

| | |
|--|-----------------|
| Szoftver laboratórium II. 3. ellenőrző dolgozat. 2013.04.09. Kurz/Terem: L1/ | 15 perc |
| Név: _____ | Összpont: _____ |

1. feladat

6 pont

A **pontozott** helyek felhasználásával **egészítse ki** az alábbi programot úgy, hogy a főprogram kommentezett sorai az **előre megadott** kimenetet (1,2 ill. 1) adják! Ügyeljen a tagváltozók elérhetőségére! (3p)

Ezt követően a főprogram minden sorához is adja meg, hogy az mit ír ki a szabványos kimenetre! (3p) Jelölje a szóköz karaktereket is! (Nem biztos, hogy minden megadott helye kell írnia!)

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

class Pont1 {
    int x;          // x koordináta
public:
    Pont1(int x = 0, int y = 0) :x(x)          { cout << "P1ctor ";}
    Pont1(const Pont1& p) :x(p.x)            { cout << "P1copy ";}
    int get() const { return x; }
    Pont1& operator=(const Pont1& p) { x = p.x; cout << "P1= "; return *this; }
    ~Pont1()                                  { cout << "P1dtor "; }
};

class Pont2 {
    Pont1 p1;
    int y;          // y koordináta
public:
    Pont2(int x = 0, int y = 0)
        : p1(x), y(y) /*.....*/          { cout << "P2ctor "; }
    Pont2(const Pont2& p2)
        : p1(p2.getx()), y(p2.y) /*....*/  { cout << "P2copy "; }
    int getx() const { return p1.get(); }
    int gety() const { return y; }
    ~Pont2()          { cout << "P2dtor "; }
};
/* ..... */
ostream& operator<<(ostream& os, const Pont2& p2) {
    os << p2.getx() << ', ' << p2.gety();
    return os;
}
/* ..... */

int main() {
    Pont1 p1(1);          cout << endl;    // P1ctor _____
    Pont2 p2(p1.get(), 2); cout << endl;    // P1ctor P2ctor _____
    Pont2 p = p2;        cout << endl;    // P1ctor P2copy _____
    cout << p             << endl;        // 1,2
    Pont2 *pp = new Pont2(p); cout << endl; // P1ctor P2copy _____
    cout << pp->getx()    << endl;        // 1
    delete pp;          cout << endl;    // P2dtor p1dtor _____
    return 0;          // P2dtor P1dtor P2dtor P1dtor P1dtor _____
}

```