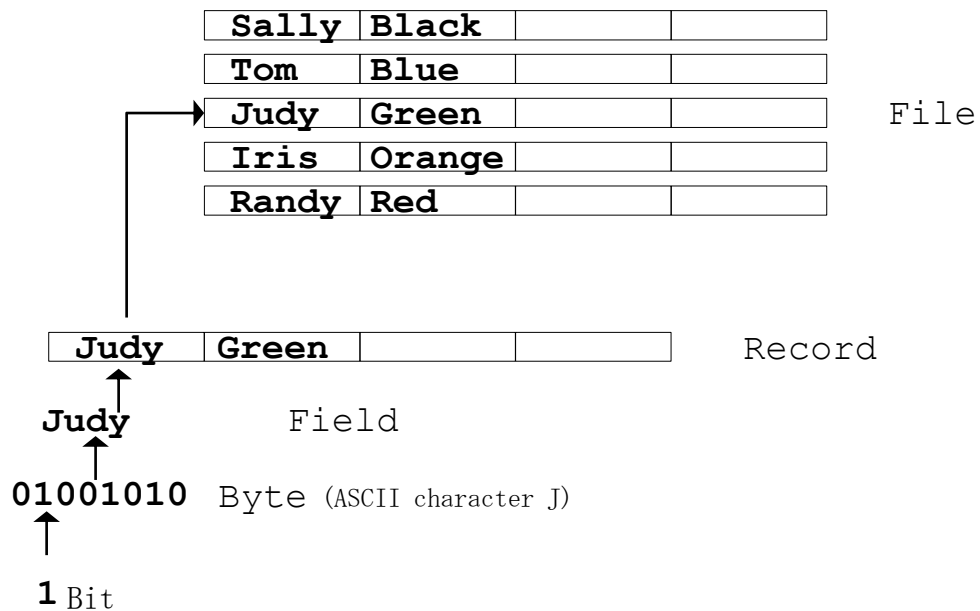


Διαχείριση Αρχείων

Ιεραρχία Δεδομένων

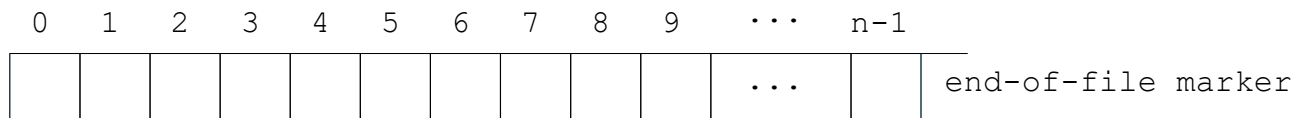
- Από μικρότερα σε μεγαλύτερα:
 - Bit (δυναδικό ψηφίο 1 ή 0)
 - Byte: 8 bits
 - Μπορεί να αποθηκεύσει έναν χαρακτήρα (**char**)
 - Unicode για μεγάλα σετ χαρακτήρων (**wchar_t**)
 - Πεδίο (Field): γκρουπ χαρακτήρων με κάποιο νόημα (πχ ένα όνομα)
 - Εγγραφή (Record): ομάδα από σχετικά πεδία (**struct** ή **class** στην C++)
 - Αρχείο (File): ομάδα σχετικών εγγραφών
 - Βάση Δεδομένων (Database): ομάδα σχετικών αρχείων

Ιεραρχία Δεδομένων



Αρχεία στην C++

- Η C++ χειρίζεται τα αρχεία ως αλληλουχία bytes
 - το τέλος σηματοδοτείται με: *end-of-file*



- Όταν ανοίγει ένα αρχείο
 - Δημιουργείται αντικείμενο και συνδέεται με ένα κανάλι (stream)
 - **cin**, **cout**, κτλ. όταν περιλαμβάνεται η **<iostream>**

Κανάλια - Streams

- Για να διαχειριστούμε ένα αρχείο:

