1. Write a **clock** class, define its private data members (**hour, minute, second**), public function members (**setHour, setMinute, setSecond**), **reSet, readTime.** Define the **constructor** to initialize it (8:00).
2. Finish the class definition, and use it in your own project

class Rational

{

public:

Rational(int nn=1,int mm=1); //constructor

Rational R\_add(Rational & A); //addition

Rational R\_sub(Rational & A); //substraction

Rational R\_mul(Rational & A); //multiplication

Rational R\_div(Rational & A); //division

void print(); //output the result, which should be a reduced form

private:

void simple( ); // [reduction](http://www.nciku.cn/search/en/reduction) of a [fraction](http://www.nciku.cn/search/en/fraction)

int m; // denominator

int n; // numerator

};

1. Write a program to build your book library **My\_library**. The book should has **name**, **pages** and so on. Define your **constructor**, **copy constructor** and **destructor**. Add some functions to your library, including **add\_book**, **delete\_book**, and more. Run your library!
2. Design and implement a **Name\_pairs** class holding(name,age)pairs where name is a **string** and age is a **double**. Represent that as a **vector<string>**(called **name**) and a **vector<double>** (called **age**) member. Provide an input operation **read\_names()** that reads a series of names. Provide a **read\_ages**() operation that prompts the user for an age for each name. Provide a **print()** operation that prints out the**(name[i],age[i]**) pairs (one per line) in the order determined by the **name** vector. Use **sort** () to sort the **name** vector in alphabetical order and reorganizes the age **vector** to match. Implement all “operations” as member functions. Test the class!
3. Redefine the operator ==and **!=** for **Name\_pairs**. Try to replace **Name\_pair::print()** with an **operator<<** (if you don’t know, just find it on the web).

**3. Questions：**

Think through the experiments, and answer the following questions.

1. How to use the public and private? When should we define a private function member?
2. When have you to define your own constructor?
3. Why we define the overloading operator? Is it