

11.3.5 CommissionEmployee– BasePlusCommissionEmployee Inheritance Hierarchy Using private Data

- We now reexamine our hierarchy once more, this time using the best software engineering practices.
- Class `CommissionEmployee` now declares data members `firstName`, `lastName`, `socialSecurityNumber`, `grossSales` and `commissionRate` as `private` (as shown previously in lines 31–36 of Fig. 11.4).

11.3.5 CommissionEmployee– BasePlusCommissionEmployee Inheritance Hierarchy Using private Data (cont.)

Changes to Class CommissionEmployee's Member Function Definitions

- In the `CommissionEmployee` constructor implementation (Fig. 11.14, lines 9–16), we use member initializers (line 12) to set the values of members `firstName`, `lastName` and `socialSecurityNumber`.
- We show how derived-class `BasePlusCommissionEmployee` (Fig. 11.15) can invoke non-private base-class member functions (`setFirstName`, `getFirstName`, `setLastName`, `getLastName`, `setSocialSecurityNumber` and `getSocialSecurityNumber`) to manipulate these data members.

```
1 // Fig. 11.14: CommissionEmployee.cpp
2 // Class CommissionEmployee member-function definitions.
3 #include <iostream>
4 #include <stdexcept>
5 #include "CommissionEmployee.h" // CommissionEmployee class definition
6 using namespace std;
7
8 // constructor
9 CommissionEmployee::CommissionEmployee(
10     const string &first, const string &last, const string &ssn,
11     double sales, double rate )
12     : firstName( first ), lastName( last ), socialSecurityNumber( ssn )
13 {
14     setGrossSales( sales ); // validate and store gross sales
15     setCommissionRate( rate ); // validate and store commission rate
16 } // end CommissionEmployee constructor
17
```

Fig. 11.14 | CommissionEmployee class implementation file:
CommissionEmployee class uses member functions to manipulate its private data. (Part I of 6.)

```
18 // set first name
19 void CommissionEmployee::setFirstName( const string &first )
20 {
21     firstName = first; // should validate
22 } // end function setFirstName
23
24 // return first name
25 string CommissionEmployee::getFirstName() const
26 {
27     return firstName;
28 } // end function getFirstName
29
30 // set last name
31 void CommissionEmployee::setLastName( const string &last )
32 {
33     lastName = last; // should validate
34 } // end function setLastName
35
```

Fig. 11.14 | CommissionEmployee class implementation file:
CommissionEmployee class uses member functions to manipulate its private data. (Part 2 of 6.)

```
36 // return last name
37 string CommissionEmployee::getLastName() const
38 {
39     return lastName;
40 } // end function getLastName
41
42 // set social security number
43 void CommissionEmployee::setSocialSecurityNumber( const string &ssn )
44 {
45     socialSecurityNumber = ssn; // should validate
46 } // end function setSocialSecurityNumber
47
48 // return social security number
49 string CommissionEmployee::getSocialSecurityNumber() const
50 {
51     return socialSecurityNumber;
52 } // end function getSocialSecurityNumber
53
```

Fig. 11.14 | CommissionEmployee class implementation file:
CommissionEmployee class uses member functions to manipulate its private data. (Part 3 of 6.)

```
54 // set gross sales amount
55 void CommissionEmployee::setGrossSales( double sales )
56 {
57     if ( sales >= 0.0 )
58         grossSales = sales;
59     else
60         throw invalid_argument( "Gross sales must be >= 0.0" );
61 } // end function setGrossSales
62
63 // return gross sales amount
64 double CommissionEmployee::getGrossSales() const
65 {
66     return grossSales;
67 } // end function getGrossSales
68
```

Fig. 11.14 | CommissionEmployee class implementation file:
CommissionEmployee class uses member functions to manipulate its private data. (Part 4 of 6.)

```
69 // set commission rate
70 void CommissionEmployee::setCommissionRate( double rate )
71 {
72     if ( rate > 0.0 && rate < 1.0 )
73         commissionRate = rate;
74     else
75         throw invalid_argument( "Commission rate must be > 0.0 and < 1.0" );
76 } // end function setCommissionRate
77
78 // return commission rate
79 double CommissionEmployee::getCommissionRate() const
80 {
81     return commissionRate;
82 } // end function getCommissionRate
83
84 // calculate earnings
85 double CommissionEmployee::earnings() const
86 {
87     return getCommissionRate() * getGrossSales();
88 } // end function earnings
89
```

Fig. 11.14 | CommissionEmployee class implementation file:
CommissionEmployee class uses member functions to manipulate its private data. (Part 5 of 6.)

```
90 // print CommissionEmployee object
91 void CommissionEmployee::print() const
92 {
93     cout << "commission employee: "
94         << getFirstName() << ' ' << getLastName()
95         << "\nsocial security number: " << getSocialSecurityNumber()
96         << "\ngross sales: " << getGrossSales()
97         << "\ncommission rate: " << getCommissionRate();
98 } // end function print
```

Fig. 11.14 | CommissionEmployee class implementation file:
CommissionEmployee class uses member functions to manipulate its private data. (Part 6 of 6.)



