

Solution Review: Implement Getter Methods

This lesson gives the solution to the previous exercise - implementing the getter methods to calculate the width and height of a rectangle.

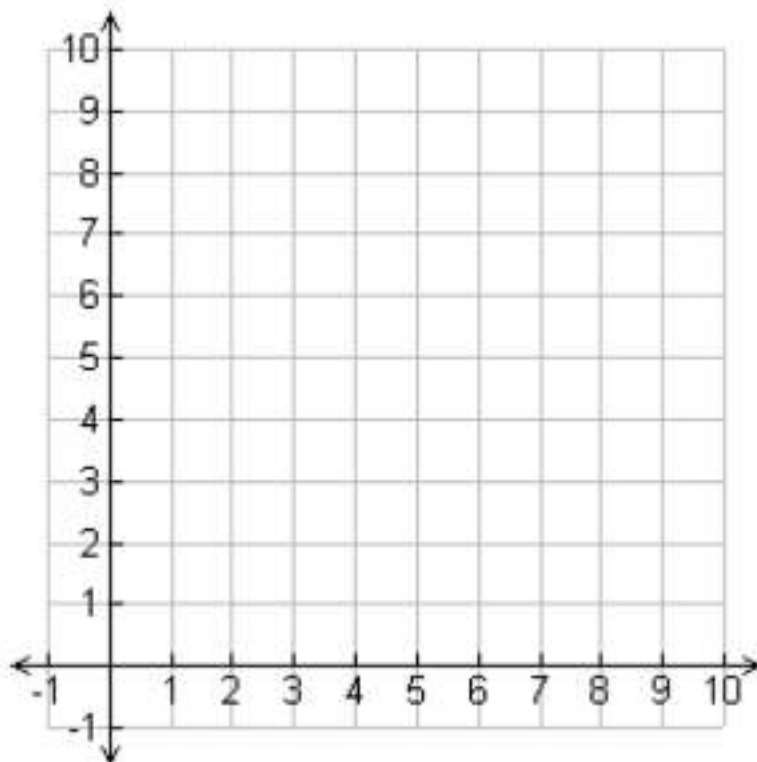
WE'LL COVER THE FOLLOWING ^

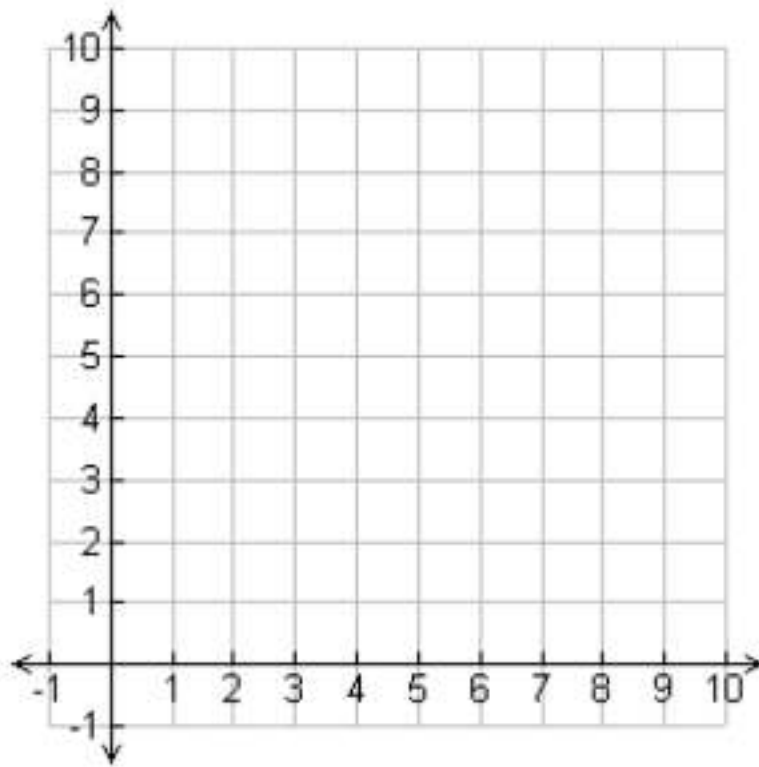
- Solution:

Solution:

The getter methods are written in the lines 11-15 in the following code playground. The `width()` and `height()` methods simply subtract the x and y coordinates respectively and return the result.

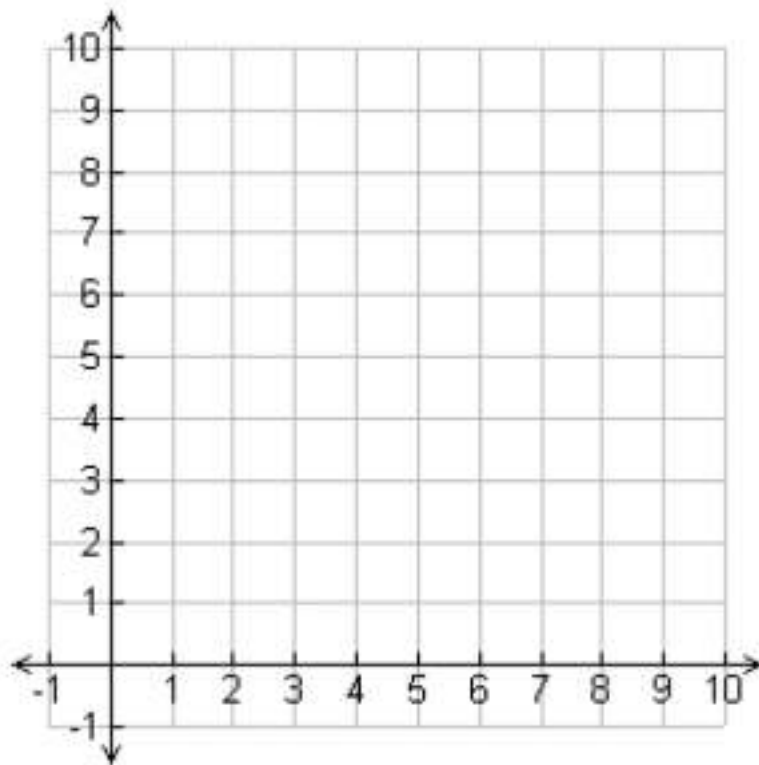
This solution is illustrated in the following figure:





