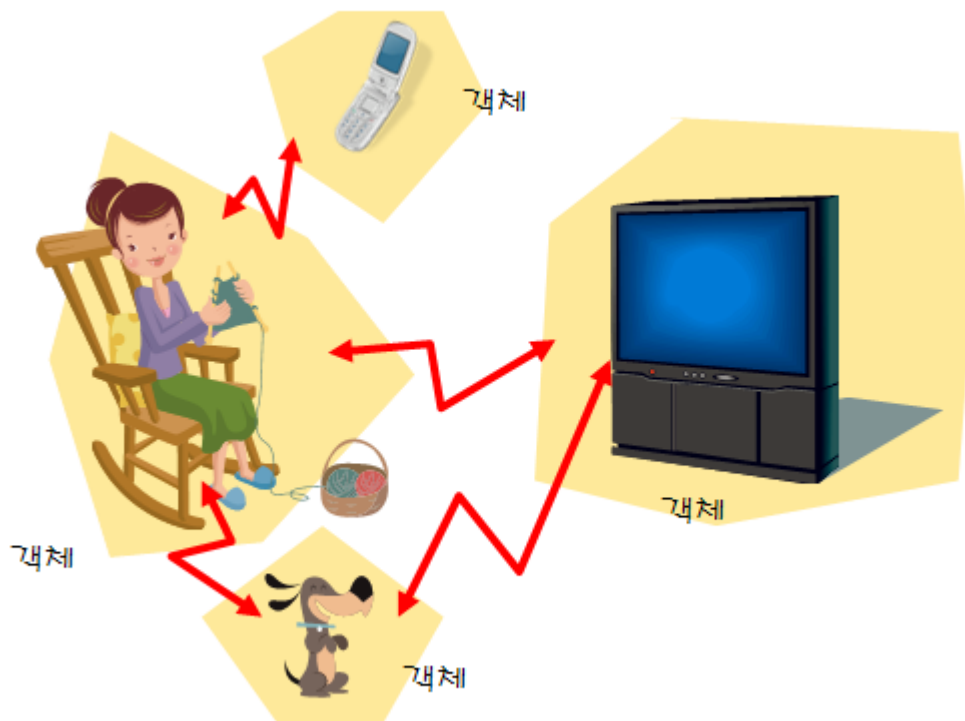




C++ Espresso

제13장 입출력과 파일처리





이번 장에서 학습할 내용



- 파일 입출력
- 텍스트 파일과 이진 파일
- 순차 파일과 랜덤 파일

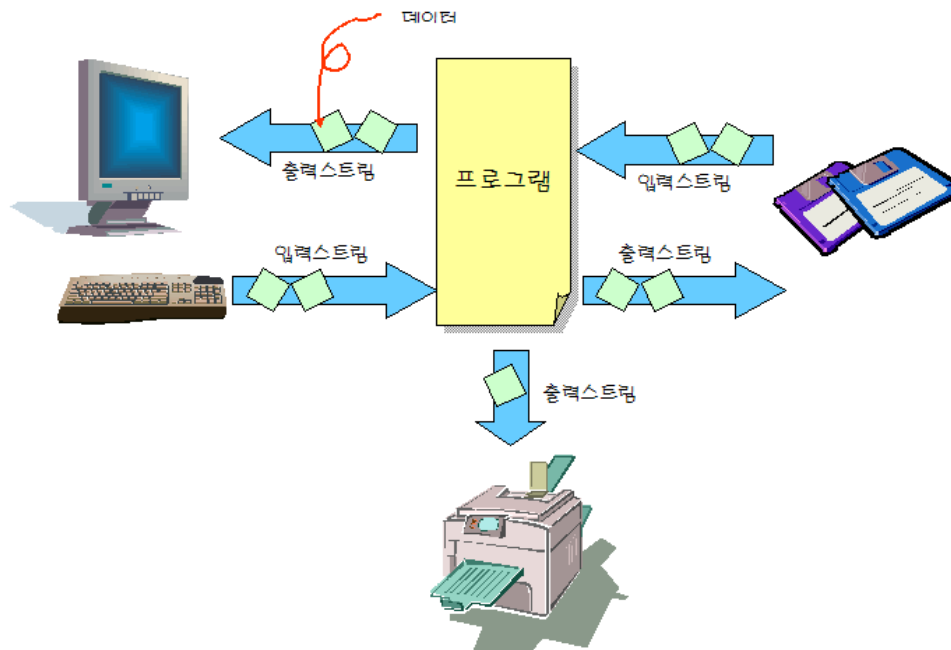
입출력과
파일처리에
대하여
살펴봅시다.





