

# Object Oriented Programming: Calling Rectangle class methods

Mr. Neat  
Java

Graphics Library: [Horstmann.com](http://Horstmann.com)

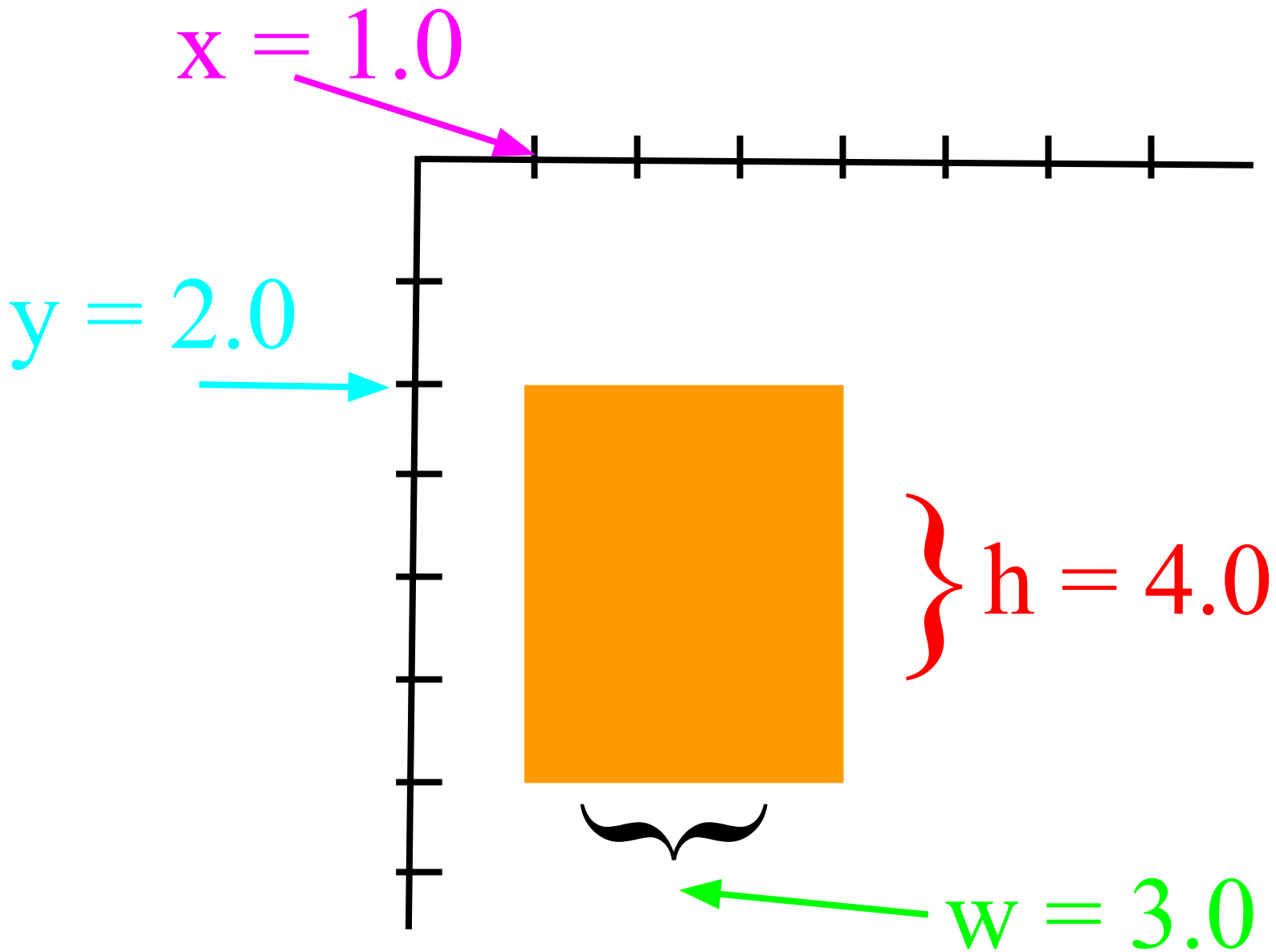
# Application Programming Interface

## API

# Calling Java Methods

General Notation:                      object name • method()

English Metafore                      noun • verb()



Rectangle **sam** = new Rectangle(1.0, 2.0, 3.0, 4.0);

# Calling Java Methods

General Notation:                      object name • method()

English Metafore                      noun • verb()

Example                      sam • draw();

