

2020-09-07

201902694 박재우

Week3

Git 주소

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

-과제 1

코드 및 실행결과

```
week3 > G+ training_01.cpp > ...
1  #include <iostream>
2  class SetValue {
3  private:
4      int x, y;
5  public:
6      void x_value(int x) {
7          this->x = x;
8      }
9      void y_value(int y) {
10         this->y = y;
11     }
12     int get_x() {
13         return x;
14     }
15     int get_y() {
16         return y;
17     }
18 };
19
20 int main() {
21     SetValue obj;
22     obj.x_value(33);
23     obj.y_value(44);
24
25     std::cout << "X = " << obj.get_x() << " ,Y = " << obj.get_y() << std::endl;
26     return 0;
27 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: zsh + □

```
→ week3 git:(master) x ls
homework_02.cpp training_01.cpp
→ week3 git:(master) x g++ training_01.cpp
→ week3 git:(master) x ./a.out
X = 33 ,Y = 44
→ week3 git:(master) x
```

결과 분석 :

SetValue class의 원소들을 private로 변경하여 추상화를 시켰다.

또한 원소의 접근을 위한 함수, 값을 저장하기위한 value 함수와 get함수를 통해서 캡슐화를 진행했다.

-과제 2

코드 및 실행결과

실행결과 분석(.c와 .pp 두가지 경우의 컴파일):

```
#include <stdio.h>

void print(int var){
    printf("Integer number: %d \n", var);
}

void print(float var){
    printf("Float number: %f \n", var);
}

void print(int var1, float var2){
    printf("Integer number: %d \n", var1);
    printf(" and float number: %f", var2);
}

int main(){
    int a = 7;
    float b = 9;

    print(a);
    print(b);
    print(a, b);

    return 0;
}
```

-.c 컴파일 결과

```
→ week3 git:(master) x ls
a.out      homework_02.c  homework_02.cpp training_01.cpp
→ week3 git:(master) x g++ homework_02.cpp
→ week3 git:(master) x ./a.out
Integer number: 7
Float number: 9.000000
Integer number: 7
and float number: 9.000000
→ week3 git:(master) x
```

-.cpp 컴파일 결과

```
→ week3 git:(master) x ls
homework_02.c  homework_02.cpp training_01.cpp
→ week3 git:(master) x g++ homework_02.c
clang: warning: treating 'c' input as 'c++' when in C++ mode, this behavior is deprecated [-Wdeprecated]
→ week3 git:(master) x ./a.out
Integer number: 7
Float number: 9.000000
Integer number: 7
and float number: 9.000000
→ week3 git:(master) x
```

결과 분석 :

.c파일을 g++로 실행했을 경우 에러는 아니지만 warning이 떴다.

이를 에러가 나타나지 않았다고 본다면 그 이유는

C++의 초기 버전은 C를 기반으로 빌드되었으며 실제로 컴파일러는 C++ 코드를 C로 변환한 다음 로컬 C 컴파일러에 의해 컴파일되었기 때문이다.