

Programozás alapjai 2. 2. Ellenőrző dolgozat. 2013.03.21. Kurz/Terem: G3/	Összpont:
Név:	Neptun:

1. Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (4p)

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

class A {
    int a;
public:
    A(int a = 6) :a(a)          { cout << "K" << a;}
    A(const A& a1):a(a1.a)     { cout << "C" << a;}
    A& operator=(int i)       { a = i; cout << "=" << i;
                               return *this; }

    int get() const           { return a; }
    void f(A a0)              { a0 << cout << "f"; }
    friend ostream& operator<<(ostream& os, const A& rhs);
    ostream& operator<<(ostream& os) const;
    ~A()                      { cout << "D"; }
};

ostream& operator<<(ostream& os, const A& rhs) {
    return os << "G" << rhs.a;
}

ostream& A::operator<<(ostream& os) const {
    return os << "L" << a;
}

inline void nl() { cout << std::endl; }

int main() {
    A a;          nl();          // _____
    A b = a;     nl();          // K6 _____
    A c(30);     nl();          // C6 _____
    A c(30);     nl();          // K30 _____
    a = 48;      nl();          // =48 _____
    c.f(b);     nl();          // C6L6fD _____
    cout << a;   nl();          // G48 _____
    a << cout;  nl();          // L48 _____
    return 0;   nl();          // DDD _____
}
```

2. Az A osztály módosítása nélkül készítsen egy olyan *= operátort, amellyel egy ilyen objektum megszorozható egy egész számmal! Ne dobjon kivételt az alábbi kódrészlet! (2p)

```
A a5(5), a6(6);
if ((a5 *= 6).get() != 30 || (a5.get() != 30) ||
    (a6 *= 8).get() != 48) throw("HIBA");
```

```
A& operator*=(A& lhs, int i) {
    return lhs = lhs.get() * i;
}
```

Programozás alapjai 2. 2. Ellenőrző dolgozat. 2013.03.21. Kurz/Terem: G3/	Összpont:
Név:	Neptun:

1. Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (4p)

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

class B {
    int b;
public:
    B(int b = 19) :b(b)          { cout << "K" << b;}
    B(const B& b0):b(b0.b)      { cout << "C" << b;}
    B& operator=(int i)        { b = i; cout << "=" << b;
                                return *this; }

    int get() const            { return b; }
    void g(B& b1)              { b1 << cout << "f"; }
    friend ostream& operator<<(ostream&, const B);
    ostream& operator<<(ostream&) const;
    ~B()                       { cout << "D"; }
};

ostream& operator<<(ostream& os, const B rhs) {
    return os << "G" << rhs.b;
}

ostream& B::operator<<(ostream& os) const {
    return os << "L" << b;
}

void nl() { cout << std::endl; }

int main() {
    B a;          nl();          // _____
    B b = a;     nl();          // K19 _____
    B c(1);      nl();          // C19 _____
    b = 14;      nl();          // K1 _____
    b = 14;      nl();          // =14 _____
    c.g(b);     nl();          // L14f _____
    cout << a;   nl();          // C19G19D _____
    a << cout;  nl();          // L19 _____
    return 0;    nl();          // DDD _____
}

```

2. A B osztály módosítása nélkül készítsen egy olyan += operátort, amellyel egy ilyen objektumhoz hozzá lehet adni egy egész számot! Ne dobjon kivételt az alábbi kódrészlet! (2p)

```
B a5(5), a6(6);
if ((a5 += 6).get() != 11 || (a5.get() != 11) ||
    (a6 += 8).get() != 14) throw("HIBA");

```

```
B& operator+=(B& lhs, int i) {
    return lhs = lhs.get() + i;
}

```