

## ■ 정오표

[p.36: 아래에서 3행]

<http://www.acornpub.co.kr/book/programming-bjarne>

->

<http://www.acornpub.co.kr/book/programming-cplus>

[p. 79: 아래에서 11행]

>>는 “cout에 출력하라”는 의미로

->

<<는 “cout에 출력하라”는 의미로

[p. 80: 아래에서 7행]

cout과 출력 연산자인 >>를 사용한다.

->

cout과 출력 연산자인 <<를 사용한다.

[p. 94: 일괄 변경]

성

->

이름

[p.102: 아래에서 2행]

```
int a = 3;
```

->

```
int a = 3;
```

[p.113 : 18행]

```
cout << <>
```

```
cout << c << ' ' << i1 << ' ' << c2 << 'Wn';
```

[p.117: 첫 행]

```
int char b1 {1000};
```



스프링 웹 개발 마스터 (세트)



미래를 바꾼 아홉 가지 알고리즘



->

char b1 {1000};

[p.131: 표 제외 아래에서 3행]

지금까지 가장 간단한

->

지금까지 가장 간단한

[p.179: 아래에서 7행]

// 잘못된 인자인 경우 'b' 반환

->

// 잘못된 인자인 경우 '-1' 반환

[p.180: 아래에서 1행]

잠정만을

->

장점만을

[p.189 : 위에서 4행]

for (int x : temps)

->

for (double x : temps)

[p.219 : 아래에서 2행]

t2.value = 3.14;^

->

t2.value = 3.14;

[p.226 : 위에서 5행]

Term 그 자체다.

->

Expression 그 자체다.

[p.249: 2행]

case '(' : case ')' : case '+' : case '-' : case ' ' : case '/' :



->

*case '(: case ')': case '+': case '-': case "": case '/':*

('- '에서 - 뒤의 공백 삭제)

[p.254 : 아래에서 9행]

그리는 !는

->

그리고 !는

[p.266: 15행]

case '- ':

->

case '-':

[p.268: 13행]

case '- ':

->

case '-':

[p.308 : 1행]

(형식을 선택할 수 있는 기능도 제동한다.)

->

(형식을 선택할 수 있는 기능도 제공한다.)

[p.312: 5행]

집작

->

집착

[p.318 : 6행]

"아주 작다"는 말은

->

"매우 작다"는 말은

[p.333 : 실습문제 1행]



use.cpp에는 my.h와 `std/libfacilities.h`를 `#include` 한다.

->

use.cpp에는 my.h를 `#include` 한다.

#### [p.333 : 실습문제 8행 - 하단 역자주]

my.h에서는 `foo`를 정의하지 않고 선언하기만 했으므로, 변수 `foo`를 사용하려면 my.cpp나 main.cpp 중의 한 곳에서 `int foo = 0;` 과 같이 `foo`를 정의해야 한다. 물론 `foo`를 전역 이름공간에서 선언했으므로 정의도 전역 이름공간에서 해야 한다.

#### [p.334 : 4행]

예상 밖의 결과라면 8.6절을 확인하자.

->

예상 밖의 결과라면 8.5.6절을 확인하자.

#### [p.340 : 7행]

Sahpe

->

Shape

#### [p.348 : 5행]

9.44절에서

->

9.4.4절에서

#### [p.358 : 1행]

Month m = Sep;

->

Month m = Month::Sep;

#### [p.372 : 13, 14행]

```
if (m == feb && d == 29 && !leapyear(y + n)) { // 윤년 처리!
```

```
    m = mar;
```

->

```
if (m == Month::feb && d == 29 && !leapyear(y + n)) { // 윤년 처리!
```

```
    m = Month::mar;
```



[p.397 : 15행]

```
if (isdigit(ch) || ch=="-") {  
->  
if(isdigit(ch) || ch=='-') {
```

[p.431 : 아래에서 8행]

```
"1.34e- 3"  
->  
"1.34e-3"
```

[p.439 : 위에서 13행]

```
// 필요한 문자 취환 수행  
->  
// 필요한 문자 치환 수행
```

[p.458 : 아래에서 1행]

```
각춘  
->  
갖춘
```

[p. 488 : 9행]

```
void Lines::Lines(initializer_list< Ist>  
-> Lines::Lines(initializer_list< Ist>
```

[p.514 : 아래에서 8행]

```
Marked_polyline(const string& m, initializer_list<Point> lst)  
->  
Marks(const string& m, initializer_list<Point> lst)
```

[p.671 : 18행]

```
arg.elem.sz  
->  
arg.elem+sz
```

[p.673 : 위에서 13행]



a.elem.sz

->

a.elem+sz

[p.678 : 위에서 6행]

fill(cin)

->

fill2(cin);

[p.690]

[p.690 : 아래에서 14행]

여기서 ch의 크기, 즉 sizeof(p) 는 100이다.