Performance Tip 11.2

Using a member function to access a data member's value can be slightly slower than accessing the data directly. However, today's optimizing compilers are carefully designed to perform many optimizations implicitly (such as inlining `set` and `get` member-function calls). You should write code that adheres to proper software engineering principles, and leave optimization to the compiler. A good rule is, "Do not second-guess the compiler."

11.3.5 CommissionEmployee– BasePlusCommissionEmployee Inheritance Hierarchy Using private Data (cont.)

Changes to Class

BasePlusCommissionEmployee's Member Function Definitions

- Class `BasePlusCommissionEmployee` has several changes to its member-function implementations (Fig. 11.15) that distinguish it from the previous version of the class (Figs. 11.10–11.11).
- Member functions `earnings` (Fig. 11.15, lines 34–37) and `print` (lines 40–48) each

```
1 // Fig. 11.15: BasePlusCommissionEmployee.cpp
2 // Class BasePlusCommissionEmployee member-function definitions.
3 #include <iostream>
4 #include <stdexcept>
5 #include "BasePlusCommissionEmployee.h"
6 using namespace std;
7
8 // constructor
9 BasePlusCommissionEmployee::BasePlusCommissionEmployee(
10     const string &first, const string &last, const string &ssn,
11     double sales, double rate, double salary )
12     // explicitly call base-class constructor
13     : CommissionEmployee( first, last, ssn, sales, rate )
14 {
15     setBaseSalary( salary ); // validate and store base salary
16 } // end BasePlusCommissionEmployee constructor
17
```

Fig. 11.15 | BasePlusCommissionEmployee class that inherits from class CommissionEmployee but cannot directly access the class's private data. (Part I of 3.)

```
18 // set base salary
19 void BasePlusCommissionEmployee::setBaseSalary( double salary )
20 {
21     if ( salary >= 0.0 )
22         baseSalary = salary;
23     else
24         throw invalid_argument( "Salary must be >= 0.0" );
25 } // end function setBaseSalary
26
27 // return base salary
28 double BasePlusCommissionEmployee::getBaseSalary() const
29 {
30     return baseSalary;
31 } // end function getBaseSalary
32
33 // calculate earnings
34 double BasePlusCommissionEmployee::earnings() const
35 {
36     return getBaseSalary() + CommissionEmployee::earnings();
37 } // end function earnings
38
```

Fig. 11.15 | BasePlusCommissionEmployee class that inherits from class CommissionEmployee but cannot directly access the class's private data. (Part 2 of 3.)

```
39 // print BasePlusCommissionEmployee object
40 void BasePlusCommissionEmployee::print() const
41 {
42     cout << "base-salaried ";
43
44     // invoke CommissionEmployee's print function
45     CommissionEmployee::print();
46
47     cout << "\nbase salary: " << getBaseSalary();
48 } // end function print
```

Fig. 11.15 | BasePlusCommissionEmployee class that inherits from class CommissionEmployee but cannot directly access the class's private data. (Part 3 of 3.)

11.3.5 CommissionEmployee– BasePlusCommissionEmployee Inheritance Hierarchy Using private Data (cont.)

BasePlusCommissionEmployee Member Function earnings

- Class `BasePlusCommissionEmployee`'s `earnings` function (Fig. 11.15, lines 34–37) redefines class `CommissionEmployee`'s `earnings` member function (Fig. 11.14, lines 85–88) to calculate the earnings of a base-salaried commission employee. It also calls `CommissionEmployee`'s `earnings` function.
 - Note the syntax used to invoke a redefined base-class member function from a derived class—place the base-class name and the binary scope resolution operator (`::`) before the base-class member-function name.
 - Good software engineering practice: If an object's member function performs the actions needed by another object, we should call that member function rather than duplicating its code body.



Common Programming Error 11.2

When a base-class member function is redefined in a derived class, the derived-class version often calls the base-class version to do additional work. Failure to use the `::` operator prefixed with the name of the base class when referencing the base class's member function causes infinite recursion, because the derived-class member function would then call itself.