



BUILDING COMPOSITION: WHOLE-PART BY EMBEDDING

The whole creates and passes data to its parts



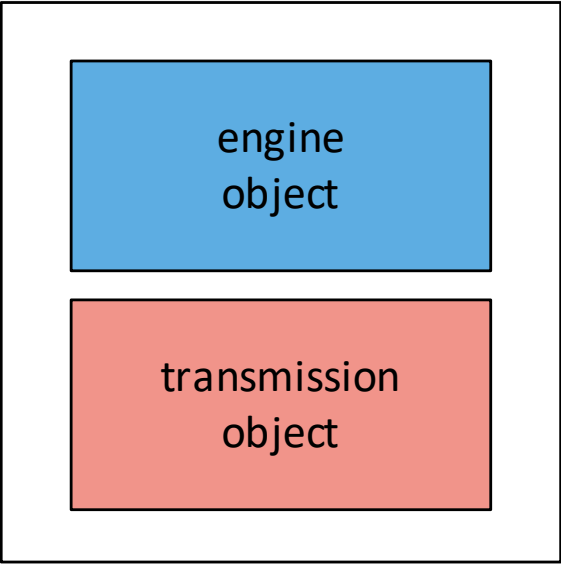
ALLOCATING MEMORY FOR WHOLE OBJECTS

```
class engine // part
{
};

class transmission // part
{
};

class car // whole
{
private:
    engine cars_engine;
    transmission cars_trans;
};
```

```
car my_car;
```



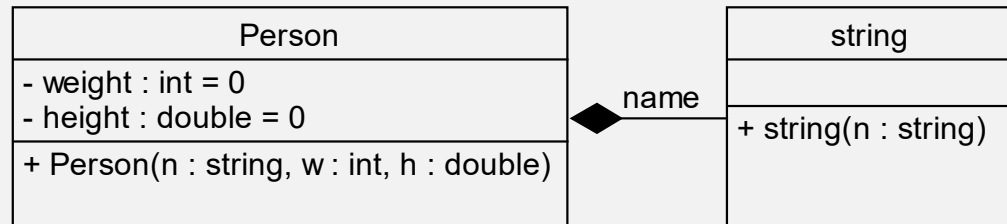
car object

COMPOSITION WITH DEFAULT CONSTRUCTORS

| Person |
|-----------------------|
| - name : string |
| - weight : int = 0 |
| - height : double = 0 |
| + Person() |

```
class Person // whole
{
    private:
        string name; // part
        int weight;
        double height;
    public:
        Person()
            : weight(0), height(0) {}
};
```

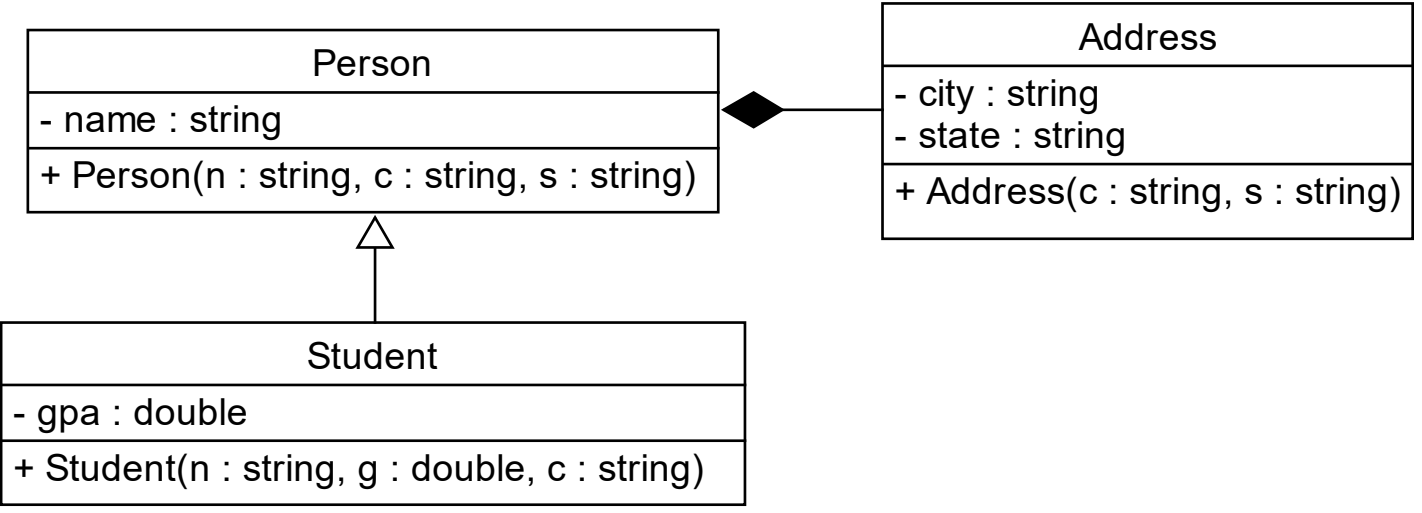
COMPOSITION WITHOUT A DEFAULT CONSTRUCTOR



```
class Person // whole
{
    private:
        string name; // part
        int weight;
        double height;
    public:
        Person(string n, int w, double h)
            : name(n), weight(), height(h) {}
};
```

