General C++ Engineering Applications

Exercise

> 4. Operator Overload

1. Vector-Class

将Vector类中的以下成员函数通过运算符重载替换：

```
Vector & copy(Vector & v);
Vector & copy(double x);
double getData(int count);
void setData(int count, double x);
Vector plus(Vector v);
Vector minus(Vector v);
double dotProduct(Vector v);
Vector scale(double s);
// left and right operator
void print(int w=8);
```

```
Vector & operator=(Vector & v);
Vector & operator=(double x);
const double operator[](int count) const;
double & operator[](int count);
Vector operator+(Vector v);
Vector operator-(Vector v);
double operator*(Vector v);
Vector operator*(double s);
friend Vector operator*(double s, Vector v);
friend ostream & operator<<(ostream & os, Vector v);
```

* 注意删除.cpp文件中原先的代码，防止重复定义;
* 注意getData与setData同样重载为[]的区别。

> 4. Operator Overload

2. Matrix-Class

将Matrix类中的以下成员函数通过运算符重载替换：

```
Matrix & copy(Matrix & v);
Matrix & copy(double x);
double getData(int m, int n);
void setData(int m, int n, double x);
Matrix plus(Matrix & m);
Matrix minus(Matrix & m);
const Vector product(Vector v);
const Matrix product(Matrix m);
// add these functions

void print(int w=8);
```

```
Matrix & operator=(Matrix & v);
Matrix & operator=(double x);
const double operator()(int m, int n) const;
double & operator()(int m, int n);
Matrix operator+(Matrix & m);
Matrix operator-(Matrix & m);
const Vector operator*(Vector v);
const Matrix operator*(Matrix m);
Matrix operator*(double s);
friend Matrix operator*(double s, Matrix m);
friend ostream & operator<<(ostream & os, Matrix m);
```

* 注意删除.cpp文件中原先的代码，防止重复定义；
* 注意getData与setData用()而不是[]重载：[]只接受一个参数。

> 4. Operator Overload

3. Vector & Matrix Test

在主程序中进行以下测试，对比注释部分与使用运算符重载方式的区别：

```
int main()
{
      cout<<"Matrix A = B:"<<endl;
      Matrix A(3,5);
      // A.copy(5.);
      A = 5.;
      // Matrix B (A);
      Matrix B = A;
      // B.print();
      cout<<B;

      cout<<"Vector v1:"<<endl;
      Vector v1(5);
      // v1.copy(2.);
      v1 = 2.;
      // v1.print();
      cout<<v1;

      cout<<"Replace Matrix B at
(2,2) with 2:"<<endl;
      // B.setData(2,2,2);
      B(2,2) = 2.;
      // B.print();
      cout<<B;

      cout<<"Vector v2 = B *
v1:"<<endl;
      // Vector v2(B.product(v1));
      Vector v2 = B * v1;
      // v2.print();
      cout<<v2;

      cout<<"A.Transpose:"<<endl;
      // A.transpose().print();
      cout<<A.transpose();

      cout<<"C = B * A:"<<endl;
      // Matrix C(B.product(A));
      Matrix C = B * A;
      // C.print();
      cout<<C;
      return 0;
}
```

> 5. File Read & Write

1. Vector & Matrix: result as file

将Vector & Matrix Test中cout的结果输出到文件中；

文件名为：*运行程序时的时间.dat*

获取系统时间的示范：(需要#include<time.h>)

```
time_t nowtime = time(0);
cout<<ctime(&nowtime)<<endl;
// 可以输出当前时间
```