



SEOIL UNIVERSITY

입출력
템플릿



SEOIL UNIVERSITY

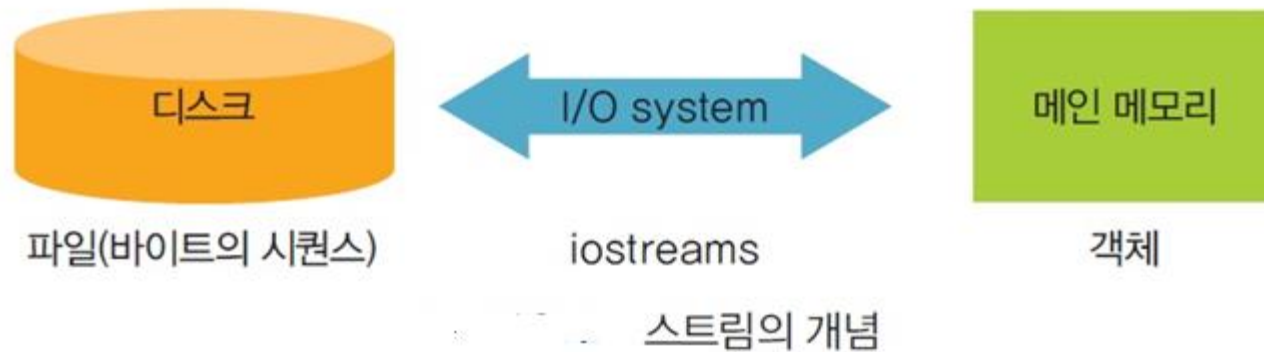
New Vision,
Young Challenge!

내게 힘이 되는대학! 서일대학교

스트림(stream)

- 스트림(stream)은 “순서가 있는 데이터의 연속적인 흐름”
- 스트림은 입출력을 물의 흐름처럼 간주

```
#include <fstream>
```



클래스	설명
ofstream	출력 파일 스트림 클래스이다. 출력 파일을 생성하고 파일에 데이터를 쓸때 사용한다.
ifstream	입력 파일 스트림 클래스이다. 파일에서 데이터를 읽을 때 사용한다.
fstream	일반적인 파일 스트림을 나타낸다.

<<과 >>을 이용한 입출력

■ 입력

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

20100001	홍길동	100
20100002	김유신	90
20100003	강감찬	80

■ 출력

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

■ score.txt

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

■ Result.txt

```
20100001 홍길동 100
```

```
#include <iostream>
#include <fstream>
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;

    ofstream os;
    os.open("result.txt");
    os << number << " " << name << " " << score << endl;

    is.close();
    os.close();
    return 0;
}
```

두가지 결과 확인

■ 1

- `os << number << " " << name << " " << score << endl;`
- `os << number << " " << name << " " << score << endl;`

■ 2

- `is >> number >> name >> score;`
- `os << number << " " << name << " " << score << endl;`

```
int main()
{
    ofstream os{ "numbers.txt" };
    if (!os) {
        cerr << "파일 오픈에 실패하였습니다" << endl;
        exit(1);
    }
    for(int i=0;i<100; i++)
        os << i << " ";
    return 0;

    // 객체 os가 범위를 벗어나면 ofstream 소멸자가 파일을 닫는다.
}
```

```
int main()
{
    ifstream is{ "numbers.txt" };
    if (!is) {
        cerr << "파일 오픈에 실패하였습니다" << endl;
        exit(1);
    }
    int number;
    while (is) {
        is >> number;
        cout << number << " ";
    }
    cout << endl;
    return 0;

    // 객체 is가 범위를 벗어나면 ifstream 소멸자가 파일을 닫는다.
}
```

