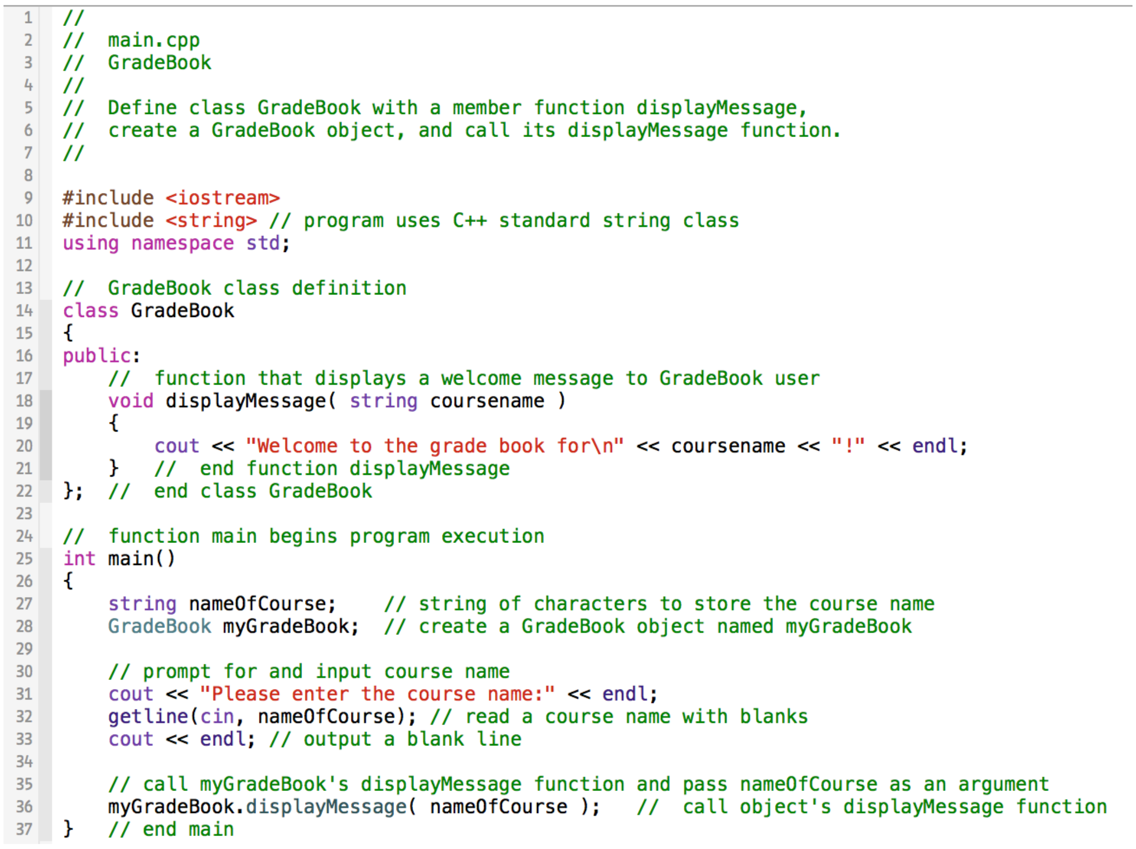
Ekaba Bisong
Programming in C++
University of Calabar

Tutorial Notes #12
May 11, 2015

Defining a Member Function with a Parameter

A member function can require one or more parameters that represent additional data it needs to perform its task. When the function is called, values - called **arguments** - are passed for each of the function's parameters.



```
1   //
2   //  main.cpp
3   //  GradeBook
4   //
5   //  Define class GradeBook with a member function displayMessage,
6   //  create a GradeBook object, and call its displayMessage function.
7   //
8
9   #include <iostream>
10  #include <string> // program uses C++ standard string class
11  using namespace std;
12
13  //  GradeBook class definition
14  class GradeBook
15  {
16  public:
17      //  function that displays a welcome message to GradeBook user
18      void displayMessage( string coursename )
19      {
20          cout << "Welcome to the grade book for\n" << coursename << "!" << endl;
21      }   //  end function displayMessage
22  };  //  end class GradeBook
23
24  //  function main begins program execution
25  int main()
26  {
27      string nameOfCourse;    // string of characters to store the course name
28      GradeBook myGradeBook;  // create a GradeBook object named myGradeBook
29
30      // prompt for and input course name
31      cout << "Please enter the course name:" << endl;
32      getline(cin, nameOfCourse); // read a course name with blanks
33      cout << endl; // output a blank line
34
35      // call myGradeBook's displayMessage function and pass nameOfCourse as an argument
36      myGradeBook.displayMessage( nameOfCourse );   //  call object's displayMessage function
37  }   // end main
```

Whats happening in the main() function?

Line 27 creates a variable of type string called *nameOfCourse* that will be used to store the course name entered by the user.

A string is actually an object of the C++ Standard Library class string. This class is defined in header <string>, and the name ***string***, like ***cout***, belongs to namespace ***std***. The using directive in line 11 allows us to simply write string in line 27 rather than *std::string*.

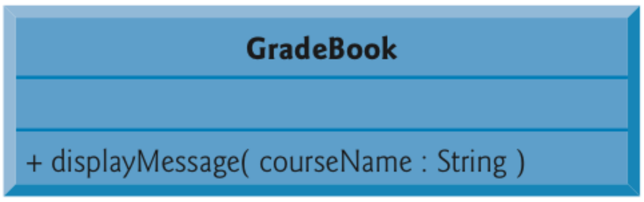
In this example, we'd like the user to type the complete course name and press Enter to submit it to the program, and we'd like to store the entire course name in the string variable *nameOfCourse*. The function call ***getline(cin, nameOfCourse)*** in line 32 reads characters (including the space characters that separate the words in the input) from the standard input stream object cin until the newline character is encountered, places the characters in the string variable *nameOfCourse* and discards the newline character.

More on Arguments and Parameters
To specify in a function definition that the function requires data to perform its task, you place additional information in the function's parameter list, which is located in the parentheses following the function name.

The number and order of arguments in a function call must match the number and order of parameters in the parameter list of the called member function's header. Also, the argument types in the function call must be consistent with the types of the corresponding parameters in the function header.

Updated UML Class Diagram for Class GradeBook

- The UML models a parameter by listing the parameter name, followed by a colon and the parameter type in the parentheses following the operation name.
- The UML has its own data types similar to those of C++.
- The UML is language independent — it's used with many different programming languages—so its terminology does not exactly match that of C++.
- For example, the UML type String corresponds to the C++ type string.



Excerpt adapted from C++ How to Program, P.J. Deitel, H.M. Deitel. -- 8th ed