# Graphics Library: Horstmann.com

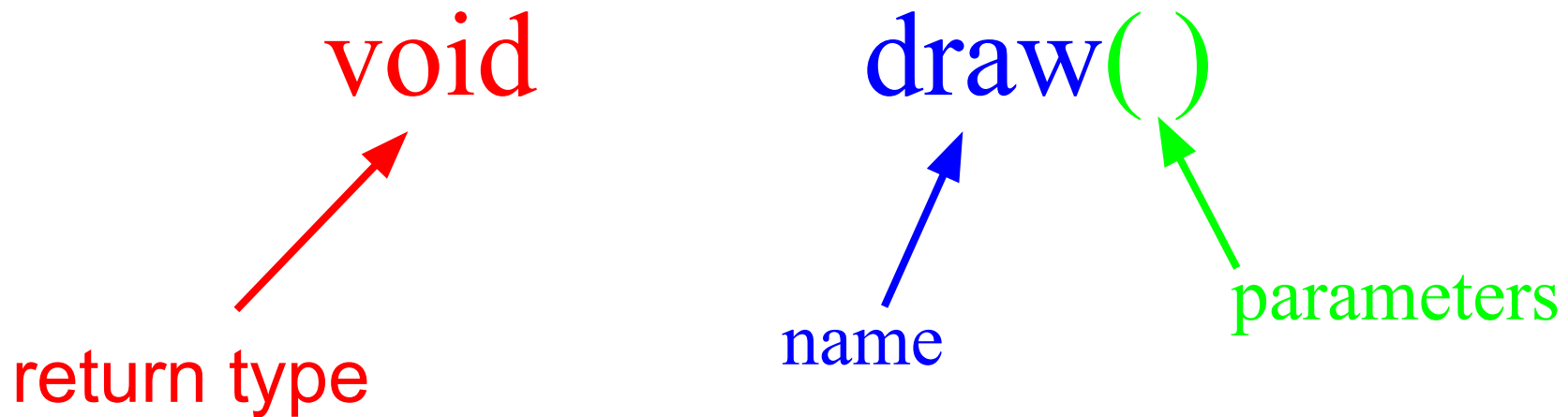
The API for the graphics package we will use:

<http://horstmann.com/sjsu/graphics/api/index.html>

(paste this link into your browser)

# Graphics Library: Anatomy of a method

Methods definition (as in API):



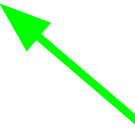
# Graphics Library: Anatomy of a method

Methods use:

sam.draw()



object previously  
constructed



method used  
by sam



# Graphics Library: other Rectangle methods

void draw()

Draws this rectangle.

void fill()

Fills this rectangle.

void setColor(Color newColor)

Sets the color of this rectangle.

# Graphics Library: other Rectangle methods

void draw()

Draws this rectangle.

void fill()

Fills this rectangle.

void setColor(Color newColor)

Sets the color of this rectangle.

# Graphics Library: other Rectangle methods

method definition:

void



return  
type

setColor(Color newColor)



name



parameters

# Graphics Library: other Rectangle methods

method definition:

**void**      **setColor**(Color newColor)

The diagram illustrates the components of the `setColor` method definition. A red arrow points from the text "return type" to the word "void". A blue arrow points from the text "name" to the text "setColor". A green arrow points from the text "parameters" to the text "Color newColor", which is enclosed in a red oval.

return  
type

name

parameters

# Graphics Library: other Rectangle methods

int fred;

double sue;

} primitives

String sam;

Color martha;

} classes

# Graphics Library: other Rectangle methods

```
Color newColor;  
martha =          ;
```



what goes here?

# Graphics Library: other Rectangle methods

BLACK

BLUE

CYAN

DARK\_GRAY

GRAY

GREEN

LIGHT\_GRAY

MAGENTA

ORANGE

PINK

RED

WHITE

YELLOW

# Graphics Library: other Rectangle methods

Let's look inside Color.java

```
public static final Color RED = new Color(255, 0, 0);  
public static final Color GREEN = new Color(0, 255, 0);  
public static final Color BLUE = new Color(0, 0, 255);  
public static final Color WHITE = new Color(255, 255, 255);  
public static final Color LIGHT_GRAY = new Color(192, 192, 192);
```

public vs. private instance variables



# Graphics Library: other Rectangle methods

Let's look inside Color.java

```
public static final Color RED = new Color(255, 0, 0);  
public static final Color GREEN = new Color(0, 255, 0);  
public static final Color BLUE = new Color(0, 0, 255);  
public static final Color WHITE = new Color(255, 255, 255);  
public static final Color LIGHT_GRAY = new Color(192, 192, 192);
```

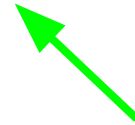
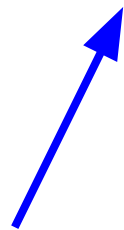
static instance variables

# Graphics Library: other Rectangle methods

Let's look inside Color.java

if we want to use the color CYAN,

Color.CYAN



Color variable name

use class name since variable is static

## Graphics Library: other Rectangle methods

```
Color newColor;  
martha = Color.CYAN;
```



what goes here?

# Graphics Library: Anatomy of a method

recall:

```
sam.draw()
```

now a method with a parameter::

```
sam.setColor(Color.CYAN)
```

sets the Rectangle sam to the color CYAN

# Graphics Library: Review

made from a Rectangle constructor:

```
Rectangle sam = new Rectangle(1.0, 2.0, 3.0, 4.0);
```



sam.draw()

one Rectangle method

# Next Lab...

Make a few colored  
Rectangles