

객체지향프로그래밍 LAB #10

<기초문제> _____

1.

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

class Point {
private: // class 안에서만 사용가능
    int x; // 멤버 변수
    int y;
public: // class 안/밖에서 사용가능
    // Point() {}
    Point(int _x, int _y) { //constructor: class와 이름이 같다,
        x = _x;
        y = _y;
    }

    /*X,Y set함수 구현(함수 1개)*/
    void setXY(int a, int b) {
        x = a;
        y = b;
    }

    /*X,Y get 함수 각각 구현(함수 2개)*/
    int getX() {
        return x;
    }
    int getY() {
        return y;
    }
    void print() {
        cout << x << ", " << y << endl;
    }
};

int main() {
    Point pt1(1, 2), pt2(3, 4);
    pt1.setXY(10, 20);
    pt1.print();
}
```

```

    pt2.print();
    cout << pt1.getX() << endl;
    cout << pt2.getY() << endl;
    return 0;
}

```



2.

```

#include <string>
#include <iostream>
using namespace std;
class Account {
    //private: //멤버변수
    string name;
    string id;
    double balance;
public: // 멤버함수(method)
    // 3개의 생성자 구현
    Account() : name(""), id(""), balance(0) { }
    Account(string a, string b) : name(a), id(b), balance(0) { }
    Account(string a, string b, double c) : name(a), id(b), balance(c) {
        if (balance < 0)
            balance = 0;
    } // name, id , balance 받아와서 초기화, balance < 0 인 경우 0으로 초기화
    void deposit(double _amt) { balance += _amt; }
    bool withdraw(double _amt) {
        if (balance - _amt < 0)
            return false;
        balance -= _amt;
        return true;
    }
    void print() {

```

```

        cout << name << ", " << id << ", " << balance << endl;
    }
};

int main() {
    Account ac1("배성호", "1002", 5000);
    Account ac2;
    ac2.print();
    cout << endl;
    ac1.print();
    cout << endl;
    int depo;
    cout << "예금할 금액을 입력하세요 : ";
    cin >> depo;
    ac1.deposit(depo);
    ac1.print(); // print() 함수로 ac1 출력
    cout << endl;
    int wdrw;
    cout << "출금할 금액을 입력하세요 : ";
    cin >> wdrw;
    if (!ac1.withdraw(wdrw))
        cout << "잔액이 부족합니다." << endl;
    ac1.print();
    return 0;
}

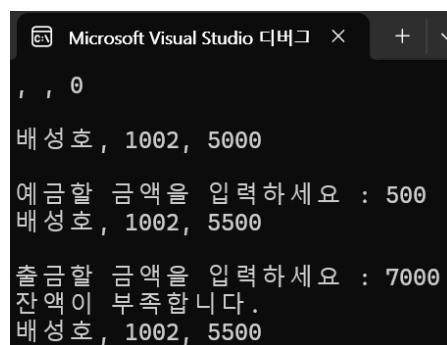
```



```

, , 0
배성호 , 1002 , 5000
예금할 금액을 입력하세요 : 20000
배성호 , 1002 , 25000
출금할 금액을 입력하세요 : 500
배성호 , 1002 , 24500

```



```

, , 0
배성호 , 1002 , 5000
예금할 금액을 입력하세요 : 500
배성호 , 1002 , 5500
출금할 금액을 입력하세요 : 7000
잔액이 부족합니다.
배성호 , 1002 , 5500

```

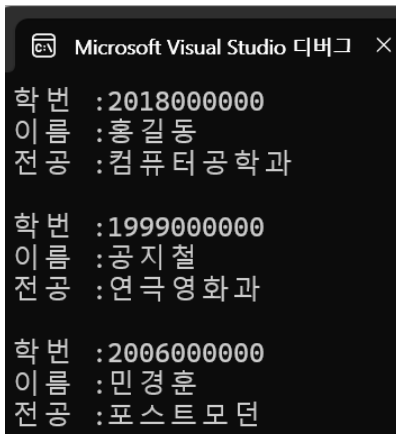
<응용문제>

1.