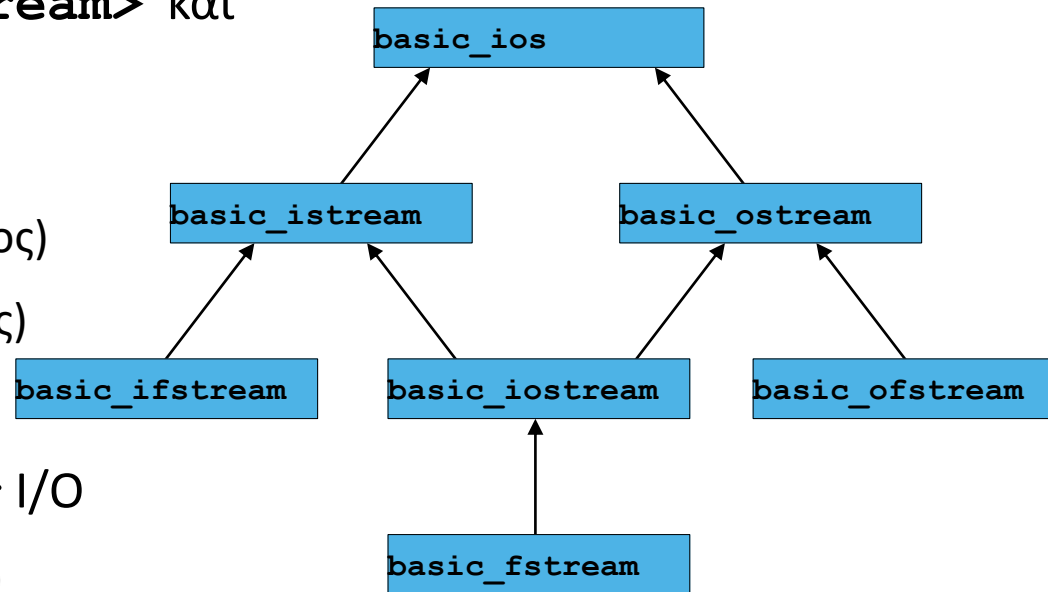
- Κάνουμε include τις `<iostream>` και `<fstream>`

