

30장. 입출력

01_ C++ 스타일의 입출력

02_ C 스타일의 입출력

cout과 cin 객체 다시 보기(1)

- cout과 cin 객체의 타입을 알아보자.

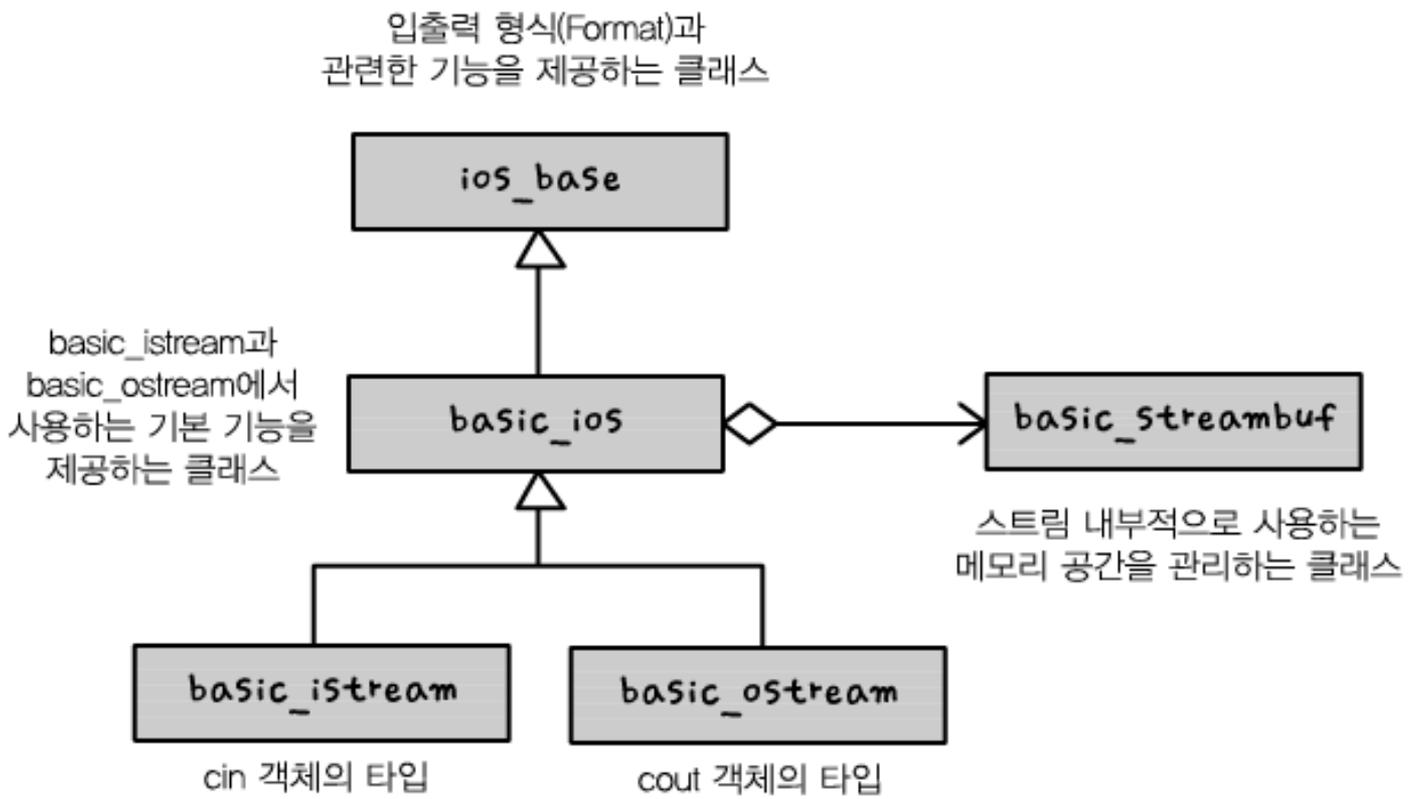
```
typedef basic_ostream<char> ostream;  
typedef basic_istream<char> istream;  
  
extern ostream cout;  
extern istream cin;
```

