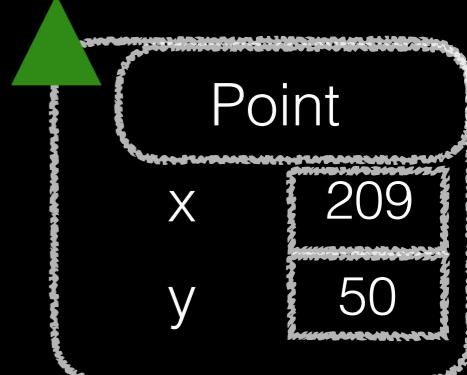
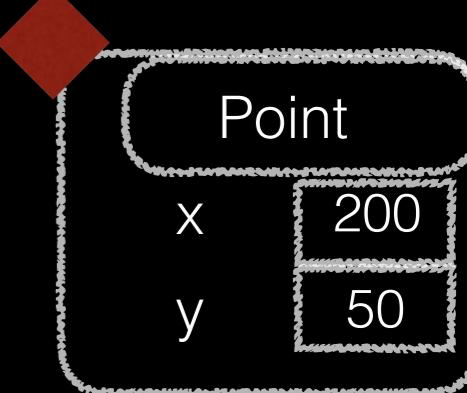
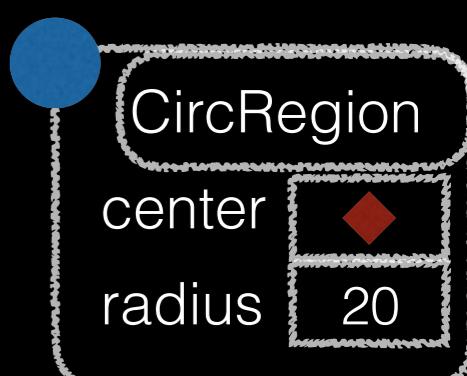
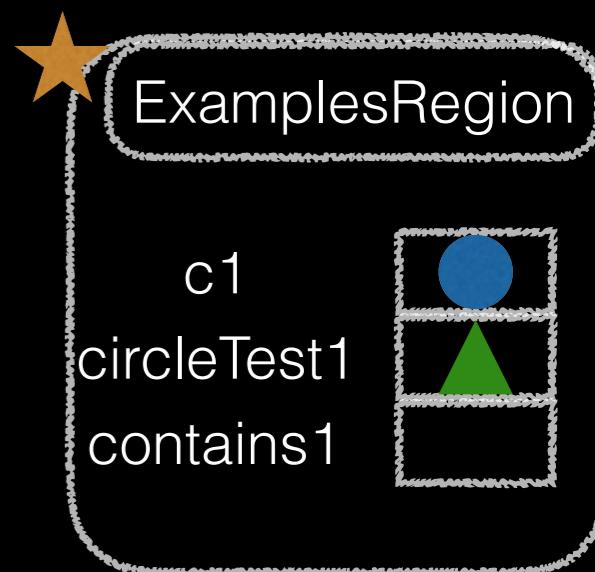


```

class Point {
    int x;
    int y;
    Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
    double distance(Point other) {
        return Math.sqrt(Math.pow(this.x - other.x, 2)
            + Math.pow(this.y - other.y, 2));
    }
}
class CircRegion {
    Point center;
    int radius;
    CircRegion(Point center, int radius) {
        this.center = center;
        this.radius = radius;
    }
    boolean contains(Point p) {
        return this.center.distance(p) < this.radius;
    }
}
class ExamplesRegion {
    CircRegion c1 = new CircRegion(new Point(200, 50), 10);
    Point circleTest1 = new Point(209, 50);
    boolean contains1 = this.c1.contains(this.circleTest1);
}

```



```
class Point {  
  
    double distance(Point other) {  
        return  
            + Mat  
    }  
}  
  
class CircRegion {  
  
    boolean contains(Point p) {  
        return  
    }  
}
```

9

true

this.c1.contains(this.circleTest1);

distance

this
other

contains

this
p

ExamplesRegion

c1

circleTest1
contains1



CircRegion

center

20

radius



Point

x

200

y

50

Point

x

209

y

50



```

class Point {

    double distance(Point other) {
        return Math.sqrt(Math.pow(this.x - other.x, 2)
            + Math.pow(this.y - other.y, 2));
    }
}

class CircRegion {

```

```

    boolean contains(Point p) {
        return this.center.distance(p) < this.radius;
    }
}

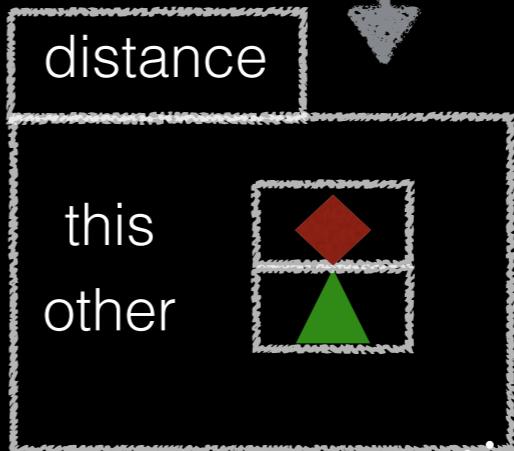
```

