



Validating Data with set Functions

GradeBook.h

```
1 // GradeBook.h
2 // GradeBook class definition. This file presents GradeBook's public
3 // interface without revealing the implementations of GradeBook's member
4 // functions, which are defined in GradeBook.cpp.
5
6 #include <string> // class GradeBook uses C++ standard string class
7 using namespace std;
8
9 // GradeBook class definition
10 class GradeBook
11 {
12 public:
13     GradeBook( string );           // constructor that initializes courseName
14     void setCourseName( string ); // function that sets the course name
15     string getCourseName();        // function that gets the course name
16     void displayMessage();         // function that displays a welcome message
17
18 private:
19     string courseName; // course name for this GradeBook
20 }; // end class GradeBook
```

GradeBook.cpp

```
1 // GradeBook.cpp
2 // GradeBook member-function definitions. This file contains
3 // implementations of the member functions prototyped in GradeBook.h.
4
5 #include <iostream>
6 #include "GradeBook.h" // include definition of class GradeBook
7 using namespace std;
8
9 // constructor initializes courseName with string supplied as argument
10 GradeBook::GradeBook( string name )
11 {
12     setCourseName( name ); // validate and store courseName
13 } // end GradeBook constructor
14
15 // function that sets the course name;
16 // ensures that the course name has at most 25 characters
17 void GradeBook::setCourseName( string name )
18 {
19     if ( name.length() <= 25 ) // if name has 25 or fewer characters
20         courseName = name; // store the course name in the object
21     if ( name.length() > 25 ) // if name has more than 25 characters
22     {
23         // set courseName to first 25 characters of parameter name
24         courseName = name.substr( 0, 25 ); // start at 0, length of 25
25         cout << "Name \"" << name << "\" exceeds maximum length (25).\n"
26              << "Limiting courseName to first 25 characters.\n" << endl;
27     } // end if
28 } // end function setCourseName
29
30 // function to get the course name
31 string GradeBook::getCourseName()
32 {
33     return courseName; // return object's courseName
34 } // end function getCourseName
35
36 // display a welcome message to the GradeBook user
37 void GradeBook::displayMessage()
38 {
39     // call getCourseName to get the courseName
40     cout << "Welcome to the grade book for\n" << getCourseName()
41          << "!" << endl;
42 } // end function displayMessage
```

- The C++ Standard Library's string class defines a member function `length` that returns the number of characters in a string object. Parameter `name` is a string object, so the call **`name.length()`** returns the number of characters in `name`.
- Standard class string provides member function **`substr`** (short for "substring") that returns a new string object created by copying part of an existing string object.

Main.cpp

```
1 // Create and manipulate a GradeBook object; illustrate validation.
2 #include <iostream>
3 #include "GradeBook.h" // include definition of class GradeBook
4 using namespace std;
5
6 // function main begins program execution
7 int main()
8 {
9     // create two GradeBook objects
10    // initial course name of gradeBook1 is too long
11    GradeBook gradeBook1( "CS101 Introduction to Programming in C++" );
12    GradeBook gradeBook2( "CS102 C++ Data Structures" );
13
14    // display each GradeBook's courseName
15    cout << "gradeBook1's initial course name is: "
16    << gradeBook1.getCourseName()
17    << "\ngradeBook2's initial course name is: "
18    << gradeBook2.getCourseName() << endl;
19
20    // modify myGradeBook's courseName (with a valid-length string)
21    gradeBook1.setCourseName( "CS101 C++ Programming" );
22
23    // display each GradeBook's courseName
24    cout << "\ngradeBook1's course name is: "
25    << gradeBook1.getCourseName()
26    << "\ngradeBook2's course name is: "
27    << gradeBook2.getCourseName() << endl;
28 }
```