1. Shapes
   1. Create a base class called “Shape” that includes 1 instance method – draw() make it virtual so that it can be overriden.
   2. Create a subclass of the Shape class called “Circle”.
      1. Make the Circle class have a private instance variable (field) – radius that determines how big the circle is. Add a public property for this field.
      2. Create a constructor that accepts a radius – BUT still allow circles to be created with no radius passed in (provide a default constructor too). If no radius is passed in, default radius should be 5.
      3. Override the draw method in the circle class so that a circle of the correct radius is drawn (or a better use of your time – just print “Circle, radius X” instead of trying to figure out how to draw a circle in ASCII art.)
   3. Create a subclass of the Shape class called “Square”.
      1. Make the square class have a private instance variable (field) – size that determines how big each side of the square is. Add a public property for this field.
      2. Create a constructor that accepts a size – BUT still allow squares to be created with no size passed in (provide a default constructor too). If no size is passed in, default size should be 10.
      3. Override the draw method in the square class so that a square of the correct size is drawn (or again – just print “Square, size X” instead of trying to figure out how to draw a square in ASCII art – though a square is pretty easy.)
   4. In the main method create an Array of type Shape to hold shapes and fill it with several different squares and circles.
   5. Loop through the array calling draw on all the shapes and verify that all shapes are drawn correctly.
   6. Change the radius and sizes of some of the existing circles and squares (remember casting).
   7. Loop through the array calling draw on all the shapes and verify that all shapes are drawn correctly with the new values.

Extra Credit:

Override the equals method for the square class and test it out