CPS235 Object Oriented Programming in C++

LAB MIDTERM EXAM *29th April 2010*

Notes:

For all of the following functions, use const where appropriate and pass objects by reference to functions wherever required.

Question 1:

In this question, you will deal with some operations on rational numbers. Recall that a number is rational if it can be represented as a fraction ***denom/num***, where both numerator ***num*** and denominator ***denom*** are integers. Your main task is to implement a class named Rational, which contains two private data members **`num(of type int)`** to represent the numerator and **`denom (of type int)`** to represent the denominator. The public interface of the class should contain the following functions:

1- A default constructor which sets both **`num`** and **`denom`** to 1
2- A 2-argument constructor to initialize **`num`** and **`denom`** to values passed in the arguments
3- An overloaded stream insertion operator **`>>`**, which takes **`num`** and **`denom`** as input from the user *(hint: this will be implemented as a non-member friend function).*
4- An overloaded stream extraction operator **`<<`**. It should display the rational number in the format **`num/denom`**. Make sure that the function prints 4 instead of 4/1, 0 instead of 0/4 and gives an error if the denom is 0. This operator should allow cascaded **`cout`** statements. i.e.,**`cout<<num1<<num2<<num3;`**
5- A **`reduce()`** function which reduces a rational number to its simplest form,
e.g., 6/24 = ¼ 4/4 = 1 , 15/3 = 3
6- A function called **`RationalToDouble()`** which converts a rational number to its equivalent decimal form (a double value) and returns that value
e.g., the function when called on an object having the value ¾ should return 0.75
7- Write functions to overload these operators: addition (+), multiplication(*),division(/) and subtraction(-). All of these functions operate on **`Rational`** objects and return a **`Rational`** object as well.
8- Generate an error whenever an attempt is made to set **`denom`** to 0 or a division by 0 is attempted.

Write a main program which has the following interface….A sample run of the intended program is given as well.

CPS235: OOP Midterm: [your full name]
0- Quit
1- Enter Rational Number
2- Print Rational Number
3- Add a Rational Number

```
4-  Subtract a Rational Number
5-  Multiply with a Rational Number
6-  Divide with a Rational Number
7-  Convert to Double
Enter your choice: 1
Enter numerator followed by denominator:   4  5
The number is: 4/5

0-  Quit
1-  Enter Rational Number
2-  Print Rational Number
3-  Add a Rational Number
4-  Subtract a Rational Number
5-  Multiply with a Rational Number
6-  Divide with a Rational Number
7-  Convert to Double
Enter your choice: 3
Enter numerator followed by denominator: 10 11
Updated rational number is: 14/15(i.e.,adds the new entry to the number previously entered
which was 4/5)

0-  Quit
1-  Enter Rational Number
2-  Print Rational Number
3-  Add a Rational Number
4-  Subtract a Rational Number
5-  Multiply with a Rational Number
6-  Divide with a Rational Number
7-  Convert to Double
Enter your choice: 5
Enter numerator followed by denominator: 4 1
Updated rational number is: 56/15(i.e .multiplies the new entry with the number previously
updated which was 14/5)

0-  Quit
1-  Enter Rational Number
2-  Print Rational Number
3-  Add a Rational Number
4-  Subtract a Rational Number
5-  Multiply with a Rational Number
6-  Divide with a Rational Number
7-  Convert to Double
Enter your choice: 6
Enter numerator followed by denominator: 0
ERROR: Cannot divide by zero
```

Question 2:

a. Create a class **`Date`** which has three private **`int`** data members **`day, month`** and **`year`**. The class has the following public functions
   i. A constructor which takes **`3 int`** arguments and assigns those arguments to the three data members. The default values of the arguments should be **`1,1,1970`**. If invalid values are entered for **`day`** and **`month`**, they should be set to the default values.
   ii. An overloaded **`operator<<`** which displays the date members

b. Create a class **`Address`** which has three private data members, all of type **`char array`** with appropriate sizes. These arrays will hold the information of **`house no. & street name, province`** and **`city`**. The class has the following public functions
   i. A 3-argument constructor, with default values **`“no name”`**
   ii. A print() function which displays all three data members.

c. Create a class employee which has a **name** of type **string**, an **address** of type **Address** and a **date_of_birth** of type **Date.** All these should be kept private in the class. The public methods include:
   i. A constructor with 3 arguments
   ii. A print() function to display all data members

d. Create a class SalariedEmployee which derives publicly from employee, and has only one private data member called **salary** of type **int**. the class should have
   i. A public constructor with 4 arguments and
   ii. A public print function to display the values of all data contained in a SalariedEmployee object
   iii. The salary data member represents the monthly salary of the employee. Write a function calculateIncrement() which returns the increment calculated as 10% of the yearly salary.

e. Create a class HourlyEmployee which derives publicly from employee, and has private data members **wages** of type **double** and **hours** of type **int.** The class should have
   iv. A public constructor with 5 arguments
   v. A public print function to display the values of all data contained in a HourlyEmployee object.
   vi. **wages** holds the amount of money earned by an employee if he works for an hour. **hours** represent the number of hours the employee works for in a day. Write a function calculateMonthlySalary() which returns the salary earned by the employee in one month. You may assume that each month is of 30 days.

In main, create an object of SalariedEmployee and display its information. Also, display the incremented salary of the employee at the end of the year. Create an object of HourlyEmployee and display its complete information including the calculated monthly salary.