Lab Purpose

Practice writing C++ object oriented programs that enable polymorphism, use inheritance, encapsulation and virtual functions.

Always bring to class

1. Gaddis' book, How-to handouts from Canvas and your class notes.
2. This assignment sheet & the grade sheet for this lab already printed out.
3. USB Flash drive(s) or other storage media.

Mandatory Instructions

In this lab assignments you will create C++ classes and demonstrate inheritance, polymorphism, encapsulation and use of the virtual functions.

1. Design a base class ***Employee*** that will contain the following data members: name (string), payRate (double) which represents an hourly rate or salaried period rate. Use lab9employee.h and lab9employee.cpp files.

   The class should contain an overloaded constructor accepting name and pay rate as parameters, ***getName*** and ***getPayRate*** accessors, and method ***pay*** which should return amount of pay (double) given hours worked as a parameter.

   ```
   double pay(double) const;
   ```

2. You will also need to design a ***Manager*** class that will inherit from ***Employee***. Use lab9manager.h and lab9manager.cpp files. This class will have one data member: salaried (bool). It should have an overloaded constructor accepting name, pay rate, and true/false value to designate salaried or not.

   The class should have a ***getSalaried*** accessor function and a ***pay*** function returning the amount of pay for a manager. If a manager is salaried, then payrate should be returned, otherwise pay should be computed as pay rate times hours worked.

   In your client code, lab9client.cpp, write a generic function ***PrintPay*** that will accept as parameters pointer to an ***Employee*** object, and hours worked. It should call the ***pay*** function and produce the following output:

   ```
   John Burke    earned    $1000.00
   Jan Kovacs    earned    $1200.00
   Tom Banks     earned    $1400.00
   ```

   Provide suitable heading, formatting for this output.

3. In ***main*** create an array of 25 elements and populate with data read from data file, lab9.txt, until you run out of data. The file contains various kinds of employees. Dynamically allocate appropriate employee object based on employee type stored in the data file as the first number in each line:

   ```
   1John Burke 25.00          ← hourly employee earning $25.00 per hour
   2Jan Kovacs 1200.00 1      ← salaried manager earning $1,200 per week
   2Tom Banks 75.00 0         ← hourly manager earning $35.00 per hour
   ```

   Create a loop that will call your generic function, ***PrintPay***, for each employee and compute pay for 40.00 hours worked, i.e., one week.

What to turn in?

Make a printout of your program to turn in by typing the following commands at the prompt:

```
$ photo lab9.log
$ ls  -l
$ cat lab9employee.h
$ cat lab9employee.cpp
$ cat lab9manager.h
$ cat lab9manager.cpp
$ cat lab9client.cpp
$ g++  lab9employee.cpp lab9manager.cpp lab9client.cpp
$ .\a.out
$ exit
```