**(c++)class 수정**

**id를 직접 입력받도록 코드를 수정했습니다**

#include <iostream>

#include <cstring>

#define SIZE 32

using namespace std;

class Student {

char name[SIZE];

int id;

public:

void Store (const char \*name, int id)

{

strcpy(this->name,name);this->id=id;

}

void Print(){cout<<"studentL name = ";

cout<<name<<"; id = "<<id <<endl;}

};

int main ()

{

Student s;

int a;

scanf("%d",&a);

s.Store("Kim",a);s.Print();

return 0;

}

**(c++)constructor 수정**

**a의 경우만 냅두고 b,c를 지웠습니다.**

#include <iostream>

#include <cstring>

using namespace std;

class Student {

char \*name;

int id;

public:

Student(const char \*n, int i);

Student(const Student&);

~Student();

void Print();

};

int main ()

{

Student \*a = new Student("Kim",123);

a->Print();

delete a;

return 0;

}

Student::Student(const char \*n,int i)

{

name = new char[strlen(n)+1];

strcpy(name,n); id=i;

}

Student::~Student()

{

if(name!=NULL) delete[] name;

name = NULL;

}

void Student::Print()

{

cout<<"("<<name<<" "<<id<<")";

}

**(c++)overloading수정**

**+뿐만 아니라 -를 추가한 코드입니다.**

#include <iostream>

using namespace std;

class Complex{

double real, imag;

public:

Complex(double r, double i);

Complex(double r){real = r;imag =0.0;};

Complex(){real = 0.0;imag=0.0;};

Complex operator+ (const Complex& c)

{

return Complex(real+c.real,imag+c.imag);

};

Complex operator- (const Complex& c)

{

return Complex(real-c.real,imag-c.imag);

};

void Print(){cout<<"("<<real<<", "<<imag<<")";};

};

int main()

{

Complex a(2.0,1.0),b(1.0),c;

c.Print();

for(int i=0;i<3;i++)

{

c=c+a-b;

c.Print();

}

}