The problem is: the “another class” has no way to know that the rect variable is now referring to a Square object -- NOT a Rectangle object.
Three possible solutions (others may be there):

A) Overriding, but dummy, i.e., giving a warning message that for the Square object, this overridden version must not be used. (Why then overriding at all? Otherwise, previous values of width/length will be used as garbage output.) And, additionally, provide a special version of setting dimension of Square only.

```
class Square extends Rectangle {
    public Square (double side){
        super(side, side);
    }
    public void set_dim_for_Square (double side){
        this.width = this.length = side;
    }
    public void set_width (double width){
            System.out.println("You are in a Square object, so please use
set_dim_for_Square(double) method.");
      this.width = this.length = 0;
    }
 public void set_length (double length){
            System.out.println("You are in a Square object, so please use
set_dim_for_Square(double) method.");
      this.width = this.length = 0;
    }
}
```

B) Using final keyword in set_width and set_len to prevent overriding at compile time. Benefit of overriding CANNOT be used in this solution.

C) Creating a new superclass named Shape and inheriting it by Rectanble and Square classes:

```
      class Shape {
    private double width;
    private double length;

    public Shape (double width, double length){
        this.width = width;
        this.length = length;
    }

    public void set_width (double width){
        this.width = width;
    }
    public void set_length (double length){
        this.length = length;
    }

    public double calculate_area (){
        return width * length;
    }
}

class Rectangle extends Shape {
    public Rectangle (double width, double length){
        super(width, length);
    }
}
```

```
class Square extends Shape {
    public Square (double width){
        super(width, width);
    }
    public void set_width (double width){
        this.width = width;
        this.length = width;
    }
    public void set_length (double length){
        this.width = length;
        this.length = length;
    }
}

class another_class {
    public static void to_another_programmer(Shape r){... ...
```