

1. LABOR

AZ ÚJ KÖRNYEZET ÉS A REFERENCIA TÍPUS

Általános információk

1. iMSc pontok

Az első laboron még nem szerezhető iMSc pont. A többi laboron szerezhető iMSc-s feladatok megoldásainak **AUT portálra** *.zip fájlba csomagoltan való feltöltési határideje az aktuális labortól számított három nap. Ha egy feladatban kérdések szerepelnek, a pontok csak akkor fogadhatók el, ha mellékletben egy **igényes** jegyzőkönyv is szerepel a kérdésekre vonatkozó válaszokkal. iMSc pont szerzésére bármely hallgató jogosult, aki az előtte lévő feladatokkal már végzett (laborvezető ellenőrzi a haladást).

2. Házi feladat

Házi feladat pontozási irányelvekről bővebben a „HF_IRANYELVEK.pdf” szól.

Kötelező feladatok

1. Emlékeztető, ismerkedés az új környezettel

Visual Studioban készíts egy új *Empty project* projektet *GettingStarted* néven, amely valósítsa meg az alábbiakat:

- *Olvasson be* egy egész számot (N), majd egy legfeljebb 10 karakterből álló szöveget (S)
- N-t és S-t adja át egy *függvénynek* (F)
- F-ben S-t a standard kimenetre *írja ki* N-szer

2. Forráskód dekompozíció

Az előző feladat F függvényét valósítsd meg kétféleképpen:

1. Külön *printing.cpp* fájlban definiáld, majd abban a fájlban, amiben használsz (*main.cpp*), *extern* kulcsszóval jelezd a fordítónak, hogy egy másik *.cpp-fájlban keresse a definíciót
2. Töröld az *extern-t*, helyette hozz létre egy *printing.h-t*
 - a. csak a deklaráció szerepeljen benne, a definíció maradjon a *printing.cpp*-ben
 - b. oldd meg, hogy többszöri include-olás esetén ne legyen többszörös deklaráció (Emlékeztető: <https://en.cppreference.com/w/cpp/preprocessor/include>)

3. Kódrészlet debuggolás

A Visual Studio eszközeit felhasználva futtasd utasításonként a következő kódrészleteket:

C- rossz	C-jó	C++
<pre>#include<stdio.h> void f(int i) { i=i+2; } int main(void) { int i=0; f(i); printf("%d\n",i); } /* A kimenet 0 */</pre>	<pre>#include<stdio.h> void f(int* pi) { (*pi)=(*pi)+2; } int main(void) { int i=0; f(&i); printf("%d\n",i); } /* A kimenet 2 */</pre>	<pre>#include<stdio.h> void f(int& i) { i=i+2; } int main(void) { int i=0; f(i); printf("%d\n",i); } /* A kimenet 2 */</pre>

4. Referencia gyakorlása

Készíts egy C++ programot, amely egy bevásárlókosár kezelését segíti! A programban két függvényt kell használnod:

- hozzaad_termek:** Referencia szerint kap egyösszár változót, és hozzáadja egy új termék árát. A termék ára normál paraméterként érkezik.
- alkalmaz_kedvezmeny:** Referencia szerint kapja azösszarat, és egy megadott százalékos kedvezményt alkalmaz rá.

Feladat részletei:

- Kérd be a felhasználótól a kosárba tett termékek darabszámát!
- Kérj be ennyi termékárát, és minden egyes alkalommal használd a hozzáad_termek függvényt az árak összeadására!
- Kérd be a kedvezmény mértékét százalékban, majd alkalmazd rá az alkalmaz_kedvezmeny függvényt!
- Írd ki a végső árat a kedvezménnyel csökkentve!

Példa futás:

```
Hány terméket szeretnél megadni? 3
Add meg az 1. termék árát: 500
Add meg a 2. termék árát: 1500
Add meg a 3. termék árát: 2500
Add meg a kedvezmény mértékét (százalékban): 10
A végső ár a kedvezménnyel: 4050.00
```

Ötletek a feladatok megvalósításához

scanf_s szabványossága

A scanf_s a Microsoft által bevezetett, biztonságosabb változata a scanf-nek, amely a puffer méretét is elvárja, például scanf_s("%49s", str, sizeof(str));. Ez a függvény csak a **C11 szabvány opcionális kiegészítéseként** érhető

el, és főként Windows/MSVC környezetben támogatott. Más fordítóprogramok, például GCC vagy Clang, alapértelmezetten nem implementálják, így hordozhatósági problémát okozhat.

Gyakorló feladatok

- [Referencia helytelen/helyes használata](#) a)-e)