

Programozás alapja 2.	1. beszámoló dolgozat. 2017.02.24. Kurz/Terem: Gy4/	Elért pontszám:
Név:	Neptun:	

Az első feladatnál minden bejelölt válasz 1 pont, ha helyes, -1 pont, ha hibás! Az esetleges negatív eredmény is összeadódik a többi feladatra kapott pontokkal. A teljes dolgozat eredménye azonban nem lehet negatív.

1. Feladat

(4p)

**Jelölje**, hogy mely kijelentés(ek) igaz(ak) a C++ nyelvre!

- |   |  |
|---|--|
| <input type="checkbox"/> Makrók használata tiltott.                             | <input type="checkbox"/> Minden osztályban meg kell írni legalább 1 konstruktort.  |
| <input type="checkbox"/> Minden operátor túlterhelhető.                         | <input type="checkbox"/> A dinamikus memóriakezelés nyelvi elemmel segített.   |
| <input type="checkbox"/> Nincs pointeraritmetika.                               | <input type="checkbox"/> Az <b>int *p = malloc(10*sizeof(int));</b> C utasítás C++ megfelelője: <b>int *p = new(10);</b> |
| <input type="checkbox"/> Névterek használata a névütközések elkerülését segíti. |  |

**Jelölje**, hogy mely kijelentés(ek) igaz(ak) az alábbi C++ kódrészletre!

```
{
    struct X {
        std::ostream& operator<<(std::ostream& os) { return os << "Kakukk"; }
        ~X() {}
    } a123;
    a123 << std::cout;
}
```

- |  |   |
|--|---|
| <input type="checkbox"/> A kódrészlet kiírja, hogy <b>Kakukk</b> | <input type="checkbox"/> A kódrészlet hibás.                                    |
| <input type="checkbox"/> <b>a123</b> egy objektum.               | <input type="checkbox"/> <b>X</b> konstruktora nem hívódik meg a kódrészletben. |
| <input type="checkbox"/> <b>X</b> -nek nincs értékadó operátora. | <input type="checkbox"/> <b>X</b> -nek nincs konstruktora.                      |

2. **Tervezz** egy olyan osztályt (Auto), ami egy autó pillanatnyi és megengedett sebességét (mindkettő valós érték) képes tárolni. Az osztálynak

(6p)

- legyen olyan konstruktora, ami paraméterként kapja az autó pillanatnyi és a megengedett sebességét!
- legyenek olyan lekérdező függvényei, amivel a pillanatnyi és a megengedett sebesség külön-külön lekérdezhető;
- legyen olyan operátora (+), ami egy olyan Auto objektumot hoz létre, amiben a pillanatnyi sebesség a jobb oldali egész operandussal növelt.

**Valósítsa** meg az osztályt C++ nyelven úgy, hogy az osztálynak legyen legalább 1 nem inline tagfüggvénye! Ne lehessen az adattagokhoz kívülről közvetlenül hozzáférni. Ügyeljen arra, hogy konstans objektumokra is jól működjön az osztály! Készítsen egy inserter (<<) operátort is, amivel egy **ostream** típusú objektumra kiírható a pillanatnyi és a megengedett sebesség! **Működjön** helyesen az alábbi kódrészlet:

```
Auto trabi(30, 50);
trabi = trabi + 10; // növeljük 10-zel
std::cout << trabi.getSpeed() << std::endl; // Kiír: 40
std::cout << trabi + 20 << std::endl; // Kiír: 60/50
```

```
class Auto {
    double sebesseg;
    double megengedett;
public:
    Auto(double s, double m) :sebesseg(s),megengedett(m) {}
    double getSpeed() const { return sebesseg; }
    double getMegengedett() const { return megengedett; }
    Auto operator+(int) const;
};
Auto Auto::operator+(int rhs) const {
    return Auto(sebesseg + rhs, megengedett);
}
ostream& operator<<(ostream &os, Auto a) {
    return os << a.getSpeed() << "/" << a.getMegengedett();
}
```

