

2020-09-07

201902694

박재우

Week3

깃 주소

<https://github.com/jwoo9928/-Object-oriented-design/tree/master/week4>

## 과제 1

### 1. 코드 및 실행화면

```
C homework01.c > main()
1  #include <stdio.h>
2
3  int main() {
4
5      char s[100];
6      printf("문자를 입력하세요(100자 이내).");
7      scanf("%s",s);
8      printf("입력된 문자는 %s 입니다.",s);
9
10
11     return 0;
12 }
13
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```
→ week4 git:(master) x gcc homework01.c
→ week4 git:(master) x ./a.out
문자를 입력하세요(100자 이내).hello
입력된 문자는 hello 입니다.
→ week4 git:(master) x
```

### 2. 결과 분석

C 언어는 string 타입이 존재하지 않아 char 배열을 선언해야 했다  
또한 cout은 printf로 변경했으며 cin은 scanf로 변경했다.  
전체적인 매커니즘은 같아 차이점은 거의 없었다.

### 3. 새로 알게 된 점

딱히 새로 알게 된 점은 존재하지 않았다.

## 과제 2

### 1. 코드 및 실행화면

```
homework02.cpp > swap(int *, int *)
1  #include<iostream>
2
3  void swap(int *first, int *second){
4      int temp = *first;
5      *first = *second;
6      *second = temp;
7  }
8  int main(){
9      int a = 2, b = 3;
10     swap(&a, &b);
11     std::cout << a << " " << b << std::endl;
12     return 0;
13 }
14
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```
→ week4 git:(master) x g++ homework02.cpp
→ week4 git:(master) x ./a.out
3 2
→ week4 git:(master) x
```

### 2. 결과 분석

기존에 작동하지 않았던 이유는

swap함수가 a, b의 주소에 접근하여 값을 변경하는 것이 아닌  
값만 가져와 함수내부에서 작동시켜 a, b의 값은 변하지 않는 데에  
있었다. 따라서 a, b의 주소값을 받아와서 역참조를 통해 직접 값을  
변경했으며 결과에도 잘 작동되었다.

### 3. 새로 알게 된 점

C++에서도 c와 같이 참조와 역참조를 같은 방식으로 사용한다는 것  
을 알게 되었다.