



# 따울림 객체 스터디

# 2주차

1

---

---

코딩  
스타일

---

---

2

---

---

연산자

---

---

3

---

---

조건문,  
반복문

---

---

---

# 1 코딩 스타일

---

## 코딩을 하는 방식

기계를 이해시키는 건 어렵지 않다.

다른 **사람도** 이해할 수 있는 코드가 좋은 코드이다.

코드를 작성한 의도를 최대한 알 수 있게끔 작성해야함  
짧다고 좋은 코드가 절대 아님!!

# 1 코딩 스타일

---

Q) 용준이는 지금  $x$ 원을 가지고 있는데, 축구에 대한 열정이 불타올라서 축구화와 축구공을 사려고 한다.

현재 돈과 축구화와 축구공의 가격을 입력 받고, 구매 후 남은 돈을 출력하시오.

---

# 1 코딩 스타일

---

## 니네가 하는거

```
#include <iostream>

using namespace std;

int main()
{
    int a, b, c;

    cin >> a >> b >> c;

    cout << a - b - c;
}
```

---

# 1 코딩 스타일

---

## 니네가 해야 하는 거

```
#include <iostream>

using namespace std;

int main()
{
    int current_money, ball_price, shoes_price;

    cin >> current_money >> ball_price >> shoes_price;

    int remained_money = current_money - (ball_price + shoes_price);

    cout << remained_money;
}
```

---

# 1 코딩 스타일

---

## 네이밍(Naming) 규칙

**camelCase** : 각 단어의 첫 글자를 대문자로 ,맨처음 문자만 소문자

**PascalCase** : 각 단어의 첫 글자를 대문자로

**snake\_case** : 각 단어를 밑줄로 구분

# 1 코딩 스타일

## 중괄호 위치

### K&R

```
if(a == 1) {  
    for (i = 0; i < 1000; i++) {  
        if (b == 10) {  
            처리();  
        } else if(c == 10) {  
            처리();  
        }  
    }  
}
```

코드의 길이가 짧아짐  
가장 많이 쓰이는 스타일

### BSD

```
if (a == 1)  
{  
    for (i = 0; i < 1000; i++)  
    {  
        if (b == 10)  
        {  
            처리();  
        }  
        else if(c == 10)  
        {  
            처리();  
        }  
    }  
}
```

가독성이 조금 더 좋아짐  
코드를 수정할 때 편리



---

# 1 코딩 스타일

---

## 코딩 컨벤션(Coding convention)

각 코딩 스타일은 서로 장단점이 있다.

하지만 여러 명의 개발자가 모인다면? => **혼란**

**그렇다면, 표준을 정하자!!**

---

# 1 코딩 스타일

---

## 구글의 c++ 코딩 컨벤션

---

이름은 최대한 그 대상의 이유를 담아야 한다.

변수는 명사형으로, 함수는 동사형으로 만든다.

약어는 사용하지 않는 것이 좋다.

---

# 1 코딩 스타일

---

## 1. 변수 네이밍

변수는\_모두\_소문자로\_쓰고\_단어는\_밑줄로\_구분  
명사형으로 쓴다

Ex) math\_score, student\_number

---

# 1 코딩 스타일

---

## 2. 클래스의 멤버 변수

1번의 변수 규칙 + 마지막에 밑줄 하나더\_

Ex) `math_score_`, `student_number_`

---

# 1 코딩 스타일

---

## 3. 함수 네이밍

각 단어의 첫 글자만 대문자로 쓴다(PascalCase)  
동사형으로 쓴다

Ex) AddStudent(), SlidingTackle()

**Let's start!**

---

# 2 연산자

---

연산자???

=> 특정한 작업을 하기 위해 사용하는 기호

# 2 연산자

## 산술 연산자

$= + - * / \%$

## 증가, 감소 연산자

$a++, ++a, b--, --b$

## 비교 연산자

$a == b$  (a와 b가 같음)

$a != b$  (a와 b가 다름)

$a > b$  (a가 b보다 큼)

$a <= b$  (a가 b보다 작거나 같음)



# 2 연산자

## 논리 연산자

!a	(a가 아님)
a && b	(a 그리고 b)
a    b	(a 또는 b)

## 비트 연산자

~a
a & b
a   b
a ^ b
a << b

# 2 스터디 방향

## 연산자 우선순위

순위	종류	연산자
1	괄호, 배열, 구조체	( ) , [ ] -> ::
2	단항 연산자	*(간접) &(주소) ! - ++ -- +(부호) -(부호) sizeof new delete
3	구조체 결합연산자	.* ->*
4	승제 연산자	* / %
5	가감 연산자	+ -
6	시프트(Shift) 연산자	<< >>
7	비교 연산자	< <= > >=
8	등가 연산자	== !=
9	비트 연산자 AND	&
10	비트 연산자 XOR	^
11	비트 연산자 OR	
12	논리 연산자 AND	&&
13	논리 연산자 OR	
14	조건 연산자	?:
15	대입 연산자	= *= /= += -= %= <<= >>= &= ^=  =
16	나열 연산자	,