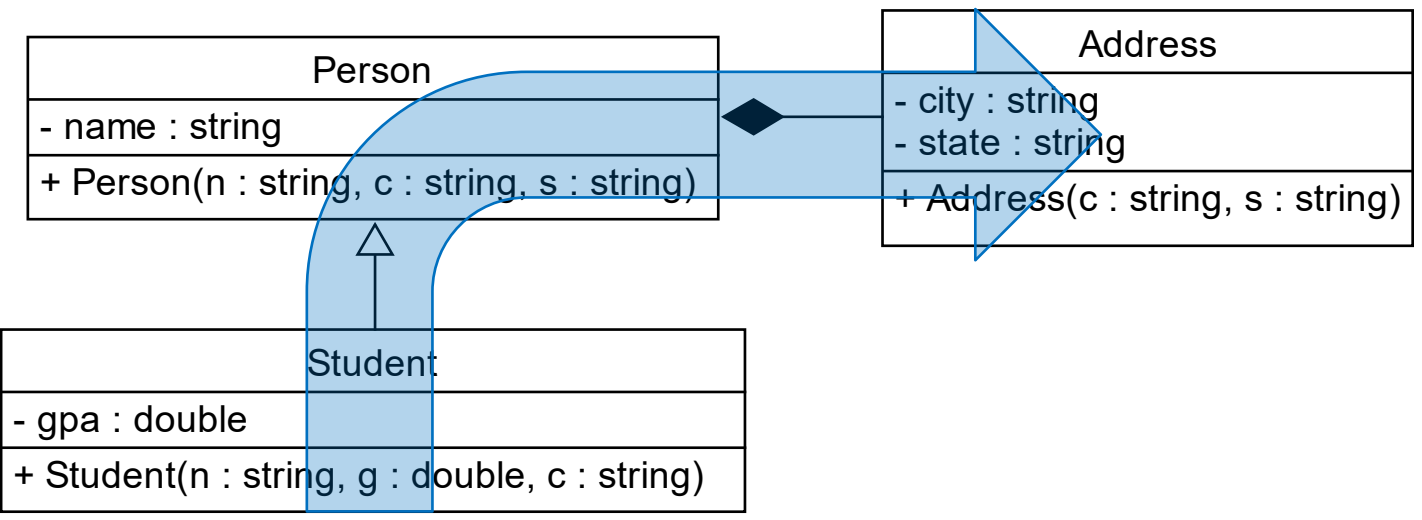


INHERITANCE & WHOLE-PART (I)





INHERITANCE & WHOLE-PART (I)