```
#include <iostream>
#include <string>
using namespace std;
class CStudent {
private:
    int sn;
    string name;
    string maj;
public:
    CStudent() : sn(2018000000), name("홍길동"), maj("컴퓨터공학과") {}
    CStudent(int a, string b, string c) : sn(a), name(b), maj(c) {}
    void setNumber(int a) {
        sn = a;
    }
    void setName(string b) {
        name = b;
    }
    void setMajor(string c){
        maj = c;
    }
    void Display() {
        cout << "학번 :" << getNumber() << endl;
        cout << "이름 :" << getName() << endl;
        cout << "전공 :" << getMajor() << endl;
        cout << endl;
    }
    int getNumber() {
        return sn;
    }
    string getName() {
        return name;
    }
    string getMajor() {
        return maj;
    }
}
```

```
};
```

```
int main() {  
    CStudent s1; // A  
    s1.Display();  
    CStudent s2(1999000000, "공지철", "연극영화과"); // B  
    s2.Display();  
    // C  
    s1.setNumber(2006000000);  
    s1.setName("민경훈");  
    s1.setMajor("포스트모던");  
    cout << "학번 :" << s1.getNumber() << endl;  
    cout << "이름 :" << s1.getName() << endl;  
    cout << "전공 :" << s1.getMajor() << endl;  
    return 0;  
}
```



```
Microsoft Visual Studio 디버그 ×  
학 번 : 2018000000  
이름 : 홍길동  
전공 : 컴퓨터공학과  
  
학 번 : 1999000000  
이름 : 공지철  
전공 : 연극영화과  
  
학 번 : 2006000000  
이름 : 민경훈  
전공 : 포스트모던
```

2.

```
#include <iostream>  
#include <string>  
using namespace std;  
class CStudent {  
private:  
    int sn;
```

```

    string name;
    string maj;
public:
    CStudent() : sn(2018000000), name("홍길동"), maj("컴퓨터공학과") {}
    CStudent(int a, string b, string c) : sn(a), name(b), maj(c) {}
    void setNumber(int a) {
        sn = a;
    }
    void setName(string b) {
        name = b;
    }
    void setMajor(string c){
        maj = c;
    }
    void Display() {
        cout << "학번 :" << getNumber() << endl;
        cout << "이름 :" << getName() << endl;
        cout << "전공 :" << getMajor() << endl;
        cout << endl;
    }
    int getNumber() {
        return sn;
    }
    string getName() {
        return name;
    }
    string getMajor() {
        return maj;
    }
};

int main() {
    CStudent s[3]; // 3명의 학생정보를 저장할 class 배열
    int inputNumber; // 키보드로 학번을 입력 받을 변수
    string inputName, inputMajor; // 키보드로 이름, 전공을 입력 받을 변수
    int length = 0; // 현재 입력된 학생의 수
    while (1) {
        int c = 0;
        if (length > 2)
            break;

```

```

cout << length + 1 << " 번째 학생 입력" << endl;
cout << "학번 : ";
cin >> inputNumber;
cout << "이름 : ";
cin >> inputName;
cout << "전공 : ";
cin >> inputMajor;
for (int i = 0; i < length; i++) {
    if (inputNumber != s[i].getNumber()) {
        continue;
    }
    cout << "※중복된 학번이 존재합니다" << endl<<endl;
    c = -1;
    break;
}
if (c == 0) {
    s[length].setNumber(inputNumber);
    s[length].setName(inputName);
    s[length].setMajor(inputMajor);
    length++;
    cout << "※입력 완료" << endl << endl;
}
}
cout << "----모든입력이 완료되었습니다----" << endl;
for (int i = 1; i <= length; i++) {
    cout << i << "학생정보" << endl;
    s[i - 1].Display();
}
return 0;
}

```

```
Microsoft Visual Studio 디버그 × + v
1 번째 학생 입력
학번 : 2017123456
이름 : 장국밥
전공 : 컴퓨터공학과
※입력 완료

2 번째 학생 입력
학번 : 2018123456
이름 : 공유
전공 : 연극영화과
※입력 완료

3 번째 학생 입력
학번 : 2018123456
이름 : 민경훈
전공 : 포스트모던
※중복된 학번이 존재합니다

3 번째 학생 입력
학번 : 2019123456
이름 : 민경훈
전공 : 포스트모던
※입력 완료

----모든입력이 완료되었습니다----
1학생 정보
학번 :2017123456
이름 :장국밥
전공 :컴퓨터공학과

2학생 정보
학번 :2018123456
이름 :공유
전공 :연극영화과

3학생 정보
학번 :2019123456
이름 :민경훈
전공 :포스트모던
```

3.

