

CS2311 Computer Programming

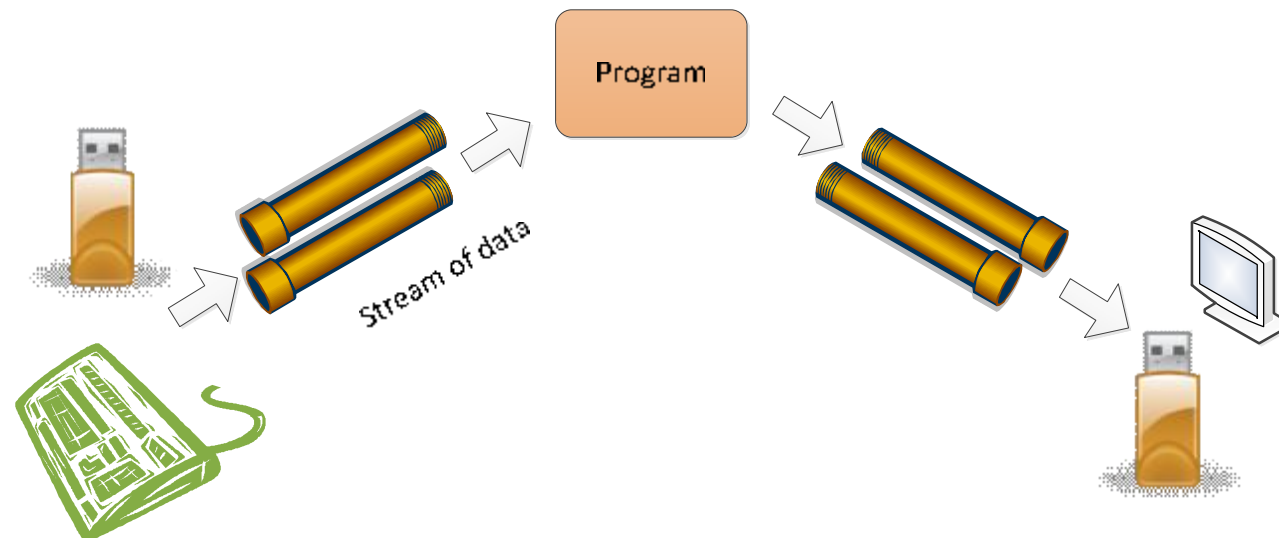
File I/O (Input/Output)

File I/O vs. Console I/O

- "Console" refers to "keyboard + screen"
- Keyboard input and screen output are volatile
- Input file can be used again and again
- useful for debugging especially when volume of data is huge
- allow off-line processing
- Output file retains the results after execution

Basic I/O – Keyboard and Screen

- Program read input from keyboard (console) or disk storage (file) and write data to screen (console) or disk storage(file)
- Sequence of inputs is conceptually treated as an object called "Stream"
- Stream – a flow (sequence) of data
- Input stream – a flow of data into your program
- Output stream – a flow of data out of your program



Streams

- **Predefined console streams in C++**
 - `#include <iostream>`
 - `cin` : input stream physically linked to the keyboard
 - `cout`: output stream physically linked to the screen
- **File streams class in C++**
 - `#include <fstream>`
 - `ifstream`: stream class for file input
 - `ofstream`: stream class for file output
- **To declare an objects of class ifstream or ofstream, use**
 - `ifstream fin; // fin is the variable name`
 - `ofstream fout; // fin is the variable name`

ifstream (input file stream)

- To declare an ifstream object
 - `ifstream fin;`
- To open a file for reading
 - `fin.open("infile.dat");`
- To read the file content
 - `fin >> x;` `//x is a variable`
- To close the file
 - `fin.close();`

ofstream (output file stream)

- To declare an ofstream object
 - `ofstream fout;`
- To open a file for writing
 - `fout.open("myfile.dat");`
- To write something to the file
 - `fout << x;` `//x is a variable`
- To close the file
 - `fout.close();`
- PS: `fin.open()` and `fout.open()` refer to different functions

Examples

Read 3 numbers from input.txt and write the sum to output.txt

```
#include <fstream>
using namespace std;
void main() {
    ifstream fin;
    ofstream fout;
    int x,y,z;
    fin.open("input.txt");
    fout.open("output.txt");
    fin >>x>>y>>z;
    fout << "The sum is "<<x+y+z;
    fin.close();
    fout.close();
}
```

3 4 7

The sum is 14

Detecting end-of-file

- Member function **eof** returns **true** if and only if we try to read from the input file which has no more data
- Only for objects of class `ifstream` (*not* `ofstream`)

E.g. `fin >> x;`

`if (fin.eof()) cout<<"Error: no more data to read"; ...`

Examples: file dump (integer only)

```
#include <iostream>
#include <fstream>
using namespace std;
void main() {
    ifstream fin;
    int x;
    fin.open("input.txt");
    while (!fin.eof()) {
        fin >> x;
        cout <<x<<" ";
    }
}
```

What if the file
contains non-
integer data?

Detecting I/O failures

- Member function `fail()` returns `true` if and only if the previous I/O operation on that stream fails. For example:
 - File not exists when opening an input stream
 - Output file is read only
 - Try to `>>` integer but data is non-numeric
 - Try to write file but disk full...etc.
- The `fail()` function exist in both `ifstream` and `ofstream`
- Actually it exist in `cin/cout` as well...
- Note: For sure you'll abort the file I/O if there's error, but please remember to close the opened file!

Examples

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

void main() {
    ifstream in1, in2;

    in1.open("infile1.dat");
    in2.open("infile2.dat");
    if (in1.fail()) {
        cout << "Input file 1 cannot be opened.\n";
    }
    ...
}
```

Examples

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

void main(){
    int data[] = {1,2,3,4,5,6,7,8,9,10};
    ofstream fout;

    fout.open("output.txt");
    if (fout.fail()) {
        cout << "Cannot open output.txt\n";    return;
    }
    for (int i=0;i<10;i++) {
        fout<<i<<endl;
        if (fout.fail()) {
            cout << "Cannot write data. Disk full?\n";
            break;
        }
    }
    fout.close(); ...
}
```

Summary

- Beside reading and writing data from and to console, program can read and write data from and to file
- ifstream and ofstream are two classes defined in <fstream>
- File must be opened before access and close after access
 - `fin.open("filename");`
 - `fin.close();`
- File I/O is similar to console I/O
 - `cin >> x;` `cout << x;`
 - `fin >> x;` `fout << x;`