



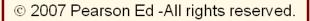


- Class Definitions and Objects
- Member Functions
- Data Members
 - Get and Set functions
 - Constructors
- Placing Classes in Separate Files
- Separating Interface from Implementation
- Data Validation
 - Ensures that data in an object is in a particular format or range.



C++ Program Structure

- . Typically C++ Programs will consist of:
 - A function main
 - One or more classes
 - Each containing data members and member functions.

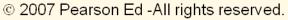




16.2 Defining a Class With a Member Function

- Class definition

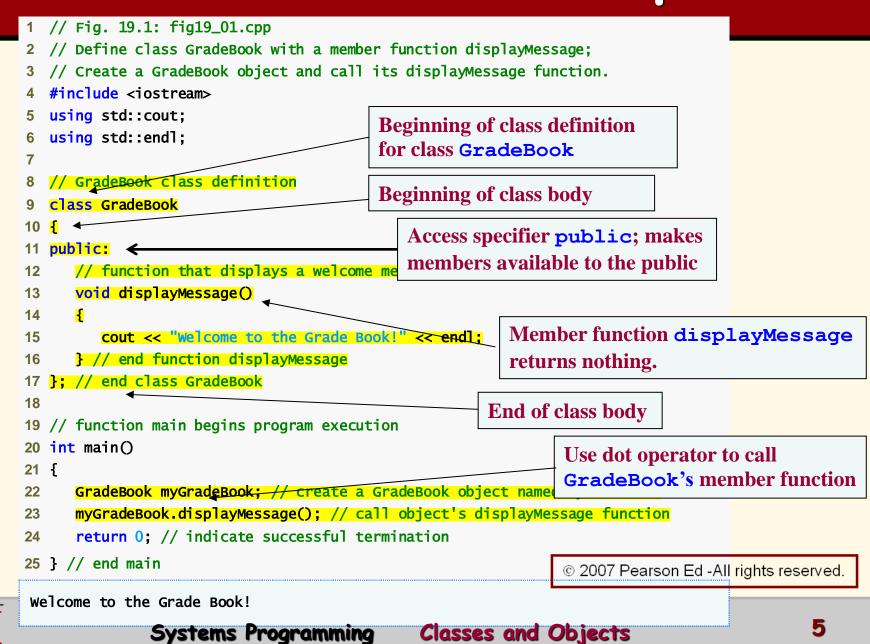
- Tells the compiler what **member functions** and **data members** belong to the class.
- Keyword class followed by the class's name.
- Class body is enclosed in braces ({})
 - $\boldsymbol{\cdot}$ Specifies data members and member functions
 - Access-specifier public:
 - Indicates that a member function or data member is accessible to other functions and member functions of other classes.



