# Hijerarhijska dekompozicija

- Međutim, izvedene klase iz različitih skupova generalizacija često imaju zajedničke instance, odnosno presek. Na primer, jedan nastavnik može biti privilegovan korisnik
- U nekim jezicima, kao što je UML, objekat može biti instanca više klasa (koje nisu u relaciji generalizacije/specijalizacije). Štaviše, objekat se može *dinamički reklasifikovati* (*reclassify*) tokom svog životnog veka: mogu mu se dodavati i oduzimati klase kojima pripada (time se dodaju ili oduzimaju sva svojstva tih klasa)
- U tradicionalnim, statički tipiziranim OO programskim jezicima, kakav je i C++, ovo nije podržano, pa objekat uvek mora biti direktna instanca jedne i samo jedne klase. Ta klasa se uvek mora odrediti u trenutku kreiranja tog objekta i objekat se ne može reklasifikovati tokom svog životnog veka
- Kako onda rešiti ovakvu situaciju? Višestrukim izvođenjem nove klase koja predstavlja presek dve osnovne:

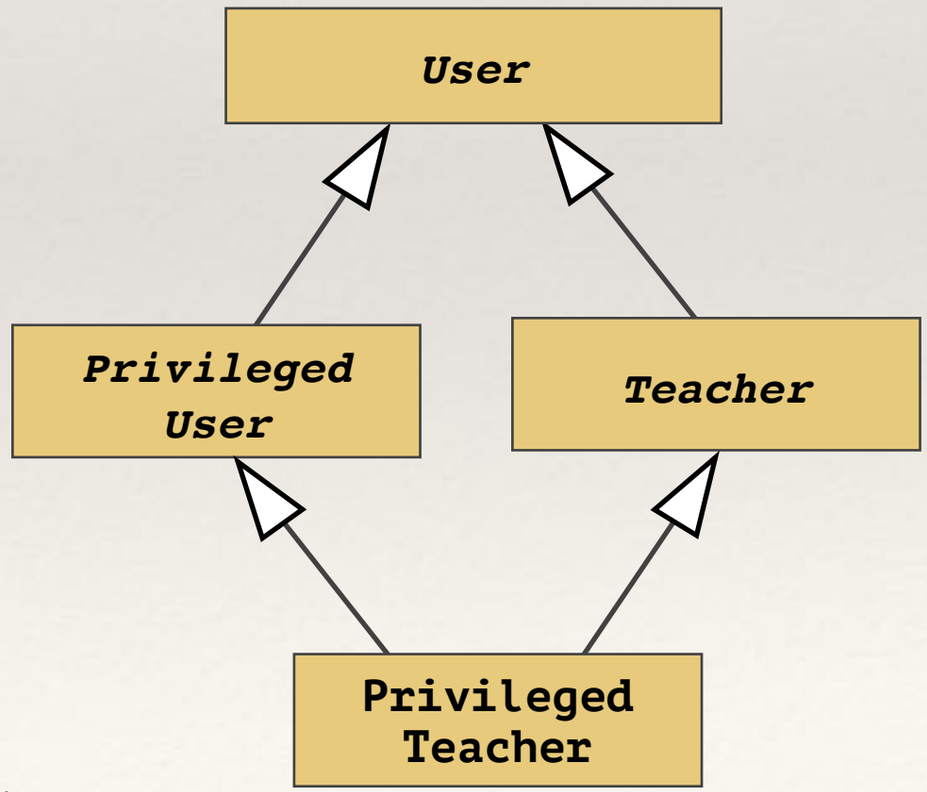
```
class User {…};

class PrivilegedUser : public User {…};
class NonprivilegedUser : public User {…};

class Teacher : public User {…};
class Student : public User {…};

class PrivilegedTeacher : public PrivilegedUser, public Teacher {…};
```


Copyright 2018-2024 by Dragan Milićev

# Hijerarhijska dekompozicija

- Međutim, kako C++ *uvek* implementira objekte izvedene klase tako da u njih ugrađuje po jedan objekat svake osnovne klase (i tako rekurzivno), sledi:
  - objekat klase *PrivilegedTeacher* u sebi ima podobjekat klase *PrivilegedUser*, a ovaj u sebi podobjekat klase *User*
  - objekat klase *PrivilegedTeacher* u sebi ima podobjekat klase *Teacher*, a ovaj u sebi podobjekat klase *User*
- Tako će objekat klase *PrivilegedTeacher* u sebi imati dva podobjekta osnovne klase *User*, što znači dva kompleta svih svojstava, što nije poželjno. Ova pojava naziva se "problem dijamanta" (*diamond problem*), zbog grafa nasleđivanja oblika romba (liči na dijamant i često se tako naziva na engleskom)
- Ovim podobjektima može se pristupiti navođenjem kvalifikovanog imena (staze) i operatora ::

```
this->PrivilegedUser::username = …;
this->Teacher::username = …;
```





Copyright 2018-2024 by Dragan Milićev