

Lab 6

C++ -- Designing and Implementing Classes

Purpose: Understand the concept of classes as types and objects as instances; object lifetimes, and invocation of methods on objects.

Grading:	Documentation & Style (indentation, spacing, etc)	5 points
	Makefile	5 points
	Test Program (thoroughly tests all Employee member functions)	10 points
	UML Diagram for the Employee Class	5 points
	Employee Class	25 points
	includes 3 data members and 7 member functions	

	Total possible	50 points

Employee Class

UML for the Employee Class

private data members

- 1) firstName – string
- 2) lastName – string
- 3) monthlySalary – float

public member functions

- 1) constructor with 3 parameters: first, last, salary
- 2) accessor for firstName
- 3) accessor for lastName
- 4) accessor for monthlySalary
- 5) mutator for firstName
- 6) mutator for lastName
- 7) mutator for monthlySalary

In Employee.h (the header file) - refer to GradeBook.h on page 608 for an example

- Include a preprocessor wrapper.
- Declare data members for first name (string), last name (string), and monthly salary (float).

Write only the prototypes for the member functions.

- Provide a constructor with three parameters and use the parameter values to initialize first name, last name and monthly salary.

Call the mutator methods from within the constructor.

- Provide accessors getFirstName, getLastName, and getMonthlySalary.
- Provide mutators setFirstName, setLastName, and setMonthlySalary. In the mutator for monthly salary, if the monthly salary is not positive, set it to 0.

In Employee.cpp - refer to GradeBook.cpp on page 609 for an example

Write the full function definitions for all of the member functions

Note: In C++ there is no class wrapped around these functions like there is in Java. Each function must have the class name and the scope resolution operator, `Employee::` in front of the function name, to designate that it is a member function of the Employee class.

In EmployeeTest.cpp

- Write a test program that demonstrates that each function of the Employee class works correctly. (You should have at least one call to each member function.)
- Create two Employee objects and display each object's full name and yearly salary.
- Give each Employee a 10 percent raise and display each Employee's full name and yearly salary again.
- Use C++ style input and output throughout your program. The C functions: `puts`, `printf`, `scanf`, etc. are not permitted in this assignment.

Makefile

Create a makefile that will compile your programs and produce an executable named EmployeeTest. The makefile should have a target called "all".