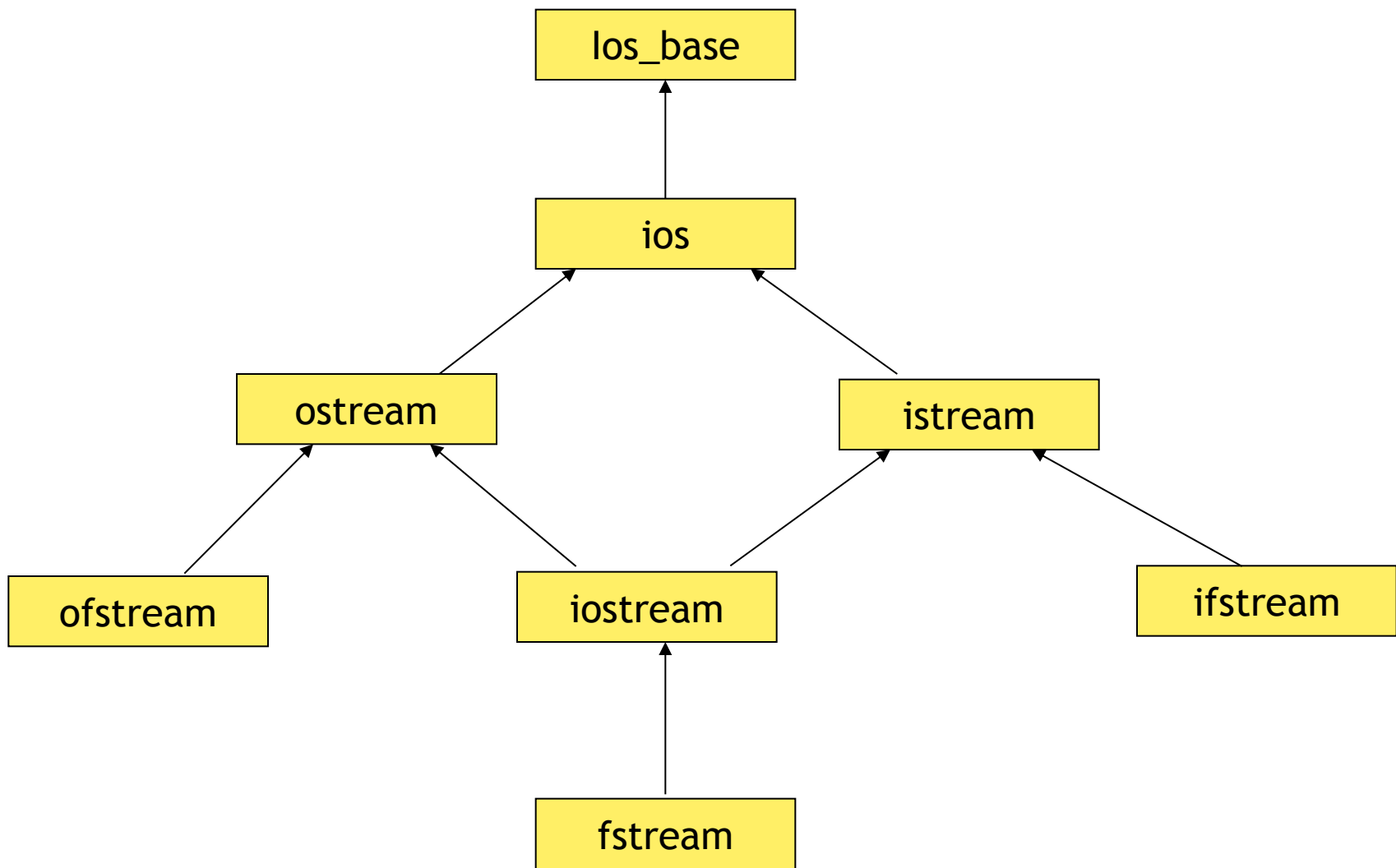
스트림 (stream)