- 스트림의 개념



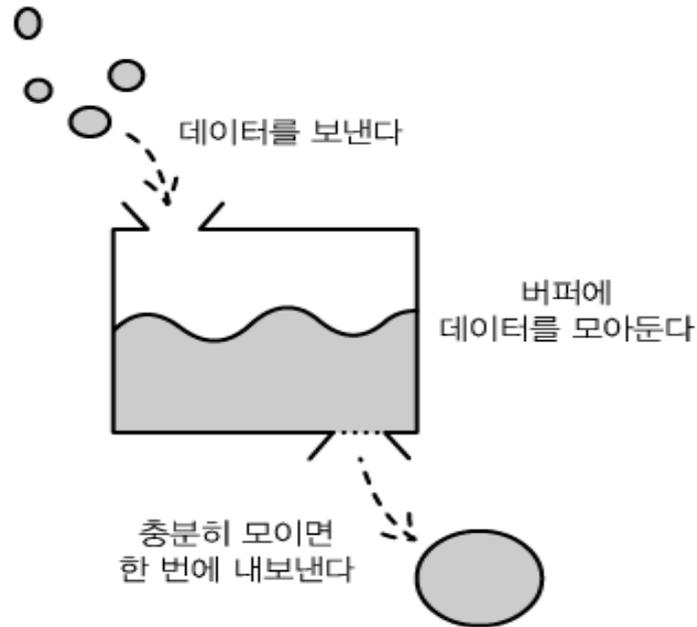
cout과 cin 객체 다시 보기(2)

- 입출력 클래스의 상속 계층도



cout과 cin 객체 다시 보기(3)

- 입출력과 버퍼링



- 버퍼에 쌓인 데이터를 강제로 방출하는 예

```
cout << "Hello, World~\n" << flush;  
cout << "Hello, World~" << endl;
```

입출력 형식 지정하기(1)

- setf() 함수의 원형

```
fmtflags setf( fmtflags f );  
fmtflags setf( fmtflags f, fmtflags mask );
```

- setf() 함수의 사용 예 (p.874 참조)

```
cout.setf( ios_base::boolalpha );  
cout << true << endl; // 1이 아닌 true가 출력된다.
```

```
cout.setf( ios_base::hex, ios_base::basefield );  
cout << 123 << endl; // 7B를 출력된다.
```

```
cout.setf( ios_base::showpoint | ios_base::showpos );  
cout << 123.0 << endl; // +123.0000이 출력된다.
```

입출력 형식 지정하기(2)

- setf() 함수로 지정한 형식을 되돌리는 예
 - 인자가 두 개인 setf() 함수를 사용한 경우에 되돌리기

```
// setf()의 반환 값을 보관한다.  
ios_base::fmtflags old_flags;  
old_flags = cout.setf( ios_base::scientific, ios_base::floatfield );  
cout << 12.34 << endl;          // 1.234000e+001을 출력한다.  
  
// 보관한 값으로 다시 세팅한다.  
cout.setf( old_flags, ios_base::floatfield );  
cout << 12.34 << endl;          // 12.34를 출력한다.
```

```
cout.setf( ios_base::showpos );  
cout << 333 << endl;            // +333  
  
cout.unsetf( ios_base::showpos );  
cout << 333 << endl;            // 333을 출력한다.
```

입출력 형식 지정하기(3)

- `width()` 함수를 사용해서 값이 출력될 공간의 크기를 지정할 수 있다.

```
cout.width( 10 );  
cout << 333 << ", " << 334 << endl; // _____ 333,334를 출력한다.
```

- `precision()` 함수를 사용해서 소수점 이하 자리 수를 지정할 수 있다.

```
cout.setf( ios_base::fixed );  
cout.precision( 2 );  
cout << 3.33333 << endl; // 3.33을 출력한다.
```

- `fill()` 함수를 사용해서 빈 칸을 채우는 문자를 지정할 수 있다.

```
cout.fill( '*' );  
cout.width( 10 );  
cout << 333 << 두이; // *****333을 출력한다.
```

조종자(Manipulator)

- 조종자를 사용하면 편하게 입출력 형식을 지정할 수 있다.
(p.878 참조)

```
cout << hex << 333 << endl;           // 14d를 출력한다.
```

```
cout << setw( 10 ) << 333 << endl;     // _____333을 출력한다.
```

```
cout << setiosflags( ios_base::scientific ) << 12.34 << endl; // 1.234000e+001  
cout << resetiosflags( ios_base::scientific ) << 12.34 << endl; // 12.34
```

스트림의 상태

- 스트림의 상태를 알아보기 위한 함수들

```
bool good() const; // true면, 정상적인 상태
bool eof() const; // true면, 파일이나 입력의 끝에 도달
bool fail() const; // true면, 예상치 못한 입력이 들어오는 등의 문제 발생
bool bad() const; // true면, 잘못된 파일 등의 문제 발생
```

- 현재 cin 객체의 상태가 정상적인지 알아보는 예

```
if ( cin.good() )
{
    // 정상 상태
}
```

- cin 객체의 상태를 정상으로 되돌리는 예

```
cin.clear();
cin.ignore( numeric_limits<streamsize>::max(), '\n' );
```

파일 입출력

- ofstream을 사용해서 파일을 여는 예 (p.881 참조)

```
ofstream file1( "test1.txt" );

ofstream file2;
file2.open( "test2.txt" );

// 출력용으로 기존 파일을 지워 버리고 새 파일을 연다.
ofstream file3( "test.txt", ios_base::out | ios_base::trunc );

// 출력용으로 기존 파일의 뒤에 내용을 추가할 수 있도록 연다.
ofstream file4( "test.txt", ios_base::out | ios_base::app );
```