9

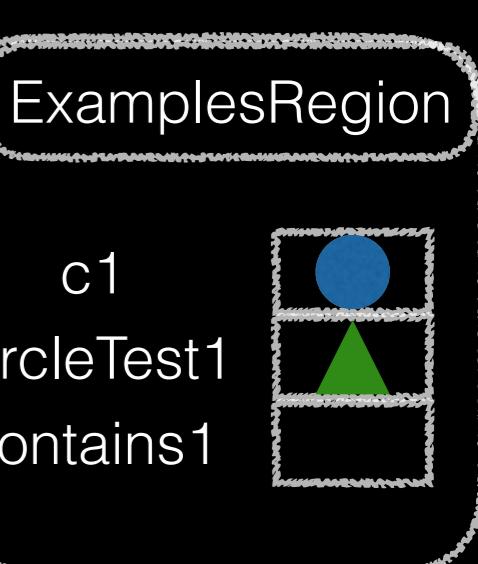
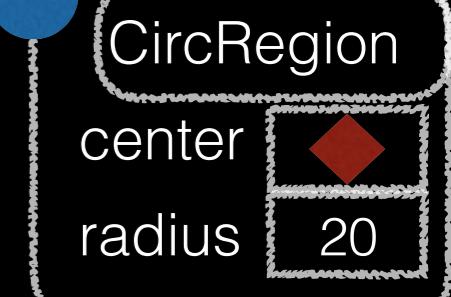
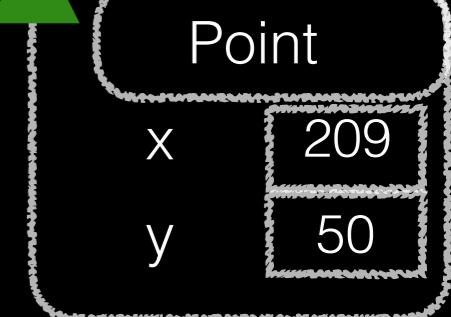
true

this.c1.contains(this.circleTest1);

Stack



Heap

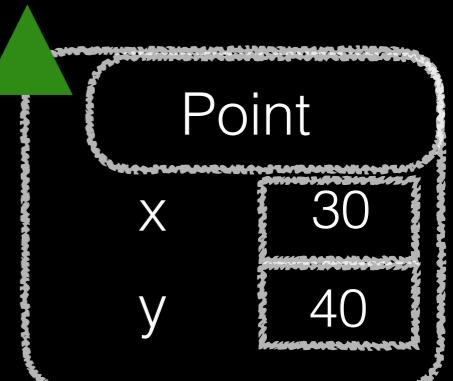
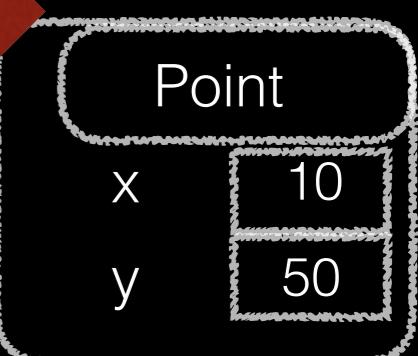
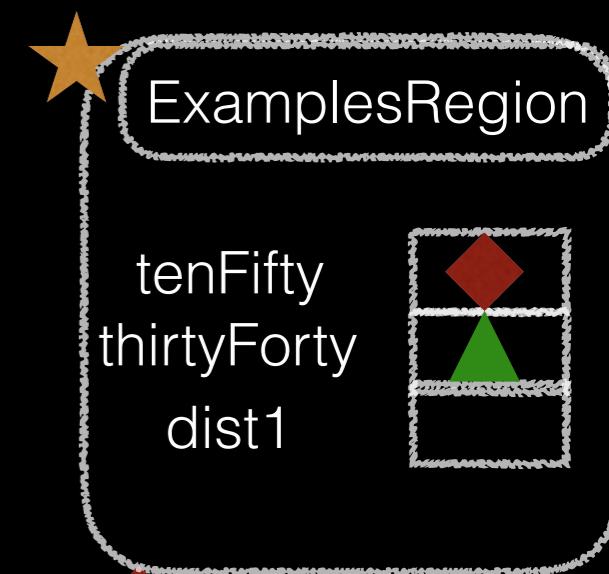


```

class Point {
    int x;
    int y;
    Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
    double distance(Point other) {
        return Math.sqrt(Math.pow(this.x - other.x, 2)
            + Math.pow(this.y - other.y, 2));
    }
}
class ExamplesRegion {
    Point tenFifty = new Point(10, 50);
    Point thirtyForty = new Point(30, 40);

    double dist1 = this.tenFifty.distance(this.thirtyForty);
}

