#include<iostream>

#include<assert.h>

using namespace std;

//class Airplane {

// int myid = 0;

// char\* model;

// char\* color;

// int capacity;

// int engines;

//

//

//public:

// Airplane() :myid(NULL), model(NULL), color(NULL), capacity(NULL), engines(NULL) {

//

// }

// Airplane(const char\* model, const char\* color,int capacity, int engines) {

// assert(id < maxcount && "Our garage is full");

// myid = ++id;

// setModel(model);

// setColor(color);

// setCapacity(capacity);

// setEngines(engines);

// }

// //Copy constructor

// Airplane(const Airplane& other) {

// cout << "Copy constructor" << endl;

// this->myid = other.myid;

// setModel(other.model);

// setColor(other.color);

// setCapacity(other.capacity);

// setEngines(other.engines);

// }

// char\* getModel() {

// return model;

// }

// void setModel(const char\* model) {

// int l = strlen(model);

// this->model = new char[l + 1];

// strcpy\_s(this->model, l + 1, model);

// }

//

// char\* getColor() {

// return color;

// }

//

// void setColor(const char\* color) {

// int l = strlen(color);

// this->color = new char[l + 1];

// strcpy\_s(this->color, l + 1, color);

// }

// int getCapacity() {

// return capacity;

// }

// void setCapacity(int capacity) {

// assert(capacity > 0 && "Capacity must be positive");

// this->capacity = capacity;

// }

//

// int getEngines() {

// return engines;

// }

//

// void setEngines(int engines) {

// assert(engines > 0 && "Engine count must be positive");

// this->engines = engines;

// }

//

// static int id;

// static int maxcount;

//

// void show() {

// cout << "+++++++++++++AIRPLANE+++++++++++++++" << endl;

// cout << "Model : " << getModel() << endl;

// cout << "Color : " << getColor() << endl;

// cout << "Capacity : " << getCapacity() << endl;

// cout << "Engine count : " << getEngines() << endl;

// }

//

// ~Airplane()

// {

// cout << "Model : " << model << " deleted" << endl;

// delete[]model;

// delete[]color;

// }

//

//};

//

//int Airplane::id = 0;

//int Airplane::maxcount = 5;

//

//class Garage {

// Airplane\* airplanes;

// int airplane\_count;

//public:

// Garage(Airplane\* airplanes, int count) {

// setCount(count);

// setAirplanes(airplanes);

// }

//

// void setCount(int count) {

// this->airplane\_count = count;

// }

// void setAirplanes(Airplane\* airplanes) {

// this->airplanes = new Airplane[airplane\_count];

// for (size\_t i = 0; i < airplane\_count; i++)

// {

// this->airplanes[i]=airplanes[i];

// }

// }

// void showGarage() {

// for (size\_t i = 0; i < airplane\_count; i++)

// {

// airplanes[i].show();

// }

// }

// ~Garage()

// {

// delete[]airplanes;

// }

//};

//

//

//void main() {

// /\*Airplane a1("Boeng", "White-Green", 120, 4);

// /\*Airplane a2(a1);

// a1.show();

// a2.show();\*/

//

// //Shallow coping

//

// //Deep coping

// Airplane a1("Boeng 1", "White-Green", 120, 4);

// Airplane a2("Boeng 2", "White-Green", 120, 4);

//

// Airplane\* airplanes = new Airplane[2]{ a1,a2 };

//

// Garage garage(airplanes, 2);

//

// garage.showGarage();

//

//}

class User {

char\* username;

char\* password;

public:

User() = default;

User(const char\* username, const char\* password) {

setUsername(username);

setPassword(password);

}

User(const User& other) {

cout << "Copy constructor" << endl;

setUsername(other.username);

setPassword(other.password);

}

char\* getUsername() {

return username;

}

void setUsername(const char\* username) {

int l = strlen(username);

this->username = new char[l + 1];

strcpy\_s(this->username, l + 1, username);

}

char\* getPassword() {

return password;

}

void setPassword(const char\* password) {

int l = strlen(password);

this->password = new char[l + 1];

strcpy\_s(this->password, l + 1, password);

}

void showUserPassword(const User& user) {

//user.setUsername("Salam");

//cout << user.getPassword() << endl;

}

void show() {

cout << "Username :" << username << endl;

cout << "Password : ";

for (size\_t i = 0; i < strlen(password); i++)

{

cout << "\* ";

}cout << endl;

}

const User& getUser(){

User user("admin", "admin");

return user;

}

~User()

{

cout << "Username : " << username << " deleted" << endl;

delete[]username;

delete[]password;

}

};

void main() {

User user("ELvin123", "Elvin123");

User user2("Admin123", "Elvin123");

user2.getUser();

}