifstream and ofstream

    - Defined in library **<fstream>**

Default File Open Modes

- **ofstream:**
  - open for output only
  - file cannot be read from
  - file created if no file exists
  - file contents erased if file exists
- **ifstream:**
  - open for input only
  - file cannot be written to
  - open fails if file does not exist

Closing a File

- Traditionally, we close a file when we're done using it:

  myfile.close();

- We can do this explicitly, but C++ streams are automatically closed at the end of the variable's lifetime (typically at the end of the function it is declared in)

Declaring Streams

- Just like declaring other variables, declare a file stream object
- Open the File

```
ofstream outFile("output.txt");   // Opens file for writing immediately

ifstream inFile("input.txt");     // Opens a file for reading immediately
```

Open the file for writing - Example

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

int main() {
    ofstream outFile("firstFile.txt");

    if (outFile.is_open()) {
        outFile << "Hello, Kurdistan!" << endl;
        outFile.close();
        cout << "Data added to the file" << endl;
    }
    else {
        cout << "Failed to open the file" << endl;
    }
    return 0;
}
```

Open the file for reading

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

int main() {
    ifstream inputFile("firstFile.txt");
    string line;

    if (inputFile.is_open()) {
        getline(inputFile, line);
        inputFile.close();

        cout << line << endl;
        cout << "File read" << endl;
    }
    else {
        cout << "Cannot open the file" << endl;
    }
    return 0;
}
```

fstream for input and output

- **fstream** object can be used for both input and output at the same time
- Create the fstream object and specify both **ios::in** and **ios::out** as the second argument

  to the open member function
- Opens the file for both reading and writing, **but it does not create the file if it doesn't already exist.**
- If you want the old contents removed each time, use **ios::trunc**

```
fstream file;
file.open("myfile.txt", ios::in | ios::out | ios::trunc);
```

- Or you can use:

```
fstream file("myfile.txt", ios::in | ios::out | ios:: trunc);
```

fstream for input and output

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

int main() {

    fstream input("city.txt", ios::in | ios::out | ios::trunc);

    if (input.is_open()) {
        input << "Hawler";
        input.close();
        cout << "Success" << endl;
    }
    else {
        cout << "Failed" << endl;
    }
    return 0;
}
```

fstream for input and output

ios :: app	Creates new file, or **append** to end of existing file
ios :: in	open for input
ios :: in	open for output

fstream with ios::app

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

int main() {

    fstream input("city.txt", ios::app);

    if (input.is_open()) {

        input << "Slemani" << endl;
        input.close();
        cout << "Success" << endl;
    }
    else {
        cout << "Failed" << endl;
    }
    return 0;
}
```

Read all lines (while loop)

```cpp
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main() {
    string x;

    ifstream readFile("city.txt");

    if (readFile.is_open()) {
        while (getline(readFile, x)) {
            cout << x << endl;
        }
        readFile.close();
        cout << "Read it Successfully" << endl;
    }
    else {
        cout << "Failed" << endl;
    }
    return 0;
}
```

Read all lines (for loop)

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

int main() {
    string x;
    ifstream readFile("city.txt");

    if (readFile.is_open()) {
        for (int i = 1; getline(readFile, x); i++) {
            cout << x << endl;
        }
        readFile.close();
        cout << "Read it Successfully" << endl;
    }
    else {
        cout << "Failed" << endl;
    }
    return 0;
}
```

Read Numbers (while loop)

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

int main() {
    ifstream file("numbers.txt");
    int number;

    if (file.is_open()) {
        while (file >> number) {
            cout << number << endl;
        }
        file.close();
    }
    else {
        cout << "Failed" << endl;
    }
    return 0;
}
```

Read Numbers (foor loop)

```cpp
#include <iostream>
#include <fstream>
using namespace std;
int main() {
    ifstream file("numbers.txt");
    int number;

    if (file.is_open()) {
        for (int i = 1; file >> number; i++) {
            cout << number << endl;
        }
        file.close();
    }
    else {
        cout << "Failed" << endl;
    }

    return 0;
}
```

Insert data into a file using for loop

```cpp
#include <iostream>
#include <fstream>
using namespace std;
int main() {
    ofstream input("city.txt");
    string name;

    if (input.is_open()) {
        for (int i = 0; i < 4; i++) {
            cout << "Input city name" << endl;
            cin >> name;
            input << name << endl;
        }
        input.close();
        cout << "Added Successfully" << endl;
    }
    else {
        cout << "Failed" << endl;
    }
    return 0;
}
```

