Homework #4

Jay Laura

This week's homework required the creation of 10 random points, four rectangles, and the computation of whether or not the point falls within the rectangle. We were instructed to reuse as much code from previous homework's as possible. I have reused the Point class, but not my Poly class from the previous weeks homework. While I could have subclassed Poly, and added a Rectangle method, I believe it makes more sense to write a simple Rectangle class for now. This choice stems from being a more functional programmer, that is, numerous reusable functions are written instead of either large classes or trees of classes. In this way, compartmentalization is achieved with excessive object generation through class instantiation.

This program therefore generates 10 random points on a Cartesian plane bounded between the origin (0,0) and (200,200). Then rectangles are generated by randomly computing the upper left and lower right vertices. From these vertices the upper right and lower left are calculated. By definition, all angles within a rectangle are 90 degrees and we can compute unknown corner vertices from known corner vertices. We then implement the ray within polygon algorithm to compute whether the point falls within the polygon. This was initially implemented by computing the determinants of matrices composed of the start and end points of line segments (rectangle) and a vector (point). Unfortunately, this method is not suitable for line segments, as each line is projected to infinity; this causes the point in polygon test to fail. Therefore a series of if statements we used to capture the bounding area of the polygon and test whether the point falls within it.

This is a computationally slow process that could be greatly improved by first computing the bounding box around the vector emanating from the point and checking for intersection. If the bounding boxes overlap, then the computation could continue to check for the number of times the vector intersects the polygon.

Finally, the output is written. This is accomplished using .join() to generate a well formatted list of rectangles and .format allow the previously generated list to be appended to the point.