

ANSI C NYELVEN**16-17. ÓRA****Műveletek adatokkal**– *értékadó műveletek*

- olyan műveletek, amikor az operátor bal oldala felveszi a jobboldali kifejezés értékét
- értékadáskor szükség esetén típus-átalakítás történik (a balérték típusára alakítunk)
- egyenlő (=)
- plusz-egyenlő (+=), mínusz-egyenlő (–=)
- szőr-egyenlő (*=), per-egyenlő (/=)

```
int c;
float f;

f = 3.1415;
c = f;           // 'c' értéke 3 lesz
c += 2;         // 'c' értéke 5 lesz
c *= 5;         // 'c' értéke 25 lesz
f = c;          // 'f' értéke 25.00 lesz
```

– *matematikai műveletek*

- olyan műveletek, amelynek operandusai csak számok (egész, valós) és karakter lehet
- lehetnek additív (+, –) vagy multiplikatív jellegűek (*, /, %)
- összeadás (+) és (–) kivonás
- szorzás (*), osztás (/), maradékos osztás (%)
- a műveletek eredménye a műveletben résztvevő operandusok típusától függ
 - egész számok osztása esetén az eredmény is egész szám lesz (3/2 => 1)
 - amennyiben a műveletben résztvevő számok eltérő típusúak, akkor típuskonverzió történik
 - implicit konverzió => a fordító automatikusan elvégzi az átalakítás (az egyik operandus miatt)
 - explicit konverzió => a felhasználó jelzi az átalakítás szükségességét

```
int c;
float f;

f = c*9/5+32;           // hibás eredmény, mert egész számokon értelmezett
f = c*9.0/5+32;        // IMPLICIT típus átalakítás (9.0 valós konstans)
f = (float) c*9/5+32;   // EXPLICIT típus átalakítás
```

– *növelő / csökkentő (léptető) műveletek*

- a műveletben megadott változó értéke eggyel nő, vagy csökken
- a művelet alkalmazható PRE és POST formában
- PRE (++a) => a változó értéke eggyel nő, majd ezt követően felhasználásra kerül
- POST (a++) => a változó értéke felhasználásra kerül, majd ezt követően értéke eggyel nő

```
int c;
int a;

c = 3;
a = ++c;           // 'a' értéke 4 lesz
a = c++;           // 'a' értéke 3 lesz

// első tíz egész szám és négyzete
for (c=0; c<10; c++) { printf("\n%d %d", c, c*c);
```

– összehasonlító műveletek

- a műveletek logikai igaz (1), vagy hamis (0) értéket szolgáltatnak
- azonos? (==), nem azonos? (!=)
- kisebb (<), kisebb, vagy egyenlő (<=)
- nagyobb (>), nagyobb, vagy egyenlő (>=)

```
5 < 3           // hamis, vagy 0
3.2 >= 3.2     // igaz, vagy 1
3.2 >= 1.9     // igaz, vagy 1
'a' < 'A'      // hamis, mert az 'a' ASCII kódja 97, az 'A' kódja 65
```

– logika műveletek

- a logikai értékeken (igaz, hamis) végzett műveletek összessége
- alapértelmezés szerint a hamis = 0, és az igaz = 1 értéket képviselnek
- a műveletek logikai igaz (1), vagy hamis (0) értéket szolgáltatnak
- egy kifejezés logikai hamisnak minősül, ha értéke 0 (nulla)
- egy kifejezés logikai igaznak minősül, ha értéke nem 0 (nem nulla)
- vannak egy és két operandusú logikai műveletek
 - egy operandusú logikai művelet: NOT (tagadás, negáció)
 - két operandusú logikai művelet: AND (és, konjunkció), OR (vagy, diszjunkció)
- tagadás (!)
- és (&&), vagy (||)

```
!(3+2)         // hamis, mert 3+2 = 5, ami IGAZ érték
5 && 0         // hamis, mert 5 IGAZ, a 0 HAMIS
5 || -1       // igaz, mert 5 IGAZ és a -1 is IGAZ
5==7 && 3     // hamis, mert 5 nem egyenlő 7-tel, azaz HAMIS
```

Házi feladat

- logikai műveletek igazságtáblája