

# Byte Align

## ● Visual C++ : 메모리 배열을 4byte로 정렬

▶ 메모리의 이점을 확보하기 위해

▶ 메모리의 복사, 전송, CPU에서의 처리 기본 단위가 4byte이기 때문

▶ 일종의 최적화로서 4byte로 정렬

# Byte Align

## 구조체

```
typedef struct person{  
    short age;  
    int height;  
    long weight;  
    char sex;  
} man;
```

```
man m;  
m.age = 20;  
m.height = 180;  
m.weight = 70;  
m.grade = 'A';
```

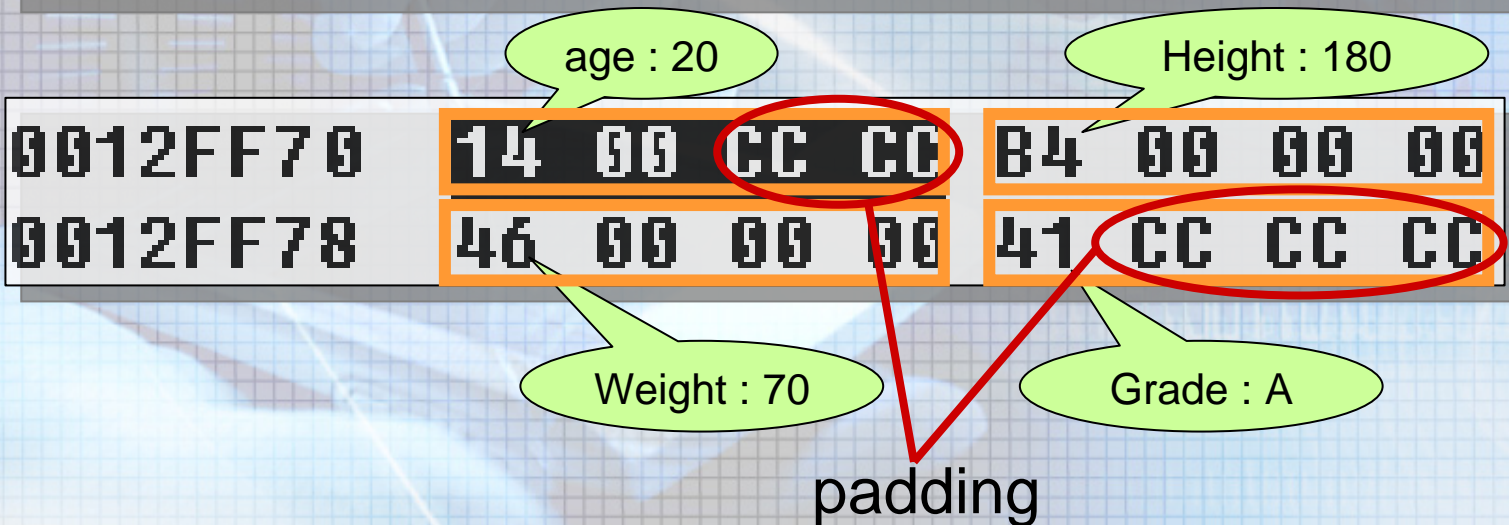


- 구조체 man의 크기 = 16 (11이 아님)

# Byte Align

## 실행결과

```
C:\연수\NETWORK\1일차\ByteAlign\Debug\ByteAlign.exe
sizeof(m) = 16, Addr = 0012ff70
sizeof(short -> m.age) = 2, Addr = 0012ff70
sizeof(int -> m.height) = 4, Addr = 0012ff74
sizeof(long -> m.weight) = 4, Addr = 0012ff78
sizeof(char -> m.grade) = 1, Addr = 0012ff7c
```



# Byte Align

## ● #pragma pack(1) 옵션

➤ 이 옵션을 추가하게 되면 padding을 안함

```
#include <stdio.h>
#pragma pack(1)

typedef struct person{
    short age;
    int height;
```

C:\W면수\NETWORK\1일차\ByteAlign\Debug\ByteAlign.

```
sizeof(m) = 11, Addr = 0012ff74
sizeof(short -> m.age) = 2, Addr = 0012ff74
sizeof(int -> m.height) = 4, Addr = 0012ff76
sizeof(long -> m.weight) = 4, Addr = 0012ff7a
sizeof(char -> m.grade) = 1, Addr = 0012ff7e
```

0012FF74	14 00	B4	00 00 00 46	00
0012FF7C	00 00	41	CC C0 FF 12	00