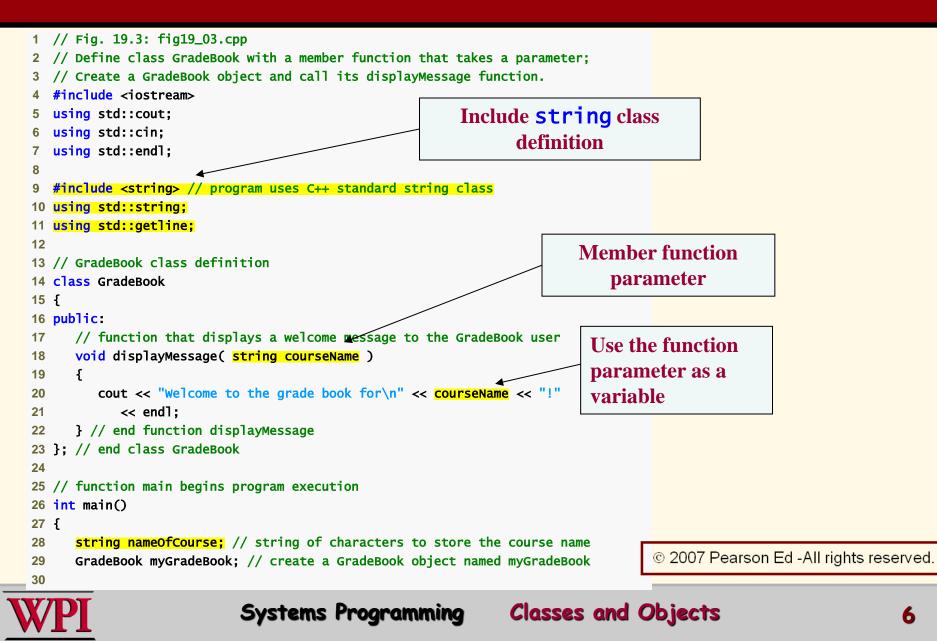




C++ Gradebook Example



Member Function Takes a Parameter



Member function takes a parameter

			getline is a library fcn	
31	// prompt for and input course <u>name</u>			
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				
<pre>38 myGradeBook.displayMessage(nameOfCourse);</pre>				
39	return 0; // indicate successful termination			
40 } // end main		Pas	sing an argument to the	
Please enter the course name:		member function		
CS101 Introduction to C++ Programming				
	Welcome to the grade book for CS101 Introduction to C++ Programming!			

© 2007 Pearson Ed -All rights reserved.



Member Function Takes a Parameter

- . A string
 - Represents a string of characters.
 - An object of C++ Standard Library class
 std::string
 - Defined in header file <string>.
- Library function getline
 - Used to retrieve input until a newline is encountered.
 - Example
 - getline(cin, nameOfCourse);
 - Inputs a line from standard input into string object nameOfCourse.



16.4 Data Members, set Functions and <u>get Functions</u>

- Local variables
 - Variables declared in a function definition's body cannot be used outside of that function body.
 - When a function terminates the values of its local variables are lost.
- . Attributes
 - Exist throughout the life of the object.
 - Are represented as data members.
 - Namely, associated with variables in a class definition.
 - Are declared inside a class definition but outside the bodies of the class's member-function definitions.
 - Each object of a class maintains its own copy of its attributes in memory.

