1. 자동차 경주 예제

```
#include <iostream>
#include <string>
using namespace std;

...

int main(){
        Car car1, car2;

        car1.init(rand() % 200, 1, "red");
        car1.show();

        car2.init(rand() % 200, 1, "yellow");
        car2.show();

        if( car1.getSpeed() > car2.getSpeed() )  // private 변수 이므로
                cout << "car1이 승리하였습니다" << endl;
        else
                cout << "car2가 승리하였습니다" << endl;

        return 0;
}
```

```
==============
속도: 41
기어: 1
색상: red
==============
==============
속도: 67
기어: 1
색상: blue
==============
car2가 승리하였습니다
```

1. 자동차 경주 예제 답

멤버함수를 class 내부에 작성

```
class Car {
private:
   int speed;          //속도
   int gear;           //기어
   string color;       //색상

public:
   void init(int s, int gear, string c)
   {
          speed = s;
          gear = g;
          color = c;
   }
```

```
   void show() {
      cout << "============== " << endl;
      cout << "속도: " << speed << endl;
      cout << "기어: " << gear << endl;
      cout << "색상: " << color << endl;
      cout << "============== " << endl;
   }

   int getSpeed()  {
      return speed;
   }

};
```

1. 자동차 경주 예제 답

멤버함수를 class 외부에 작성

```
class Car {
private:
   int speed;          //속도
   int gear;           //기어
   string color;       //색상
public:
   void init(int s, int gear, string c);
   void show();
   int getSpeed();
};

int Car::getSpeed()
{
            return speed;
}
```

```
void Car::init(int s, int g, string c)
{
     speed = s;
     gear = g;
     color = c;
}
void Car::show() {
     cout << "========== " << endl;
     cout << "속도: " << speed << endl;
     cout << "기어: " << gear << endl;
     cout << "색상: " << color << endl;
     cout << "========== " << endl;
}
```

2. 램프 예제

- 집에서 사용하는 스탠드에 관한 프로그램 → DeskLamp class
- 멤버변수

```
bool isOn;    // private , true or false
```

- 멤버함수 → public

```
void turnOn();      // 램프를켠다. → isOn 을 true로 설정
void turnOff();     // 램프를끈다. → isOn 을 false로 설정
void print();       // 현재상태를 출력 → isOn 값을 보고
                    // 출력문을 결정
```

```
int main() {
          // 객체생성
          DeskLamp lamp;

          lamp.turnOn();
          lamp.print();
          lamp.turnOff();
          lamp.print();
          return 0;
}
```

```
램프가 켜짐
램프가 꺼짐
```

참고

```
x = 1

y = (x >= 0 ? 100 : -100);       // ( 조건문 ? 값_1 : 값_2)

cout << y << endl
```

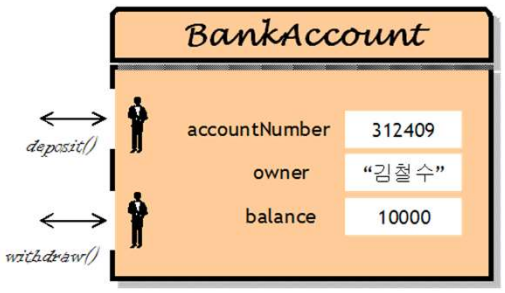
2. 램프 예제 답

```
#include <iostream>
#include <string>
using namespace std;

class DeskLamp {
private:
        bool isOn;          // 켜짐이나꺼짐과같은램프의상태
public:
        void turnOn();      // 램프를켠다.
        void turnOff(); // 램프를끈다.
        void print();       // 현재상태를출력
};

void DeskLamp::turnOn(){    isOn = true;        }
void DeskLamp::turnOff(){   isOn = false;       }
void DeskLamp::print(){
        cout << "램프가" << (isOn == true ? "켜짐" : "꺼짐") << endl;
}
```

3. 은행 계좌 예제



3. 은행 계좌 예제

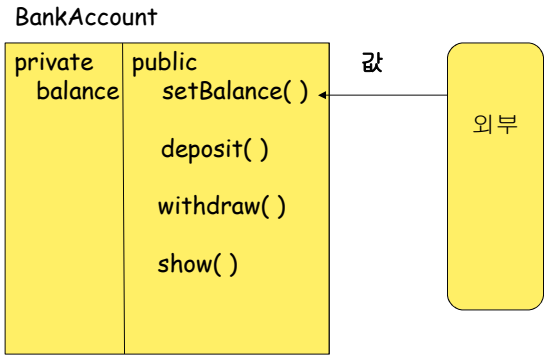
- 은행 계좌 → BankAccount class
- 멤버변수 : private
  int balance; // 잔고 → 책에는 예금주, 계좌번호 있으나 사용 안함.