```



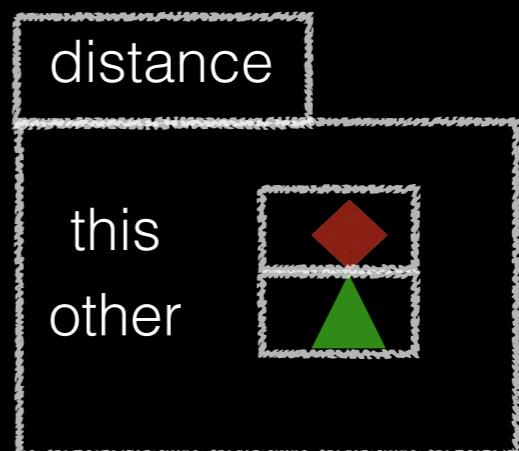
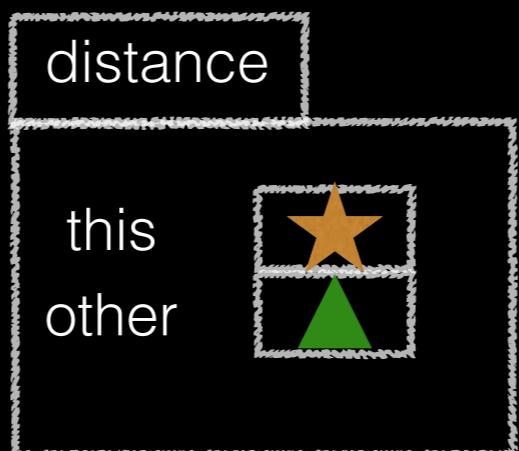
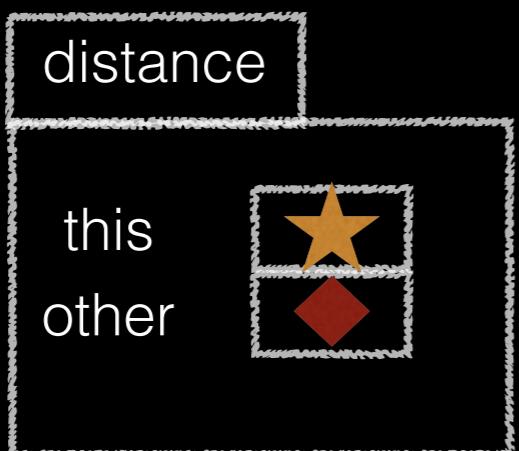
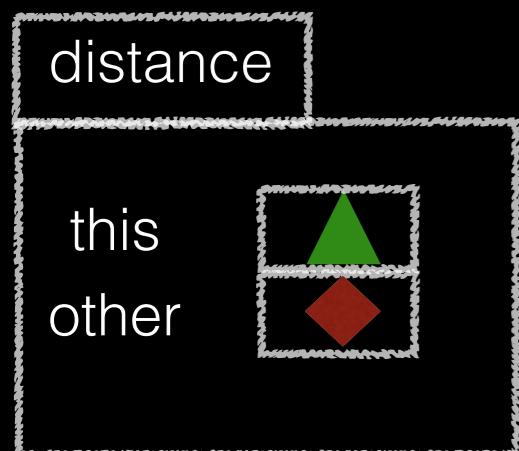
Which of these is the stack frame that's created for the call to `distance`?

A

B

C

D



```

class Point {
    int x;
    int y;
    Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
    double distance(Point other) {
        return Math.sqrt(Math.pow(this.x - other.x, 2)
            + Math.pow(this.y - other.y, 2));
    }
}
class ExamplesRegion {
    Point tenFifty = new Point(10, 50);
    Point thirtyForty = new Point(30, 40);
    Point anotherP = tenFifty;

    double dist1 = this.tenFifty.distance(this.anotherP);
}