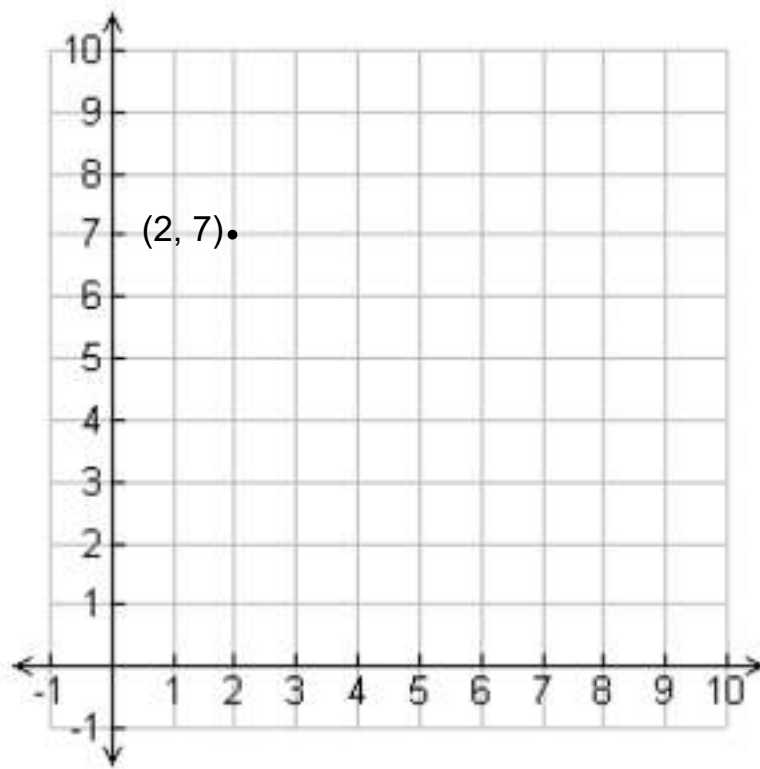
`Rectangle (x1, y1, x2, y2)`

2 of 22



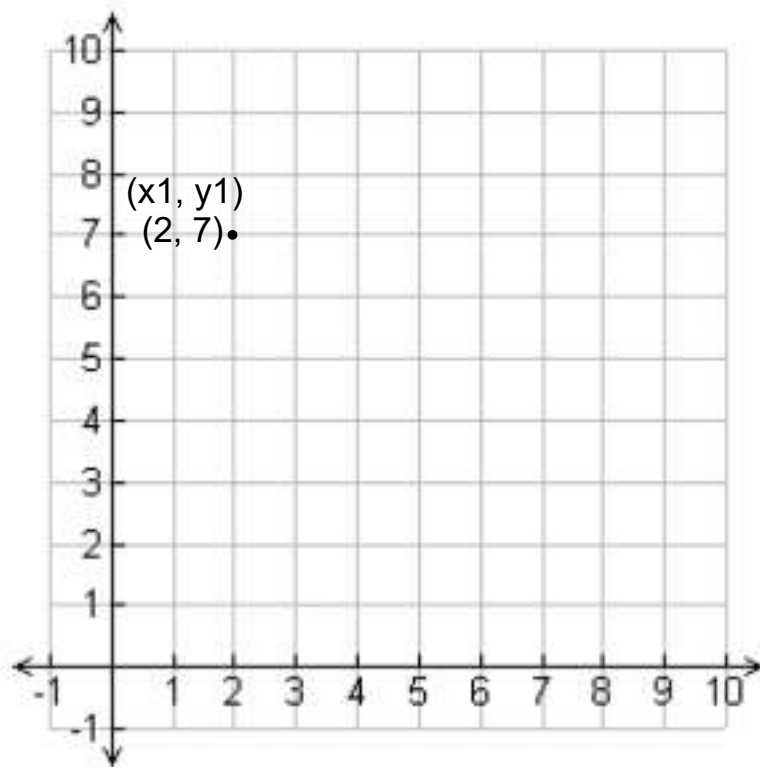
`r = Rectangle (2, 7, 8, 4)`

3 of 22



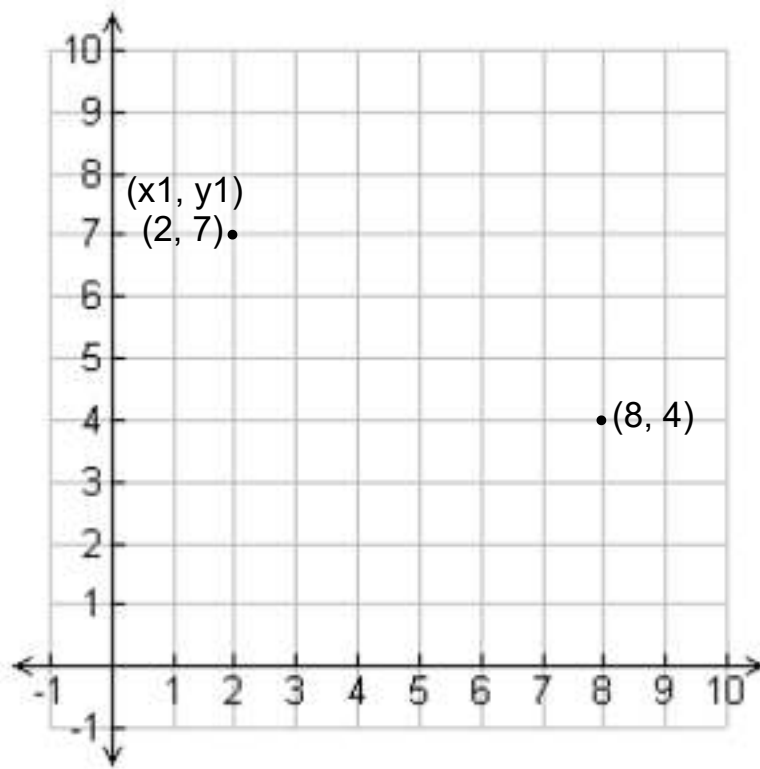
`r = Rectangle (2, 7, 8, 4)`

4 of 22



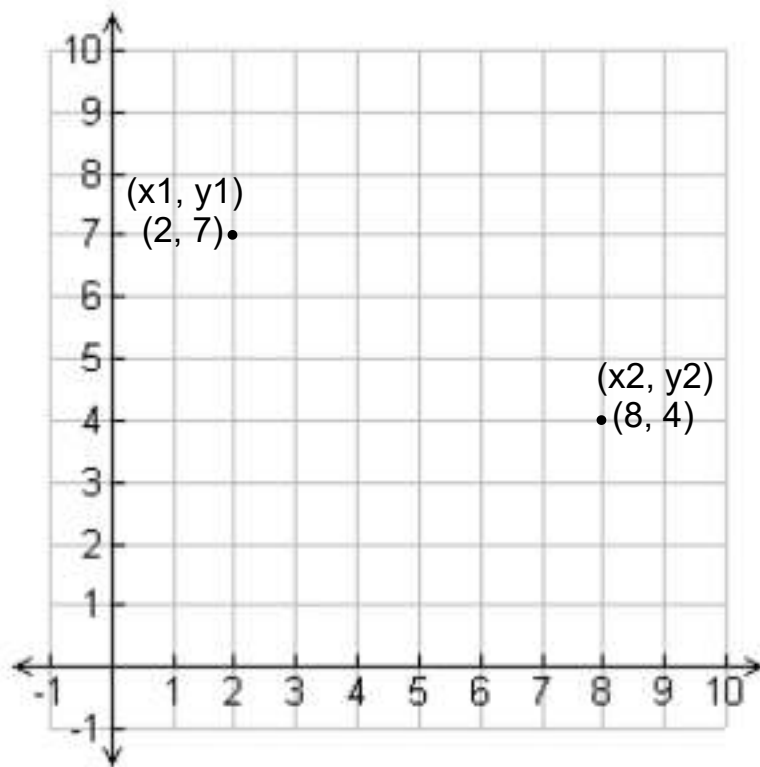