- 멤버함수 : public

```
        void setBalance(int amount);     // balance에 대한 설정자
        void deposit(int amount);        // 입금 함수 → 잔고 변화
        void withdraw(int amount);       // 출금 함수 → 잔고 변화
        void print();                    // 현재 잔고 출력

int main() {
        BankAccount account;
        account.setBalance(0);           // 잔고 0원으로 초기화
        account.deposit(100);  // 100 원 저금
        account.print();
        account.withdraw(80); // 80 원 인출
        account.print();
        return 0;
}
```

잔액은 100입니다.
잔액은 20입니다.



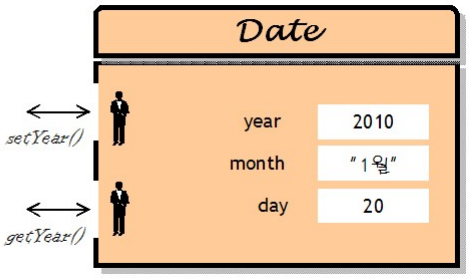
3. 은행 계좌 예제 답

```
class BankAccount {          // 은행계좌
private:
    int balance;           // 잔액을 표시하는 변수
public:
    void setBalance(int amount);   // balance에 대한 설정자
    void deposit(int amount);   // 저금함수
    void withdraw(int amount);// 인출함수
    void print();              // 현재상태출력
};

void BankAccount::setBalance(int amount) {    balance = amount;  }
void BankAccount::deposit(int amount) {    balance += amount; }
void BankAccount::withdraw(int amount){ balance -= amount; }
void BankAccount::print(){
    cout << "잔액은" << balance << "입니다." << endl;
}
```

4. 날짜 예제

- 날짜



4. 날짜 예제

```
#include <iostream>
#include <string>
using namespace std;

class Date {
private:
        int year;
        int month;
        int day;
public:
        int getYear();
        void setYear(int y);
        int getMonth();
        void setMonth(int m);
        void setDay(int d);
        int getDay();
        void print();
};
int Date::getYear()
{
        return year;
}
```

4. 날짜 예제

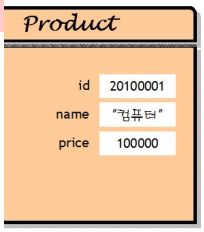
```
void Date::setYear(int y){    year = y;   }
int Date::getMonth(){         return month;  }

void Date::setMonth(int m){  month = m;    }
int Date::getDay(){           return day;     }

void Date::setDay(int d){     day = d;     }
void Date::print(){
        cout << year << "년" << month <<  "월" << day << "일" << endl;
}
int main()
{
        Date date;
        date.setYear(2010);
        date.setMonth(1);
        date.setDay(20);
        date.print();
        return 0;
}
```

2010년 1월 20일

5. 상품 예제



- 상품(Product class)
- 멤버변수 : private
  int id; // 상품 번호
  string name; int price;
- 멤버함수 : public
  void input(); // 키보드에서 3개 입력 받아 멤버변수에 저장
  void print(); // 3 개 멤버변수값 출력
  bool isCheaper(Product other); // 자신이 타상품(인자) 보다 싼지 아닌지
- main

```
int main() {
        Product p1, p2;
        p1.input();  // 사용자가 입력
        p2.input();
        if( p1.isCheaper(p2) ){
                p1.print();
                cout << "이더쌉니다\n";
        }
        else {
                p2.print();
                cout << "이더쌉니다\n";
        }
        return 0;
}
```

```
상품의 일련 번호: 1
상품의 이름: 모니터
상품의 가격: 100000

상품의 일련 번호: 2
상품의 이름: 컴퓨터
상품의 가격: 20000000

상품 번호 1
상품의 이름: 모니터
상품의 가격: 100000
이 더 쌉니다
```

5. 상품 예제 답

```
#include <iostream>
#include <string>
using namespace std;

class Product {
private:
        int id;
        string name;
        int price;
public:
        void input();
        void print();
        bool isCheaper(Product other);
};
```

```
void Product::print()
{
  cout << " 상품번호" << id << endl
     << " 상품의이름: " << name
     << " 상품의가격: " << price << endl;
}

void Product::input()
{
        cout << "상품의일련번호: ";
        cin >> id;
        cout << "상품의이름: ";
        cin >> name;
        cout << "상품의가격: ";
        cin >> price;
}
```