```

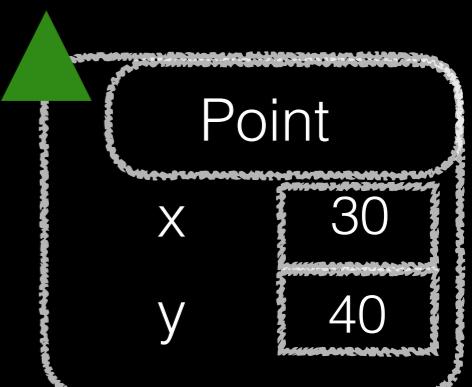
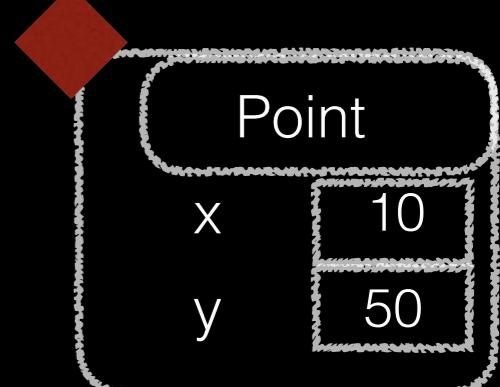
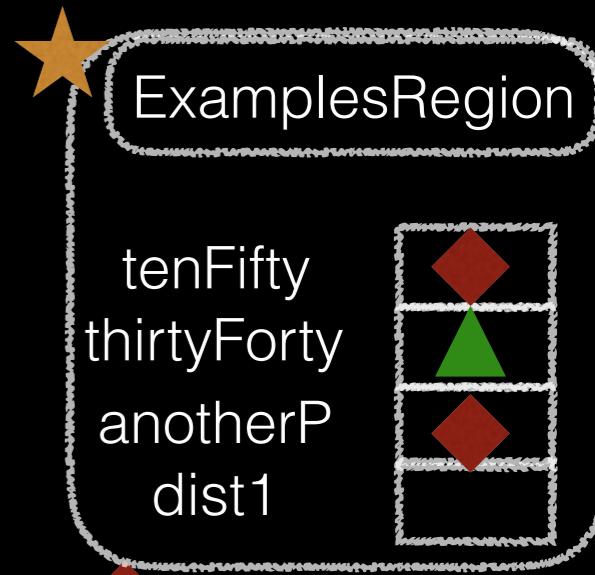
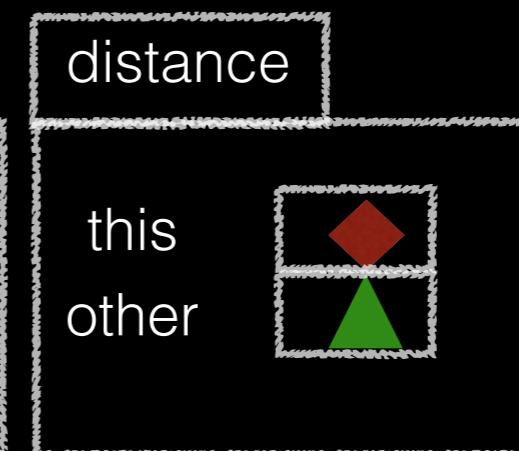
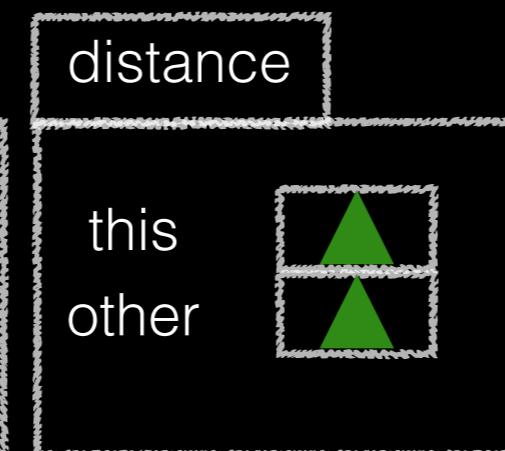
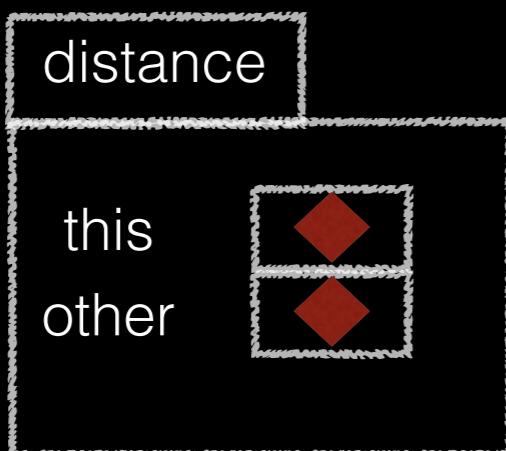
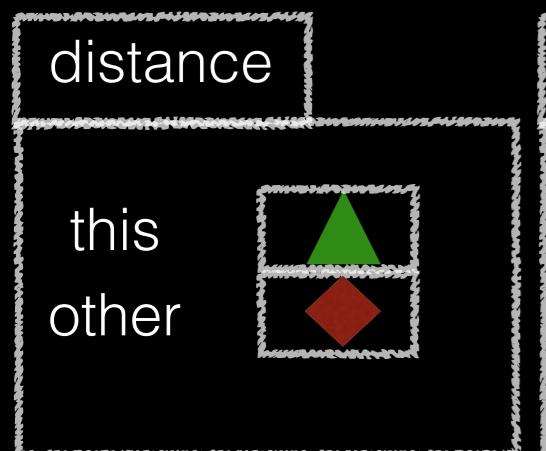
Which of these is the stack frame that's created for the call to `distance`?

