Lvrednosti

- Neki ugrađeni operatori zahtevaju kao neki svoj operand lvrednost, pa se kao taj operand može pojaviti samo izraz koji jeste lvrednost; ako operator ima bočni efekat nad operandom, onda taj operand mora biti lvrednost
- Činjenica da rezultat nekog operatora jeste lvrednost prestavlja činjenicu da ta vrednost upućuje na neki "čvrst" entitet, koji ima svoj identitet, a koja se onda može upotrebiti kao operand operatora koji zahteva lvrednost
- Tako prevodilac proverava ispravnost izraza. Na primer:

- Operator dodele = zahteva kao svoj levi operand lvrednost, da bi nad njim proizveo bočni efekat, ali ne zahteva da njegov desni operand bude lvrednost; operator + ne proizvodi lvrednost:

```
a = (b+c)
(b+c) = a
```

Ispravno pod uslovom da je *a* ime varijable, što jeste lvrednost; desni operand operatora = ne mora biti lvrednost

Greška u prevođenju, jer levi operand operatora = nije lvrednost

- I rezultat operatora dodele jeste lvrednost, koja se odnosi na istu stvar na koju i njegov levi operand (koji svakako jeste lvrednost), pa se rezultat ovog operatora može upotrebiti tamo gde se zahteva lvrednost; na primer:

```
(a = b) = c
```

Rezultat izraza *a=b* jeste lvrednost koja se odnosi na *a*, pa se njemu dodeljuje vrenost *c*

Lvrednosti

- Operator *&* zahteva operand koji jeste lvrednost, ali mu rezultat nije lvrednost. Nasuprot tome, operator indirekcije * ne zahteva da operand bude lvrednost, ali mu rezultat jeste lvrednost:

```
&a
&a = …
&(a+3)
*(p+3)
*(p+3) = …
```

- Operator *a<<b* daje rezultat koji predstavlja binarnu vrednost celobrojnog operanda *a* pomerenu za *b* bita ulevo. Kako znati da li taj operator proizvodi bočni efekat ili ne, tj. da li menja (pomera) svoj levi operand? Ako ga menja, taj operand bi morao biti lvrednost, ali ovaj operator ne zahteva da levi operand bude lvrednost, pa onda sigurno ne proizvodi bočni efekat

- Rezultat poziva funkcije je lvrednost akko funkcija vraća referencu na lvrednost, na primer:

```
int& f ();
f() += 1;
X& X::operator+=(int);
X x;
X* p = &(x += 3);
```