Classes...again we do this.

The Extra Weekly Assignment for this week is a modified version of problem 16.7 in the 4th edition of Deitel and Deitel. It is designed to provide extra practice with constructors and member functions in preparation for the final exam.

A demo program, `ewa08_demo_2008.out`, is available in ~EngH192/Students for you to copy to your home directory and run as an example.

Not only that, but a half completed version of the program is available as well. All you have to do is use the existing `Rectangle` sections as guidelines for filling in the blanks and completing the appropriate `Isosceles` sections. Everything else is taken care of for you. The file is named `ewa08_half_2008.cpp` and is available in ~EngH192/Students for you to copy to your home directory and modify as needed. Of course, if you want, you can do it completely from scratch. ;)

The classes for two geometric shapes have been defined for you below. You are to write the definitions for each of the member functions and write a `main()` function that creates a `Rectangle` object and an `Isosceles` object and supports the following commands.

1. a, A -- Set the length and width of the rectangle
2. b, B -- Show the length and width of the rectangle
3. c, C -- Calculate the area of the rectangle
4. d, D -- Calculate the perimeter of the rectangle
5. e, E -- Show the area of the rectangle
6. f, F -- Show the perimeter of the rectangle
7. g, G -- Set the side and base of the isosceles triangle
8. h, H -- Show the side and base of the isosceles triangle
9. i, I -- Calculate the area of the isosceles triangle
10. j, J -- Calculate the perimeter of the isosceles triangle
11. k, K -- Show the area of the isosceles triangle
12. l, L -- Show the perimeter of the isosceles triangle
13. q, Q -- Quit

All of your commands, except for Quit, must invoke the appropriate member functions. The “set” member functions for each class must not allow values less than or equal to zero or greater than twenty and must display an error message in the event the user attempts to enter values outside the valid range. You will probably want a `while` loop and a `switch-case` or `if-else if` structure to control the overall logic of the program. NOTE: The `Show()` member functions display the appropriate values while `Perimeter()` and `Area()` calculate the appropriate values and store the results in the corresponding variables within the object (i.e. `Perimeter()` and `Area()` don't display anything).
Once your program is running correctly, submit the source code along with output that demonstrates that the program can successfully perform all of its functions, including the fact that the constructors properly initialize their corresponding objects.

HINT 1: The Numbers example in Lecture 25 is a very good example for getting started with this problem, particularly the constructor aspect of it.

class Rectangle
{
    public:
        Rectangle (float, float, float, float);
        void setDimensions ();
        void showDimensions ();
        void showPerim ();
        void showArea ();
        void Perimeter ();
        void Area ();

    private:
        float side1;
        float side2;
        float area;
        float perim;
};

class Isosceles
{
    public :
        Isosceles (float, float, float, float);
        void setDimensions ();
        void showDimensions ();
        void showPerim ();
        void showArea ();
        void Perimeter ();
        void Area ();

    private:
        float side1;
        float side2;
        float area;
        float perim;
};