11.3.1 Creating and Using a `CommissionEmployee` Class

- `CommissionEmployee`'s class definition (Figs. 11.4–11.5).
- `CommissionEmployee`'s `public` services include a constructor and member functions `earnings` and `print`.
- Also includes `public` *get* and *set* functions that manipulate the class's data members `firstName`, `lastName`, `socialSecurityNumber`, `grossSales` and `commissionRate`.
  - These data members are `private`, so objects of other classes cannot directly access this data.
  - Declaring data members as `private` and providing non-`private` *get* and *set* functions to manipulate and validate the data members helps enforce good software engineering.

```
1  // Fig. 11.4: CommissionEmployee.h
2  // CommissionEmployee class definition represents a commission employee.
3  #ifndef COMMISSION_H
4  #define COMMISSION_H
5
6  #include <string> // C++ standard string class
7
8  class CommissionEmployee
9  {
10 public:
11    CommissionEmployee( const std::string &, const std::string &,
12       const std::string &, double = 0.0, double = 0.0 );
13
14    void setFirstName( const std::string & ); // set first name
15    std::string getFirstName() const; // return first name
16
17    void setLastName( const std::string & ); // set last name
18    std::string getLastName() const; // return last name
19
20    void setSocialSecurityNumber( const std::string & ); // set SSN
21    std::string getSocialSecurityNumber() const; // return SSN
22
23    void setGrossSales( double ); // set gross sales amount
24    double getGrossSales() const; // return gross sales amount
```

Fig. 11.4 | CommissionEmployee class header. (Part 1 of 2.)

```
25
26     void setCommissionRate( double ); // set commission rate (percentage)
27     double getCommissionRate() const; // return commission rate
28
29     double earnings() const; // calculate earnings
30     void print() const; // print CommissionEmployee object
31  private:
32     std::string firstName;
33     std::string lastName;
34     std::string socialSecurityNumber;
35     double grossSales; // gross weekly sales
36     double commissionRate; // commission percentage
37  }; // end class CommissionEmployee
38
39  #endif
```

Fig. 11.4 | CommissionEmployee class header. (Part 2 of 2.)

```
1  // Fig. 11.5: 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 {
13    firstName = first; // should validate
14    lastName = last; // should validate
15    socialSecurityNumber = ssn; // should validate
16    setGrossSales( sales ); // validate and store gross sales
17    setCommissionRate( rate ); // validate and store commission rate
18 } // end CommissionEmployee constructor
19
```

Fig. 11.5 | Implementation file for `CommissionEmployee` class that represents an employee who is paid a percentage of gross sales. (Part 1 of 5.)

```
20  // set first name
21  void CommissionEmployee::setFirstName( const string &first )
22  {
23     firstName = first; // should validate
24  } // end function setFirstName
25
26  // return first name
27  string CommissionEmployee::getFirstName() const
28  {
29     return firstName;
30  } // end function getFirstName
31
32  // set last name
33  void CommissionEmployee::setLastName( const string &last )
34  {
35     lastName = last; // should validate
36  } // end function setLastName
37
```

Fig. 11.5 | Implementation file for `CommissionEmployee` class that represents an employee who is paid a percentage of gross sales. (Part 2 of 5.)

```
38  // return last name
39  string CommissionEmployee::getLastName() const
40  {
41     return lastName;
42  } // end function getLastName
43
44  // set social security number
45  void CommissionEmployee::setSocialSecurityNumber( const string &ssn )
46  {
47     socialSecurityNumber = ssn; // should validate
48  } // end function setSocialSecurityNumber
49
50  // return social security number
51  string CommissionEmployee::getSocialSecurityNumber() const
52  {
53     return socialSecurityNumber;
54  } // end function getSocialSecurityNumber
55
```

Fig. 11.5 | Implementation file for `CommissionEmployee` class that represents an employee who is paid a percentage of gross sales. (Part 3 of 5.)

```
56  // set gross sales amount
57  void CommissionEmployee::setGrossSales( double sales )
58  {
59     if ( sales >= 0.0 )
60        grossSales = sales;
61     else
62        throw invalid_argument( "Gross sales must be >= 0.0" );
63  } // end function setGrossSales
64
65  // return gross sales amount
66  double CommissionEmployee::getGrossSales() const
67  {
68     return grossSales;
69  } // end function getGrossSales
70
71  // set commission rate
72  void CommissionEmployee::setCommissionRate( double rate )
73  {
74     if ( rate > 0.0 && rate < 1.0 )
75        commissionRate = rate;
76     else
77        throw invalid_argument( "Commission rate must be > 0.0 and < 1.0" );
78  } // end function setCommissionRate
```

