

Opseg važenja bloka

- ❖ I klasa može biti definisana unutar tela funkcije - tzv. *lokalna klasa* (*local class*). Takva klasa nije dostupna van opsega važenja te funkcije i ima neka ograničenja: ne može imati statičke podatke članove, njene funkcije članice moraju biti definisane unutar definicije te klase i drugo
- ❖ Ovakve klase tipično se prave da bi se na licu mesta napravili ad-hoc objekti koji zadovoljavaju neki interfejs i samo redefinišu neke polimorfne operacije, tj. objekti su neke klase izvedene iz neke apstraktne klase, s tim da su takve pojave bitne samo na datom mestu, a ne šire, odnosno takvi objekti se prave samo tu, dok ih svi ostali vide samo kroz te apstraktne, generalizovane interfejse, a nikad kao instance te lokalne klase (pa je zato ta klasa i lokalna, enkapsulirana u funkciju)
- ❖ Na primer, neka funkcija očekuje kao parametar objekat koji zadovoljava neki interfejs, kako bi pozvala neku njegovu operaciju (tzv. *callback* mehanizam):

```
class ISubscriber {  
public:  
    virtual void notify (Message*);  
};  
  
class Publisher {  
public:  
    static void subscribe (ISubscriber*);  
};  
  
void Component::Component () {  
    struct Sub : public ISubscriber {  
        Sub (Component* c) : comp(c) {}  
        virtual void notify (Message* msg) { if (comp) comp->processMessage(msg); }  
        Component* comp;  
    };  
    Publisher::subscribe(new Sub(this));  
}
```

Da bi se neko prijavio da prima objave (notifikacije), mora da zadovolji ovaj interfejs, tj. (re)definiše operaciju *notify*

Da bi se neko prijavio da prima objave (notifikacije), mora da se prijavi ovom “oglašivaču” operacijom *subscribe*

Da bi se komponenta prijavila “oglašivaču”, pravi jedan ad-hoc objekat lokalne klase *Sub*, koja služi samo da redefiniše polimorfnu operaciju *notify*, kao i da sačuva pokazivač na komponentu kojoj treba da prosledi poruku kad je primi objavom

Opseg važenja klase

- ❖ Oblast važenja imena deklarisanog unutar definicije klase je od tačke deklaracije do ostatka definicije te klase, ali se proteže i kroz tela svih funkcija članica te klase i izvedenih klasa, čak i ako su ta tela navedena van definicije klase ili pre date deklaracije
- ❖ Unutar definicije klase mogu se navoditi deklaracije ne samo objekata i funkcija, nego i tipova (*typedef* sinonimi, enumeracije, klase itd.); na primer:

```
class DatabaseManager {  
public:  
    enum DBStatus { ok, failed, refused };  
    DatabaseManager (const char* name);  
    DBStatus openConnection();  
    DBStatus closeConnection();  
    DBStatus getConnectionStatus();  
    DBStatus performQuery (const char* sqlQuery);  
  
private:  
    char* name;  
    ...  
};
```