



operator<< AND operator>> WITH INHERITANCE

Chaining I/O operators with inheritance hierarchies



REVIEWING THE INSERTER PATTERN: THE Person CLASS INSERTER

```
class Person
{
    private:
        name : string;
    public:
        friend ostream& operator<<(ostream& out, Person& me)
        {
            out << me.name << endl;
            return out;
        }
};
```

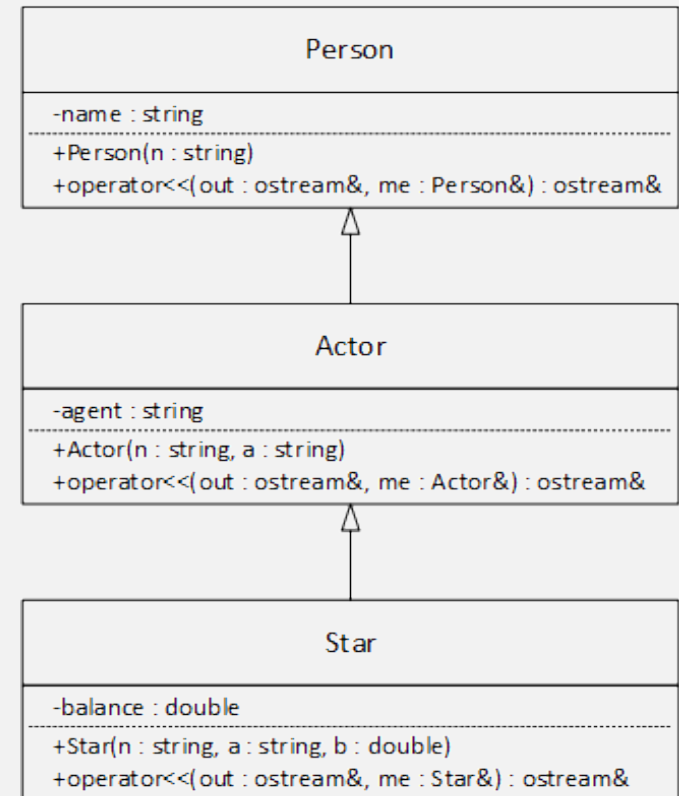
FUNCTION CHAINING: ACTOR TO PERSON

```
friend ostream& operator<<(ostream& out, Person& me)
{
    out << me.name << endl;
    return out;
}
```

```
friend ostream& operator<<(ostream& out, Actor& me)
{
    out << (Person &)me << " " << me.agent << endl;
    return out;
}
```

. . .

```
Actor a;
cout << a;
```



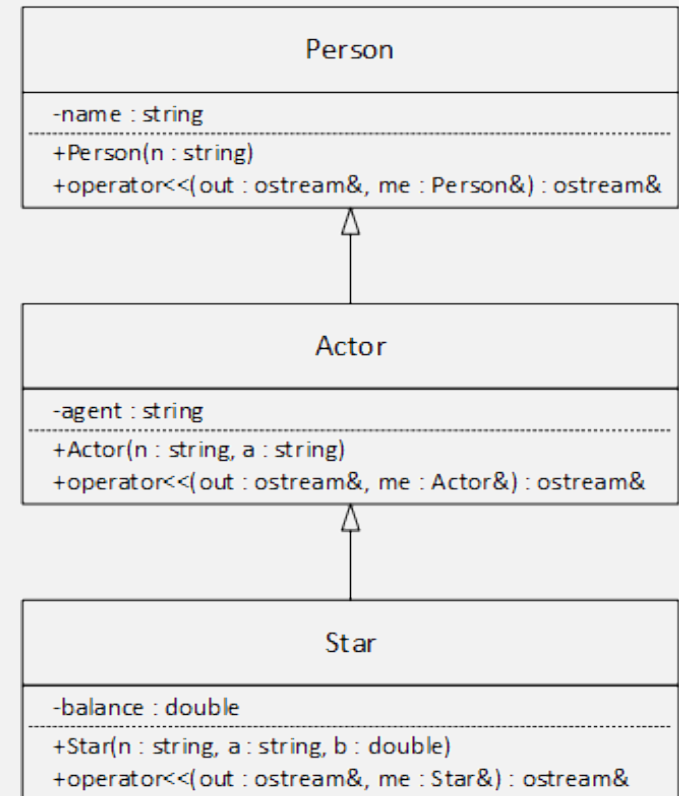
FUNCTION CHAINING: STAR TO ACTOR

```
friend ostream& operator<<(ostream& out, Actor& me)
{
    out << (Person &)me << " " << me.agent << endl;
    return out;
}
```

```
friend ostream& operator<<(ostream& out, Star& me)
{
    out << (Actor &)me << " " << me.balance << endl;
    return out;
}
```

. . .

```
Star s;
cout << s;
```





INHERITANCE AND THE EXTRACTOR

```
friend istream& operator>>(istream& in, Person& me)
{
    getline(in, me.name);
    return in;
}
```

```
friend istream& operator>>(istream& in, Actor& me)
{
    in >> (Person &)me;
    getline(in, me.agent);
    return in;
}
```

```
friend istream& operator>>(istream& in, Star& me)
{
    in >> (Actor &)me;
    in >> me.balance;
    return in;
}
```