# 2 연산자

## 결과값 예측(소정의 상품)

1. 

```
int a = 20 - 3 * 7;
```

*cout << a;*
2. 

```
int b = 15 + 9 % 6;
```

*cout << b;*
3. 

```
int c = 35;  
c /= 10;
```

*cout << c;*
4. 

```
int d = 3;  
int e = 10;  
e *= d + 4;
```

*cout << e;*

---

# 3 조건문

---

## 조건문 (if)

```
if ( 조건 ) 조건의 결과가 참이라면  
{  
    (내용) 내용을 실행  
}  
  
거짓이면 무시!
```

---

# 3 조건문

---

## 조건문 (if + else)

```
if ( 조건 ) 조건의 결과가 참이라면
{
    (내용1) 내용1을 실행
}
else 거짓이면
{
    (내용2) 내용2를 실행
}
```

# 3 조건문

---

## 조건문 (if + else if)

```
if ( 조건 1 )    조건1의 결과가 참이라면
{
    (내용1)      내용1을 실행
}
else if(조건2)   거짓이라면 조건 2를 확인하고
{
    (내용2)      조건2가 참이면 내용2를 실행
}
else             조건 1,2가 전부 거짓이면
{
    (내용3)      내용3을 실행
}
```

# 3 조건문

## 조건문 (switch)

```
int num = 3;
switch (num) 조건
{
    case 1:
        cout << "1";
        break;
    case 2:
        cout << "2";
        break;
    case 3:
        cout << "3";
        break;
    default:
        cout << "0";
        break;
}
```

**조건의 결과에 따라 해당하는 case의 위치부터 실행된다**

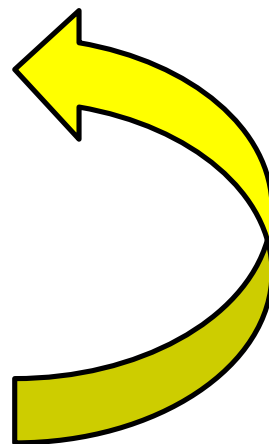
**조건에 해당하는 num이 3이므로 case 3이 실행된다**

**break : 현재 case를 벗어나게 함 (까먹지 말 것)**

# 4 반복문

## for 반복문

**변수 설정**    **조건**    **증감**  
for ( int i=0; i < 10; i++)  
{ **조건이 성립되면 내용을 실행**  
    **( 내용 )**  
}



**변수를 증감시키고  
다시 조건을 검사**



# 4 반복문

## for 반복문

```
int main()
{
    for (int i = 0; i < 5; i++)
    {
        cout << i << "\n";
    }
}
```

C:\WINDOWS\system32\cmd.exe

```
0
1
2
3
4
계속하려면 아무 키나 누르십시오 . . .
```

# 4 반복문

## 다중 for문

```
for ( int i = 0; i < 10; i++)  
{  
    for ( int j = 0; j < 10; j++)  
    {  
        (내용)  
    }  
}
```

내용은 총  $10 * 10 = 100$ 번 실행됨

# 4 반복문

## 다중 for문

```
int main()
{
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            cout << i;
        }
        cout << "\n";
    }
    return 0;
}
```

C:\WINDOWS\system32\cmd.exe

```
00000
11111
22222
33333
44444
계속하려면 아무 키나 누르십시오 . . .
```

# 4 반복문

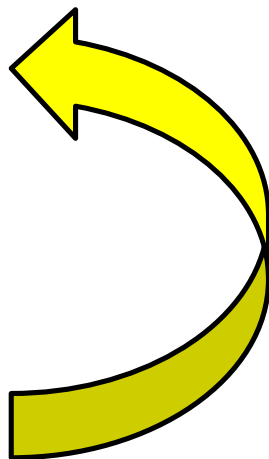
## while 반복문

```
while (조건)
```

```
{
```

```
( 내용 )
```

```
}
```



조건이 성립되면 내용을 실행  
내용을 전부 실행하고 나면  
다시 위로 돌아가서 조건을 검사  
조건이 참인 동안 계속해서 반복

# 4 반복문

## while 반복문

```
int main()
{
    int num = 0;
    while (num < 5)
    {
        cout << num << "\n";
        num++;
    }
}
```

C:\WINDOWS\system32\cmd.exe

```
0
1
2
3
4
계속하려면 아무 키나 누르십시오 . . .
```

# 4 반복문

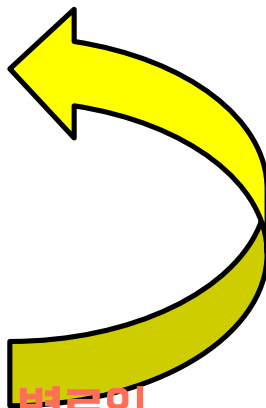
## do-while 반복문 (별로임.. 되도록이면 쓰지 말 것)

```
do  
{
```

( 내용 )

```
} while ( 조건 );
```

세미콜론도 붙여야됨.. 별로임..



내용을 무조건 한 번 실행한 다음  
조건을 검사  
그 이후는 while과 동일

# 4 반복문

## do-while 반복문

```
int main()
{
    int num = 0;
    do
    {
        cout << num << "\n";
        num++;
    } while (num < 5);
}
```

C:\WINDOWS\system32\cmd.exe

```
0
1
2
3
4
계속하려면 아무 키나 누르십시오 . . .
```

---

# 5 과제

---

**BAEKJOON ONLINE JUDGE**

<https://acmicpc.net>





**감사합니다.**

Made by 규정

---

---