- 스트림 (stream)은 “순서가 있는 데이터의 연속적인 흐름”이다.
- 스트림은 입출력을 물의 흐름처럼 간주하는 것이다.





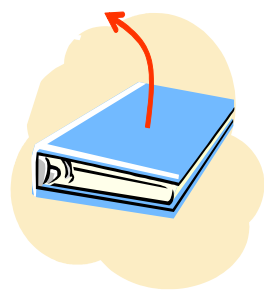
입출력 관련 클래스들



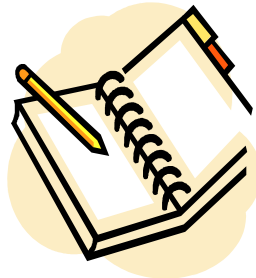


파일 처리의 순서

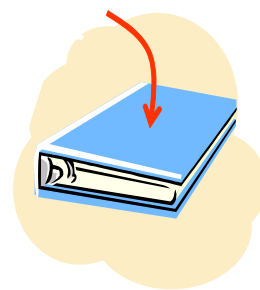
- 파일을 다룰 때는 반드시 다음과 같은 순서를 지켜야 한다.



파일 열기



파일 읽기와 쓰기



파일 닫기



<<과 >>을 이용한 입출력

- 입력

```
ifstream is;  
is.open("score.txt");  
int number;  
is >> number;
```

- 출력

```
ofstream os;  
os.open("result.txt");  
os << number;
```



예제 #1

- 학생 데이터를 파일에 저장하고 읽어보자.

```
int number;           // 학번  
char name[30];        // 이름  
int score;            // 성적
```





예제



```
#include <iostream>
#include <fstream>           // 파일 입출력을 하려면 헤더 파일을 포함하여야 한다.
using namespace std;

int main()
{
    ifstream is;
    is.open("score.txt");
    if( !is ) {               // ! 연산자 오버로딩
        cerr << "파일 오픈에 실패하였습니다" << endl;
        exit( 1 );
    }

    int number;               // 학번
    char name[30];           // 이름
    int score;                // 성적
    is >> number >> name >> score;
```




예제



```
ofstream os;  
os.open("result.txt");  
os << number << " " << name << " " << score << endl;  
  
is.close();  
os.close();  
return 0;  
}
```



score.txt

```
20100001 홍길동 100  
20100002 김유신 90  
20100003 강감찬 80
```



result.txt

```
20100001 홍길동 100
```



멤버 함수 이용 입출력

```
#include <iostream>
#include <fstream>           // 파일 입출력을 하려면 헤더 파일을 포함하여야 한다.
using namespace std;

int main()
{
    ifstream is;
    is.open("score.txt");
    if( !is ) {               // ! 연산자 오버로딩
        cerr << "파일 오픈에 실패하였습니다" << endl;
        exit( 1 );
    }
    char c;
    is.get(c);                // 하나의 문자를 읽는다.
    while(! is.eof() )
    {
        cout << c;
        is.get(c);
    }
    is.close();
    return 0;
}
```



멤버 함수 이용 입출력

파일 score.txt

```
20100001 홍길동 100  
20100002 김유신 90  
20100003 강감찬 80
```

실행 결과

```
20100001 홍길동 100  
20100002 김유신 90  
20100003 강감찬 80  
계속하려면 아무 키나 누르십시오 . . .
```



키보드에서 받은 문자 저장



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

// 파일 입출력을 하려면 헤더 파일을 포함하여야 한다.

```
int main()
{
    ofstream os;
    char c;
    os.open("test.txt");
    while( cin.get(c) )
    {
        os.put(c);
    }
    os.close();
    return 0;
}
```



키보드에서 받은 문자 저장

파일 test.txt

This is a test.

