

Programozás alapjai 2. Zárthelyi 2018.03.22. „A”		Alíírás:

1. Adott a következő kódrészlet.

```
class Instrument {
public:
    char* unique_id;

    Instrument(const char* param) {
        unique_id = new char[strlen(param) + 1];
        strcpy_s(unique_id, strlen(param) + 1, param);
    }
};

void Play(Instrument i) {
    cout << "Playing the " << i.unique_id << endl;
}

int main() {
    Instrument pianol = "Piano_123";
    Play(pianol);

    return 0;
}
```

Feladat	Max. pontszám	Pontszám
1	5	
2	5	
3	5	
4	10	
Σ	25	
<i>iMSC</i>	5	

a) Mely pontokon történik a fenti kódban az Instrument osztállyal kapcsolatos konstruktor hívás? Konkrétan melyik konstruktor(ok)ról van szó?

b) A fenti kód futtatása esetén két különböző hiba miatt is memóriakezelési probléma adódik. Magyarázza el, konkrétan mi okozza ezeket. Egészítse ki fentebb a kódot függvénnyel vagy függvényekkel, hogy megfelelően működjön. A már leírt függvényeken ne változtasson. Figyeljen arra, hogy a módosítás ne hozzon be újabb memóriakezelési hibát. (Az adattagok láthatósága szándékos a kód rövidsége érdekében.)

2. Láthatóság

a) Mi az a két, láthatóságot módosító kulcsszó, amelyet eddig tanultunk? Milyen típusú tagok láthatóságát módosítja? Melyik kulcsszó pontosan kinek enged hozzáférést?

b) Demonstrációként készítsen egy egyszerű Point osztályt, amely x és y tagváltozókat tartalmaz, amelyeknek a módosítását csak 0 és 100 közötti értékekre engedi, és osztályon kívülről x és y értéke megismerhető és megváltoztatható. A friend mechanizmust ne használja!

3. Konstans tagfüggvények

- a) Mit nevezünk konstans tagfüggvénynek? Mi a szintaktikája?
- b) Milyen tagfüggvényt hívhat a konstans tagfüggvény, honnan lehet őt hívni?
- c) Miért, mikor van rá szükség?
- d) Írjon egy egyszerű példát konstans tagfüggvényre, és mutassa be a használatát olyan módon, amikor nélküle nem fordulna az alkalmazás.

4. Tervezési feladat

A KamuBolt nevű vállalkozás termékei árusításához szoftverrendszert készített. Az egyes termékeket (*Product*) tartja nyilván. Minden terméknek eltárolja az árát (*price*). (A termék egyéb azonosítóival, nevével stb. most nem foglalkozunk). A cég időnként a teljes árukészletére akciót hirdet, ahol 0-100 közötti kedvezményt ad minden termékre (*salePercent*). Emellett időnként módosítja a termék árakat is a (*) sorban jelölt módon.

a) A feladat a *Product* osztály olyan szintű megvalósítása, hogy az alábbi kódrészlet hibátlanul lefusson.

b) Mit és hogyan kell ahhoz megvalósítani, hogy a (**) kiíratás jól működjön? (Valósítsa is meg).

```
Product funyiro(35000); //a kezdeti ar 35.000 Ft.  
Product::setSalePercent(250); //Hibás érték esetén ne történjen semmi, ne állítsuk el.  
Product::setSalePercent(50); //A Kamubolt 50% arleszallitast hirdet minden termekre!  
int plus1 = 15000; //Eloszor ennyivel akarnak arat emelni a funyiro eseten  
int plus2 = 10000; //Majd meg ennyivel  
funyiro += plus1 += plus2; //(*)  
cout << funyiro.getCurrentPrice(); //(**) kiiratjuk a termék aktualis kedvezmenyes arat („30000“)
```

Használja a dőlt betűs osztály- és tag neveket.

5. iMSC feladat: +5 iMSC pontért

Egészítse ki az osztályt oly módon, hogy a lenti kódrészletben levő összehasonlítások működjenek. (A lentiektől eltérő összehasonlításokat nem kell tudni végezni.)

```
const Product gatya(15000);  
if (gatya > funyiro)  
    cout << "A gatya akciós ára dragabb, mint a funyiroe" << endl;  
if (gatya > 35000)  
    cout << "A gatya akciós ára dragabb 35.000-nél" << endl;
```