

Programozás alapja 2.	1. ellenőrző dolgozat.	2016.03.08.	Kurz/Terem: G4/	Elért pontszám:
Név:		Neptun:		

Az első két 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. Jelölje, hogy mely kijelentés(ek) igaz(ak) a C++ nyelvre! (2p)

- | | |
|---|---|
| <input type="checkbox"/> Makrók használata tiltott. | <input type="checkbox"/> Minden osztályban meg kell írni legalább 1 konstruktort. |
| <input type="checkbox"/> Minden operátor túlterhelhető. | <input checked="" type="checkbox"/> A dinamikus memóriakezelés nyelvi elemmel segített. |
| <input type="checkbox"/> Nincs pointeraritmetika. | <input checked="" type="checkbox"/> A referencia típusú változókat definiálásukkor mindig inicializálni kell. |

2. Jelölje, hogy mely kijelentés(ek) igaz(ak) az alábbi C++ kódrészletre! (2p)

```
{ struct S {}; S* s1 = new S; }
```

- | | |
|--|--|
| <input type="checkbox"/> s1 egy objektum. | <input type="checkbox"/> S konstruktora nem hívódik meg a kódrészletben. |
| <input checked="" type="checkbox"/> *s1 egy objektum. | <input checked="" type="checkbox"/> S konstruktora 1-szer hívódik meg a kódrészletben. |
| <input type="checkbox"/> S-nek nincs értékadó operátora. | <input type="checkbox"/> S destruktora 1-szer hívódik meg a kódrészletben. |

3. **Tervezzén** 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 a pillanatnyi sebességet a jobboldali operandus értékével (double) megnöveli.

Valósítsa meg az osztályt C++ nyelven úgy, hogy az osztálynak legyen legalább 1 nem inline tagfüggvénye! Ne lehessen az adatokhoz kívülről közvetlenül hozzáférni. 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 += 10; // növeljük 10-zel
std::cout << trabi.getSpeed() << std::endl; // Kiír: 40
trabi += 20;
std::cout << trabi << std::endl; // Kiír: 60/50
```

Egy lehetséges megoldás:

```
class Auto {
    double sebesseg;
    double megengedett;
public:
    Auto(double sebesseg, double megengedett) : sebesseg(sebesseg),
                                                megengedett(megengedett) {}

    double getSpeed() const { return sebesseg; }
    double getMegengedett() const { return megengedett; }
    Auto& operator+=(double rhs);
};

Auto& Auto::operator+=(double rhs) {
    sebesseg += rhs;
    return *this;
}

ostream& operator<<(ostream &os, Auto a) {
    os << a.getSpeed() << "/" << a.getMegengedett();
    return os;
}
```

