

File operation

➤ File is an external collection of related data treated as a unit

It is used to keep a record of our data.

Why do we need files?

Batching processing vs. interactive processing

➤ Read from external file

A stream needs to be created to connect the external files to the program.

ifstream : input stream connects the input data file to the program

ofstream : output stream connects the program to the output files

Input and output streams are defined in <fstream>

➤ Create output stream – writing to external file

notes: 1. If the file does not exist beforehand, it will be created.

2. If the file exists beforehand, all information in the file will be lost after we opened a output stream to it.

3. Make sure the file is properly closed after the information is written to the file.

4. The settings used for formatted output can also be used with the user created output streams

Example 1

```
#include <fstream>
```

```
#include <cmath>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    ofstream outfile;
```

```
    int value=10;
```

```
    outfile.open("ex1.result");
```

```
    outfile<<fixed;
```

```
    outfile<<showpoint;
```

```
    outfile<<setprecision(2);
```

```
    outfile << setw(10) << "Value" << setw(15) << "Square"
        << setw(15) << "Square Root" << endl;
```

```
    outfile << setw(10) << value << setw(15) << pow(double(value), 2.0)
        << setw(15) << sqrt(double(value)) << endl;
```

```
    value = value + 10;
```

```
    outfile << setw(10) << value << setw(15) << pow(double(value), 2.0)
        << setw(15) << sqrt(double(value)) << endl;
```

```
    value = value + 10;
```

```
    outfile << setw(10) << value << setw(15) << pow(double(value), 2.0)
```

```

        << setw(15) << sqrt(double(value)) << endl;

    outfile.close();

    return 0;
}

```

Example 2: Input and output file streams in a program

```

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

int main()
{
    float    length;
    float    width;
    float    area;
    ifstream inputFile;
    ofstream outputFile;

    inputFile.open("rectangle.data");
    outputFile.open("result");

    inputFile >> length >> width;

    area = length * width;

    outputFile << "The width of the rectangle is " << width << endl;
    outputFile << "The length of the rectangle is " << length << endl;
    outputFile << "The area of the rectangle is  " << area << endl;

    inputFile.close();
    outputFile.close();

    return 0;
}

```