`r = Rectangle (2, 7, 8, 4)`

5 of 22



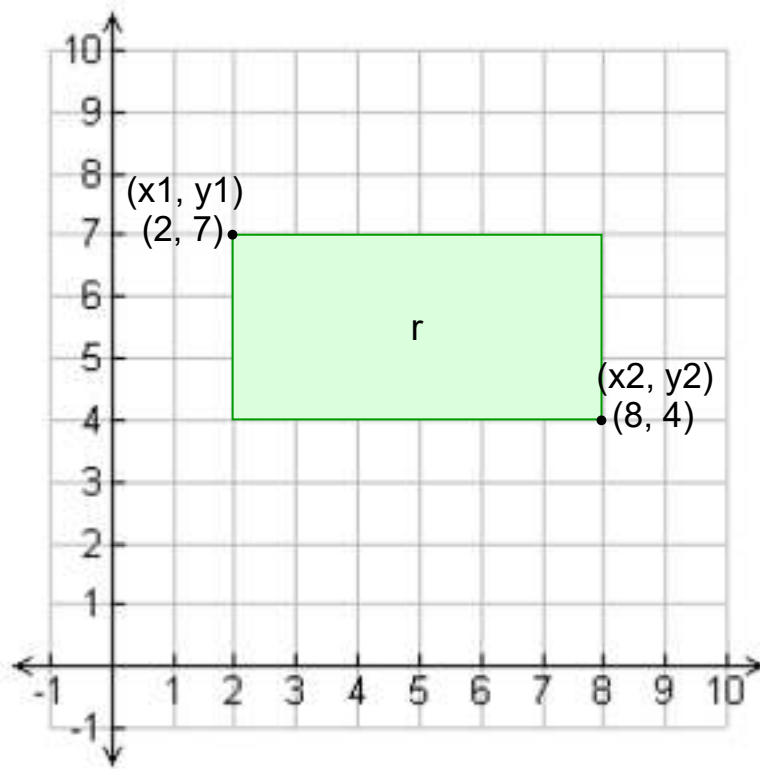
`r = Rectangle (2, 7, 8, 4)`

6 of 22



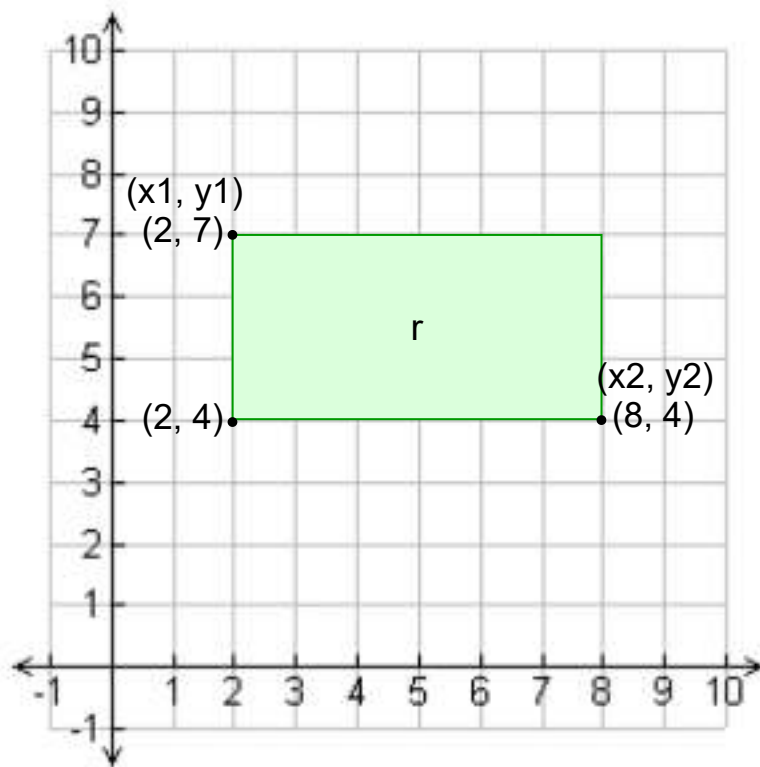
`r = Rectangle (2, 7, 8, 4)`

7 of 22

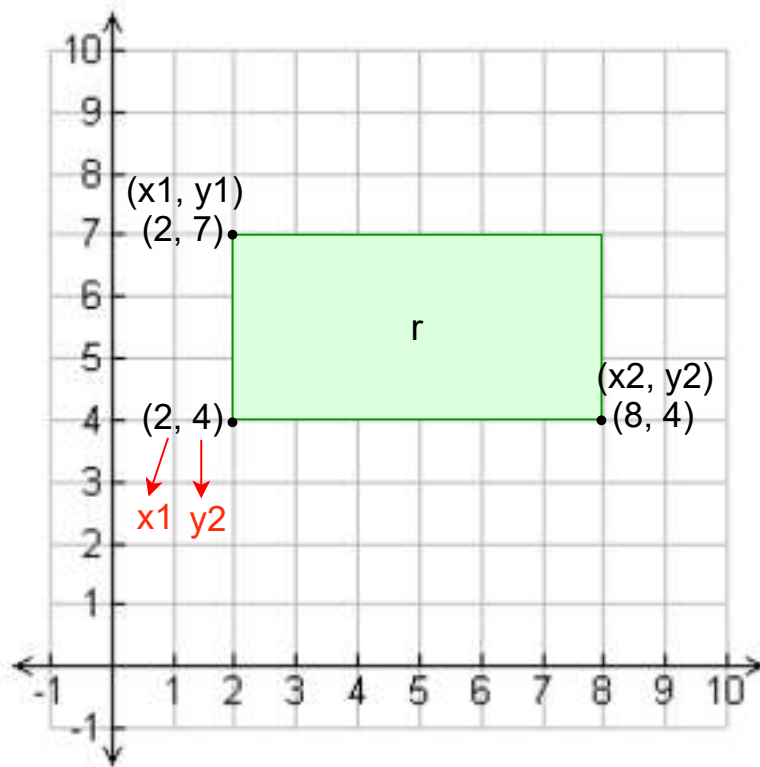


`r = Rectangle (2, 7, 8, 4)`

