ICS4U **Assignment – Rectangle Class** Apr 8, 2010

Suppose that rectangles in a Cartesian plane are represented by a class `Rectangle` that has four private fields:

- integer left – the x-coordinate of the left edge
- integer bottom – the y-coordinate of the bottom edge
- integer width – the width of the rectangle
- integer height – the height of the rectangle

1. Define the class.

2. Write a constructor method that has four parameters representing the four fields of the class. The constructor should replace any negative length parameters with zero.

3. Write a `toString` method. For the rectangle with a lower left corner located at (3, 2) and having a width of 4 and a height of 5, the method should return "`base: (3,2) w:4 h:5`".

4. Write an instance method, `area`, that returns the area of a rectangle.

5. Write an instance method, `perimeter`, that returns the perimeter of a rectangle.

6. Write a class method, `intersection`, that has two Rectangle parameters. The method should return the rectangle formed by the area common to the two rectangles (i.e., the rectangle formed where they overlap with each other). If they do not intersect, the method should return a rectangle where all fields are zero. If the rectangles only touch, but do not overlap, then the width or height should be zero, but all other parameters should be properly calculated and stored.

7. Write a class method, `totalPerimeter`, that has two Rectangle parameters. The method should return the total perimeter of the figure formed by the two rectangles. It should only count those portions that are on the edges of the exterior of the resulting figure. If one rectangle is completely contained by the other, then return the perimeter of the outer rectangle. If the rectangles do not intersect, the method should return the sum of the individual perimeters.

8. Write an instance method, `contains`, that has one parameter of type Rectangle. The method should return true if every point of the specified rectangle (i.e., passed by the explicit parameter) is on or within the implicit parameter (i.e., the object invoking the instance method). It should return false otherwise.

   For example, `a.contains(b);` would return true if the rectangle `b` is entirely within `a`.

You are welcome to design and write additional code to help you accomplish this task, but you must hide them using encapsulation.