

객체지향설계 #Week03

제출일 : 20.09.17

학번 이름 : 201902721 유찬희

GitHub 주소 : <https://github.com/HanCiHu/OOP>

1.

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

- private를 이용하여 변수 x, y를 추상화 하였고 x, y에 직접 접근이 불가능하므로 get함수와 set함수를 이용해서 캡슐화 하였습니다.

2.

```
1      #include <stdio.h>
2
3      void print(int var){
4          printf("Integer number: %d \n", var);
5      }
6
7
8      void print(float var){
9          printf("Float number: %f \n", var);
10     }
11
12
13     void print(int var1, float var2){
14         printf("Integer number: %d \n", var1);
15         printf(" and float number: %f", var2);
16     }
17
18     int main(){
19         int a = 7;
20         float b = 9;
21
22         print(a);
23         print(b);
24         print(a, b);
25
26         return 0;
27     }
```

- 위의 코드를 g++을 이용하여 컴파일 하면 오류없이 컴파일이 되는 반면 gcc를 이용하여 컴파일을 하면 오류를 발생합니다.
- C언어는 함수의 이름을 가지고 함수를 구분하고, C++은 함수의 이름과 매개변수의 개수를 가지고 함수들을 구분하기 때문에 g++을 이용하면 오류가 발생하지 않고 gcc를 이용하면 오류가 발생하는 것 같습니다.
- 그러나 확장자가 .c일 때 g++로 컴파일을 하면 컴파일이 정상적으로 되는데 이것은 C++이 C언어를 객체지향적으로 확장시킨 것이기 때문에 컴파일러가 C언어의 컴파일을 지원해서 오류가 발생하지 않는 것 같습니다.