

Destruktor

- ❖ Destruktor (*destructor*) je posebna nestatička funkcija članica klase koja se poziva uvek na kraju životnog veka objekta. Svrha destruktora je da oslobodi resurse koje je objekat eventualno zauzimao tokom svog životnog veka
- ❖ Destruktor se deklariše kao funkcija članica klase sa imenom te klase i znakom ~ ispred imena klase:

```
struct X {  
    ~X ();  
    ...  
};
```

Destruktor

- ❖ Destruktor nikada nema parametre, pa klasa ima najviše jedan destruktor (ne može se preklopiti). Destruktor nema povratni tip. Destruktor ima pokazivač *this*, kao i svaka nestatička funkcija članica
- ❖ Destruktor se poziva implicitno uvek na kraju životnog veka objekta, bez obzira na taj životni vek:
 - na kraju izvršavanja programa, za statičke objekte
 - po završetku izvršavanja niti, za objekte sa životnim vekom vezanim za nit (*thread local*)
 - po izlasku iz opsega važenja bloka, za automatske objekte napravljene u tom bloku i privremene objekte čiji je životni vek produžen jer je za njih vezana referenca
 - izrazom *delete*, za objekte sa dinamičkim životnim vekom
 - završetkom celog izraza, za privremene objekte napravljene u tom izrazu
 - prilikom razmotavanja steka kod bačenog izuzetka, za sve automatske objekte koji su napravljeni, a čiji se blokovi napuštaju do hvatanja izuetka
- ❖ Destruktor se može pozvati i eksplicitno, za objekte koji su napravljeni pomoću operacije *placement new*. Ako se destruktor pozove eksplicitno za objekat čiji se destruktor kasnije poziva i implicitno, ponašanje je nedefinisano

Destruktor

- ❖ Ako klasa X nema eksplicitno deklarisan destruktor (tj. korisnički definisan destruktor), prevodilac će implicitno deklarirati destruktor koji je javan i *inline*
- ❖ Programer može zahtevati automatsko generisanje destruktora koji bi prevodilac implicitno deklarirao specifikatorom *=default*
- ❖ Ako iz bilo kog razloga nije moguće uništavanje podobjekata osnovnih klasa i objekata članova (npr. jer su im destruktori nedostupni), destruktor se smatra obrisanim (nije moguć njegov implicitan ili eksplicitan poziv)
- ❖ Ako implicitno deklarisan ili podrazumevani destruktor nije obrisao, prevodilac generiše njegovu definiciju sa praznim telom
- ❖ Kada se uništava neki objekat, najpre se izvršava telo destruktora, a nakon toga se implicitno pozivaju destruktori objekata članova (ako su objekti klasa) i destruktori osnovnih klasa, uvek po redosledu obrnutom od redosleda inicijalizacije