

Programozás alapja 2.	1. ellenőrző dolgozat.	2016.03.07.	Kurz/Terem: G2/	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)

- Az `if` feltételében is lehet változót deklarálni.  
`pl: if (int i )`
- `inline` függvényben nem lehet `for` utasítás.
- A függvények egymásba ágyazhatók. A belső függvények a `scope (::)` operátorral érhetőek el.
- A destruktorkor mindig `void` típusú, amit nem kell kiírni.
- Konstans tagfüggvény nem változtathatja meg az objektum állapotát.
- `realloc` helyett `renew` utasítást kell használni.

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

```
{ struct S {}; new S[3]; S s1;}
```

- `S` egy osztály.
- `S` egy objektum.
- `S`-nek nincs konstruktora.
- `S` konstruktora 3-szor hívódik meg a kódrészletben.
- `S` konstruktora 4-szer hívódik meg a kódrészletben.
- `S` destruktora 4-szer hívódik meg a kódrészletben.

3. **Tervezz** egy olyan osztályt (Datum), ami idő (hónap, nap) tárolására alkalmas! Az osztálynak (6p)

- legyen paraméter nélkül hívható konstruktora, ami január 1-et (hó:1, nap:1) állít be;
- legyen olyan beállító függvénye (set), amivel a hónap és a nap is beállítható; amennyiben csak a hónapot adják meg paraméterként, úgy adott hónap elsejét állítson be a függvény;
- legyenek olyan lekérdező függvényei, amelyekkel a hónap és a nap külön-külön lekérdezhető;
- legyen olyan operátora (--), amivel a napot lehet csökkenteni, ha az nagyobb, mint 1;

**Valósítsa** meg az osztályt C++ nyelven úgy, hogy az osztálynak legyen legalább 1 nem inline tagfüggvénye! Ne lehessen az adatokhoz kívülről közvetlenül hozzáférni. A beállító függvény dobjon **const int** hibát, ha a megadott nap nagyobb, mint 31, vagy a hónap nagyobb, mint 12!

**Működjön** helyesen az alábbi kódrészlet:

```
Datum n0, n0501, n1224;
n0501.set(05); // május 1
n1224.set(12, 25); // karácsony
--n1224; // karácsony este
```

Egy lehetséges megoldás:

```
class Datum {
    int honap;
    int nap;
public:
    Datum() :honap(1), nap(1) {}
    void set(unsigned int, unsigned int = 0);
    int getHonap() const { return honap; }
    int getNap() const { return nap; }
    Datum operator--();
};

void Datum::set(unsigned int h, unsigned int n) {
    if (h > 12 || n > 31) throw 1;
    honap = h;
    nap = n;
}

Datum Datum::operator--() {
    if (nap > 1) --nap;
    return *this;
}
```

