Data Hiding
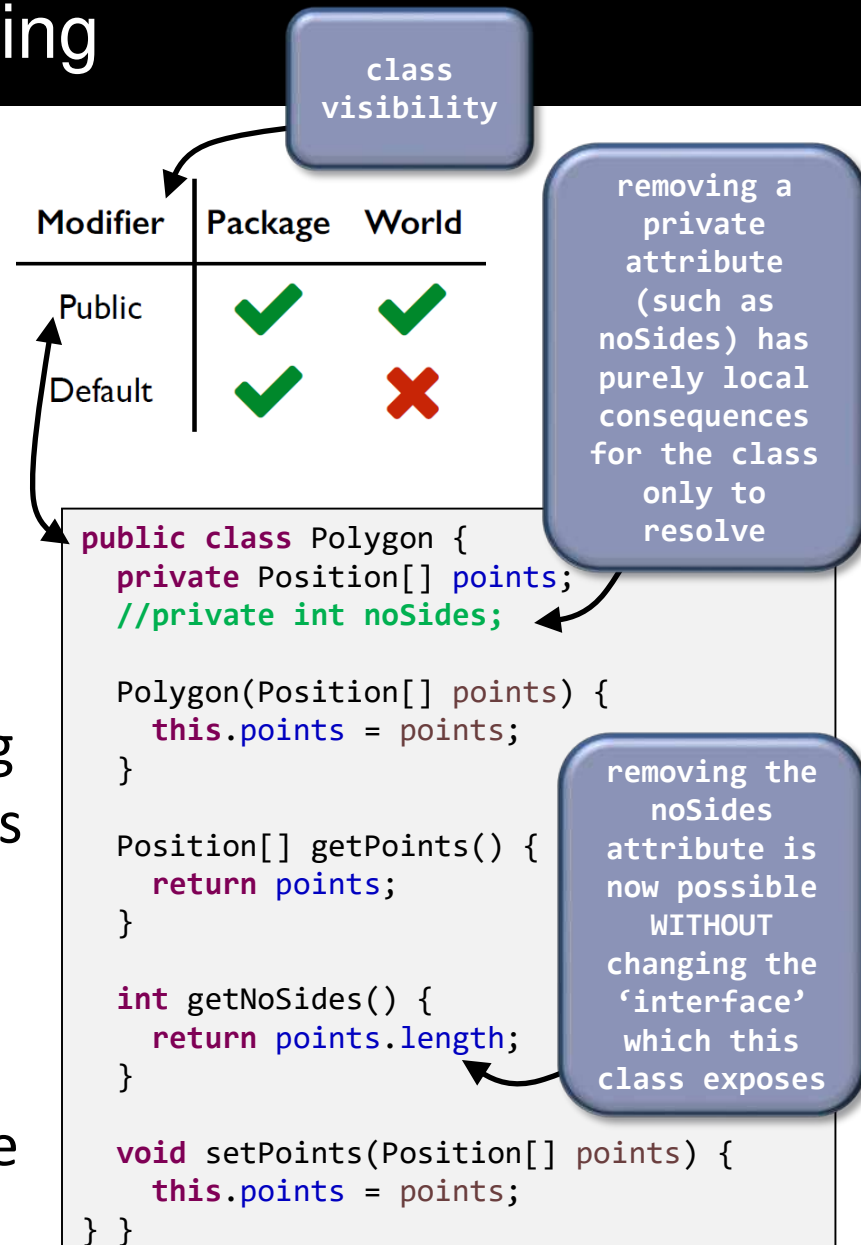
- you should **routinely hide** your attributes and methods as much as possible using `private`

- now that we've made our implementation of attributes `private`, there's another benefit: we can change the implementation **without** changing the way objects access to our class (i.e. we may want to remove an attribute completely)

- using `public` makes elements globally accessible (also applicable to classes)

class visibility

Modifier	Package	World
Public	✔	✔
Default	✔	✘

removing a private attribute (such as noSides) has purely local consequences for the class only to resolve

```
public class Polygon {
  private Position[] points;
  //private int noSides;

  Polygon(Position[] points) {
    this.points = points;
  }

  Position[] getPoints() {
    return points;
  }

  int getNoSides() {
    return points.length;
  }

  void setPoints(Position[] points) {
    this.points = points;
  }
} }
```

removing the noSides attribute is now possible WITHOUT changing the 'interface' which this class exposes

some slide content used from N. Wu