

# Opseg važenja klase

❖ Ime iz oblasti važenja klase može se koristiti samo na jedan od sledećih načina:

- Unutar oblasti važenja iste klase ili oblasti važenja izvedenih klasa:

```
DatabaseManager::DatabaseManager(const char* nm) {  
    ...  
    name = new char[sizeof(nm)+1];  
    ...  
}
```

Ime *name* je u opsegu važenja

- Kao desni operand operatora `.` kom je levi operand objekat te klase ili klase izvedene iz te klase:

```
DatabaseManager* p = new DatabaseManager(...);  
...  
(*p).openConnection();
```

- Kao desni operand operatora `->` kom je levi operand pokazivač na objekat te klase ili klase izvedene iz te klase:

```
p->openConnection();
```

# Opseg važenja klase

- Kao desni operand operatora `::` kom je levi operand ta klasa ili klasa izvedena iz te klase. Na primer, čest je slučaj da redefinisana operacija u izvedenoj klasi treba da uradi isto što i metoda osnovne klase, ali i još nešto (u bilo kom redosledu); drugim rečima, da treba da pozove metodu iz osnovne klase:

```
class Base {
public:
    virtual void f();
};

class Derived : public Base {
public:
    virtual void f();
};

void Derived::f () {
    ...
    Base::f();
    ...
}
```

- ❖ Operator `::`, koji je inače asocijativan sleva nadesno, navodi da prevodilac vrši *kvalifikovanu potragu* (*qualified lookup*) na sledeći način: `Base::f` znači da se najpre sprovodi nekvalifikovana potraga za imenom `Base`; pronađe se to ime i njegova deklaracija, odnosno deklaracija klase `Base`; zatim se u oblasti važenja te klase traži deklaracija za ime `f`; ako se ne nađe takva deklaracija, traži se u njenoj osnovnoj klasi, ako je ima itd.