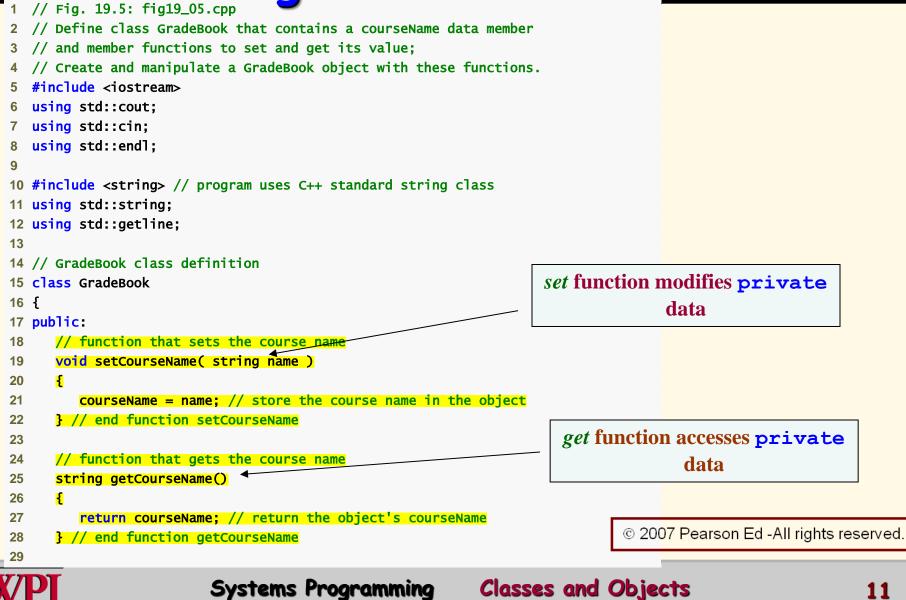


16.4 Data Members, set Functions and <u>get Functions</u>

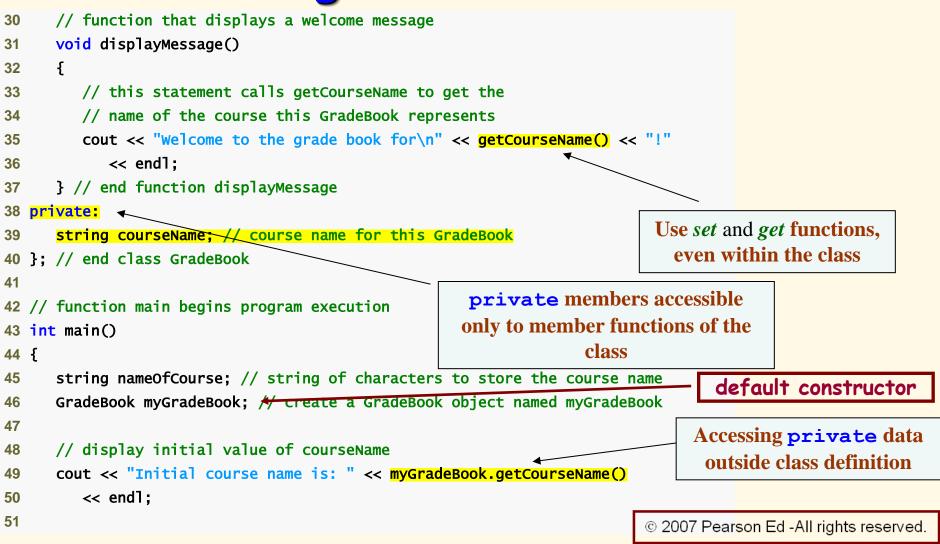
- Access-specifier private
 - Makes a data member or member function accessible only to member functions of the class.
 - private is the default access for class members.
 - "information hiding" is an object-oriented tenet.
- . Returning a value from a function
 - A function that specifies a return type other than void
 - Returns a value to its calling function.



16.4 Data Members, set Functions and get Functions 1 // Fig. 19.5: fig19_05.cpp



16.4 Data Members, set Functions and <u>get Functions</u>





16.4 Data Members, set Functions and <u>get Functions</u>

// prompt for, input and set course name					
	<< endl:				
54 getline(cin, nameOfCourse); // read a course name with blanks 55 myGradeBook.setCourseName(nameOfCourse); // set the course name					
56					
58 myGradeBook.displayMessage(); // display message with new course name					
60 } // end main					
,,	Modifying private data outside class				
tial course name is:	definition				
Please enter the course name: cs101 Introduction to C++ Programming					
come to the grade book for	default setting from constructor is an empty string!!				
	<pre>cout << endl; // outputs a blank line myGradeBook.displayMessage(); // display return 0; // indicate successful terminat // end main tial course name is:</pre>				

© 2007 Pearson Ed -All rights reserved.



Software Engineering Observation 16.1

. As a rule of thumb, data members should be declared private and member functions should be declared **public**. (We will see that it is appropriate to declare certain member functions private, if they are to be accessed only by other member functions of the class.)



Data Members, set Functions and get Functions

- Software engineering with set and get functions:
 - public member functions that allow clients of a class to set or get the values of private data members.
 - set functions are sometimes called mutators and get functions are sometimes called accessors.
 - Allows the creator of the class to control how clients access private data.
 - Should also be used by other member functions of the same class.



^{© 2007} Pearson Ed -All rights reserved.

Initializing Objects with Constructors

- . Constructors
 - Functions used to initialize an object's data when it is created.
 - The call is made **implicitly** by the compiler when the object is created.
 - Must be defined with the same name as the class.
 - Cannot return values.
 - Not even void !!
 - A default constructor has no parameters.
 - The compiler will provide one when a class does not explicitly include a constructor.
 - A compiler's default constructor only calls constructors of data members that are objects of classes.



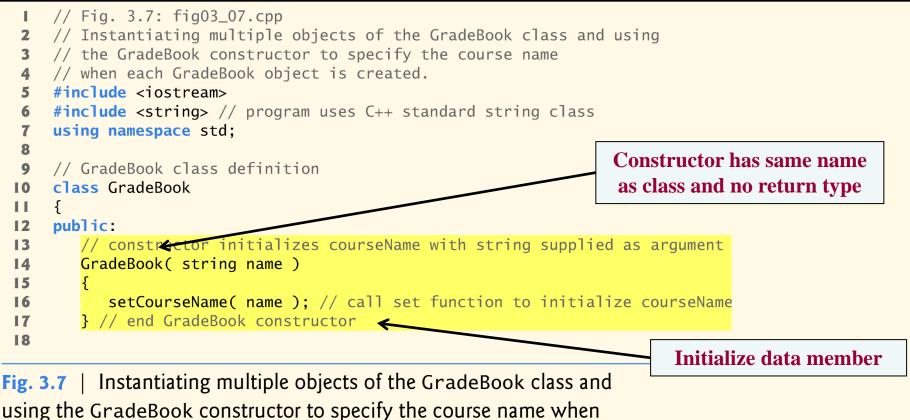
16.5 Initializing Objects with Constructors