```
#include <iostream>
```

```
#include <string>
```

```
#include <vector>
```

```
using namespace std;
```

```
class Student {
```

```
private:
```

```
    int sn;
```

```
    string name;
```

```
    string maj;
```

```
    vector<string> sub;
```

```
    vector<char> gr;
```


public:

```
Student() : sn(0), name("default"), maj("depart") {}
Student(string a, int b, string c) : sn(b), name(a), maj(c) {}
void setID(int a) {
    sn = a;
}
void setName(string b) {
    name = b;
}
void setDpt(string c){
    maj = c;
}
void print() {
    cout << getName() << " " << getNumber() << " " << getMajor() << endl;
}
void addGrade(string a, char b) {
    sub.push_back(a);
    gr.push_back(b);
}
void printGrades() {
    for (int i = 0; i < sub.size(); i++) {
        cout << sub[i] << " " << gr[i] << endl;
    }
    cout << "GPA : " << getGPA();
}
float getGPA() {
    float s = 0;
    if (gr.size() == 0) {
        return 0.0;
    }
    for (int i = 0; i < gr.size(); i++) {
        if (gr[i] == 'A')
            s += 4;
        else if (gr[i] == 'B')
            s += 3;
        else if (gr[i] == 'C')
            s += 2;
        else if (gr[i] == 'D')
            s += 1;
        else if (gr[i] == 'F')
```

```

        s += 0;
    }
    return (s/ float(gr.size()));
}

void getYear(int year) {
    string d = to_string(sn);
    d = d.substr(0, 4);
    int a = year - stoi(d)+1;
    if (a == 1)
        cout << "Freshmen(1학년)" << endl;
    else if (a == 2)
        cout << "Sophomore(2학년)" << endl;
    else if (a == 3)
        cout << "Junior(3학년)" << endl;
    else if (a == 4)
        cout << "Senior(4학년)" << endl;
    else
        cout << "About to graduate(5학년) " << endl;
}

int getNumber() {
    return sn;
}

string getName() {
    return name;
}

string getMajor() {
    return maj;
}

};

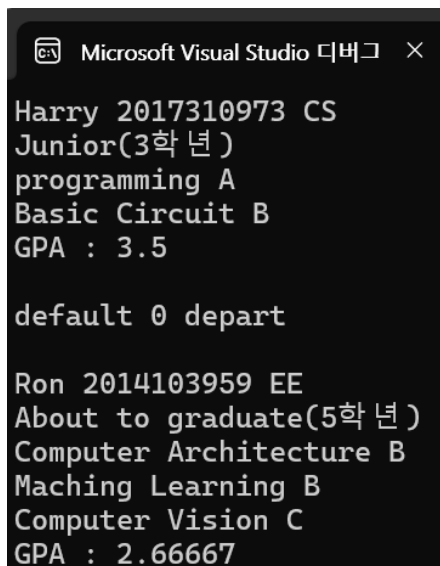
int main() {
    Student Harry("Harry", 2017310973, "CS");
    Harry.print();
    Harry.getYear(2019);
    Harry.addGrade("programming", 'A');
    Harry.addGrade("Basic Circuit", 'B');
    Harry.printGrades();
    cout << "₩₩₩₩";
    Student Ron;

```

```

    Ron.print();
    cout << "Wn";
    Ron.setName("Ron");
    Ron.setID(2014103959);
    Ron.setDpt("EE");
    Ron.print();
    Ron.getYear(2019);
    Ron.addGrade("Computer Architecture", 'B');
    Ron.addGrade("Maching Learning", 'B');
    Ron.addGrade("Computer Vision", 'C');
    Ron.printGrades();
    cout << "WnWn";
    return 0;
}

```



Microsoft Visual Studio 디버그

```

Harry 2017310973 CS
Junior(3학년)
programming A
Basic Circuit B
GPA : 3.5

default 0 depart

Ron 2014103959 EE
About to graduate(5학년)
Computer Architecture B
Maching Learning B
Computer Vision C
GPA : 2.66667

```