

Statički životni vek

- ❖ Za ovakve nelokalne varijable koje se inicijalizuju i dinamičkom inicijalizacijom, tačan trenutak izvršavanja te dinamičke inicijalizacije nije precizno određen: ona može, ali ne mora biti izvršena pre početka izvršavanja funkcije *main*, jer prevodilac ne može to uvek da obezbedi (zbog nezavisnog prevođenja fajlova)
- ❖ Jedino što se garantuje jeste to da su statičke varijable propisno inicijalizovane pre nego što se na bilo koji način pristupi bilo kojoj statičkoj varijabli definisanoj u istoj jedinici prevođenja
- ❖ Zbog ovoga, korišćenje nelokalnih statičkih objekata klasa (statičkih podataka članova ili statičkih objekata unutar prostora imena) nije bezbedno: ukoliko se njima pristupa na mestu izvan fajla u kom su definisani, može se dogoditi da se taj pristup radi pre nego što su oni propisno inicijalizovani
- ❖ Zato se ne preporučuje definisanje statičkih objekata klasa koji nisu lokalni, posebno ako njihovi konstruktori imaju neke vidljive efekte. Na primer:

```
// File: a.cpp  
#include "a.h"
```

```
A sa;  
A::A () : m(true) {}
```

sa je globalni statički objekat; vreme njegove inicijalizacije nije precizno određeno

```
// File: b.cpp  
#include "a.h"
```

```
bool f (A* pa) { return pa->m; }  
int main () {  
    if (f(&sa)) ...  
}
```

Kada se ovo izvršava, *sa* možda nije uopšte inicijalizovan, pa će *f* možda vratiti *true*, a možda i *false*

Fajl *a.h* sadrži samo sledeće:

```
struct A {  
    bool m;  
    A();  
};  
extern A sa;
```

Statički životni vek

- ❖ Sa druge strane, inicijalizacija lokalnih statičkih varijabli je precizno definisana: takva varijabla inicijalizuje se kada kontrola toka prvi put naiđe na njenu definiciju; ako više niti nailazi na tu definiciju, samo prva će izvršiti inicijalizaciju. Svako naredno izvršavanje preskače (ignoriše) tu definiciju
- ❖ Ukoliko izvršavanje nikada ne dođe do ovakve definicije, statički objekat neće biti ni inicijalizovan (pa stoga ni uništen pozivom destruktora), na primer ako se funkcija ne pozove ili ne izvrši grana u kojoj je definicija
- ❖ S obzirom na svoj statički životni vek, statičke lokalne varijable nadživljavaju izvršavanje bloka (pa i poziv funkcije) u kom su deklarisanе, iako imaju lokalni opseg važenja, pa nisu dostupne van te oblasti. Na primer:

```
int a = 1;

void f () {
    int b = 1;

    static int c=1;

    cout<<" a = "<<a++;
    cout<<" b = "<<b++;
    cout<<" c = "<<c++<<'\\n' ;

}

int main () {
    while (a<4) f();
}
```