- Any constructor that takes no arguments is called a default constructor.
- A class gets a default constructor in one of two ways:
 - The compiler implicitly creates a default constructor in a class that does not define a constructor. Such a constructor does not initialize the class's data members, but does call the default constructor for each data member that is an object of another class. An uninitialized variable typically contains a "garbage" value.
 - You explicitly define a constructor that takes no arguments. Such a default constructor will call the default constructor for each data member that is an object of another class and will perform additional initialization specified by you.
- If you define a constructor with arguments, C++ will not implicitly create a default constructor for that class.

Copyright @ Pearson, Inc. 2013. All Rights Reserved.



Constructor Example



each GradeBook object is created. (Part 1 of 3.)

Copyright @ Pearson, Inc. 2013. All Rights Reserved.



Constructor Example

```
// function to set the course name
19
       void setCourseName( string name )
20
21
       Ł
22
          courseName = name; // store the course name in the object
       } // end function setCourseName
23
24
25
       // function to get the course name
26
       string getCourseName()
27
       {
          return courseName; // return object's courseName
28
       } // end function getCourseName
29
30
31
       // display a welcome message to the GradeBook user
32
       void displayMessage()
33
       {
34
          // call getCourseName to get the courseName
          cout << "Welcome to the grade book for\n" << getCourseName()
35
36
             << "!" << endl:
       } // end function displayMessage
37
```

Fig. 3.7 | Instantiating multiple objects of the GradeBook class and using the GradeBook constructor to specify the course name when each GradeBook object is created. (Part 2 of 3.)

Copyright @ Pearson, Inc. 2013. All Rights Reserved.



Constructor Example

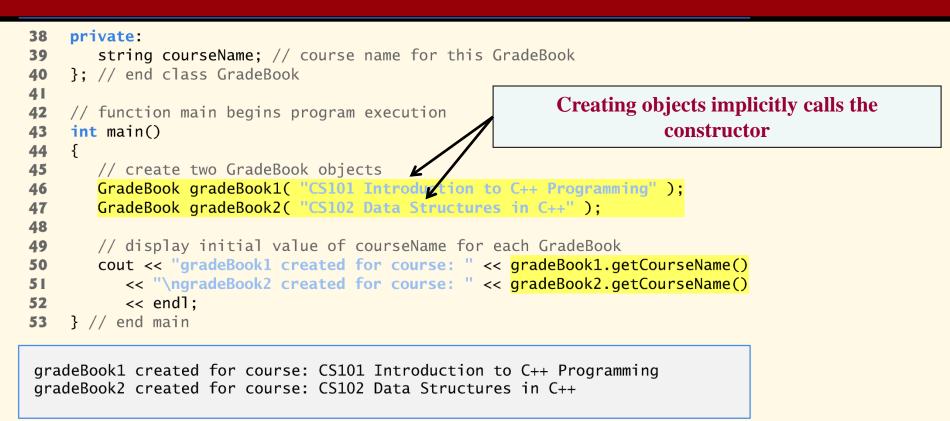


Fig. 3.7 | Instantiating multiple objects of the GradeBook class and using the GradeBook constructor to specify the course name when each GradeBook object is created. (Part 3 of 3.)

Copyright @ Pearson, Inc. 2013. All Rights Reserved.



Placing a Class in a Separate File for Reusability

- .cpp file is known as a source-code file.
- Header files
 - Separate files in which class definitions are placed.
 - Allows compiler to recognize the classes when used elsewhere.
 - Generally have .h filename extensions
- Driver files
 - A program used to test software (such as classes).
 - Contains a main function so it can be executed.

