

Programozás alapja 2.	1. beszámoló dolgozat. 2019.02.28. Kurz/Terem: L01/	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.

Labor e.d. (beugró): 1. feladtból min 2 pont.

1. Feladat

(4p)

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

- | | |
|--|--|
| <input type="checkbox"/> Minden programot <code>using namespace std;</code> direktívával kell kezdeni. | <input type="checkbox"/> Függvényprototípus használata kötelező, ha a függvény a használat előtt nincs definiálva. |
| <input type="checkbox"/> Referencia típusú változó nem adható át paraméterként. | <input type="checkbox"/> A throw utasítás végtelen ciklusból való kilépésre való. |
| <input type="checkbox"/> A new képes kivételt generálni. | <input type="checkbox"/> Kivételt csak abban a fájlban lehet elfogni, ahol a throw utasítás van. |
| <input type="checkbox"/> A névterek egymásba ágyazhatók. | <input type="checkbox"/> Minden programot <code>#include <iostream></code> direktívával kell kezdeni. |

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

```
{ int i; while (std::cin >> i); }
```

- Addig olvas be egész számokat, amíg az inputon egész formátumnak megfelelő adatok érkeznek
- A kódrészlet hibás, mert az **i** változó nem kap értéket.
- Végtelen ciklus.

2. Tervezzén egy olyan osztályt (**Ido**), ami idő (óra, perc) tárolására alkalmas! Az osztálynak

(6p)

- legyen paraméter nélkül hívható konstruktora, ami 12 óra 0 percet állít be;
- legyen olyan konstruktora is, amivel az óra és a perc is megadható;
- legyen olyan operátora (+), amivel egy időponthoz egész (int) órát lehet adni jobbról! Az eredmény az összeadás műveletnél megszokott módon új objektumban keletkezzen! Amennyiben az összeadás során keletkező objektumban az óra értéke meghaladná a 24 órát, úgy dobjon **const char*** kivételt!

Valósítsa meg az osztályt C++ nyelven úgy, hogy az osztálynak legyen legalább 1 nem inline tagfüggvénye! Ne lehessen az adattagokhoz kívülről közvetlenül hozzáférni.

Működjön helyesen az alábbi kódrészlet:

```
Ido t0, t10(10, 0), t23(23, 05); // 12:00, 10:00, 23:05 időpontokat hoz létre
t0 = t10 + 1; // t0-ba 11:00 kerül, t10 változatlan marad
t23 + 2; // kivétel keletkezik
```

```
class Ido {
    int ora;
    int perc;
public:
    Ido(int o = 12, int p = 0) :ora(o), perc(p) {}
    Ido operator+(int rhs) const;
};
Ido Ido::operator+(int rhs) const {
    if (ora + rhs > 24) throw "Ora > 24";
    return Ido(ora + rhs, perc);
}
```

Programozás alapja 2.	1. beszámoló dolgozat. 2019.02.28.	Kurz/Terem: L02/	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.

Labor e.d. (beugró): 1. feladtból min 2 pont.

1. Feladat

(4p)

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

- A struktúra egy osztály.
- A C nyelvben ismert realloc függvényt a **renew** operátor helyettesíti.
- A **new** sosem generál kivételt.
- A névterek nem ágyazhatók egymásba.
- Az értékadó operátor nem terhelhető túl.
- Referencia típusú változó átadható paraméterként.
- A **delete[]** egy operátor.
- Egy változót többször is lehet definiálni, de deklarálni csak egyszer lehet.

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

```
void swap(int& a, int& b) {
    int c = a;
    a = b;
    b = c; }
```

- A függvény felcseréli a referencia paraméterként kapott két változó adatát.
- A függvény az első paraméterként kapott változóba másolja a második paraméter adatát.
- A függvény hibás, mert lokális változó referenciáját képi.

3. **Tervezz** egy olyan osztályt (**Honap**), ami hónap sorszámának tárolására alkalmas! Az osztálynak

(6p)

- legyen paraméter nélkül hívható konstruktora, ami 3. hónapot állít be;
- legyen olyan konstruktora is, amivel megadható a hónap;
- legyen olyan operátora (+), amivel egy Honaphoz egész (int) hónapot lehet adni jobbról! Az eredmény az összeadás műveletnél megszokott módon új objektumban keletkezzen! Amennyiben az összeadás során keletkező objektumban a hónap értéke meghaladná a 12-t, úgy dobjon **const char*** kivételt!

Valósítsa meg az osztályt C++ nyelven úgy, hogy az osztálynak legyen legalább 1 nem inline tagfüggvénye! Ne lehessen az adattagokhoz kívülről közvetlenül hozzáférni.

Működjön helyesen az alábbi kódrészlet:

```
Honap h3, h10(10); // 3. és 10. hónapot hoz létre
h3 = h10 + 1; // h3-ba 11. hónap, kerül, h10 változatlan
h3 + 2; // kivétel keletkezik
```

```
class Honap {
    int ho;
public:
    Ho(int h = 3) :ho(h) {}
    Ho operator+(int rhs) const;
};
Ho Ho::operator+(int rhs) const {
    if (ho + rhs > 12) throw "Ho > 12";
    return Ho(ho + rhs);
}
```

Programozás alapja 2.	1. ellenőrző dolgozat.	2019.03.01.	Kurz/ Terem : L03/	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.

Labor e.d. (beugró): 1. feladatból min 2 pont.