Insert data into a file using for loop and getline

```cpp
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main() {
    ofstream input("info.txt");
    string name;

    if (input.is_open()) {
        for (int i = 0; i < 4; i++) {
            cout << "Input your full name" << endl;
            getline(cin, name);
            input << name << endl;
        }
        input.close();
        cout << "Added Successfully" << endl;
    }
    else {
        cout << "Failed" << endl;
    }
    return 0;
}
```

Ask the user to stop inserting data

```cpp
int main() {
    ofstream input("info.txt");
    string name;
    bool flag = true;

    if (input.is_open()) {
        while (flag) {
            cout << "Input your full name (type 'exit' to stop)" << endl;
            getline(cin, name);

            if (name == "exit") {
                flag = false;
            }
            else {
                input << name << endl;
            }
        }
        input.close();
        cout << "Added Successfully" << endl;
    }
    else {
        cout << "Failed" << endl;
    }
    return 0;
}
```

Insert different data type into file

```
int  a=50;
double  b=4.9;
string c ="hello";
myfile << a << b << c << endl;
```

Insert random numbers and find the largest

```cpp
#include <iostream>
using namespace std;
#include <fstream>
int main(){
    srand(time(0));
    ofstream x("random.txt");
    if (x.is_open()){
        for (int i = 0; i < 10; i++){
            int number = rand() % 100;
            x << number << endl;
        }
        x.close();
    } else {
        cout << "Faild" << endl;
    }

    ifstream read("random.txt");
    int largest = 0;
    int number;
    if (read.is_open()){
        while (read >> number){
            if (number > largest){
                largest = number;
            }
        }
    } else {
        cout << "Failed" << endl;
    }
    cout << largest << endl;
}
```

Reading a Text File Word by Word

```cpp
#include <iostream>
#include <fstream>
using namespace std;
int main() {
    string x;
    ifstream readFile("info.txt");

    if (readFile.is_open()) {
        while (readFile >> x) {
            cout << x << endl;
        }

        readFile.close();
        cout << "Read it Successfully" << endl;
    }
    else {
        cout << "Failed" << endl;
    }
    return 0;
}
```

Inside the file

```
Tishk International University
Grade one
ProgrogrammingII
Welcome to our university
```

Output

```
Tishk
International
University
Grade
one
ProgrogrammingII
Welcome
to
our
university
```

Exercise

- How can we modify these two codes to find the number of words and lines?

```cpp
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main() {
    string x;
    ifstream readFile("info.txt");
     if (readFile.is_open()) {
       while(getline(readFile,x)){
            cout <<x << endl;
        }
         readFile.close();
         cout<<"Read it Successfully"<<endl;
     } else {
        cout<<"Failed"<<endl;
     }
    return 0;
}
```

```cpp
#include <iostream>
#include <fstream>
using namespace std;
int main() {
    string x;
    ifstream readFile("info.txt");
     if (readFile.is_open()) {
       while(readFile>>x){
            cout <<x << endl;
        }
         readFile.close();
         cout<<"Read it Successfully"<<endl;
     } else {
        cout<<"Failed"<<endl;
     }
    return 0;
}
```

Counting number of words in the file

```
int main() {
    string x;
    int counter = 0;

    ifstream readFile("info.txt");

    if (readFile.is_open()) {
        while (readFile >> x) {
            counter++;
        }

        readFile.close();
    }
    else {
        cout << "Failed" << endl;
    }
    cout << "This file has " << counter << " words." << endl;
    return 0;
}
```

Counting number of lines in the file

```
int main() {
    string x;
    int counter = 0;

    ifstream readFile("info.txt");

    if (readFile.is_open()) {
        while (getline(readFile, x)) {
            counter++;
        }

        readFile.close();
    }
    else {
        cout << "Failed" << endl;
    }
    cout << "This file has " << counter << " lines." << endl;
    return 0;
}
```

Exercise

- How to modify this code to insert all data word by word to a vector and print the vector?

```
int main() {
    string x;
    int counter = 0;

    ifstream readFile("info.txt");

    if (readFile.is_open()) {
        while (readFile >> x) {
            counter++;
        }

        readFile.close();
    }
    else {
        cout << "Failed" << endl;
    }

    cout << "This file has " << counter << " words." << endl;
    return 0;
}
```

Inserting all data word by word to a vector and print the vector

```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
using namespace std;
int main() {
    string x;
    vector<string> v;

    ifstream readFile("info.txt");

    if (readFile.is_open()) {
        while (readFile >> x) {
            v.push_back(x);
        }
        readFile.close();
    }
    else {
        cout << "Failed" << endl;
    }

    for (int i = 0; i < v.size(); i++) {
        cout << v[i] << endl;
    }
    return 0;
}
```

Activities and Next Lecture's Topic

Activities

- Review this lecture note
- Practice

References

- Gaddis, T. (2014). Starting out with C++: Early objects (7th ed.). Pearson Education.

Thank You!