

Semantika vrednosti od C++17

- ❖ Do verzije C++17, optimizacija izostavljanja kopiranja bila je neobavezna i važilo je *as-if* pravilo: čak i kada se kopiranje izbegava, odgovarajući konstruktor kopije ili premeštanja morao je da bude definisan za navedene situacije inicijalizacije privremenim objektima, čak i kada je sigurno da se ti konstruktori nikada ne pozivaju (jer prevodilac uvek vrši ovu optimizaciju i programer to može da zna)
- ❖ Zato su morali da se prave prestupi i definišu ovakvi konstruktori i za klase koje prirodno ne treba da budu takve da se njihovi objekti mogu kopirati ili premeštati, ili su pravljena drugačija zaobilazna rešenja, ako se želi prenos po vrednosti
- ❖ Počev od verzije C++17, optimizacija izbegavanja kopiranja (*copy elision*) obavezna je (a ne samo opcionalna) na sledećim mestima:
 - u naredbi *return*, kada je operand ove naredbe čdvrednost (*prvalue*) istog klasnog tipa (ignorišući cv-kvalifikacije) kao i povratni tip funkcije (RVO):

```
X f () {  
    return X();  
}  
f();
```

Samo jedan poziv podrazumevanog konstruktora X()

- pri inicijalizaciji varijable, kada je inicijalizator čdvrednost (*prvalue*) istog klasnog tipa (ignorišući cv-kvalifikacije) kao i varijabla:

```
X x = X(X(f()));
```

Samo jedan poziv podrazumevanog konstruktora X() koji inicijalizuje x

- ❖ Odgovarajući konstruktori kopije i premeštanja čak više ne moraju ni da postoje niti da budu dostupni, jer semantika jezika garantuje da se oni ne pozivaju na ovim mestima
- ❖ Optimizacija NRVO je i dalje opcionalna: ako se ona ne vrši, poziva se odgovarajući konstruktor premeštanja ili kopije prilikom inicijalizacije povratne vrednosti ako je operand naredbe *return* imenovana automatska varijabla (ali ne parametar)

Semantika vrednosti od C++17

- ❖ Zapravo je formalna semantika povratnih vrednosti poziva funkcija (uključujući i rezultate operatorskih funkcija koje se pozivaju za preklopljene operatore) značajno promenjena od verzije C++17: ako funkcija vraća rezultat koji nije referenca, taj rezultat *nije privremeni objekat*, kao što je to ranije bilo, nego samo *vrednost (value)*, i to *čdvrednost (prvalue)*
- ❖ Čdvrednost predstavlja “potrebu za inicijalizacijom”, ali ne obavezno privremenog objekta (koja se opciono može i preskočiti): ta inicijalizacija obaviće se onako kako je definisano operandom naredbe *return* koja vraća tu vrednost, a obaviće se u zavisnosti od konteksta u kom se koristi
- ❖ Iz funkcija se tako, formalno, vraćaju samo vrednosti i one se dalje prosleđuju kao argumenti / operandi ili inicijalizatori dalje u izrazima ili inicijalizacijama
- ❖ Privremeni objekat se pravi samo u određenim situacijama, odnosno u zavisnosti od konteksta, kada se ta vrednost *materijalizuje* u privremeni objekat (*temporary materialization*)
- ❖ Materijalizacija privremenog objekta se maksimalno odlaže do trenutka kada se taj objekat mora napraviti, a sve do tada se samo koristi vrednost