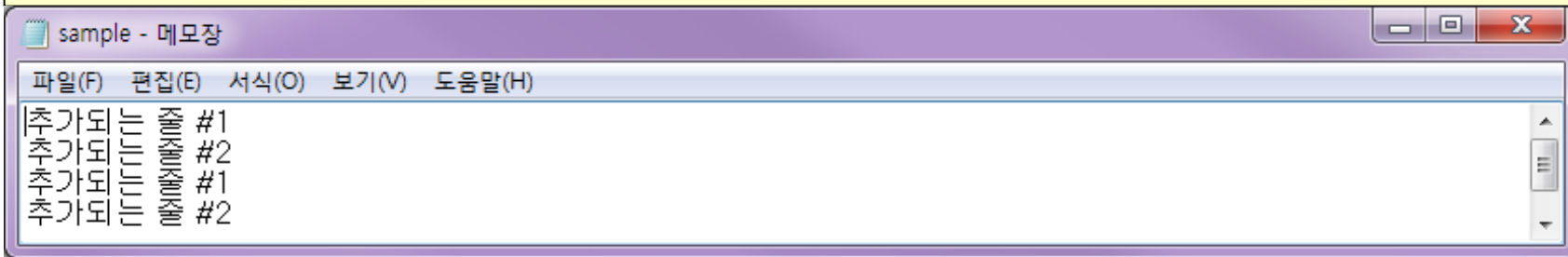

ios::in	입력을 위하여 파일을 연다.
ios::out	출력을 위하여 파일을 연다.
ios::binary	이진 파일 입출력을 위하여 파일을 연다.
ios::ate	파일의 끝을 초기 위치로 한다.
ios::app	파일의 끝에 추가된다.
ios::trunc	새로운 내용으로 교체된다.

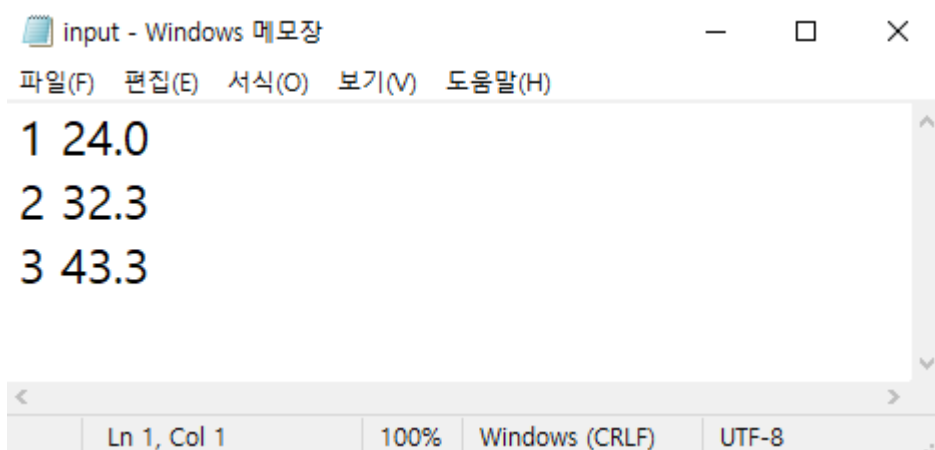
```
int main()
{
    using namespace std;

    ofstream os("input.txt", ios::app);
    if (!os)
    {
        cerr << "파일 오픈에 실패하였습니다" << endl;
        exit(1);
    }

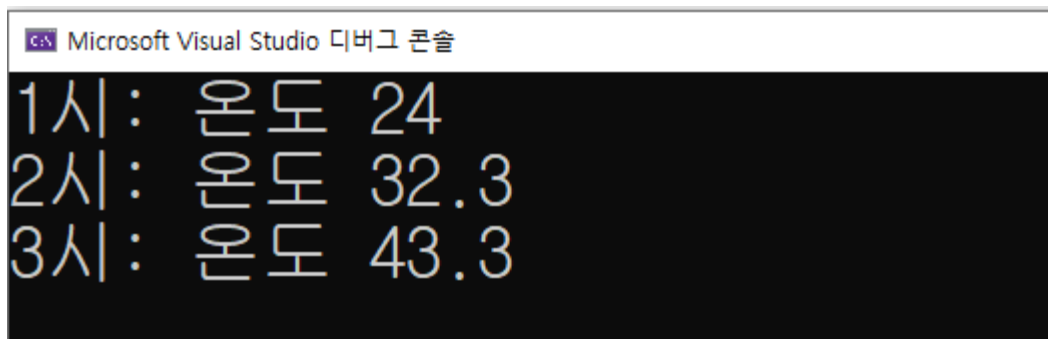
    os << "추가되는 줄 #1" << endl;
    os << "추가되는 줄 #2" << endl;

    return 0;
}
```





```
input - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
1 24.0
2 32.3
3 43.3
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```



