
```
36
37 // display a welcome message to the GradeBook user
38 void GradeBook::displayMessage() const
39 {
40     // call getCourseName to get the courseName
41     cout << "Welcome to the grade book for\n" << getCourseName()
42         << "!" << endl;
43 } // end function displayMessage
```

Fig. 3.16 | Member-function definitions for class GradeBook with a *set* function that validates the length of data member courseName. (Part 3 of 3.)

```
1 // Fig. 3.17: fig03_17.cpp
2 // Create and manipulate a GradeBook object; illustrate validation.
3 #include <iostream>
4 #include "GradeBook.h" // include definition of class GradeBook
5 using namespace std;
6
7 // function main begins program execution
8 int main()
9 {
10     // create two GradeBook objects;
11     // initial course name of gradeBook1 is too long
12     GradeBook gradeBook1( "CS101 Introduction to Programming in C++" );
13     GradeBook gradeBook2( "CS102 C++ Data Structures" );
14
15     // display each GradeBook's courseName
16     cout << "gradeBook1's initial course name is: "
17         << gradeBook1.getCourseName()
18         << "\ngradeBook2's initial course name is: "
19         << gradeBook2.getCourseName() << endl;
20
21     // modify gradeBook1's courseName (with a valid-length string)
22     gradeBook1.setCourseName( "CS101 C++ Programming" );
```

Fig. 3.17 | Creating and manipulating a GradeBook object in which the course name is limited to 25 characters in length. (Part 1 of 2.)

```
23
24 // display each GradeBook's courseName
25 cout << "\ngradeBook1's course name is: "
26     << gradeBook1.getCourseName()
27     << "\ngradeBook2's course name is: "
28     << gradeBook2.getCourseName() << endl;
29 } // end main
```

Name "CS101 Introduction to Programming in C++" exceeds maximum length (25).
Limiting courseName to first 25 characters.

```
gradeBook1's initial course name is: CS101 Introduction to Pro
gradeBook2's initial course name is: CS102 C++ Data Structures
```

```
gradeBook1's course name is: CS101 C++ Programming
gradeBook2's course name is: CS102 C++ Data Structures
```

Fig. 3.17 | Creating and manipulating a GradeBook object in which the course name is limited to 25 characters in length. (Part 2 of 2.)

3.8 Validating Data with *set Functions* (*cont.*)

- A `public set` function such as `setCourseName` should carefully scrutinize any attempt to modify the value of a data member (e.g., `courseName`) to ensure that the new value is appropriate for that data item.
- A *set* function could return a value indicating that an attempt was made to assign invalid data to an object of the class.
- A client could then test the return value of the *set* function to determine whether the attempt to modify the object was successful and to take appropriate action if not.



Software Engineering Observation 3.3

Making data members `private` and controlling access, especially write access, to those data members through `public` member functions helps ensure data integrity.



Error-Prevention Tip 3.4

The benefits of data integrity are not automatic simply because data members are made **private**—you must provide appropriate validity checking and report the errors.