Fig. 11.5 | Implementation file for `CommissionEmployee` class that represents an employee who is paid a percentage of gross sales. (Part 4 of 5.)

```
79
80  // return commission rate
81  double CommissionEmployee::getCommissionRate() const
82  {
83     return commissionRate;
84  } // end function getCommissionRate
85
86  // calculate earnings
87  double CommissionEmployee::earnings() const
88  {
89     return commissionRate * grossSales;
90  } // end function earnings
91
92  // print CommissionEmployee object
93  void CommissionEmployee::print() const
94  {
95     cout << "commission employee: " << firstName << ' ' << lastName
96        << "\nsocial security number: " << socialSecurityNumber
97        << "\ngross sales: " << grossSales
98        << "\ncommission rate: " << commissionRate;
99  } // end function print
```

Fig. 11.5 | Implementation file for `CommissionEmployee` class that represents an employee who is paid a percentage of gross sales. (Part 5 of 5.)

11.3.1 Creating and Using a `CommissionEmployee` Class (cont.)

`CommissionEmployee` Constructor

- The `CommissionEmployee` constructor definition *purposely does not use member-initializer syntax* in the first several examples of this section, so that we can demonstrate how `private` and `protected` specifiers affect member access in derived classes.
  - Later in this section, we'll return to using member-initializer lists in the constructors.

11.3.1 Creating and Using a `CommissionEmployee` Class (cont.)

`CommissionEmployee` Member Functions `earnings` and `print`

- Member function `earnings` calculates a `CommissionEmployee`'s earnings.
- Member function `print` displays the values of a `CommissionEmployee` object's data members.

Testing Class `CommissionEmployee`

- Figure 11.6 tests class `CommissionEmployee`.

```
1  // Fig. 11.6: fig11_06.cpp
2  // CommissionEmployee class test program.
3  #include <iostream>
4  #include <iomanip>
5  #include "CommissionEmployee.h" // CommissionEmployee class definition
6  using namespace std;
7
8  int main()
9  {
10    // instantiate a CommissionEmployee object
11    CommissionEmployee employee(
12       "Sue", "Jones", "222-22-2222", 10000, .06 );
13
14    // set floating-point output formatting
15    cout << fixed << setprecision( 2 );
16
17    // get commission employee data
18    cout << "Employee information obtained by get functions: \n"
19       << "\nFirst name is " << employee.getFirstName()
20       << "\nLast name is " << employee.getLastName()
21       << "\nSocial security number is "
22       << employee.getSocialSecurityNumber()
23       << "\nGross sales is " << employee.getGrossSales()
24       << "\nCommission rate is " << employee.getCommissionRate() << endl;
```

Fig. 11.6 | CommissionEmployee class test program. (Part 1 of 3.)

```
25
26      employee.setGrossSales( 8000 ); // set gross sales
27      employee.setCommissionRate( .1 ); // set commission rate
28
29      cout << "\nUpdated employee information output by print function: \n"
30         << endl;
31      employee.print(); // display the new employee information
32
33      // display the employee's earnings
34      cout << "\n\nEmployee's earnings: $" << employee.earnings() << endl;
35   } // end main
```

Fig. 11.6 | CommissionEmployee class test program. (Part 2 of 3.)

```
Employee information obtained by get functions:

First name is Sue
Last name is Jones
Social security number is 222-22-2222
Gross sales is 10000.00
Commission rate is 0.06

Updated employee information output by print function:

commission employee: Sue Jones
social security number: 222-22-2222
gross sales: 8000.00
commission rate: 0.10

Employee's earnings: $800.00
```

Fig. 11.6 | `CommissionEmployee` class test program. (Part 3 of 3.)

11.3.2 Creating a `BasePlusCommissionEmployee` Class Without Using Inheritance

- We now discuss the second part of our introduction to inheritance by creating and testing (a completely new and independent) class `BasePlusCommissionEmployee` (Figs. 11.7–11.8), which contains a first name, last name, social security number, gross sales amount, commission rate *and* base salary.

```
1  // Fig. 11.7: BasePlusCommissionEmployee.h
2  // BasePlusCommissionEmployee class definition represents an employee
3  // that receives a base salary in addition to commission.
4  #ifndef BASEPLUS_H
5  #define BASEPLUS_H
6
7  #include <string> // C++ standard string class
8
9  class BasePlusCommissionEmployee
10 {
11 public:
12    BasePlusCommissionEmployee( const std::string &, const std::string &,
13       const std::string &, double = 0.0, double = 0.0, double = 0.0 );
14
15    void setFirstName( const std::string & ); // set first name
16    std::string getFirstName() const; // return first name
17
18    void setLastName( const std::string & ); // set last name
19    std::string getLastName() const; // return last name
20
21    void setSocialSecurityNumber( const std::string & ); // set SSN
22    std::string getSocialSecurityNumber() const; // return SSN
23
```

Fig. 11.7 | BasePlusCommissionEmployee class header. (Part 1 of 2.)