You must create three files for this project, a .h file for the class definitions, a .cpp file for the bodies of the class functions, and a .cpp file containing the main program and all required global functions (if any). These files should be named "MoreOOP.h" "MoreOOP.cpp" and "main.cpp".

In your .h file, you will create two classes, a scaled point and a scaled rectangle. All data items for both classes will be protected, all functions for both classes will be public.

The scaled point will have two members, the X and Y coordinates, which will be integers, and two static members, the Numerator and Denominator, which will be integers. The Numerator and denominator must be initialized to 1, NOT in the constructor!

The Scaled Point class will have the following class functions:

GetX – Get with scaling – Returns (integer) (X*Numerator)/Denominator
GetY – Same as GetX, but for Y coordinate.
SetX – Has one integer argument and sets X=(Arg*Denominator)/Numerator
SetY – Same, but for Y coordinate.

It will also have the following static functions:

int Scale(int Value,int Mul,int Div) – Computes (Value*Mul)/Div
int Scale(int Value) – Computes (Value*Numerator)/Denominator
void SetScale(int NewNumerator, int NewDenominator) – Sets numerator & denominator
int GetNumerator(void) – Self explanatory.
int GetDenominator(void) – Self explanatory.

The scaled point class will have overloads for the assignment operator and for the += operator. The += overload will add an integer (the same integer) to both coordinates. It will have a copy constructor, a default constructor and a constructor that takes two integer arguments, the new X and Y coordinates. No constructor will change the value of any static element. The scaled point class will have a default destructor which does nothing. The scaled point class will have one friend, the scaled rectangle class.

The scaled rectangle class will have two data items, both of which will be scaled points. The scaled points will give the upper left and lower right corners of the rectangle.

The scaled rectangle class will have four constructors, the default, a copy constructor, and two specialized constructors, one that takes two scaled points (upper left and lower right) and one that takes four integers (left, top, right, and bottom.) It will overload the assignment and += operators. The += operator will take an integer argument and add that integer (using +=) to both points. There will be accessor functions that return the upper-left and lower-right points. There will be mutator functions that supply new values for the two points. There will be two mutator functions for each point, one that takes a scaled point for an argument, and one that takes two integers.

The += operator is used to offset points and rectangles by a fixed amount in both directions. You could add another += overload to offset by different amounts in both directions using a statement of the form myRect += ScalePoint(3,4); or myPoint += ScalePoint(-3,7);.

The main program must do the following.

Open a text file named “MoreOOP.txt”. Read one integer from this file. This will give you the number of rectangles in the file. Next, read all the rectangles in the file. Each rectangle consists of four integers, so for each rectangle, read four integers. These integers are in the following order: Left, Top, Right, Bottom.

After all data has been read, print out the rectangles in the form:

0: 1,2,3,4
1: 5,6,7,8

and so forth, where the first number is the index and the next for integers are Left, Top, Right and Bottom.

Set the scaling to Numerator=5, Denominator=1 and print out everything again.

Add 10 to each rectangle using += and print out everything again.

Don’t forget to delete any memory that you allocate using the “new” command.