5. 상품 예제 답

```
bool Product::isCheaper(Product other)
{
        if( price < other.price )
                return true;
        else
                return false;
}
```

```
p1.isCheaper(p2)
```

```
p2.isCheaper(p1)
```

```
                p1
price = 100,000
isCheaper(Product other){  // other = p2
    if( price < other.price )
        return true;
    else
        return false;
}
```

```
                p2
price = 2,000,000
isCheaper(Product other){  // other = p1
    if( price < other.price )
        return true;
    else
        return false;
}
```

Private 멤버 접근

p1.isCheaper(p2)

p1
```
price = 100,000
isCheaper(Product other){  // other = p2
    if( price < other.price )
        return true;
    else
        return false;
}
```

p2
```
price = 2,000,000
isCheaper(Product other){
    if( price < other.price )
        return true;
    else
        return false;
}
```

- private 는 외부에서 접근 못함(내부에서만 접근) → p2.price 를 p1 함수에서 접근 ?
- 내부 → 클래스 내부 → 같은 클래스의 객체들은 서로 private 접근 가능
  - Product p1, p2; → 서로 private 접근 가능
  - Car p3; → p3는 p1, p2 의 private 접근 못함

Report

- 201 쪽 연습문제 3번, 6번
- 제출 파일 이름 : 수업활동 일지에 제출

```
void main(){
    p3( );
    p6( );
}
```

연습문제 3 주사위

```
주사위의 숫자는 4입니다.
주사위의 숫자는 4입니다.
주사위의 숫자는 5입니다.
주사위의 숫자는 3입니다.
주사위의 숫자는 6입니다.
주사위의 숫자는 3입니다.
주사위의 숫자는 3입니다.
주사위의 숫자는 2입니다.
주사위의 숫자는 6입니다.
주사위의 숫자는 1입니다.
Press any key to continue
```

```
int main()
{
        Dice D1;
        srand( time(NULL) );
        for(int i = 0; i < 10;  i++){
                D1.roll();              // 주사위를 던지고 숫자를 출력
                cout<<"주사위의 숫자는 "<<D1.getFace()<<"입니다."<<endl;
        }
        return 0;
}
```

- 멤버변수
  - int face; // 임의의 1~6 값이 주사위 값으로 저장
- 멤버함수
  - roll() → 주사위 던지는 행위 → 임의의 1~6 값이 주사위 값으로 저장(책 참조), rand(...)
  - getFace( ) → 현재 주사위 값을 출력
- srand(time(NULL));
  - rand( ) 는 실행시 마다 같은 값 발생하기에
  - srand(time(NULL)) 실행하면 rand() 함수의 seed 가 변경되어 rand() 함수 수행시 다른 값 발생
  - **time(...)** 사용하려면 **"#include <ctime>"** 해주어야 함.

연습문제 6 복소수

```
int main(){
        double r1, i1, r2, i2;                  Complex c1, c2, c3;
        cout<<"1번째 실수부와 허수부를 입력하세요 :";      cin>>r1>>i1;
        cout<<"2번째 실수부와 허수부를 입력하세요 :";      cin>>r2>>i2;

        cout<<"====================================="<<endl;
        c1.setComplex(r1, i1);          cout<<"복소수는 : ";          c1.Print();
        c2.setComplex(r2, i2);          cout<<"복소수는 : ";          c2.Print();

        c3.Add(r1, r2, i1, i2);         cout<<"합은 : ";              c3.Print();
        c3.Sub(r1, r2, i1, i2);         cout<<"뺄셈은 : ";            c3.Print();
        cout<<"====================================="<<endl;

        return 0;
}
```

```
1번째 실수부와 허수부를 입력하세요 :8 13.9
2번째 실수부와 허수부를 입력하세요 :4 4.0
=================================
복소수는 : 8 + 13.9i
복소수는 : 4 + 4i
합은 : 12 + 17.9i
뺄셈은 : 4 + 9.9i
=================================
Press any key to continue
```

연습문제 8 : 은행 계좌 예제

- 앞의 은행 계좌 예제 4 프로그램에 이체 기능 추가(이체만 생각하자)

- 은행 계좌 → BankAccount class
- 멤버변수 (private) → int balance; // 잔고

- 멤버함수(public)
  - void print(); // 현재 상태(잔고) 출력
  - void setBalance(int amount); // balance에 대한 설정자(값 저장 멤버함수)

  이체 함수는 ??

  void transfer(이체 대상, 이체금액);
  // 이체 대상→ 통장
  // 이체 하면 → 보내는 사람, 받는 사람 모두 통장 잔고 변화해야 함

  void transfer(BankAccount a, int amount);