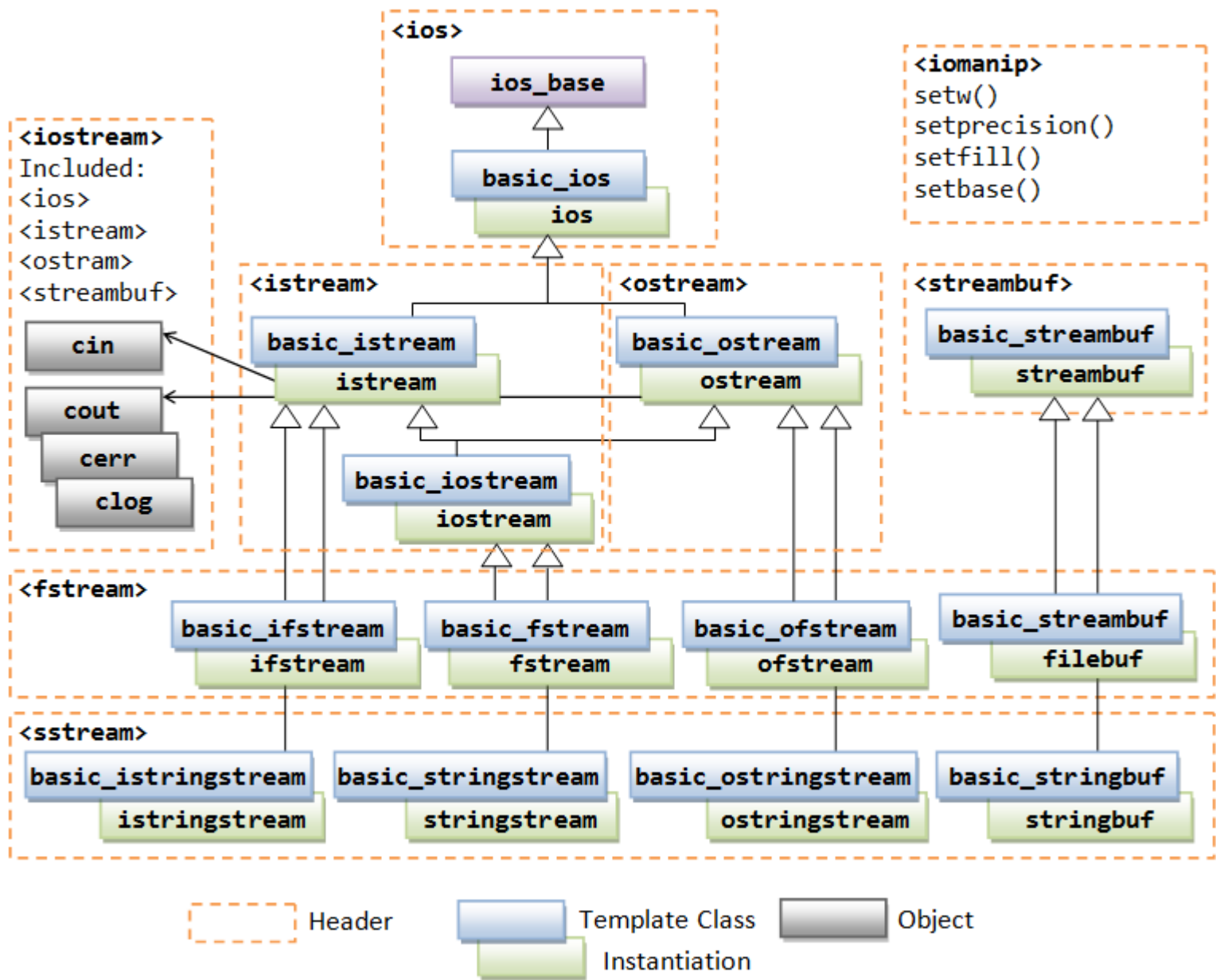
- Class templates

- `basic_ifstream` (είσοδος)
- `basic_ofstream` (έξοδος)
- `basic_fstream` (I/O)

- `typedefs` ειδικά για `char` I/O

- `ifstream` (είσοδος `char`)
- `ofstream` (εξοδος `char`)
- `fstream` (`char` I/O)





Άνοιγμα Αρχείου

- Η C++ δεν αναγκάζει συγκεκριμένη δομή στο αρχείο
 - Η έννοια μιας “εγγραφής” καθορίζεται/υλοποιείται από τον προγραμματιστή
- Για να ανοίξει το αρχείο δημιουργείται αντικείμενο
 - δημιουργεί «γραμμή επικοινωνίας» από το αντικείμενο στο αρχείο
 - Κλάσεις
 - `ifstream` (μόνο για είσοδο)
 - `ofstream` (μόνο για έξοδο)
 - `fstream` (I/O)
 - Οι δημιουργοί παίρνουν ως παραμέτρους το όνομα του αρχείου (*file name*) και τον τρόπο (*file-open mode*)

```
ofstream outClientFile( "filename", fileOpenMode );
```
 - Για να ανοίξουμε ένα αρχείο αργότερα (αφού δημιουργήσουμε το αντικείμενο):

```
ofstream outClientFile;  
outClientFile.open( "filename", fileOpenMode);
```



Τρόποι ανοίγματος αρχείου

- Τρόποι ανοίγματος αρχείου (File-open modes)

Mode	Description
<code>ios::app</code>	Εγγραφή στο τέλος του αρχείου
<code>ios::ate</code>	Άνοιγμα αρχείου για εγγραφή και μεταφορά στο τέλος του αρχείου. Μπορεί να γίνει εγγραφή σε οποιοδήποτε σημείο του αρχείου.
<code>ios::in</code>	Άνοιγμα αρχείου για είσοδο
<code>ios::out</code>	Άνοιγμα αρχείου για εγγραφή
<code>ios::trunc</code>	Διαγραφή των περιεχόντων του αρχείου αν υπάρχει (προκαθορισμένη συμπεριφορά και του <code>ios::out</code>)
<code>ios::binary</code>	Ανοίγει αρχείο για δυαδική ανάγνωση/εγγραφή (π.χ., όχι κείμενο)

- **ofstream** για έξοδο

- `ofstream outFile("persons.dat", ios::out);`
- `ofstream outFile("persons.dat"); //ios:out default`

- **ifstream** για είσοδο

- `ifstream inFile("persons.dat", ios::in);`
- `ifstream inFile("persons.dat"); //ios:in default`

- Επίσης:

- `fstream myfile;`
- `myfile.open ("example.bin", ios::out | ios::app | ios::binary);`



Άνοιγμα/κλείσιμο αρχείου

- Operations
 - Overloaded **operator!**
 - **!outClientFile**
 - Returns nonzero (true) if **badbit** or **failbit** set
 - Opened non-existent file for reading, wrong permissions
 - Closing file
 - **outClientFile.close()**
 - Automatically closed when destructor called

