Object Oriented Programing Lab

BSCS(Fall 2015)

Lab # 6

Friday, April 8, 2016

Instructions: For support and guidance you can CONTACT ME AND TA's. Discussion with peers is strictly prohibited.

Task# 1: NumDays Class

Design a class called **NumDays**. The class's purpose is to store a value that represents a number of work hours and convert it to a number of days. For example, 8 hours would be converted to 1 day, 12 hours would be converted to 1.5 days, and 18 hours would be converted to 2.25 days. The class should have a constructor that accepts a number of hours, as well as member functions for storing and retrieving the hours and days.

The class should also have the following overloaded operators:

- + Addition operator. When two **NumDays** objects are added together, the overloaded + operator should return the sum of the two objects' hours members.
- - Subtraction operator. When one **NumDays** object is subtracted from another, the overloaded − operator should return the difference of the two objects' hours members.
- ++ Prefix and postfix increment operators. These operators should increment the number of hours stored in the object. When incremented, the number of days should be automatically recalculated.
- Prefix and postfix decrement operators. These operators should decrement the number of hours stored in the object. When decremented, the number of days should be automatically recalculated.

Task #2: Time Off

NOTE: This assignment assumes you have already completed Programming Task #1.

Design a class named **TimeOff.** The purpose of the class is to track an employee's sick leave, vacation, and unpaid time off. It should have, as members, the following instances of the **NumDays** class described in Task #1:

maxSickDays: A NumDays object that records the maximum number of days of sick leave the employee may take.

sickTaken: A NumDays object that records the number of days of sick leave the employee has already taken.

maxVacation: A NumDays object that records the maximum number of days of paid vacation the employee may take.

vacTaken: A NumDays object that records the number of days of paid vacation the employee has already taken.

maxUnpaid: A NumDays object that records the maximum number of days of unpaid vacation the employee may take.

unpaidTaken: A NumDays object that records the number of days of unpaid leave the employee has taken.

Additionally, the class should have members for holding the **employee's name** and **identification number.** It should have an appropriate constructor and member functions for storing and retrieving data in any of the member objects.

Input Validation: Company policy states that an employee may not accumulate more than 240 hours of paid vacation. The class should not allow the **maxVacation** object to store a value greater than this amount.

Task# 3: Personnel Report

NOTE: This assignment assumes you have already completed Programming Task# 1 and 2.

Write a program that uses an instance of the **TimeOff** class you designed in **Task#2**. The program should ask the user to enter the number of months an employee has worked for the company. It should then use the **TimeOff** object to calculate and display the employee's maximum number of sick leave and vacation days. Employees earn 12 hours of vacation leave and 8 hours of sick leave per month.

Task# 4: Month Class

Design a class named Month . The class should have the following private members:
• **name:** A string object that holds the name of a month, such as "January," "February," etc.

• **monthNumber:** An integer variable that holds the number of the month. For example, January would be 1, February would be 2, etc. Valid values for this variable are 1 through 12.

In addition, provide the following member functions:

• A default constructor that sets **monthNumber** to 1 and **name** to “January.”

• A constructor that accepts the name of the month as an argument. It should set name to the value passed as the argument and set **monthNumber** to the correct value.

• A constructor that accepts the number of the month as an argument. It should set **monthNumber** to the value passed as the argument and set **name** to the correct month name.

• Appropriate set and get functions for the **name** and **monthNumber** member variables.

• Prefix and postfix overloaded ++ operator functions that increment **monthNumber** and set **name** to the name of next month. If **monthNumber** is set to 12 when these functions execute, they should set **monthNumber** to 1 and **name** to “January.”

• Prefix and postfix overloaded −− operator functions that decrement **monthNumber** and set **name** to the name of previous month. If **monthNumber** is set to 1 when these functions execute, they should set **monthNumber** to 12 and **name** to “December.”