A

B

C

D

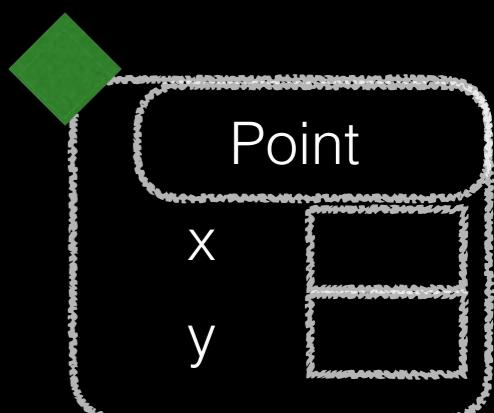
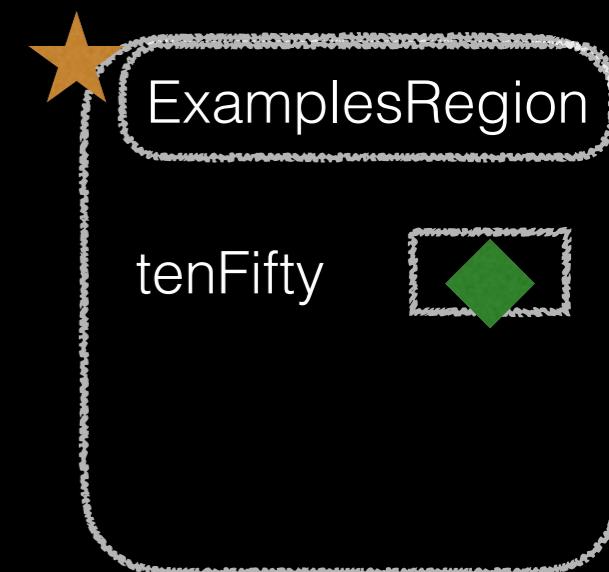
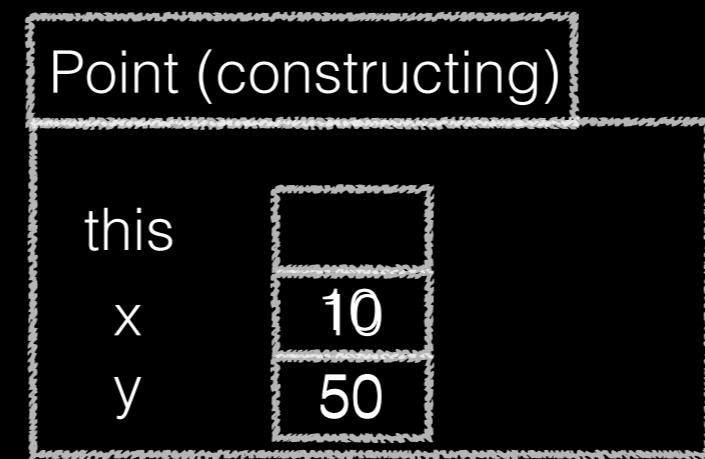


```

class Point {
    int x;
    int y;
    Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

class ExamplesRegion {
    Point tenFifty = new Point(10, 50);
}

```



Constructors:

1. Are special methods, called when **new** is used
2. Are passed the newly-constructed object as **this**, and any arguments
3. Typically assign values into fields using **this.field = value**

