

## Constructing Objects

## Rectangular Shapes and Rectangle Objects

- Objects of type `Rectangle` *describe* rectangular shapes

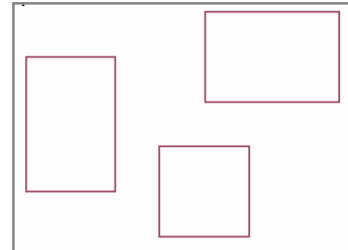


Figure 9:  
Rectangular Shapes

## Rectangular Shapes and Rectangle Objects

- A `Rectangle` object isn't a rectangular shape—it is an object that contains a set of numbers that describe the rectangle

Rectangle	Rectangle	Rectangle
x = 5	x = 35	x = 45
y = 10	y = 30	y = 0
width = 20	width = 20	width = 30
height = 30	height = 20	height = 20

Figure 10:  
Rectangular Objects

## Constructing Objects

- `new Rectangle(5, 10, 20, 30)`
- Detail:
  1. The `new` operator makes a `Rectangle` object
  2. It uses the parameters (in this case, 5, 10, 20, and 30) to initialize the data of the object
  3. It returns the object
- Usually the output of the `new` operator is stored in a variable

```
Rectangle box = new Rectangle(5, 10, 20, 30);
```

## Constructing Objects

- The process of creating a new object is called *construction*
- The four values 5, 10, 20, and 30 are called the *construction parameters*
- Some classes let you construct objects in multiple ways

```
new Rectangle()  
// constructs a rectangle with its top-left corner  
// at the origin (0, 0), width 0, and height 0
```

## Syntax 2.3: Object Construction

```
new ClassName(parameters)
```

**Example:**

```
new Rectangle(5, 10, 20, 30)  
new Rectangle()
```

**Purpose:**

To construct a new object, initialize it with the construction parameters, and return a reference to the constructed object

## Accessor and Mutator Methods

- **Accessor** method: does not change the state of its implicit parameter

```
double width = box.getWidth();
```

- **Mutator** method: changes the state of its implicit parameter

```
box.translate(15, 25);
```

## Accessor and Mutator Methods

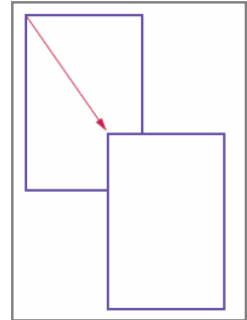


Figure 11:  
Using the `translate` Method to Move a  
Rectangle