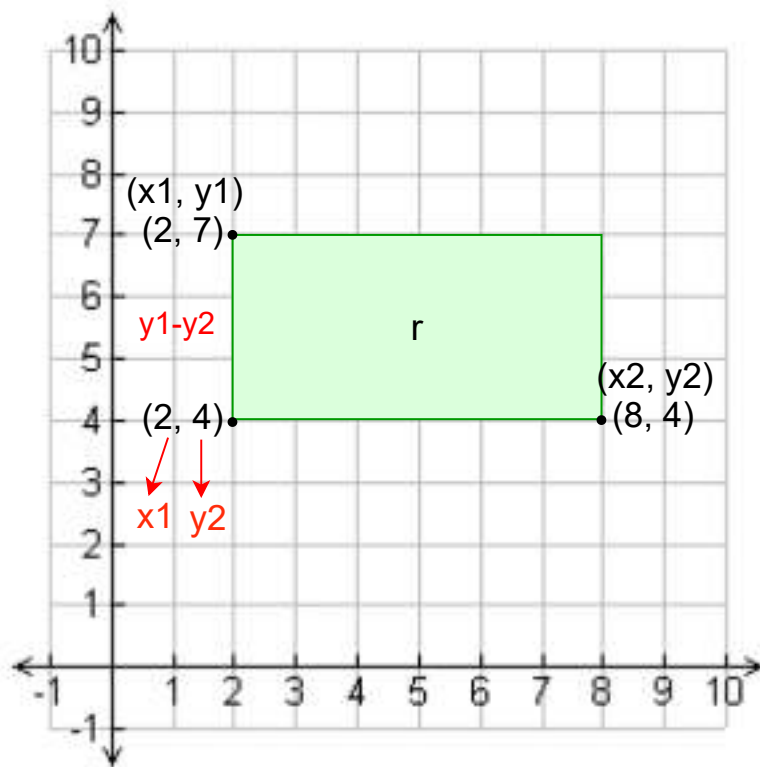
8 of 22



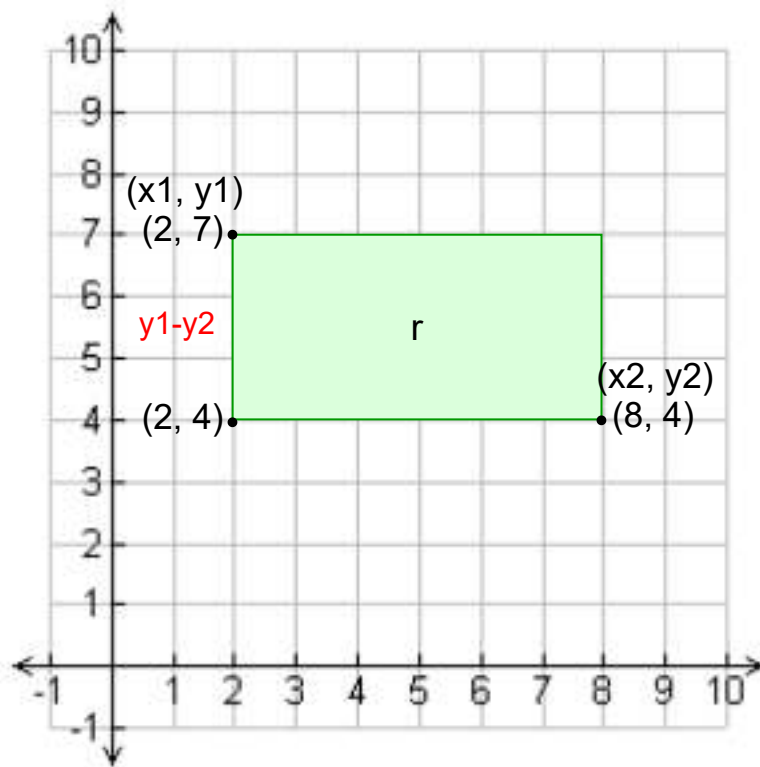
9 of 22



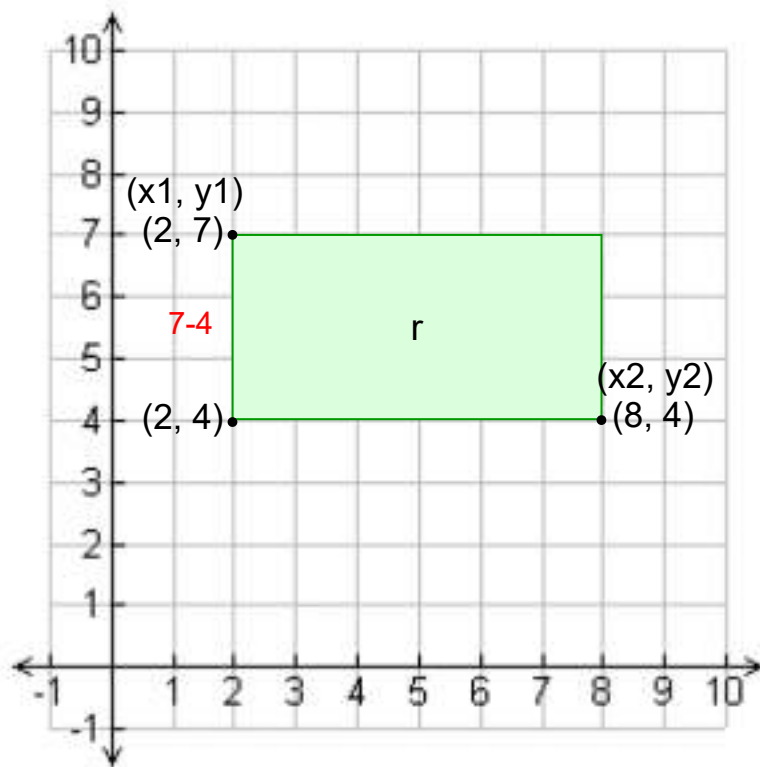
10 of 22



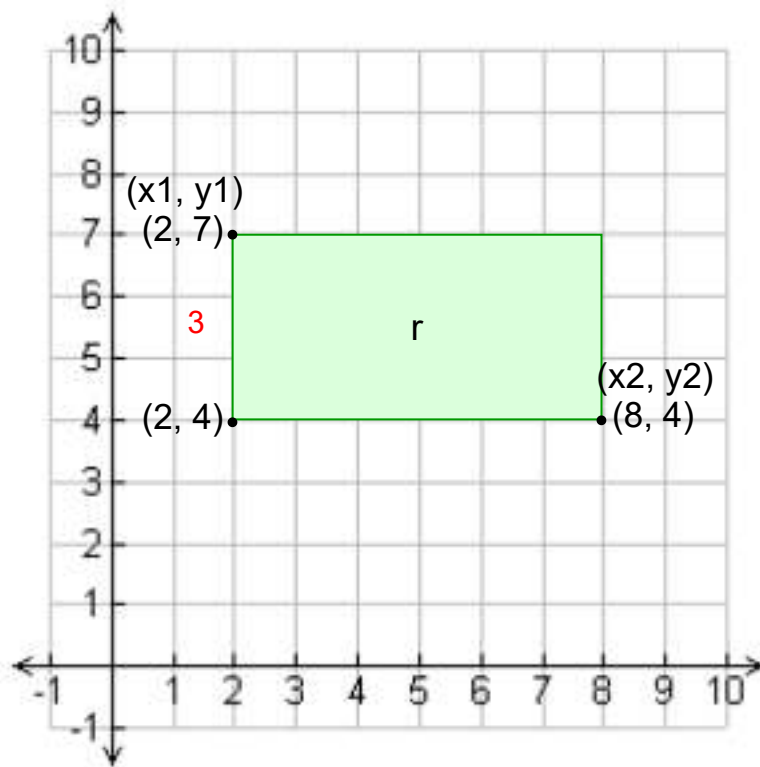
11 of 22



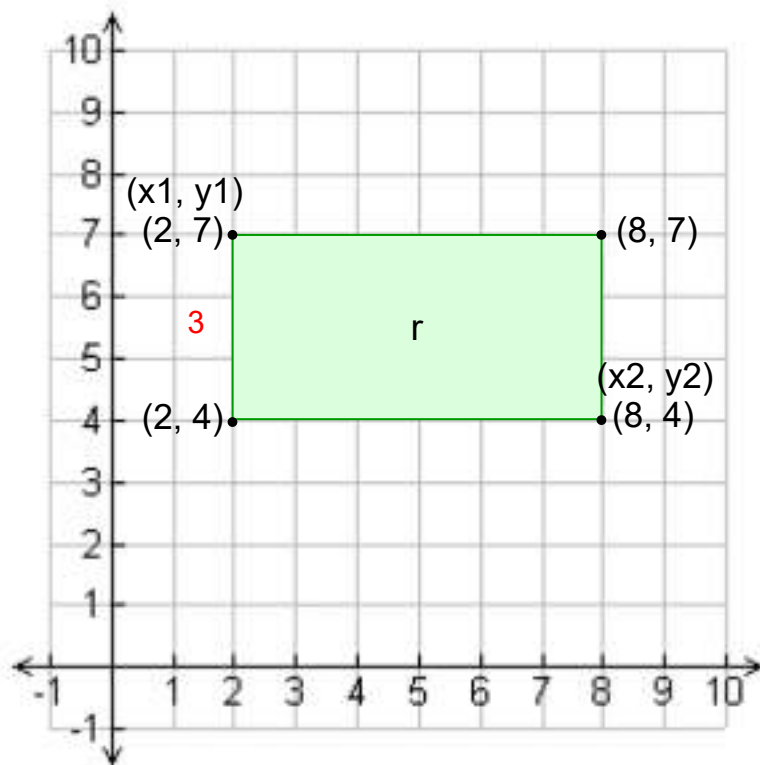
12 of 22