- 파일을 여는데 성공했는지 확인하는 예

```
ofstream file( "test.txt" );
if ( false == file.is_open() )
{
    // 파일을 여는데 실패
}
```

C 스타일의 입출력 (1)

- printf(), scanf()를 사용한 콘솔 입출력

```
#include <stdio.h>

int main()
{
    printf("세 개의 숫자를 입력하세요~\n");

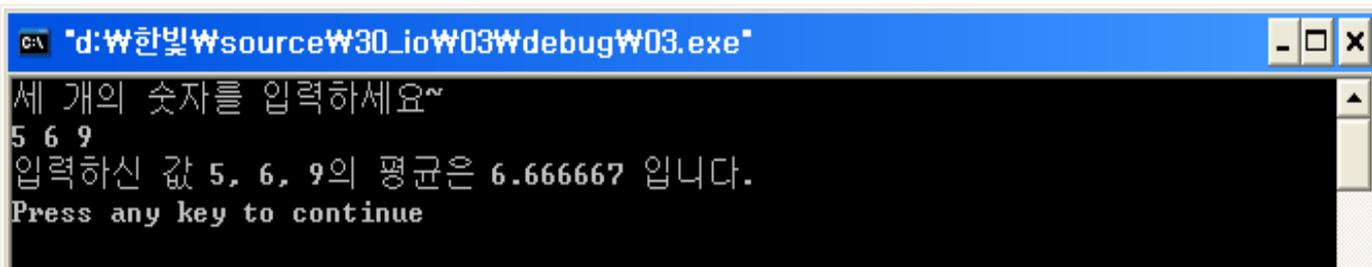
    // 세 개의 정수를 입력받는다.
    int a, b, c;
    scanf("%d%d%d", &a, &b, &c);

    float ave = float(a + b + c) / 3.0f;

    printf("입력하신 값 %d, %d, %d의 평균은 %f 입니다.\n", a, b, c, ave);

    return 0;
}
```

- 실행 결과

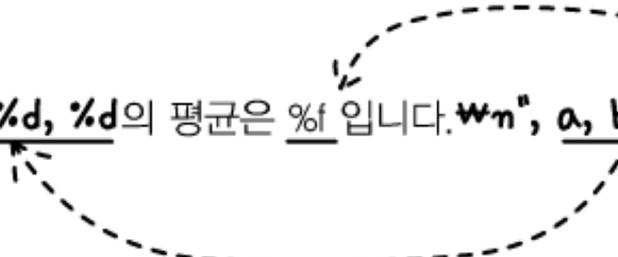


```
C:\ "d:\한빛\source\30_io\03\debug\03.exe"
세 개의 숫자를 입력하세요~
5 6 9
입력하신 값 5, 6, 9의 평균은 6.666667 입니다.
Press any key to continue
```

C 스타일의 입출력 (2)

- printf()의 사용

```
printf("입력하신 값 %d, %d, %d의 평균은 %f 입니다.wn", a, b, c, ave);
```



- printf()와 scanf() 함수에 사용하는 특수 문자

문자	의미
c	매칭하는 인자를 문자로 출력한다. (charater)
d	매칭하는 인자를 10진수로 출력한다. (decimal)
o	매칭하는 인자를 8진수로 출력한다. (octal)
u	매칭하는 인자를 부호 없는 정수로 출력한다. (unsigned)
x	매칭하는 인자를 16진수로 출력한다. 7abc처럼 소문자로 출력. (hexadecimal)
X	매칭하는 인자를 16진수로 출력한다. 7ABC처럼 대문자로 출력.
e	매칭하는 인자(실수)를 1.23e+6처럼 출력한다. e는 소문자로 출력. (exponential)
E	매칭하는 인자(실수)를 1.23E+6처럼 출력한다. E는 대문자로 출력.
f	매칭하는 인자(실수)를 123.56처럼 출력한다.(floating)
s	매칭하는 인자(문자열의 주소)를 문자열로 출력한다. (string)

C 스타일의 입출력 (3)

- printf()의 기본적인 사용

```
printf("%d ", 65); // 65
printf("%o ", 65); // 101
printf("%x ", 65); // 41
printf("%c ", 65); // A

printf("%f ", 12.34); // 0.000000
printf("%e ", 12.34); // 1.234000e+001

printf("%s 미디어 ", "한빛"); // 한빛 미디어
```

- 형식을 지정하는 예

```
printf( "%10d", 333 ); // _____333 출력
printf( "%010d", 333 ); // 0000000333 출력
printf( "%+10d", 333 ); // _____+333 출력
printf( "%-10d", 333 ); // 333_____ 출력
printf( "%.2f", 12.345678 ); // 12.35 출력
```

Q&A

