

A csoport

1. feladat

1. Sorolj fel legalább 3-at azok közül a dolgok közül, amik ahhoz szükségesek, hogy egy programozási nyelvről azt mondhassuk, hogy támogatja az OOP-t. (4 van összesen)
 - a. öröklődés (inheritance), polimorfizmus (polymorphism), egységbezárás (encapsulation), adat rejtés (data hiding)
2. Mire szolgál a konstruktor?
 - b. Feladata a dinamikus adattagok létrehozása és az adattagok kezdőértékkel való inicializálása.

B csoport

1. feladat

1. Sorolj fel legalább három konstruktor típust
 - a. default, paraméteres, másoló és move konstruktor
2. Mire szolgál a destruktork?
 - b. a lefoglalt erőforrások felszabadítására szolgál.

2. feladat (Mindenkinek)

Írj egy header fájlt az alábbi programkód alapján! A string-ek másolását szeretnénk elkerülni, ezért használjunk konstans referencia szerinti paraméter átadást, illetve függvény visszatérési értéket. A *getObjectCounter()* függvény vissza adja, hogy hány példányt hoztunk létre az osztályból.

main.cpp

```
#include "a.hpp"
#include "a.hpp"
int main(int argc, char const *argv[])
{
    std::string name = "Jack";
    // a bemeneti paramétert a HEAP-en szeretnénk tárolni, úgyhogy hozzuk létre dinamikusan
    A a1(1), a2(2), a3(3);
    a1.setName(name);
    A a4 = a1;
    A a5(5);
    a5 = a4;
    std::cout << a1.getName() << std::endl; // _Jack_
    std::cout << A::getObjectCounter() << std::endl;
    return 0;
}
```

a.hpp

```
#ifndef _A_HPP_
#define _A_HPP_

#include <iostream>

class A
{
private:
    std::string name;
    int* p;
    static int counter;
public:
    A(int a) : p(new int(a)) { counter++; }

    ~A() { delete p; }

    A(const A& rhs)
    {
        p = new int;
        *p = *rhs.p;
        name = rhs.name;
    }

    A& operator=(const A& rhs)
    {
        if(this != &rhs)
        {
            p = new int;
            *p = *rhs.p;
            name = rhs.name;
        }
        return *this;
    }

    inline void setName(std::string& name_) { name = "_" + name_ + "_"; }
    inline const std::string& getName() const { return name; }
    inline static int getObjectCounter() { return counter; }
};

int A::counter = 0;

#endif
```

3. feladat (Mindenkinek)

a) mit ír ki az alábbi kódrészlet?

'x' value is: 1

'x' value is: 1

b) az alábbi struktúra definíció alapján minek felel meg a `Foo f = 5;` utasítás?

implicit konstuktor hívás

```
#include <iostream>
#define log(var_1, var_2) std::cout << var_1 << " " << var_2 << std::endl
struct Foo
{
    int x;
    Foo(int i = 3) : x(i) { }
    inline void modify(int x) const { x = x; }
    inline void print() const { log("'x' value is: ", x); }
};
int main(int argc, char const *argv[])
{
    Foo f(1);
    f.modify(3);
    f.print();

    const Foo f1 = 1;
    f1.modify(7);
    f1.print();

    return 0;
}
```