

# Kopiranje objekata

- ❖ Kada je privremeni objekat kojim se vrši inicijalizacija operand naredbe *return*, ova ista optimizacija izostavljanja kopiranja naziva se *optimizacija povratne vrednosti* (*return value optimization, RVO*); na primer:

```
string concat (const string& s1, const string& s1) {  
    return s1+s2;  
}
```

RVO: rezultat izraza iza *return* biće konstruisan u prostoru privremenog objekta koji ova funkcija vraća, pa neće biti pozivan konstruktor kopije

- ❖ Moguće su i ovakve višestruke, vezane optimizacije koje izbegavaju višestruka kopiranja
- ❖ U opštem slučaju, ukratko, ako je *X* klasni tip, navedene optimizacije izgledaju ovako:

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

RVO: samo jedan jedini poziv podrazumevanog konstruktora *X()* koji inicijalizuje privremeni objekat koji je rezultat konačnog izraza *f()*

```
f();
```

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

Samo jedan jedini poziv podrazumevanog konstruktora *X()* koji inicijalizuje objekat *x*

- ❖ Do verzije jezika C++17, ovo su bile opcione (neobavezne) optimizacije, iako ih većina prevodilaca sprovodi; od verzije C++17 one su obavezne (uvek se sprovode)
- ❖ Programi čija semantika zavisi od toga da li se ove optimizacije sprovode ili ne, odnosno u kojima postoje bočni efekti konstruktora kopije i operatora dodele kopiranjem nisu dobri

# Kopiranje objekata

- ❖ Još jedna varijanta izbegavanja kopiranja jeste i dalje opciona (i u verziji C++17) i naziva se *optimizacija imenovane povratne vrednosti* (*named return value optimization, NRVO*): ako je operand naredbe *return* ime (ali ne nepostojanog) automatskog objekta koji nije parametar te funkcije ili parametar *catch* bloka, i koji je istog tipa kao povratni tip funkcije (uz ignorisanje cv-kvalifikacije), zapravo se taj automatski objekat izgrađuje (inicijalizuje) u prostoru objekta koji se vraća iz funkcije; sve operacije nad tim automatskim objektom vrše se u tom memorijskom prostoru. Na primer:

```
string join (const string& s1, const string& s2) {  
    string s = s1 + " " + s2;  
    return s;  
}
```

- ❖ U opštijem slučaju:

```
T f () {  
    T t;  
    ...  
    return t;  
}
```

- ❖ Ova optimizacija implementira se na isti opisani način: funkcija prima kao skriveni parametar adresu prostora u koji treba da vrati rezultat; pošto na osnovu koda tela funkcije prevodilac može lako da zaključi da se iza *return* imenuje automatski objekat, sve operacije nad tim objektom (uključujući i poziv konstruktora) može da usmeri na taj memorijski prostor, odnosno tu adresu smatra adresom tog automatskog objekta