

Preklopljeni operatori

- ❖ Jedan aspekt može biti veoma bitan pri odlučivanju o tome da li neki preklopljeni operator treba napraviti kao nestatičku funkciju članicu ili kao funkciju nečlanicu (ako je to moguće):
 - poziv nestatičke funkcije članice za neki izraz ne dozvoljava “promociju” tog izraza, odnosno bilo kakvu implicitnu konverziju levog operanda operatora `::` zahteva se da klasa kojoj pripada taj objekat, ili neka njena osnovna klasa, ima navedenu funkciju, inače poziv nije dozvoljen:

```
complex complex::operator+ (complex c) {...}  
complex::complex (double d) {...}
```

Konstruktor konverzije, definiše implicitnu konverziju iz *double* u *complex*

```
complex c1(3.,4.); double d = 5.0;  
complex c2 = d+c1;
```

Greška u prevođenju: tumači se kao *d.operator+(c1)*, a levi operand operatora `+` ne konvertuje se u objekat

- pri pozivu funkcije nečlanice vrše se dozvoljene i definisane implicitne konverzije argumenata; za navedeni primer klase *complex*, ukoliko se operator `+` definiše kao funkcija nečlanica i definiše navedeni konstruktor konverzije iz tipa *double*, moguće su sve varijante poziva za različite tipove argumenata:

```
complex operator+ (complex c1, complex c2) {...}  
complex::complex (double d) {...}
```

```
complex c1(3.,4.); double d = 5.0; int i = 3;  
complex c2 = d+c1, c3 = c2+i, c4 = c2+c3;
```

Preklopljeni operatori

❖ Zbog toga je uobičajena praksa da se operatori preklapaju na sledeći način:

- operatori kod kojih operandi nisu na neki način ravnopravni, simetrični, prvenstveno zbog toga što dati operator ima bočni efekat, odnosno menja svoj operand, definišu se kao nestatičke funkcije članice; na primer, iako operatori složene dodele (`+=` i ostali) ili operatori `++` i `--` mogu, što se tiče jezika, da se definišu i kao funkcije nečlanice, obično se definišu kao nestatičke funkcije članice jer menjaju svoj (levi) operand, pa zato zahtevaju to da taj operand bude baš objekat nad kojim deluju, a ne neka konvertovana vrednost (privremeni objekat):

```
complex complex::operator+= (complex c) {...}
```

```
complex c1(3.,4.), c2;  
c2 += c1;
```

- operatori kod kojih su operandi na neki način ravnopravni (simetrični), odnosno koji ne menjaju svoje operande (npr. binarni aritmetički i relacioni operatori, operatori koji rade sa bitima itd.), preklapaju se kao funkcije nečlanice:

```
complex operator+ (complex c1, complex c2) {...}  
complex::complex (double d) {...}
```

```
complex c1(3.,4.); double d = 5.0; int i = 3;  
complex c2 = d+c1, c3 = c2+i, c4 = c2+c3;
```