1. Feladat

(4p)

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

- | | |
|--|--|
| <input type="checkbox"/> inline függvénynek nem lehet default paramétere. | <input type="checkbox"/> Függvénysablonnak nem lehet paramétert átadni. |
| <input type="checkbox"/> realloc helyett renew utasítást kell használni. | <input type="checkbox"/> A ciklus feltételében is lehet változót deklarálni. Pl: <code>while (int i = f())...</code> |
| <input type="checkbox"/> Destruktornak csak konstans paramétere lehet. | <input type="checkbox"/> Konstans tagfüggvény nem változtathatja meg az objektum állapotát. |

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

```
{ struct S { int a; }; S so; }
```

- | | |
|---|---|
| <input type="checkbox"/> S egy osztály. | <input type="checkbox"/> so destruktorosa sosem hívódik meg, mert nincs. |
| <input type="checkbox"/> S egy objektum. | <input type="checkbox"/> so adattagja inicializálatlan. |
| <input type="checkbox"/> S minden adattagja privát | |

2. Tervezz egy osztályt (Tort) törtek tárolására! Az osztály a törtet két egész számként (számláló, nevező) tárolja! Az osztálynak

(6p)

- legyen paraméter nélkül hívható konstruktora, ami a számlálót 0-ra, a nevezőt 1-re állítja;
- legyen olyan konstruktora is, amivel a számláló és a nevező is beállítható;
- legyen olyan operátora (/), amivel egy ilyen tört egy egész számmal elosztható! Az eredmény az osztás műveletnél megszokott módon új objektumban keletkezzen! Nullával való osztás esetén dobjon **const char*** kivételt!

Valósítsa meg az osztályt C++ nyelven úgy, hogy az osztálynak legyen legalább 1 nem inline tagfüggvénye! Ne lehessen az adattagokhoz kívülről közvetlenül hozzáférni.

Működjön helyesen az alábbi kódrészlet:

```
Tort t0, t1(1), t05(1, 2);           // 0/1, 1/1, 1/2
t0 = t1 / 2;                        // t0-ba 1/2 kerül, t1 változatlan marad
t05 = t05 / 0;                     // kivétel keletkezik
```

```
class Tort {
    int szamlalo;
    int nevezo;
public:
    Tort(int szamlalo=0, int nevezo=1) :szamlalo(szamlalo), nevezo(nevezo) {}
    Tort operator/(int) const;
};
Tort Tort::operator/(int n) const {
    if (n == 0) throw "Div 0";
    return Tort(szamlalo, n*nevezo);
}
```

Programozás alapja 2.	1. ellenőrző dolgozat.	2019.03.01.	Kurz/Terem: L04/	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.

Labor e.d. (beugró): 1. feladtból min 2 pont.

1. Feladat

(4p)

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

- | | |
|--|--|
| <input type="checkbox"/> A private kulcsszó struktúrában nem szerepelhet. | <input type="checkbox"/> A scope (::) operátorral a privát adatok is elérhetők bárki számára. |
| <input checked="" type="checkbox"/> A new[] képes kivételt generálni. | <input checked="" type="checkbox"/> A new[] operátorral foglalt memóriaterületet delete[] operátorral kell felszabadítani. |
| <input type="checkbox"/> inline függvényben nem lehet ciklus. | <input type="checkbox"/> A névterek nem ágyazhatók egymásba. |

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

```
{ class C { int x; }; C co;}
```

- | | |
|--|---|
| <input type="checkbox"/> C egy objektum. | <input type="checkbox"/> C konstruktora nem hívódik meg, mert nincs |
| <input checked="" type="checkbox"/> C egy osztály. | <input type="checkbox"/> A kódrészletben memóriaszivárgás lép fel. |
| <input checked="" type="checkbox"/> C minden adattagja privát. | |

2. Tervezz egy olyan osztályt (Vektor), ami síkbeli pontok helyvektorának tárolására használható! A helyvektort X,Y koordinátákkal tároljuk. Az osztálynak

(6p)

- legyen paraméter nélkül hívható konstruktora, ami mindkét koordinátát nullára állítja;
- legyen olyan konstruktora is, amivel mind az X, mind az Y koordináta beállítható;
- legyen olyan operátora (/), amivel egy vektort el lehet osztani egy egész számmal! Az osztás során mind az X, min az Y koordináta osztódik. Az eredmény az osztás műveletnél megszokott módon új objektumban keletkezzen! Nullával való osztás esetén dobjon **const char*** kivételt!

Valósítsa meg az osztályt C++ nyelven úgy, hogy az osztálynak legyen legalább 1 nem inline tagfüggvénye! Ne lehessen az adattagokhoz kívülről közvetlenül hozzáférni. **Működjön** helyesen az alábbi kódrészlet:

```
Vektor v0, v1(2), v15(1, 5); // (0, 0); (2,0); (1,5)
v0 = v1 / 2; // v0-ba (1,0) kerül, v1 változatlan marad
v15 = v15 / 0; // kivétel keletkezik
```

```
class Vektor {
    double x;
    double y;
public:
    Vektor(double x = 0, double y = 0) :x(x), y(y) {}
    Vektor operator/(int rhs) const;
};
Vektor Vektor::operator/(int rhs) const {
    if (rhs == 0) throw "div 0";
    return Vektor(x/rhs, y/rhs);
}
```