```
using System;
public class Quad:IComparable<Quad>//implement IComparable
{
    private string name;//instance field
    public Quad(string na)
    {
        name = na;//set value of instance field
    }
    //implement CompareTo to make list sortable
    //in this case, the items are sorted by name
    public int CompareTo(Quad other)
    {
        if (this.name.CompareTo(other.name) < 0)
        {
            return -1;
        }
        else
        {
            return 1;
        }
    }//put default code inside Perimeter
    public virtual string Perimeter()
    {
        return $"The perimeter of {name} is ";
    }
}
public class Square: Quad
{
    private double sideLength;
    public Square(string n, double s):base(n)
    {
        sideLength = s;
    }
    //override Perimeter, calling the base portion
    //and then adding refinement with 4*sideLength
    public override string Perimeter()
    {
        return base.Perimeter()+4*sideLength;
    }
}
public class Rectangle: Quad
{
    private double sideOne, sideTwo;
    public Rectangle(string n, double s1, double s2) : base(n)
    {
        sideOne = s1; sideTwo = s2;
    }
    //override Perimeter, calling the base portion
    //and then adding refinement with 2sideOne+2sideTwo
    public override string Perimeter()
    {
        return base.Perimeter() + (2*sideOne+2*sideTwo);
    }
```

```
54  }
```