Object-Oriented Programming II

객체지향프로그래밍 II

Exercise for chapter 3 (week 1)

Chapter 3: Introduction to Classes and Objects

Exercise for:

1. Defining a Class
2. Defining a Member Function with a Parameter
3. Data Members, *set* Functions and *get* Functions

Ex3-1: Defining a Class (1)

```
1  // Fig. 9.1: Time.h
2  // Declaration of class Time.
3  // Member functions are defined in Time.cpp
4
5  // prevent multiple inclusions of header file
6  #ifndef TIME_H
7  #define TIME_H
8
9  // Time class definition
10 class Time
11 {
12 public:
13    Time(); // constructor
14    void setTime( int, int, int ); // set hour, minute and second
15    void printUniversal(); // print time in universal-time format
16    void printStandard(); // print time in standard-time format
17 private:
18    int hour; // 0 - 23 (24-hour clock format)
19    int minute; // 0 - 59
20    int second; // 0 - 59
21 }; // end class Time
22
23 #endif
```

Ex3-2: Defining a Class (2)

```
1  // Fig. 3.1: fig03_01.cpp
2  // Define class GradeBook with a member function displayMessage;
3  // Create a GradeBook object and call its displayMessage function.
4  #include <iostream>
5  using std::cout;
6  using std::endl;
7
8  // GradeBook class definition
9  class GradeBook
10 {
11 public:
12    // function that displays a welcome message to the GradeBook user
13    void displayMessage()
14    {
15       cout << "Welcome to the Grade Book!" << endl;
16    } // end function displayMessage
17 }; // end class GradeBook
18
19 // function main begins program execution
20 int main()
21 {
22    GradeBook myGradeBook; // create a GradeBook object named myGradeBook
23    myGradeBook.displayMessage(); // call object's displayMessage function
24    return 0; // indicate successful termination
25 } // end main
```

Beginning of class definition for class **GradeBook**

Beginning of class body

Access specifier **public**; makes members available to the public

Member function **displayMessge** returns nothing

End of class body

Use dot operator to call **GradeBook**'s member function

```
Welcome to the Grade Book!
```

매개변수와 인수

함수 매개변수 (parameter)

: 함수가 업무를 실행하는데 필요한 정보

함수 인수 (argument)

✓ 함수의 매개변수를 위해 함수 호출시 공급된 값

- 인수 값은 함수 매개변수로 복사된다.

참고: string 객체

string 객체

- ✓ 문자열 저장 및 활용을 위한 클래스
- ✓ C++ 표준 라이브러리의 일부임 (std::string)
  - 헤더파일 <string>에서 정의됨

getline 함수

- ✓ newline 문자를 만날 때 까지 입력된 문자열을 읽어들임
- ✓ 예: getline(cin, nameOfCourse);
  - 표준 입력(키보드)으로부터 문자열 입력을 받아 string 객체인 nameOfCourse에 저장

Ex3-3: 매개변수와 인수 사용

```
1  // Fig. 3.3: fig03_03.cpp
2  // Define class GradeBook with a member function that takes a parameter;
3  // Create a GradeBook object and call its displayMessage function.
4  #include <iostream>
5  using std::cout;
6  using std::cin;
7  using std::endl;
8
9  #include <string> // program uses C++ standard string class
10 using std::string;
11 using std::getline;
12
13 // GradeBook class definition
14 class GradeBook
15 {
16 public:
17    // function that displays a welcome message to the GradeBook user
18    void displayMessage( string courseName )
19    {
20       cout << "Welcome to the grade book for\n" << courseName << "!"
21          << endl;
22    } // end function displayMessage
23 }; // end class GradeBook
24
25 // function main begins program execution
26 int main()
27 {
28    string nameOfCourse; // string of characters to store the course name
29    GradeBook myGradeBook; // create a GradeBook object named myGradeBook
30
```

Include `string` class definition

Member function parameter

Use the function parameter as a variable

Ex3-3: 매개변수와 인수 사용 (cont.)

```
31    // prompt for and input course name
32    cout << "Please enter the course name:" << endl;
33    getline( cin, nameOfCourse ); // read a course name with blanks
34    cout << endl; // output a blank line
35
36    // call myGradeBook's displayMessage function
37    // and pass nameOfCourse as an argument
38    myGradeBook.displayMessage( nameOfCourse );
39    return 0; // indicate successful termination
40 } // end main
```

Passing an argument to the member function

```
Please enter the course name:
CS101 Introduction to C++ Programming

Welcome to the grade book for
CS101 Introduction to C++ Programming!
```

지역변수와 멤버변수

지역 변수 (local variables)

- ✓ 함수 안에서 선언된 변수
  - 함수 밖에서 사용 할 수 없다.
- ✓ 함수가 소멸될 때 같이 소멸된다.

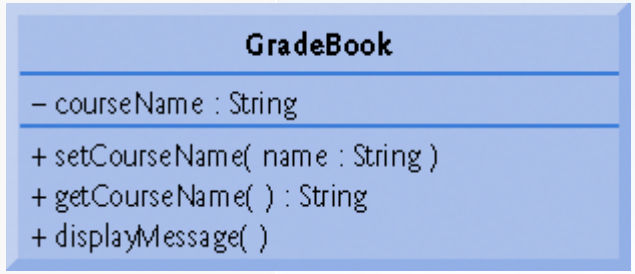
멤버 변수 (member variables)

- ✓ 객체가 살아있는 동안만 존재
- ✓ 데이터 멤버로 표현
  - 클래스 정의에 선언된 변수
- ✓ 클래스의 각 객체는 속성의 복사본을 보유한다.