연습문제 8 : 은행 계좌 예제

```
class BankAccount {
private:
    int balance;

public:
    void transfer(BankAccount otherAccount, int amount);
    void setBalance(int amount);
    void print();
};

void BankAccount::setBalance(int amount){    balance = amount;    }

void BankAccount::transfer(BankAccount otherAccount, int amount){
    balance -= amount;
    otherAccount.balance += amount;
}

void BankAccount::print(){ cout << "잔액은 " << balance << "입니다." << endl;  }
```

연습문제 8 : 은행 계좌 예제

```
int main() {
    BankAccount account1, account2;
    account1.setBalance(1000);
    account2.setBalance(100);

    cout << "계좌 1의 잔액은>> ";        account1.print();    // 1000
    cout << "계좌 2의 잔액은>> ";        account2.print();    // 100

    cout << "→ account 1에서 100원을 account 2로 이체하기" << endl;
    account1.transfer(account2, 100);

    cout << "계좌 1의 잔액은>> ";        account1.print();    // 900
    cout << "계좌 2의 잔액은>> ";        account2.print();    // 100 ??

    return 0;
}
```

문제점 이유 ? → 3 장에서 설명한 내용, call by value
매개변수가 일반 변수일 때와 객체일 때 동일한 원리로 수행

예제

```
void dec_by_v(int v) {                  // int v = time
    v--;
    return;
}
void dec_by_r(int& r) {                 // int &r = time;
    r--;
    return;
}

void dec_by_p(int* p) {                 // int *p = &time
    --(*p);
    return;
}

int main() {
    int time = 10;
    dec_by_v(time);     cout << time;        10
    dec_by_r(time);     cout << time;        9
    dec_by_p(&time);    cout << time;        8

    return 0;
}
```

	주소	값
time r → x	1000	10 → 9 → 8
	1001	
v → x	1002	10 → 9 → x
	1003	
p → x	1004	1000 → x
	1005	
	1006	

연습문제 8 : 은행 계좌 예제(오동작 이유)

```
main(){...
    cout << "-> 계좌 1에서 100원을 계좌 2로 이체하기" << endl;
    account1.transfer(account2, 100);
...
}

void BankAccount::transfer(BankAccount otherAccount, int amount){
    balance -= amount;
    otherAccount.balance += amount;
}
```

BankAccount otherAccount = account2;

account2	account1	otherAccount
balance = 100	balance = 1000 → 900	balance = 100→200
transfer(...){ ... }	transfer(...){ ... }	transfer(...){ ... }

지역객체소멸

연습문제 8 : 은행 계좌 예제(수정 내용)

```
main(){...
    cout << "-> 계좌 1에서 100원을 계좌 2로 이체하기" << endl;
    account1.transfer(account2, 100);
...
}

void BankAccount::transfer(BankAccount &otherAccount, int amount){
    balance -= amount;
    otherAccount.balance += amount;
}
```

BankAccount &otherAccount = account2;

account1	account2
balance = 1000 → 900	balance = 100 → 200
transfer(...){ ... }	transfer(...){ ... }

연습문제 8 : 은행 계좌 예제(완성)

```
class BankAccount {
private:
    int balance;

public:
    void transfer(BankAccount &otherAccount, int amount);
    void setBalance(int amount);
    void print();
};

void BankAccount::setBalance(int amount){balance = amount;     }

void BankAccount::transfer(BankAccount &otherAccount, int amount){
    balance -= amount;
    otherAccount.balance += amount;
}

void BankAccount::print(){  cout << "잔액은 " << balance << "입니다." << endl; }
```

일반적으로 함수가 매개변수로 객체를 받을 때는 참조자로 받음

연습문제 8 은행 계좌 예제 전체 코드

```
class BankAccount {// 은행계좌
private:
   int balance;// 잔액을 표시하는 변수

public:
   void setBalance(int amount);
   void deposit(int amount) ;
   void withdraw(int amount) ;
   void transfer(BankAccount &a, int amount);
   void print();            // 현재 잔액 출력
};

void BankAccount::print(){
   cout << "잔액은 " << balance << "입니다."
        << endl;
}
void BankAccount::setBalance(int amount) {
   balance = amount;
}
```

```
void BankAccount::deposit(int amount) {
   balance += amount;
}
void BankAccount::withdraw(int amount){
   balance -= amount;
}

void BankAccount::transfer(BankAccount &x, int amount){
   balance  = balance - amount;
   x.balance = x.balance + amount
}
/*  멤버함수 이용 구현
void BankAccount::transfer(BankAccount &x, int amount){
   withdraw(amount);
   x.deposit(amount);
}
*/
// &x 대신 x 사용하면 ?? → No
```