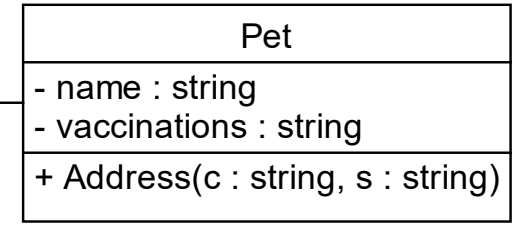
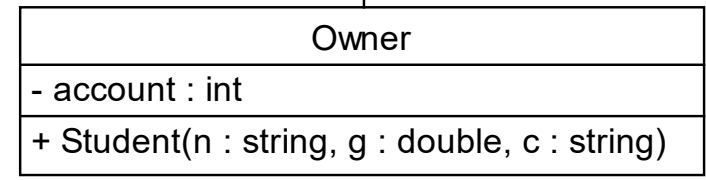
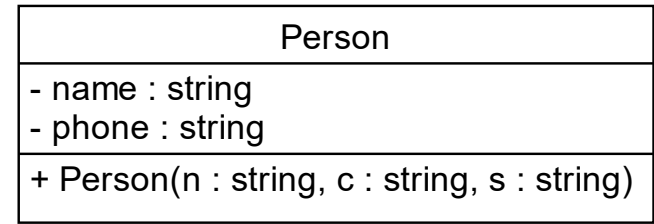
INHERITANCE & WHOLE-PART (I) CONTINUED

```
class Person // superclass and whole
{
    private:
        string name;
        Address addr;
    public:
        Person(string n, string c, string s) : addr(c, s), name(n) {}
};

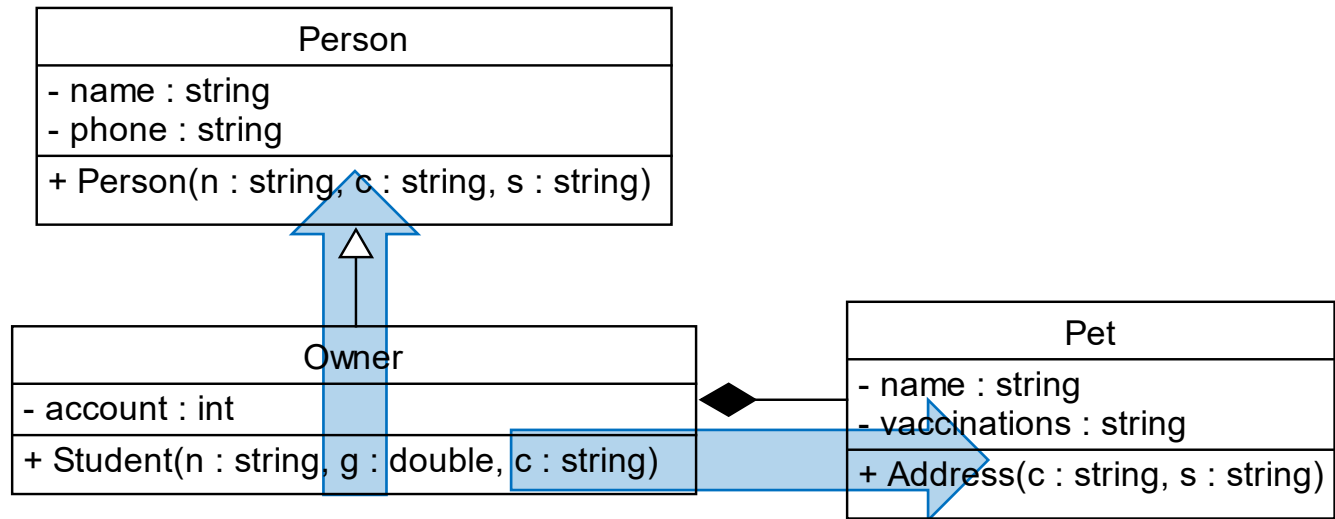
class Student : public Person // subclass
{
    private:
        double gpa;
    public:
        Student(string n, double g, string c, string s) : Person(n, c, s), gpa(g) {}
}
```



INHERITANCE & WHOLE-PART (2)



INHERITANCE & WHOLE-PART (2)



INHERITANCE & WHOLE/PART (2) CONTINUED

```
class Person
{
    private:
        string name;
        string phone;
    public:
        Person(string n, string p)
            : name(n), phone(p) {}
};

class Owner : public Person
{
    private:
        int account;
        Pet my_pet;
    public:
        Owner(string n, string p, int a, string pn, string v) : Person(n, p), my_pet(pn, v), account(a) {}
};

class Pet
{
    private:
        string name;
        string vaccinations;
    public:
        Pet(string n, string v)
            : name(n), vaccinations(v) {}
};
```