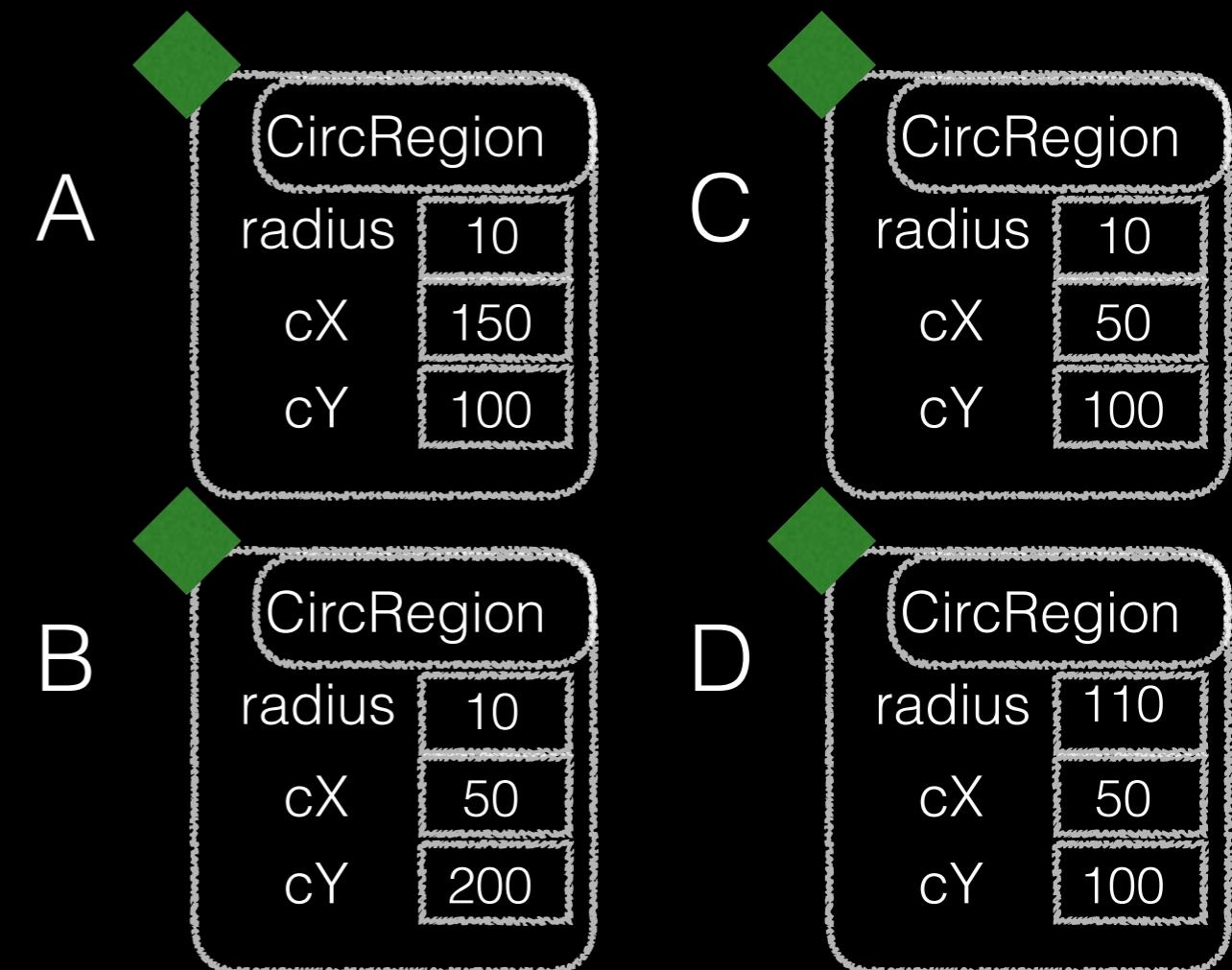
When **new** is used

1. A fresh object, with a new reference, is created with uninitialized fields
2. The constructor with parameters that match the arguments is called
3. The whole **new** expression evaluates to the new reference

```

class CircRegion2 {
    int radius;
    int cX;
    int cY;
    CircRegion2(int radius, int cX, int cY) {
        this.radius = radius;
        this.cX = cX + 100;
        this.cY = cY;
    }
}
class ExamplesRegion {
    CircRegion2 cr2 = new CircRegion(10, 50, 100);
}

```



Constructors:

1. Are special methods, called when **new** is used
2. Are passed the newly-constructed object as **this**, and any arguments
3. Typically assign values into fields using **this.field = value**

