【Effective CPP】Day6(2)

Book	Effective C++
Author	
Summary	
Date	@2022/05/13

【Ch2】Constructors, Destructors, and Assignment Operators

Item 12: Copy all parts of an object

The compiler-generated copying functions(copy constructor and copy assignment operator) copy all the data of the object being copied.

However, if we want to define our own copying functions, we need to make sure that we copy all the data members of the object.

For example, we define a `Customer` class like following:

```
void logCall(const std::string& funcName);

class Customer {
public:
  ...
  Customer(const Customer& rhs);
  Customer& operator=(const Cutomer& rhs);

private:
  std::string name;
};
```

And have a class `PriorityCustomer` inherits it:

```
class PriorityCustomer : public Customer {
public:
  PriorityCustomer(const PriorityCustomer& rhs);
  PriorityCustomer& operator=(const PriorityCustomer& rhs);
```

```
private:
  int priority;
};
```

We need to copy both the base parts and the derived parts of the class

```
PriorityCustomer(const PriorityCustomer& rhs) : Customer(rhs), priority(rhs.priority) {
  logCall("PriorityCustomer copy constructor");
}

PriorityCustomer& PriorityCustomer::operator=(const PriorityCustomer& rhs) {
  Customer::operator=(rhs); // Assign base class parts;
  priority = rhs.priority;
  return *this;
}
```

Things to Remember

1. Copying functions should be sure to copy all of an object's data members and all of its base class parts.
2. Don't try to implement one of the copying functions in terms of the other. Instead, put common functionality in a third function that both call.