실행 결과

This is a test.

^Z

Press any key to continue



각 줄에 번호를 붙이는 예제



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

int main()
{
    ifstream is;
    ofstream os;
    is.open("score.txt");
    if( is.fail() ){
        cerr << "파일 오픈 실패" << endl;
        exit(1);
    }
    os.open("result.txt");
    if( os.fail() ){
        cerr << "파일 오픈 실패" << endl;
        exit(1);
    }
}
```



각 줄에 번호를 붙이는 예제



```
char c;
int line_number=1;
is.get(c);
os << line_number << ": ";
while(! is.eof() )
{
    os << c;
    if( c == '\n' ){
        line_number++;
        os << line_number << ": ";
    }
    is.get(c);
}
is.close();
os.close();
return 0;
}
```



각 줄에 번호를 붙이는 예제

파일 score.txt

```
20100001 홍길동 100  
20100002 김유신 90  
20100003 강감찬 80
```

파일 result.txt

```
1: 20100001 홍길동 100  
2: 20100002 김유신 90  
3: 20100003 강감찬 80
```




setf와 unsetf 멤버 함수를 이용한 출력 형식 지정

fmtflags setf(fmtflags); 멤버 함수의 사용 (fmtflags는 int형과 동일)

```
int main(void)
{
    bool bTF = true;
    int i = 16;
    double d = 12.0;

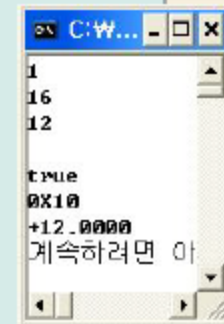
    cout << bTF << endl;
    cout << i << endl;
    cout << d << endl << endl;
    // 플래그 동시 설정
    cout.setf(ios_base::boolalpha | ios_base::showbase |
              ios_base::showpoint | ios_base::uppercase | ios_base::showpos);

    cout.unsetf(ios_base::dec);
    cout.setf(ios_base::hex);    // 16진수 설정

    cout << bTF << endl;
    cout << i << endl;
    cout << d << endl;

    return 0;
}
```

플래그 상수	설명	용도
boolalpha	bool 값을 true와 false로 표현	입출력
showbase	정수값 출력 시 진법 표시 접두어 사용	출력
showpoint	실수값 출력 시 소수점 표기	출력
uppercase	16진수 출력 시 X를 대문자로 표기 실수 과학적 표기 시 E를 대문자로 표기	출력
showpos	양수 앞에 + 부호 표기	출력
skipws	입력 시 공백 문자 무시 (디폴트 On)	입력





setf와 unsetf 멤버 함수를 이용한 출력 형식 지정

※ fmtflags setf(fmtflags, fmtflags); 멤버 함수의 사용

■ 관련 그룹별 지정

첫번째 매개변수	두번째 매개변수	설명	용도
dec, oct, hex	basefield	10진수, 8진수, 16진수 표기	입출력
fixed, scientific	floatfield	실수의 소수점 표기와 과학적 표기	출력
left, right, internal	adjustfield	왼쪽 정렬, 오른쪽 정렬, 부호와 진법점두어는 왼쪽 정렬 이고 값은 오른쪽 정렬	출력

```
int main(void)
{
    bool bTF = true;
    int i = 16;
    double d = 12.0;
```

```
    cout << bTF << endl;
    cout << i << endl;
    cout << d << endl << endl;
```