->

여기서 ch의 크기, 즉 sizeof(ch) 는 100이다.

[p.690 : 아래에서 11행]

여기서 p는 &ch[0]으로 초기화되며, sizeof(ch)는 (100이 아닌) 4다.

->

여기서 p는 &ch[0]으로 초기화되며, sizeof(p)는 (100이 아닌) 4다.

[p.720: 아래에서 3행]

템플릿 사용자의 몫은 아니다.

->

템플릿 사용자의 몫은 아니다.

[p.787 : 아래에서 13행]

p가 여전히 4의 값을 갖는 요소를 가리킨다는 점에 주목하자.

->

q가 여전히 4의 값을 갖는 요소를 가리킨다는 점에 주목하자.

[p.802 : 아래에서 6행]

기본적으로 항등 비교는 ==를, 순서 비교는 >(~보다 작다)을 이용해서 수행한다.

->

기본적으로 항등 비교는 ==를, 순서 비교는 <(~보다 작다)을 이용해서 수행한다.



[p.910 : 아래에서 7행]

for (const auto& x : m)

->

for (const auto& x : \*m)

[p.911 : 3행]

for (const auto& x : m)

->

for (const auto& x : \*m)

[p.916 : 2행]

TX 77845를 처리할 수 없다.

->

TX 77845를 처리할 수 없다.

[p.917 : 함수 10행]

regex pat {R"(\w{2}\s\*\d{5}(- \d{4})?)"};

=>

regex pat {R"(\w{2}\s\*\d{5}(-\d{4})?)"};

[p.924 : 아래에서 4행]

[a- zA- Z]

->

[a-zA-Z]

[p.927 : 2행]

include

->



include

[p 968 : 9행]

Using cmplx

->

using cmplx

[p.1063 : 첫 번째 함수]

아래 이미지 참고

```
template<class Iter, class T>
bool b2(Iter first, Iter last, const T& value)
{
    // check if [first:last) is a sequence:
    if (last<first) throw Bad_sequence();

    // check if the sequence is ordered:
    if (2<=last-first)
        for (Iter p = first+1; p<last; ++p)
            if (*p<*(p-1)) throw Not_ordered();

    // all's OK, call binary_search:
    return binary_search(first,last,value);
}
```

[p.1065 : 3행]

binary\_sort

->

binary\_search

[p.1068 : 예제 코드]

using namespace std;





->  
using namespace std;  
using namespace std::chrono;

[p.1071 : 3행]

쉬어야

->

쉬워야

[p.1101 : 5행]

/\* s2와 s1에서 n 바이트를 복사 (strcmp와 비슷함): \*/

->

/\* s1과 s2를 n 바이트만큼 비교 (strcmp와 비슷함): \*/

[p.1101 : 11행]

함수를

->

함수들을

[p.1130 : 아래에서 9행]

123 // double: 0.123

->

[p.690 : 아래에서 14행]

여기서 ch의 크기, 즉 sizeof(p) 는 100이다.

->

여기서 ch의 크기, 즉 sizeof(ch) 는 100이다.

[p.690 : 아래에서 11행]

여기서 p는 &ch[0]으로 초기화되며, sizeof(ch)는 (100이 아닌) 4다.

->

여기서 p는 &ch[0]으로 초기화되며, sizeof(p)는 (100이 아닌) 4다.

.123 // double: 0.123



댓글 0개


정렬 기준 날짜



댓글 달기...

 Facebook 댓글 플러그인



에이콘출판(주) | (07967) 서울시 양천구 국회대로 287 (목동) acornpub@acornpub.co.kr  02-2653-7600  02-2653-0433 | [사이트맵](#)

acorn publishing Co. ©2013. All Rights Reserved