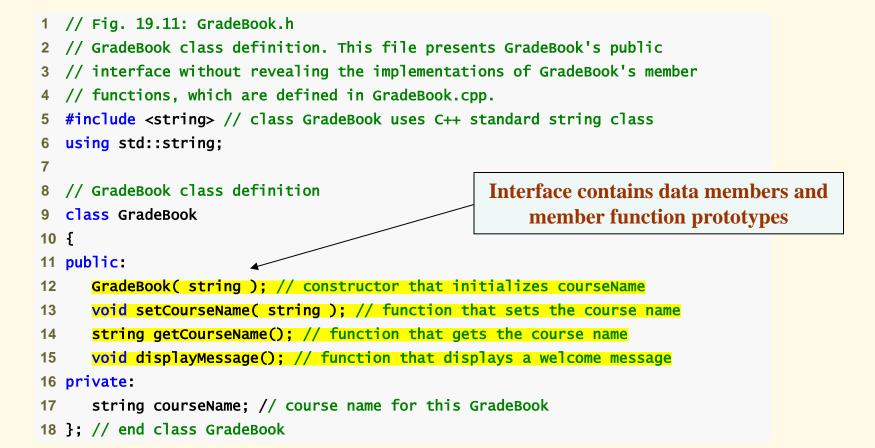


- . Interface
 - Describes what services a class's clients can use and how to request those services.
 - without revealing how the class carries out the services.
 - a class definition that lists only member function names, return types and parameter types.
 - e.g., function prototypes
 - A class's interface consists of the class's **public** member functions (services).
- · Separating interface from implementation:
 - Client code should not break if implementation changes, as long as the interface stays the same.



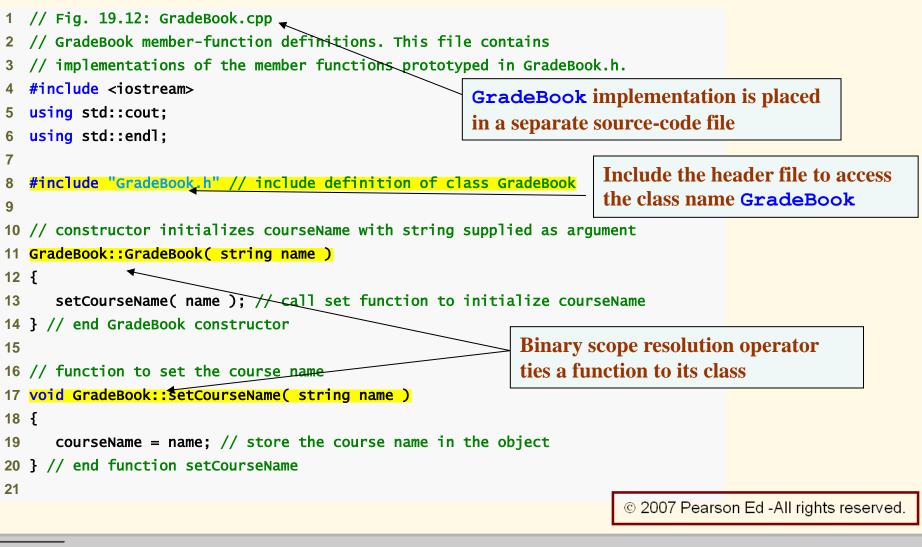
- Define the member functions outside the class definition, in a separate source-code file.
 - In a source-code file for a class
 - Use binary scope resolution operator (::) to tie each member function to the class definition.
 - Implementation details are hidden.
 - Client code does not need to know the implementation.
- . In a header file for a class
 - The function prototypes describe the class's public interface.





© 2007 Pearson Ed -All rights reserved.