Ex3-4: private 멤버 변수

```
1  // Fig. 3.5: fig03_05.cpp
2  // Define class GradeBook that contains a courseName data member
3  // and member functions to set and get its value;
4  // Create and manipulate a GradeBook object with these functions.
5  #include <iostream>
6  using std::cout;
7  using std::cin;
8  using std::endl;
9
10 #include <string> // program uses C++ standard string class
11 using std::string;
12 using std::getline;
13
14 // GradeBook class definition
15 class GradeBook
16 {
17 public:
18    // function that sets the course name
19    void setCourseName( string name )
20    {
21       courseName = name; // store the course name in the object
22    } // end function setCourseName
23
24    // function that gets the course name
25    string getCourseName()
26    {
27       return courseName; // return the object's courseName
28    } // end function getCourseName
29
```

<UML Diagram>



set function modifies **`private`** data

get function accesses **`private`** data

Ex3-4: private 멤버 변수 (cont.)

```
30    // function that displays a welcome message
31    void displayMessage()
32    {
33       // this statement calls getCourseName to get the
34       // name of the course this GradeBook represents
35       cout << "Welcome to the grade book for\n" << getCourseName() << "!"
36          << endl;
37    } // end function displayMessage
38 private:
39    string courseName; // course name for this GradeBook
40 }; // end class GradeBook
41
42 // function main
43 int main()
44 {
45    string nameOfCourse; // string of characters to store the course name
46    GradeBook myGradeBook; // create a GradeBook object named myGradeBook
47
48    // display initial value of courseName
49    cout << "Initial course name is: " << myGradeBook.getCourseName()
50       << endl;
51
```

Use *set* and *get* functions, even within the class

private members accessible only to member functions of the class

Accessing **private** data outside class definition

Ex3-4: private 멤버 변수

```
52    // prompt for, input and set course name
53    cout << "\nPlease enter the course name:" << endl;
54    getline( cin, nameOfCourse ); // read a course name with blanks
55    myGradeBook.setCourseName( nameOfCourse ); // set the course name
56
57    cout << endl; // outputs a blank line
58    myGradeBook.displayMessage(); // display message with new course name
59    return 0; // indicate succes
60 } // end main
```

Modifying **`private`** data outside class definition

```
Initial course name is:

Please enter the course name:
CS101 Introduction to C++ Programming

Welcome to the grade book for
CS101 Introduction to C++ Programming!
```

Ex3-5: Account class

3.12 ***(Account Class)*** Create an Account class that a bank might use to represent customers' bank accounts. **Include a data member** of type int to represent the account balance. **Provide a constructor** that receives an initial balance an d uses it to initialize the data member. The constructor should **validate** the initial balance to ensure that it is greater than or equal to 0. If not, set the balance to 0 and display an error message indicating that the initial balance was inval id. **Provide three member functions**. Member function **credit** should add an amount to the current balance. Member function **debit** should withdraw money from the Account and ensure that the debit amount does not exceed the Accou nt 's balance. If it does, the balance should be left unchanged and the function should print a message indicating "Debi t amount exceeded account balance. "Member function **getBalance** should return the current balance.

Create a program that creates two Account objects and tests the member functions of class Account.

```
account1 balance: $50
account2 balance: $25

Enter withdrawal amount for account1: 35

attempting to subtract 35 from account1 balance

account1 balance: $15
account2 balance: $25

Enter withdrawal amount for account2: 50

attempting to subtract 50 from account2 balance

Debit amount exceeded account balance.

account1 balance: $15
account2 balance: $25
```

Ex3-6: Invoice class

3.13 ***(Invoice Class)*** Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should **include four data members** - a part number (type string), a part description (type string), a quantity of the item being purchased (type int) and a price per item (type int). Your class **should have a constructor** that initializes the four data members. Provide a **set** and a **get** function for each data member. In addition, provide a member function named **getInvoiceAmount** that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as an int value. If the quantity is not positive, it should be set to 0 . If the price per item is not positive, it should be set to 0 .

Write a test program that demonstrates class Invoice 's capabilities.

```
Part number: 12345
Part description: Hammer
Quantity: 100
Price per item: $5
Invoice amount: $500

quantity cannot be negative. quantity set to 0.

Invoice data members modified.

Part number: 123456
Part description: Saw
Quantity: 0
Price per item: $10
Invoice amount: $0
```

HW #1 : Classes and Objects(1)

- EX3-5와 EX3-6 를 완성
- Deadline: 2021-03-17(Wed) 23:59

Report format

- Report cover format :
  - Title
    - HW#1. Classes and Objects (1)
  - Information
    - Department(학과)
    - Class(분반)
    - Name(이름)
    - Date (제출일자)
    - Prof. Name (교수명)

Report format

- Report contents:
  - Problem
    - 문제를 서술함
  - Solution
    - 프로그램 소스코드, UML 다이어램, Comment 등을 포함
  - Discussion
    - 문제를 해결하면서 이해한 내용, 어려웠던 점, 바라는 점 등을 기술