```
Microsoft Visual Studio 디버그 콘솔
1시: 온도 24
2시: 온도 32.3
3시: 온도 43.3
```

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

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

    int hour;
    double temperature;

    while (is >> hour >> temperature) {
        cout << hour << "시: 온도 " << temperature << endl;
    }
    return 0;
}
```

```
#include<vector>
class TempData {
public:
    int hour;
    double temperature;
};

int main()
{
    ifstream is{ "input.txt" };
    if (!is) {                // ! 연산자 오버로딩
        cerr << "파일 오픈에 실패하였습니다" << endl;
        exit(1);
    }
    vector<TempData> temps;
```

```
int hour;
double temperature;
while (is >> hour >> temperature) {
    temps.push_back(TempData{ hour, temperature });
}
for ( TempData t : temps) {
    cout << t.hour << "시: 온도 " << t.temperature << endl;
}
return 0;
}
```

좌표

- 함수에 파일명만 지정하면 현재 작업 디렉터리(working directory)에서 파일을 연다.
- ("hello.txt", "w") 현재 작업 디렉터리의 hello.txt 파일
 - 절대 경로 : `ifstream is("c:\\Win\\Wininput");`
 - 상대 경로 : `ifstream is{ "in \\Wininput.txt" };` 상대 경로는 작업 디렉터리를 기준의 파일
 - 상대경로 기준 `..\\` 는 부모디렉토리를 뜻함
 - `ifstream is{ "..\\Win\\Wininput.txt" };`
- 리눅스 및 OS X는 경로를 표현할 때 `\\` 대신 `/`를 사용
 - `fopen("/home/project/hello.txt", "w");`

함수 템플릿

- C++에서도 하나의 형틀을 만들어서 다양한 코드를 생산해 내도록 할 수 있는데 이것을 템플릿이라고 한다. 함수 템플릿(function template)은 함수를 찍어내기 위한 형틀이다.

int버전으로 필요하시다구요.