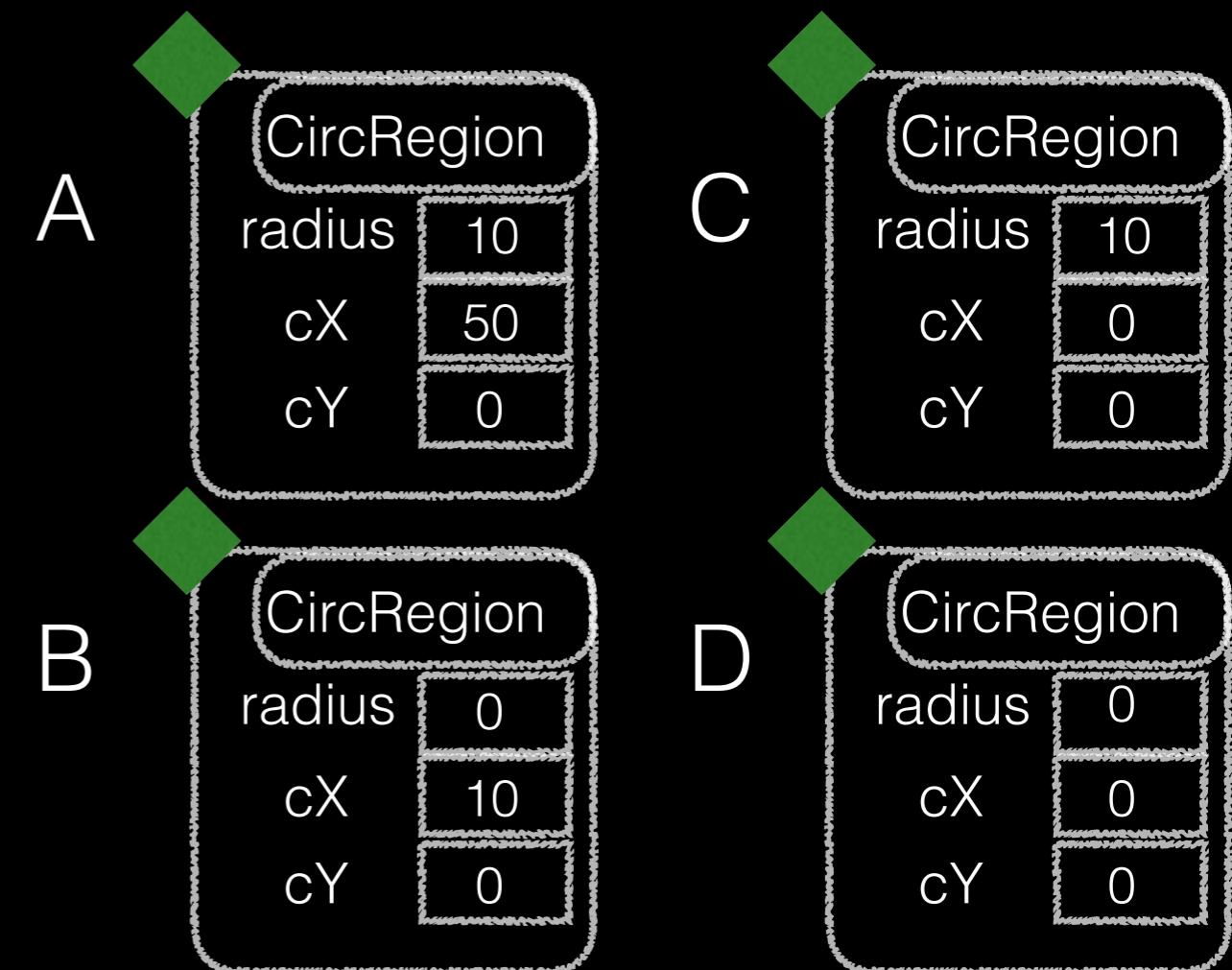
When **new** is used

1. A fresh object, with a new reference, is created with uninitialized fields
2. The constructor with parameters that match the arguments is called
3. The whole **new** expression evaluates to the new reference

```

class CircRegion2 {
    int radius;
    int cX;
    int cY;
    CircRegion2(int radius) {
        this.radius = radius;
        this.cX = 0;
        this.cY = 0;
    }
}
class ExamplesRegion {
    CircRegion2 cr2 = new CircRegion(10);
}

```



Constructors:

1. Are special methods, called when **new** is used
2. Are passed the newly-constructed object as **this**, and any arguments
3. Typically assign values into fields using **this.field = value**

When **new** is used

1. A fresh object, with a new reference, is created with uninitialized fields
2. The constructor with parameters that match the arguments is called
3. The whole **new** expression evaluates to the new reference

```

class CircRegion2 {
    int radius;
    int cX;
    int cY;
    CircRegion2(int radius, int cX, int cY) {
        this.radius = radius;
        this.cX = cX;
        this.cY = cY;
    }
    CircRegion2(int radius) {
        this.radius = radius;
        this.cX = 0;
        this.cY = 0;
    }
}
class ExamplesRegion {
    CircRegion2 cr2 = new CircRegion(10);
}

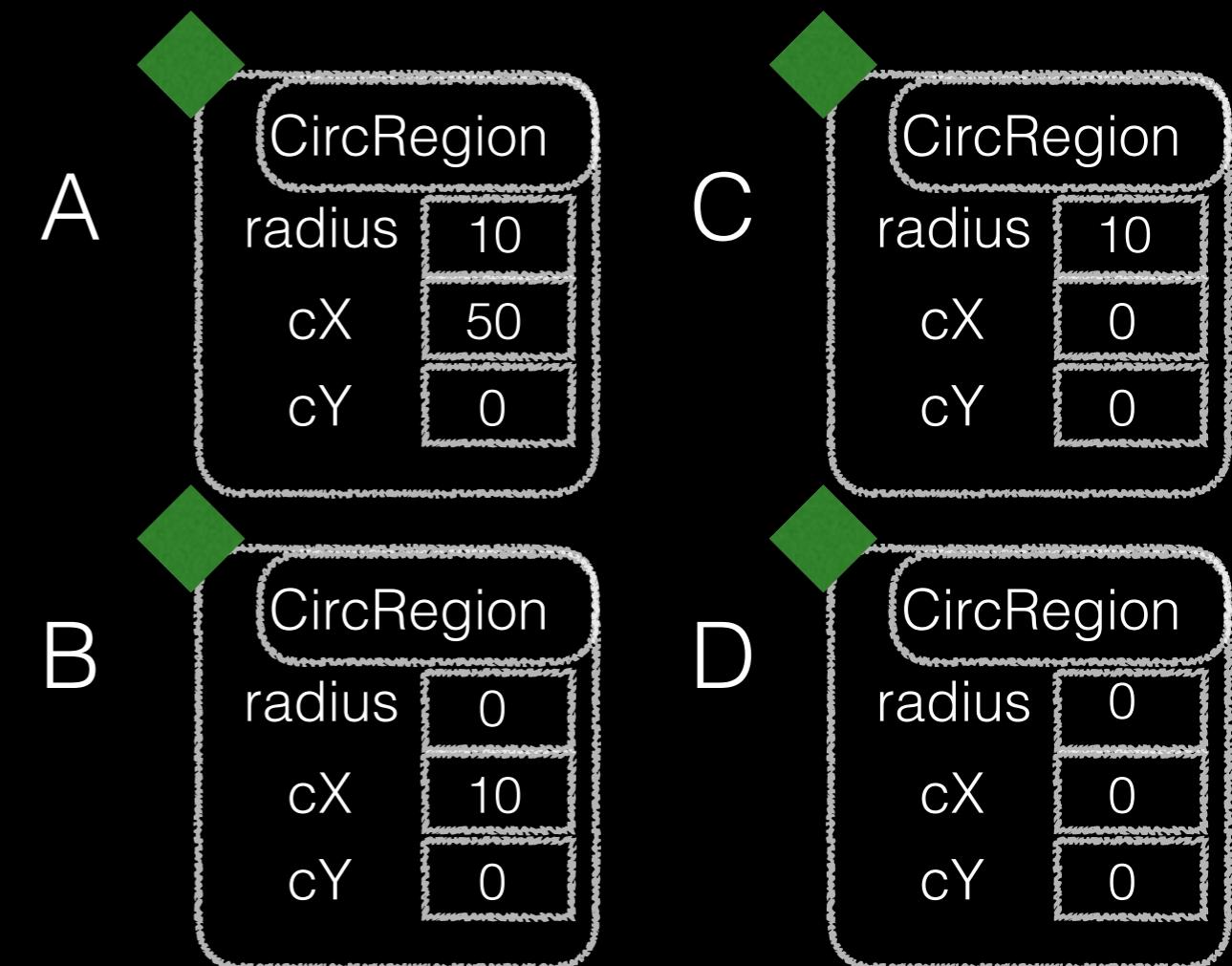
```