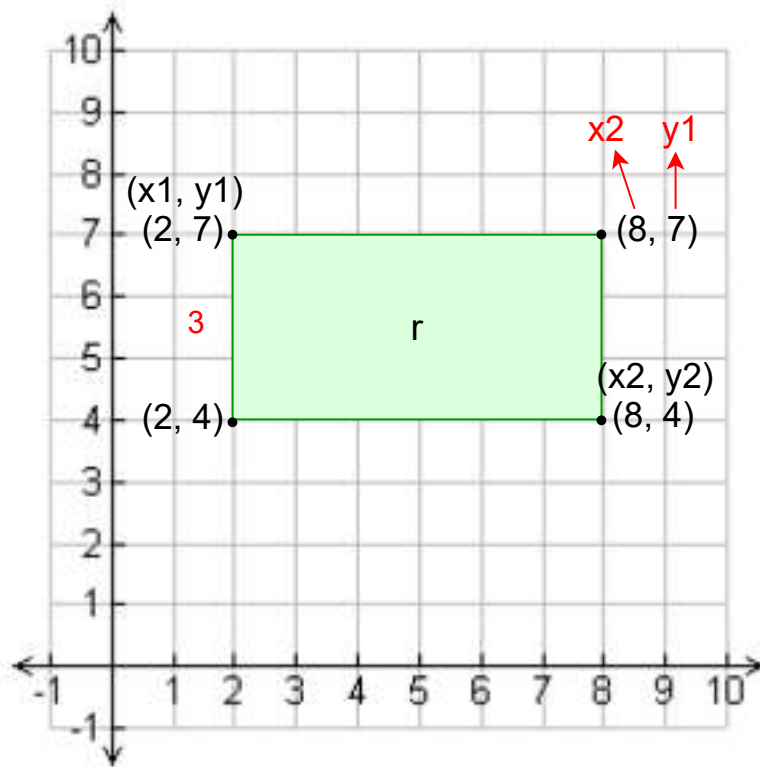
13 of 22



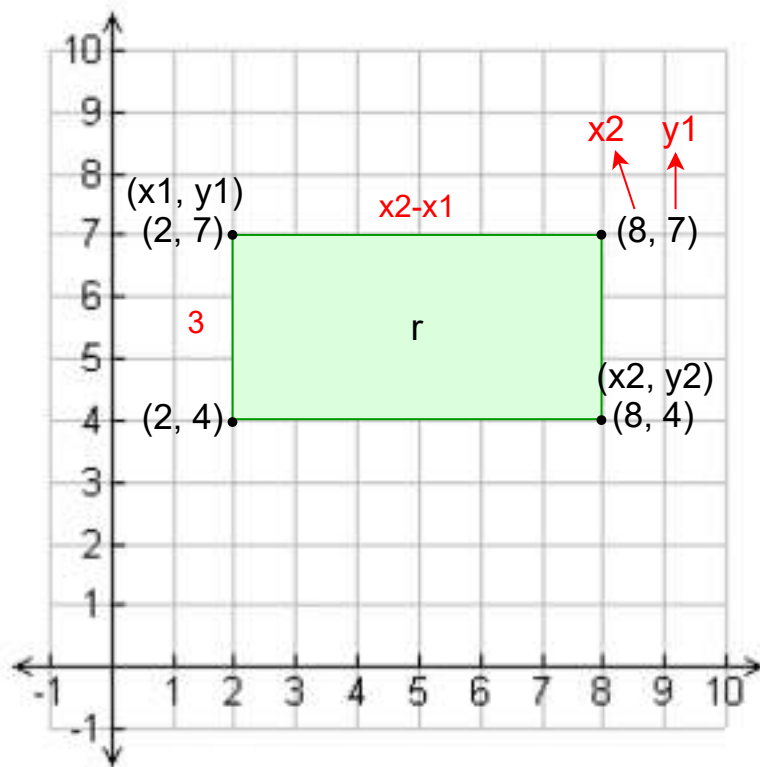
14 of 22



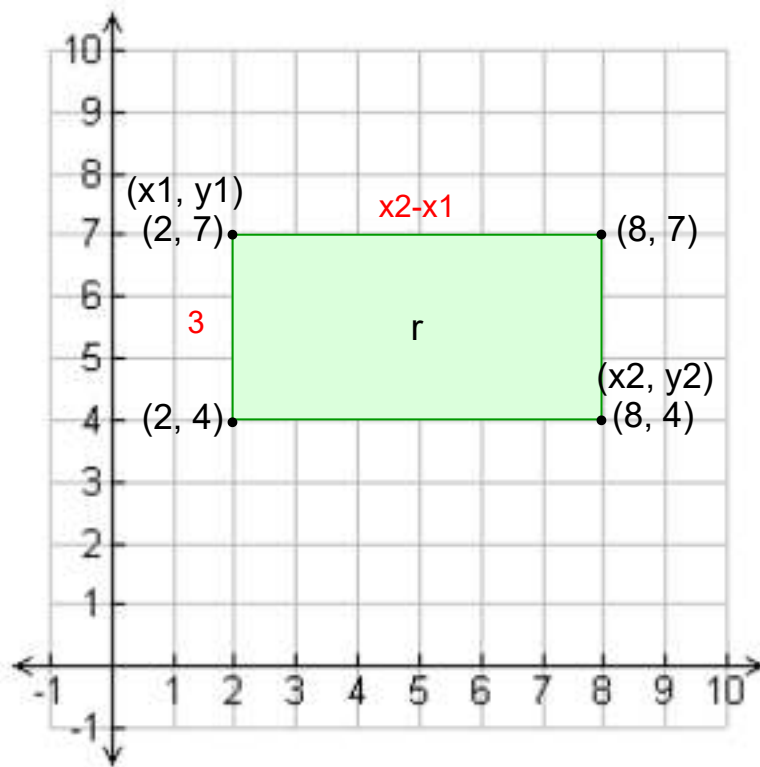
15 of 22



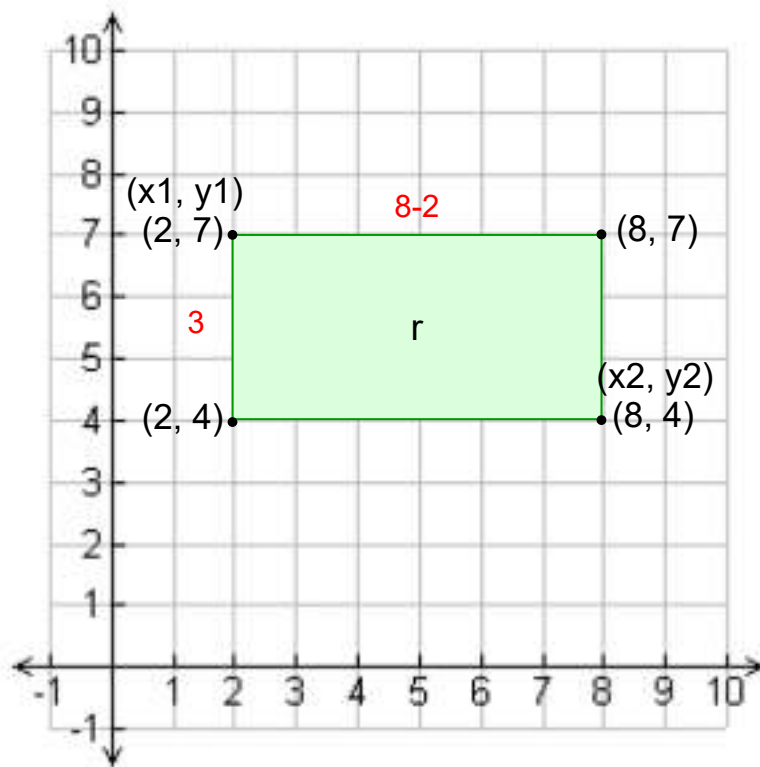
16 of 22



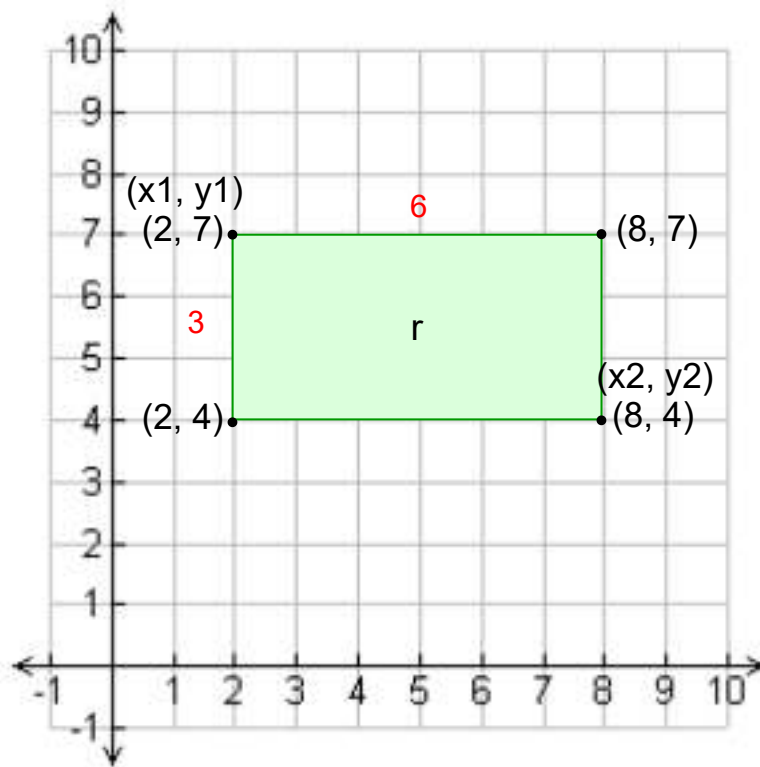
17 of 22