템플릿함수

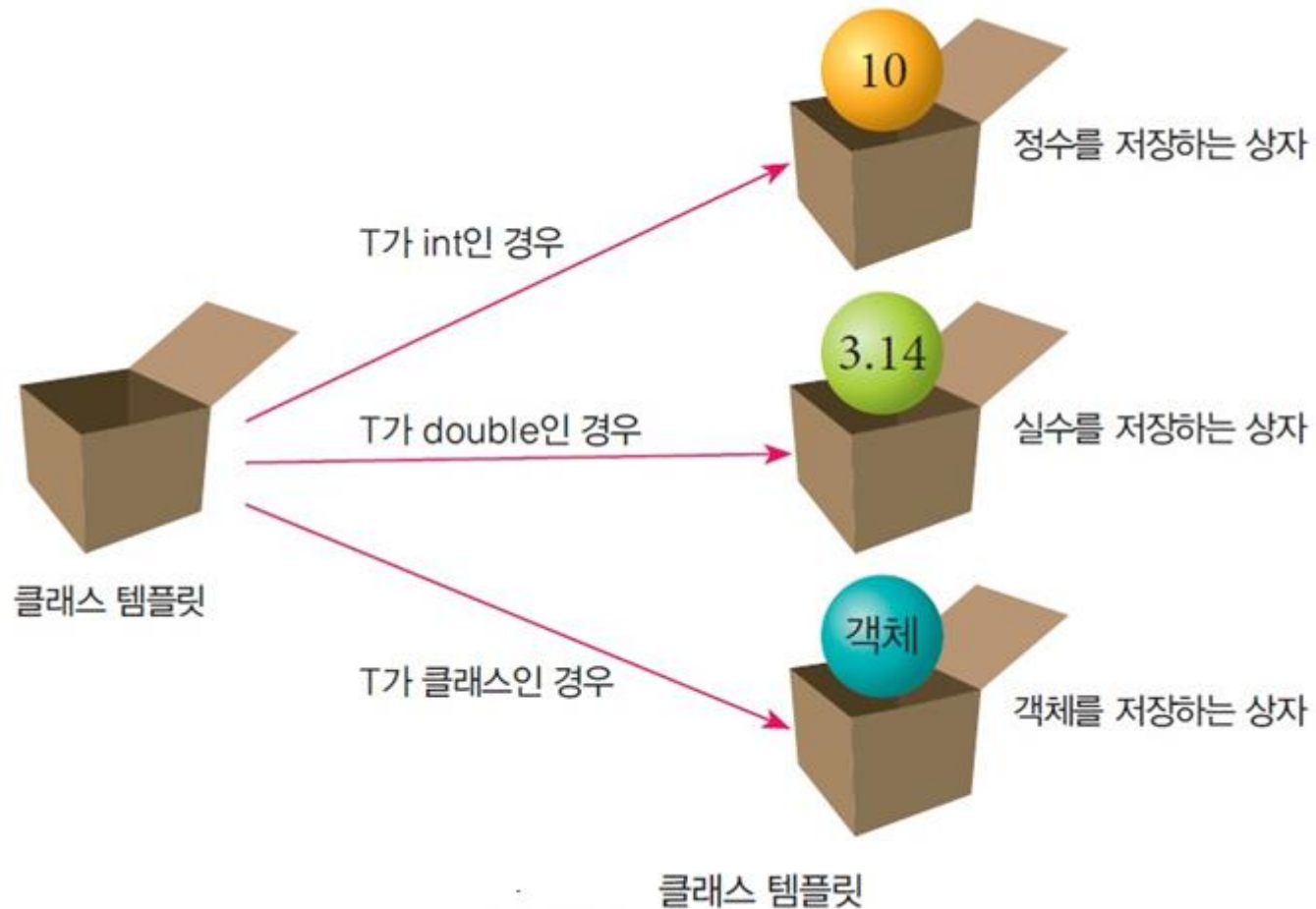
```
_____ get_max(____x, ____ y)
{
    if( x > y)    return x;
    else return y;
}
```



```
int get_max(int x, int y)
{
    if( x > y)    return x;
    else return y;
}
```