```
    cout.setf(ios_base::boolalpha | ios_base::showbase |
        ios_base::showpoint | ios_base::uppercase | ios_base::showpos);
```

```
    cout.setf(ios_base::hex, ios_base::basefield);
    cout.setf(ios_base::scientific, ios_base::floatfield);
```

```
    cout << bTF << endl;
    cout << i << endl;
    cout << d << endl;
```

```
    return 0;
}
```

16진수 설정

과학적 표기법

```
C:\WIN... - _ X
1
16
12

true
0X10
+1.200000E+001
계속하려면 아무 키를 누르세요
```



setf이외의 멤버 함수를 이용한 출력 형식 지정

▣ 출력 형식 지정을 위한 ostream 멤버 함수

함수	설명	디폴트값
int width(int i)	최소 필드 너비 조정 조정 후 첫 번째 출력 후에는 디폴트 값으로 자동 환원	출력 내용 과 동일
char fill(char c)	필드 내의 공백 자리에 채워질 문자 설정	공백
int precision(int p)	실수 출력 시 출력되는 총 자릿수, 출력 형식이 fixed 또는 scientific이라면 소수점 이하 자릿수	6

```
int main(void)
{
    cout.width(10);
    cout << "hello" << endl;
    cout.fill('%');
    cout.width(10);
    cout << "hello" << endl << endl;

    cout.setf(ios::left, ios::adjustfield);
    cout.width(10);
    cout << "hello" << endl << endl;
}
```

```
cout.width(10);
cout.precision(4);
cout << 123.1234567 << endl;
cout.width(10);
cout.precision(6);
cout << 123.1234567 << endl;

return 0;
}
```



입출력 조작자를 이용한 출력 형식 지정

▣ 입출력 조작자의 종류

입출력 조작자	대응 서식 플래그	설명	용도
boolalpha noboolalpha	setf(ios_base::boolalpha) unsetf(ios_base::boolalpha)	bool 값의 true, false 표현	입출력
showbase noshowbase	setf(ios_base::showbase) unsetf(ios_base::showbase)	진법 표시 접두어 사용	출력
showpoint noshowpoint	setf(ios_base::showpoint) unsetf(ios_base::showpoint)	소수점 표기	출력
uppercase nouppercase	setf(ios_base::uppercase) unsetf(ios_base::uppercase)	16진수 X, 과학적 표기 E 대문자 사용	출력
showpos noshowpos	setf(ios_base::showpos) unsetf(ios_base::showpos)	양수 앞에 + 부호 표기	출력
skipws noskipws	setf(ios_base::skipws) unsetf(ios_base::skipws)	입력 시 공백 문자 무시 (디폴트 값 : On)	입력
dec hex oct	setf(ios_base::dec, ios_base::basefield)	진수 표기	입출력
fixed scientific	setf(ios_base::fixed, ios_base::floatfield)	실수 표기	출력
left internal right	setf(ios_base::left, ios_base::adjustfield)	정렬 방법	출력



입출력 조작자를 이용한 출력 형식 지정

입출력 조작자의 종류 (계속)

입출력 조작자	대응 서식 플래그	설명	용도
setw(int)	width(int)	필드 너비 조정, 이후 한 번의 출력 후 디폴트로 환원됨	출력
setfill(char)	fill(char)	공백 자리 채움 문자 지정	출력
setprecision(int)	precision(int)	실수 출력 자릿수 설정	출력
endl		newline 문자 출력, 스트림 비움	출력
flush		스트림을 비움	출력
setiosflags	setf		
resetiosflags	unsetf		

```
#include <iostream>
#include <iomanip>          // setfill, setw
using namespace std;

int main(void)
{
    cout << hex << 100 << endl;    // 16진수 출력
    cout << oct << 10 << endl;     // 8진수 출력

    cout << setfill('X') << 100 << setw(10) << endl; // 채움문자, 필드 설정
    cout << 100 << " hi " << endl;

    return 0;
}
```

