

# Operatori

- ❖ Neka od (brojnih i složenih) pravila redosleda izračunavanja (*order of evaluation*) su:
  - sva izračunavanja vrednosti i bočnih efekata celog izraza završavaju se pre nego što počne izračunavanje vrednosti i bočnih efekata narednog izraza
  - izračunavanje svih argumenata funkcije i svih operacija pre poziva funkcije završava se pre bilo kog izvršavanja unutar pozvane funkcije; analogno važi za povratak iz funkcije
  - izračunavanje vrednosti (ali ne obavezno i bočnih efekata) svih operanada bilo kog operatora završava se pre izračunavanja vrednosti (ali ne obavezno i bočnih efekata) tog operatora
  - za neke operatore definisan je i sekvenciran redosled izračunavanja operanada (za neke samo izračunavanja vrednosti, za neke i bočnih efekata): ++, --, &&, ||, ?:, , (zarez), =, @= (operatori složene dodele), [], >>, <<, dok za ostale nije
- ❖ Logički operatori && (logičko I) i || (logičko ILI) izračunavaju se po pravilu “kratkog spoja” (*short-circuit evaluation*): ako prvi operand definitivno određuje vrednost rezultata, drugi se i ne izračunava:

**if** (p && p->f()) ...

Ovo je bezbedno, jer ako pokazivač *p* ima vrednost *null*, drugi operand operatora && neće ni biti izračunavan, pa ovaj pokazivač neće biti korićen neispravno

---

# Lvrednosti

---

- ❖ Zbog čega su sledeći izrazi ispravni:

```
&a  
*(p+3)  
x += 1
```

a sledeći nisu?

```
&(i+3)  
(i+j) += k
```

- ❖ Intuitivno je jasno da u nekim od ovih situacija operacija nije dozvoljena jer njen operand ne predstavlja nekakav “čvrst” objekat koji se može identifikovati, koji zauzima neku memoriju, ali je to suviše neformalno tumačenje koje prevodilac ne može da sprovodi: potrebna su precizna, formalna pravila kojima prevodilac može da proverava ispravnost ovakvih izraza
- ❖ Još je u jeziku C postojao kocept (on vodi poreklo iz nekih starijih jezika) tzv. *lvrednosti* (*lvalue*), kao “nečega što može da stoji sa leve strane znaka =” i koji se koristio za ovakvu namenu; ovaj koncept u jeziku C++ ima složenije značenje, ali i veći značaj, pošto se operatori mogu preklapati, pa operandi mogu biti i objekti klasa
- ❖ Pored sintaksne provere i provere tipova operanada, prevodilac proverava i svojstvo *lvrednosti* za izraze