

Hijerarhijska dekompozicija

- ❖ Treba primetiti da konverzija pokazivača na izvedenu klasu u pokazivač na osnovnu klasu daje kao rezultat pokazivač iste vrednosti u slučaju jednostrukog izvođenja klasa, ali može da daje promenjenu vrednost u slučaju višestrukog izvođenja; analogno važi i za konverziju nadole:

```
class D1 : public B2 {...};
```

```
class D2 : public B1, public B2 {...};
```

- ❖ Kao i uvek, eksplicitna statička konverzija nadole ne proverava činjenicu da li se iza pokazivača zaista krije objekat tražene klase, pa stoga nije bezbedna iz ugla prevodioca, osim ako se to ne obezbedi nekim drugim načinom, odnosno logikom programa

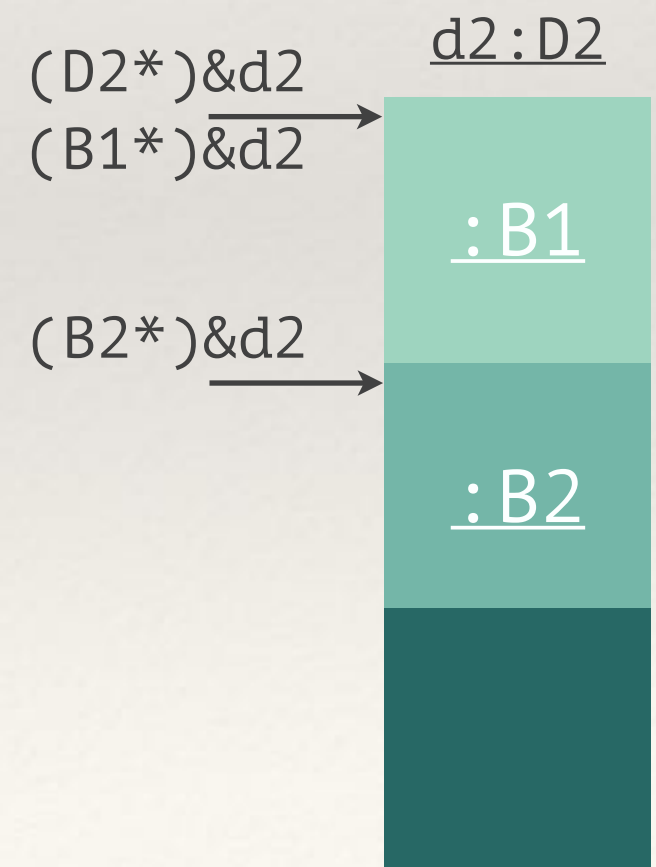
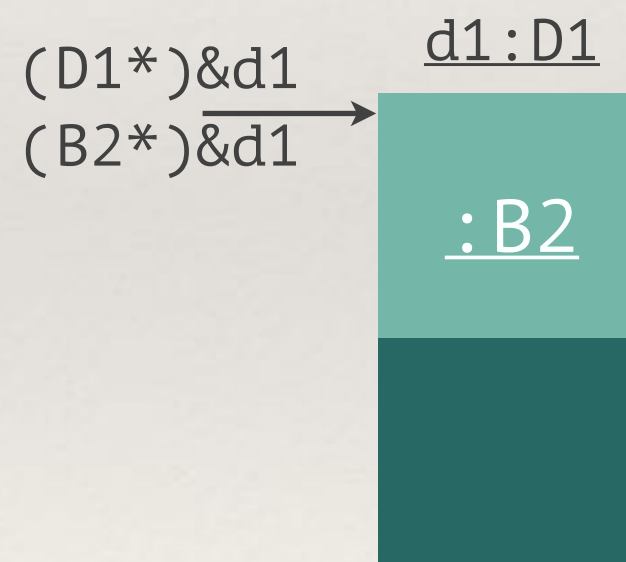
- ❖ Dinamička konverzija je bezbednija, jer dinamički proverava ovo, i vraća *null* vrednost ako objekat nije tražene klase. Ona se može koristiti za konverzije nadole, nagore, ili bočno:

```
D2* d = new D2;
```

```
B1* b1 = d;
```

```
B2* b2 = dynamic_cast<B2*>(b1);
```

U slučaju da pokazivač *b1* ne ukazuje na objekat tipa *B2*, vratiće *null* vrednost



Hijerarhijska dekompozicija

- ❖ U skupovnoj logici, osnovna klasa (generalizacija) predstavlja nadskup skupa objekata njene izvedene klase (specijalizacije). Ali šta ako je osnovna klasa apstraktna?
- ❖ Pošto ona tada nema svoje direktne instance, ne postoje elementi tog skupa koji nisu ujedno i elementi skupa predstavljenog nekom od izvedenih klasa. Prema tome, apstraktna klasa predstavlja *uniju* skupova predstavljenih izvedenim klasama
- ❖ A šta ako izvedene klase, posmatrane kao skupovi, imaju presek, odnosno zajedničke elemente?
- ❖ Ovo se obično dešava ako se generalizacije / specijalizacije prave po različitim, ortogonalnim kriterijumima
- ❖ Na primer, u nekom školskom sistemu, *korisnici* (*User*) mogu biti klasifikovani po pravima pristupa na *privilegovane* (*Privileged User*) i *neprivilegovane* (*NonprivilegedUser*); sa druge strane, mogu se klasifikovati prema svojoj ulozi (*role*) na *nastavnike* (*Teacher*) i *učenike* (*Student*)
- ❖ Na jeziku UML, ovakve različite grupe relacija generalizacija / specijalizacija nazivaju se *skupovima generalizacija* (*generalization set*)
- ❖ Ako su izvedene klase kao skupovi disjunktne, tj. nemaju presek (zajedničke instance), skup generalizacija naziva se *isključiv* (*exclusive*); ako je takva osnovna klasa apstraktna, ona predstavlja uniju disjunktne podskupova - *particiju* (*partition*)

