

---

# Operatori *new* i *delete*

---

❖ Kao što je već objašnjeno, izraz oblika

**new** (placement\_params) T (init)

uvek radi sledeće tri stvari:

1. Poziva neku od ugrađenih globalnih alokatorskih funkcija oblika

**void\* ::operator new** (size\_t size, placement\_params)

**void\* ::operator new []** (size\_t size, placement\_params)

2. Inicijalizuje objekat odgovarajućom inicijalizacijom

3. Vraća pokazivač na napravljeni dinamički objekat

❖ Ovaj postupak je uvek isti i ne može se promeniti, u smislu skupa i redosleda koraka, ali se poziv alokatorske funkcije u prvom koraku može preusmeriti na drugu funkciju, pa i onu definisanu posebno za neku klasu preklapanjem alokatorske funkcije *new* za tu klasu

❖ U prvom koraku, ugrađene alokatorske funkcije *::new* podrazumevano rade sa slobodnim delom memorije u kom pronalaze slobodan prostor tražene veličine *size* i alociraju ga

❖ Ove funkcije mogu da se zamene drugim, korisnički definisanim, sa istim potpisom; dovoljno je definisati neku od njih u nekom fajlu

# Operatori *new* i *delete*

- ❖ Posmatrajmo izgled slobodnog prostora tokom sukcesivnih operacija alokacije i dealokacije delova memorije različite veličine: kada se dealocira neki dinamički objekat, u oslobođeni prostor njegove veličine može se alocirati nov dinamički objekat koji je najčešće manji (može biti i jednak, što je manje verovanto, ali nikako veći) od tog dela; tako ostaje *fragment* slobodne memorije koji možda nije dovoljan da se u njega smesti bilo koji nov objekat
- ❖ Posle dužeg rada programa može se dogoditi to da je slobodna memorija rascepkana (fragmentirana) na male slobodne delove, pa iako je ukupno ima sasvim dovoljno, nijedan slobodan deo nije dovoljno velik da bi se u njega smestio traženi objekat; tada program više ne može alocirati prostor za traženi novi dinamički objekat i više ne radi regularno
- ❖ Ovakav problem naziva se *fragmentacija memorije* (*memory fragmentation*)
- ❖ Ovaj problem nije primetan ni značajan kod programa koji rade neko vreme i onda se gase, npr. kod interaktivnih uslužnih programa, ali kod programa koji rade neprekidno, npr. onih ugrađenih u hardver koji radi neprekidno (*embedded software*) može da bude veoma ozbiljan