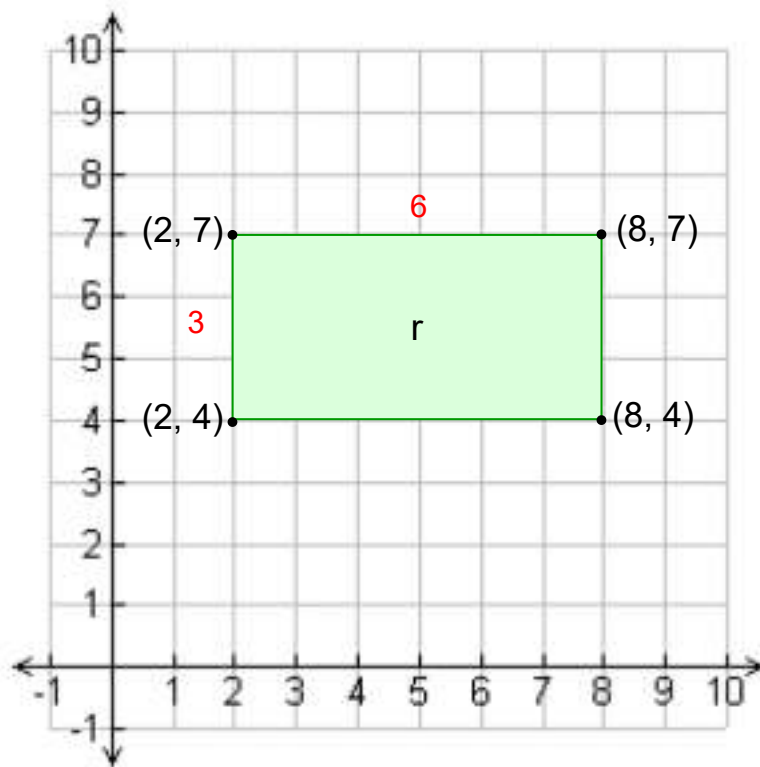
18 of 22



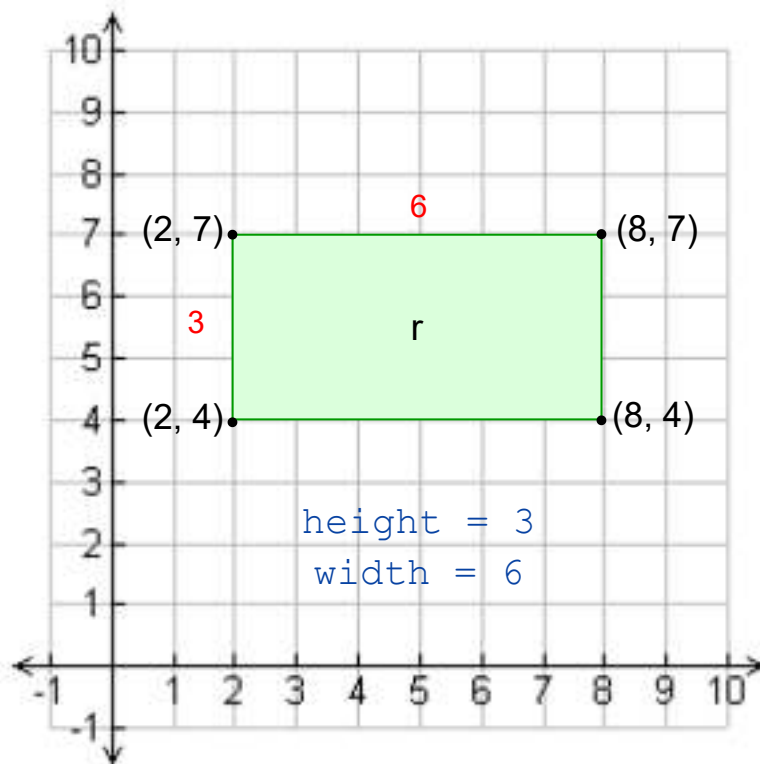
19 of 22



20 of 22



21 of 22



22 of 22

—

[]

The following code illustrates the concept:

```
class Rectangle:
    def __init__(self, x1, y1, x2, y2): # class constructor
        if x1<x2 and y1>y2:
            self.x1 = x1 # class variable
            self.y1 = y1 # class variable
            self.x2 = x2 # class variable
            self.y2 = y2 # class variable
        else:
            print("Incorrect coordinates of the rectangle!")

    def width(self):
        return self.x2-self.x1

    def height(self):
        return self.y1-self.y2

rectangle = Rectangle(2, 7, 8, 4)
print(rectangle.width())
print(rectangle.height())
```



Now, let's expand this challenge by implementing the area and perimeter methods in the next lesson.