Classes and Objects

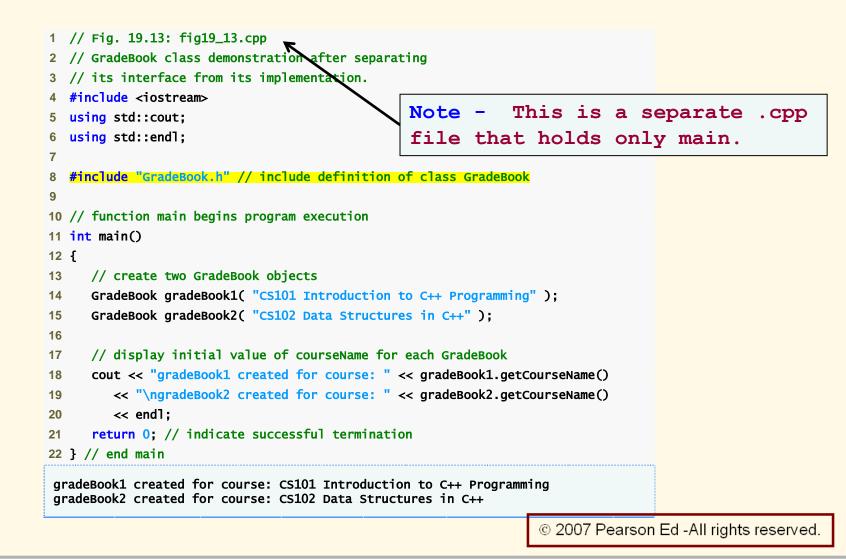
25

Systems Programming



```
22 // function to get the course name
23 string GradeBook::getCourseName()
24 {
      return courseName; // return object's courseName
25
26 } // end function getCourseName
27
28 // display a welcome message to the GradeBook user
29 void GradeBook::displayMessage()
30 {
     // call getCourseName to get the courseName
31
32
     cout << "Welcome to the grade book for\n" << getCourseName()
         << "!" << endl:</pre>
33
34 } // end function displayMessage
```





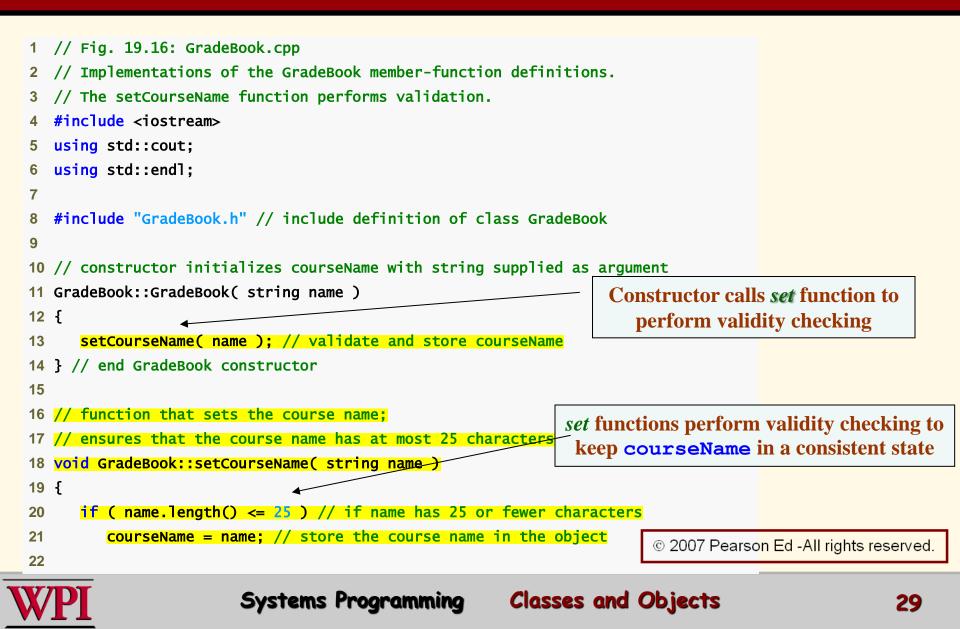


19.10 Validating Data with set Functions

- set functions can validate data.
 - Known as validity checking.
 - Keeps object in a consistent state.
 - The data member contains a valid value.
 - Can return values indicating that attempts were made to assign invalid data.
- string member functions
 - length returns the number of characters in the string.
 - substr returns specified substring within the string.



Validating Data with set Functions



Validating Data with set Functions

```
if ( name.length() > 25 ) // if name has more than 25 characters
23
     £
24
        // set courseName to first 25 characters of parameter name
25
         courseName = name.substr( 0, 25 ); // start at 0, length of 25
26
27
         cout << "Name \"" << name << "\" exceeds maximum length (25).\n"
28
29
            << "Limiting courseName to first 25 characters.\n" << endl;</pre>
30
     } // end if
31 } // end function setCourseName
32
33 // function to get the course name
34 string GradeBook::getCourseName()
35 {
      return courseName; // return object's courseName
36
37 } // end function getCourseName
38
39 // display a welcome message to the GradeBook user
40 void GradeBook::displayMessage()
41 {
42
     // call getCourseName to get the courseName
     cout << "Welcome to the grade book for\n" << getCourseName()
43
         << "!" << endl:</pre>
44
45 } // end function displayMessage
```

© 2007 Pearson Ed -All rights reserved.



Review of Classes and Objects

- . Introduced class definitions and objects
 - Public versus private access into class.
- Syntax for member functions
- Syntax data members
 - Get and Set functions
 - Constructors
- · Placing classes in separate files
- Separating interface from implementation
- Data validation in set functions.

