

# 3주차 Cpp

13조 소재현, 전대원

C와 C++의 입출력 문법 차이

Printf, scanf -> std::cout 로 대체한다

cout 과 << 시프트연산자를 통해 c에서 문법을 한층 더 간략하게 한다

std::cout    std는 이름공간 , cout 는 이름이다

서로 구분하기 위해 ::를 사용한다

○ 예제 2-5    Pair Programing

<소재현> 코딩

<전대원> 노트 정리

처음 실행과정에서 문제점

<소재현> : { char password; cout <-- 이 코드에서 char이 문자 하나로만 처리되어서

문자열로서 저장되는 char [] 방식으로

바꾸어 password를 모두 담을 수 있게 했다

<소재현> if() 다음에 문자열끼리 서로 비교하는 명령어를 물어봤음 이후 strcmp 명령어를 사용했음

<전대원> while 반복문에 대해 else 다음에 break를 거는 것이 맞다고 생각하고 실행하였으나 암호가 틀렸을 때 반복문의 종지를 보고 Else 다음에 break를 삭제하였다

## Pair Programing 문제 8.

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
cout << "5 명의 이름을 ';'으로 구분하여 입력하세요";
```

```
char a[50];  
char b[50];  
char c[50];  
char d[50];  
char e[50];
```

```
cin.getline(a, 50, ';');  
cin.getline(b, 50, ';');  
cin.getline(c, 50, ';');  
cin.getline(d, 50, ';');  
cin.getline(e, 50, ';');
```

```
int aa = strlen(a); //문자열 길이 비교를 위한 함수 사용  
int bb = strlen(b);  
int cc = strlen(c);  
int dd = strlen(d);  
int ee = strlen(e);
```

```
char* longname = a;  
int max = aa;
```

```
if (bb > max) {  
    longname = b;  
    max = bb;  
}  
if (cc > max) {  
    longname = c;  
    max = cc;  
}
```

```

    if (dd > max) {
        longname = d;
        max = dd;
    }
    if (ee > max) {
        longname = e;
        max = ee;
    }

    cout << "1 : " << a << endl; // c++에서 다음줄로
    바꾸어 주는 함수 endl 사용
    cout << "2 : " << b << endl;
    cout << "3 : " << c << endl;
    cout << "4 : " << d << endl;
    cout << "5 : " << e << endl;

    cout << "가장 긴 이름은 " << longname;

}

```

---

<소재현> : 위 문제의 접근을 위해  
 먼저 ; 로 이름을 구분해야 하므로 수업내용에서 배운  
 cin.getline() 함수를 이용해서 입력문자를 구분.

<전대원> : 이후 가장 긴 이름을 찾아야 함으로 각각의  
 문자열에 길이 판별하여 주는 함수 strlen() 함수를 통해  
 문자열별 길이를 알아냄

<소재현> : 이후 가장 긴 이름을 표현할 함수 longname  
 과 가장 길이의 크기가 큰 함수를 max 로 표현

<전대원, 소재현> : 위 변수를 가지고 조건문을 통해  
각각을 비교 후 가장 긴 이름 도출

<전대원, 소재현> : c++에서 다음줄로 바꾸어 주는 함수  
endl 사용으로 코드종료