

Name:

Sec:

07/18/2006

Lab 6 Using Class Constructors

C++ does not automatically initialize user defined variables (e.g. classes and structures that you create). To guarantee that the data members of a class are initialized, you use constructors. The constructor without parameters is called the default constructor.

Constructor functions have the same names as the class, have no return type, and are automatically executed when a class object enters its scope. (They cannot be called directly). When there are multiple constructors, the names are the same, but the parameter lists must differ. This is an example of what is known as function overloading, have multiple functions with the same name but with different parameter lists. Overloading a classes constructor can be a very useful and powerful technique for writing programs.

Objectives

In this lab, you define a class that contains a default constructor, and some constructors with parameters..

After completing this lab, you will be able to:

- Write a default constructor function.
- Write a constructor function with parameters.

Instructions

1. We will create a simple employee class in this lab. Create an Employee class that contains the following data members:
 - The private string data members named lastName and firstName
 - A private double data member named rateOfPay
2. Now we will write a default constructor for your Employee class. A class constructor is simply a class function that has the same name as the name of the class, but does not have a return type. To create a constructor for your Employee class, declare a public method called Employee(), that takes no parameters. The implementation of your Employee() constructor should set the firstName and lastName parameters to "John" and "Doe" respectively and set the rateOfPay to be 6.50 (minimum wage in dollars per hour).
3. Overload the Employee constructor by declaring a second constructor that takes some parameters. This constructor will take 2 strings and a double

as parameters (representing the first name, last name and pay rate respectively for a new Employee). Your implementation of the second, overloaded, Employee constructor should take these 3 parameters and initialize the firstName, lastName and rateOfPay member variables respectively.

4. Also implement a public member function called displayPay(). This member function should take a single integer as a parameter. It will not return any value but will instead display a pay check for the employee. The integer parameter is the number of hours that the employee it to be payed for. So if we pass in 40 as the number of hours worked to the displayPay() member function, the function should calculate the paycheck by multiplying 40 by the rate of pay, and display a pay check to the console, like this:

Pay to the order of: John Doe 40hrs x \$6.50/hr = \$260.00

5. Create a main function and demonstrate your Employee class by creating a default Employee, and 2 other Employees using your overloaded constructor. Display the pay checks of your employees using the displayPay() member method.

Lab 6 Finished

You have now completed Lab 6. If your program compiles and runs correctly and you have successfully uploaded your source file to the eCollege online submission site, then you are done.