CS 271 / 462
Programming Assignment 8
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 includes 3 data members and 7 member functions	25 points
		-------------
	Total possible	50 points

Textbook Chapters

- Chapter 15 - Intro to C++
- Chapter 16 - Introduction to Classes, Objects, and Strings

Documentation and Style (does not apply to the makefile)

- Each program must have header comments.
- Each program must have explanatory inline comments.
- Each program must follow the course guidelines for documentation and style.

All Programs must compile. Any program that does not compile or a makefile that has errors will receive a grade of zero.

The programs you submit must be your own work. See the syllabus for the course policy on cheating. Submissions that contain evidence of cheating will receive a grade of zero.

Step 1: Create the UML Diagram for the Employee Class

Use Microsoft Word as shown in class. Insert a table with 3 rows and 1 column.

There are examples in the Oct 20 lecture notes. Here is the information you need:

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

Step 2: Create the makefile

Files included in this assignment are Employee.h, Employee.cpp, and EmployeeTest.cpp. You must have separate targets for all, Employee.o, EmployeeTest.o, and EmployeeTest. The name of the executable is EmployeeTest.

Use g++ -std=c++11 (not gcc) as the compiler command.

Step 3: Create Employee.h (the header file)

For examples, refer to GradeBook.h on page 611 or the Student.h file from Monday, Oct 25 lecture notes.

- Include a preprocessor wrapper.
- Include any C++ libraries that are needed.
- Write the using namespace statement.
- Write the class definition for the Employee class.

  Include a private section with data members for first name (string), last name (string), and monthly salary (float).
  Include a public section with prototypes for the member functions.

In the class definition, write only the <u>prototypes</u> for the member functions.

Principle of Least Privilege

- Accessors should use const to prevent the function from changing the calling object.

Step 4: Create Employee.cpp

- Include any C++ libraries that are needed.
- Include the .h file.
- Write the **using namespace std;** statement.
- Write the full function definitions for all of the member functions.

  Before each function name, write the class name and :: (binary scope resolution operator).
- The constructor's 3 parameters will be used to initialize firstName, lastName and monthlySalary.
- The mutator functions will use the parameter to change the value of a data member.
- The mutator for monthlySalary must include data validation. If the parameter is not positive, set monthlySalary to 0.
- The accessors will return the value of a data member.

Step 5: Create EmployeeTest.cpp

- Include any C++ libraries that are needed.
- Include the .h file.
- Write the **using namespace std;** statement.
- 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. Direct calls, not indirect calls.)
- 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.

  The following C language components are not allowed:

  NO scanf
  NO printf or puts
  NO C libraries
  NO \n
  NO gcc in the makefile

  One point will be deducted from your assignment score for each occurrence of a prohibited C component.

Submit 5 files:

1) Microsoft Word document (.docx or .doc) with the UML diagram (Microsoft Office 365 is free to all students.)
2) makefile
3) Employee.h
4) Employee.cpp
5) EmployeeTest.cpp

Do not zip or tar the files.

It is your responsibility to upload the correct files and to ensure that the uploaded files contain the work that you want to have graded.