일반화 프로그래밍의 개념

클래스 템플릿

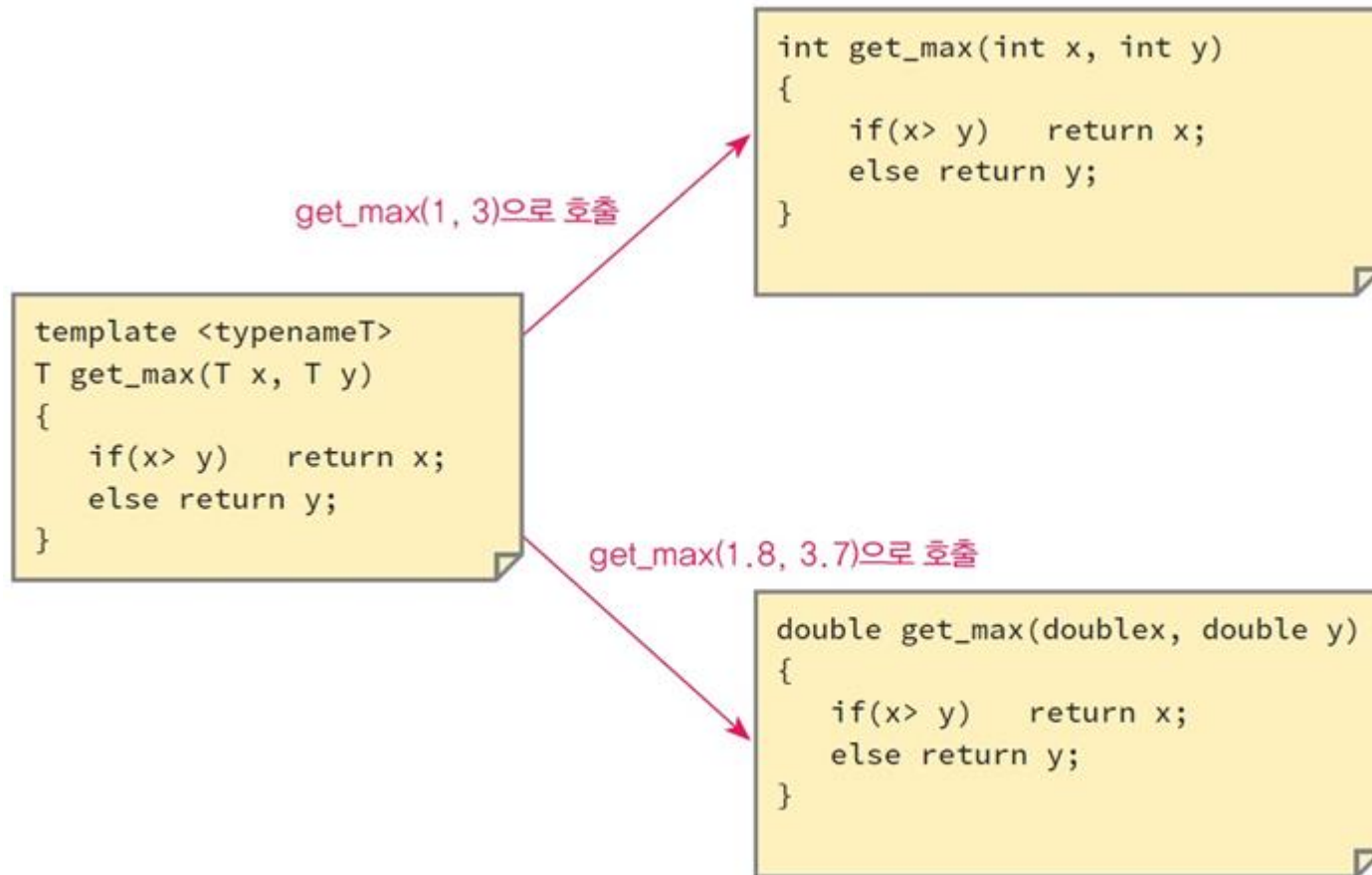




템플릿 프로그래밍

```
template<typename T>  
T get_max(T x, T y)  
{  
    if( x > y ) return x;  
    else return y;  
}
```

템플릿 프로그래밍



템플릿 함수



템플릿 프로그래밍

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

template <typename T>
T get_sum(T x)
{
    x = x + 10;
    return x;
}

int main()
{
    cout << get_sum(3) << endl;

    return 0;
}
```



템플릿 프로그래밍

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

template <typename T>
T get_max(T x, T y)
{
    if (x > y) return x;
    else return y;
}

int main()
{
    cout << get_max(1, 3) << endl;
    cout << get_max(1.2, 3.9) << endl;
    return 0;
}
```

C:\Windows\system32\cmd.exe

```
3
3.9
계속하려면 아무 키나 누르십시오 . . .
```

함수 템플릿과 함수 오버로딩

```
template <typename T>
T get_sum(T x)
{
    x = x + 10;
    return x;
}

template <typename T>
T get_sum(T x, T y)
{
    if (x > y) return x;
    else return y;
}

int main()
{
    cout << get_sum(3) << endl;
    cout << get_sum(1.2, 3.9) << endl;

    return 0;
}
```



클래스 템플릿

클래스 템플릿

```
template <typename T>
class 클래스이름
{
    ...// T를 어디서든지 사용할 수 있다.
}
```

예제

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

template <typename T>
class Box {
    T data; // T는 타입(type)을 나타낸다.
public:
    Box() { }
    void set(T value) {
        data = value;
    }
    T get() {
        return data;
    }
};
```


예제

```
int main()
{
    Box<int> b1;
    b1.setBox(100);
    b1.view();

    Box<double> b2;
    b2.setBox(3.14159);
    b2.view();;

    return 0;
}
```



C:\Windows\system32\cmd.exe

```
100
3.14159
계속하려면 아무 키나 누르십시오 . . .
```



SEOIL UNIVERSITY

감사합니다.