

Inicijalizacija

❖ Statička inicijalizacija radi sledeće:

- Ukoliko je dozvoljeno, vrši se *konstantna inicijalizacija* (*constant initialization*); u praksi, ova inicijalizacija obavlja se za vreme prevođenja tako što prevodilac izračunava konstantne izraze koji su inicijalizatori, a izračunate vrednosti upisuju se u prostor alociran za varijable u prevedenim fajlovima; na primer:

```
const int n = 5;  
int x = 2*n+1, *p = &x;  
size_t s = sizeof(X);
```

- U onim slučajevima u kojima se ne vrši konstantna inicijalizacija, vrši se *inicijalizacija nulom* (*zero initialization*); u praksi, za ove varijable ne odvajaju se prostor u prevedenim fajlovima, već program u izvršavanju koristi uslugu operativnog sistema koja alocira segment memorije inicijalizovan nulama; na primer:

```
int n, *p;
```

- ❖ Dinamička inicijalizacija vrši se po redosledu koji određuju specifična pravila jezika; za nelokalne varijable koje ne spadaju u neke posebne kategorije (npr. statički podaci članovi šablonskih klasa ili *inline* varijable pod određenim uslovima), redosled je određen redosledom definisanja u jednoj jedinici prevođenja
- ❖ Ako dinamička inicijalizacija ne menja vrednost nijednog objekta iz prostora imena pre njegove inicijalizacije, i ako bi statička inicijalizacija proizvela isti rezultat kao i dinamička, prevodiocu je dozvoljeno da dinamičku inicijalizaciju obavi kao statičku, zapravo za vreme prevođenja (rana dinamička inicijalizacija, *early dynamic initialization*)
- ❖ Kao što je već objašnjeno, dinamička inicijalizacija ne mora biti završena pre početka izvršavanja funkcije *main*
- ❖ Ako dinamička inicijalizacija baci izuzetak, poziva se funkcija *std::terminate*

Inicijalizacija

- ❖ Inicijalizacija lokalne statičke varijable vrši se kada bilo koje izvršavanje (u bilo kojoj niti) prvi put naiđe na njenu definiciju; svaki sledeći nailazak preskače inicijalizaciju
- ❖ Ako ova inicijalizacija baci izuzetak, statička varijabla se ne smatra inicijalizovanom i inicijalizacija će biti pokušana pri sledećem nailasku izvršavanja na istu definiciju
- ❖ Nestatički podatak član može biti inicijalizovan na dva načina:
 - u listi inicijalizatora članova (*member initializer list*) u konstruktoru klase:

```
X::X (int j) : i(j+1) {}
```

- podrazumevanim inicijalizatorom člana (*default member initializer*) u definiciji klase, koji se koristi ako se član izostavi iz liste inicijalizatora članova u konstruktoru; ako se član sa podrazumevanim inicijalizatorom pojavi i u listi inicijalizatora članova, ta podrazumevana inicijalizacija se ignoriše:

```
struct X {  
    int i = 1;  
    string s{'H', 'e', 'l', 'l', 'o'};  
  
    X () {}  
  
    X (int j) : i(j+1) {}  
};
```