Παράδειγμα – Έξοδος

```
#include <iostream>
using std::cout;
using std::cin;
using std::ios;
using std::cerr;
using std::endl;
#include <fstream>
using std::ofstream;

int main(){
    ofstream outClientFile("persons.dat", ios::out );
    if ( !outClientFile ) {
        cerr << "File could not be opened" << endl;
        return 1;
    }
    char name[30];
    int age;
    while ( cin >> name >> age )
        outClientFile << name << ' ' << age << endl;
    return 0;
}
```



Παράδειγμα – Είσοδος

```
#include <iostream>
using std::cout;
using std::ios;
using std::cerr;
using std::endl;
#include <fstream>
using std::ifstream;
#include <iomanip>
#include <cstdlib>

int main(){
    ifstream inClientFile("persons.dat", ios::in );
    if ( !inClientFile ) {
        cerr << "File could not be opened" << endl;
        return 1;
    }
    char name[ 30 ];
    int age;
    while ( inClientFile >> name >> age )
        cout << name << ' ' << age << endl;
    return 0;
}
```



Παράδειγμα-Είσοδος-getline()

```
1 // reading a text file
2 #include <iostream>
3 #include <fstream>
4 #include <string>
5 using namespace std;
6
7 int main () {
8     string line;
9     ifstream myfile ("example.txt");
10    if (myfile.is_open())
11    {
12        while ( getline (myfile,line) )
13        {
14            cout << line << '\n';
15        }
16        myfile.close();
17    }
18
19    else cout << "Unable to open file";
20
21    return 0;
22 }
```

Η getline επιστρέφει το ίδιο το ifstream. Θα είναι false αν έχουμε φτάσει στο τέλος του αρχείου ή αν υπάρχει κάποιο σφάλμα.

Δείκτες σε θέση αρχείου

- Δείκτες σε θέση στο αρχείο
 - Number of next byte to read/write
 - Συναρτήσεις αλλαγής θέσης του δείκτη
 - **seekg** (seek get της **istream** class)
 - **seekp** (seek put της **ostream** class)
 - Οι κλάσεις διαθέτουν "get " και "put" δείκτες
 - **seekg** και **seekp** παίρνουν ως ορίσματα:
 - Offset: αριθμός bytes
 - Direction (προκαθορισμένη τιμή **ios::beg**)
 - **ios::beg** – σε σχέση με την αρχή του αρχείου
 - **ios::cur** – σε σχέση με την τρέχουσα θέση
 - **ios::end** – σε σχέση με το τέλος

(1) `istream& seekg (streampos pos);`

(2) `istream& seekg (streamoff off, ios_base::seekdir way);`

- **tellg** και **tellp**
 - `location = fileObject.tellg()`
 - επιστρέφει την τρέχουσα θέση

Παραδείγματα seekg

- Παραδείγματα
 - `fileObject.seekg(0)`
 - Πηγαίνει στην αρχή του αρχείου (location 0), προκαθορισμένο:
`ios::beg`
 - `fileObject.seekg(n)`
 - Πηγαίνει στο n-οστό byte από την αρχή
 - `fileObject.seekg(n, ios::cur)`
 - Μεταφέρεται n bytes μπροστά
 - `fileObject.seekg(y, ios::end)`
 - Πηγαίνει πίσω y bytes από το τέλος
 - `fileObject.seekg(0, ios::end)`
 - Πηγαίνει στο τελευταίο byte
 - `seekp` (όμοια)



Binary προσπέλαση

- Λειτουργίες
 - `write (char * memory_block, int size);`
 - `read (char * memory_block, int size);`



Ανάγνωση Binary

```
1 // reading an entire binary file
2 #include <iostream>
3 #include <fstream>
4 using namespace std;
5
6 int main () {
7     streampos size;
8     char * memblock;
9
10    ifstream file ("example.bin", ios::in|ios::binary|ios::ate);
11    if (file.is_open())
12    {
13        size = file.tellg();
14        memblock = new char [size];
15        file.seekg (0, ios::beg);
16        file.read (memblock, size);
17        file.close();
18
19        cout << "the entire file content is in memory";
20
21        delete[] memblock;
22    }
23    else cout << "Unable to open file";
24    return 0;
25 }
```