Constructors:

1. Are special methods, called when **new** is used
2. Are passed the newly-constructed object as **this**, and any arguments
3. Typically assign values into fields using **this.field = value**

When **new** is used

1. A fresh object, with a new reference, is created with uninitialized fields
2. The constructor with parameters that match the arguments is called
3. The whole **new** expression evaluates to the new reference



```

class CircRegion2 {
    int radius;
    int cX;
    int cY;
    CircRegion2(int radius, int cX, int cY) {
        this.radius = radius;
        this.cX = cX;
        this.cY = cY;
    }
    CircRegion2(int radius, int coord) {
        this.radius = radius;
        this.cX = coord;
        this.cY = coord;
    }
}
class ExamplesRegion {
    CircRegion2 cr2 = new CircRegion(10, 100);
}

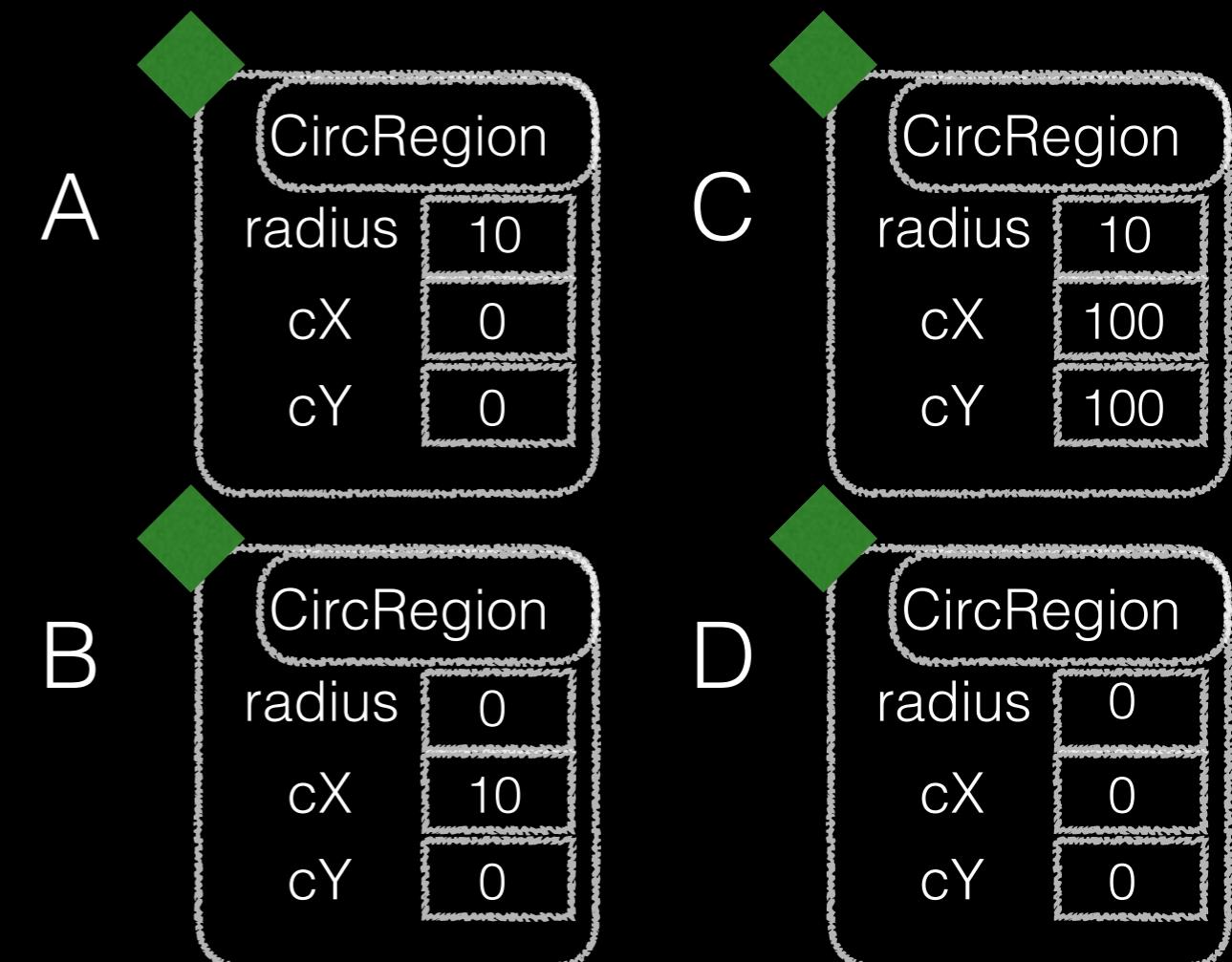
```

Constructors:

1. Are special methods, called when **new** is used
2. Are passed the newly-constructed object as **this**, and any arguments
3. Typically assign values into fields using **this.field = value**

When **new** is used

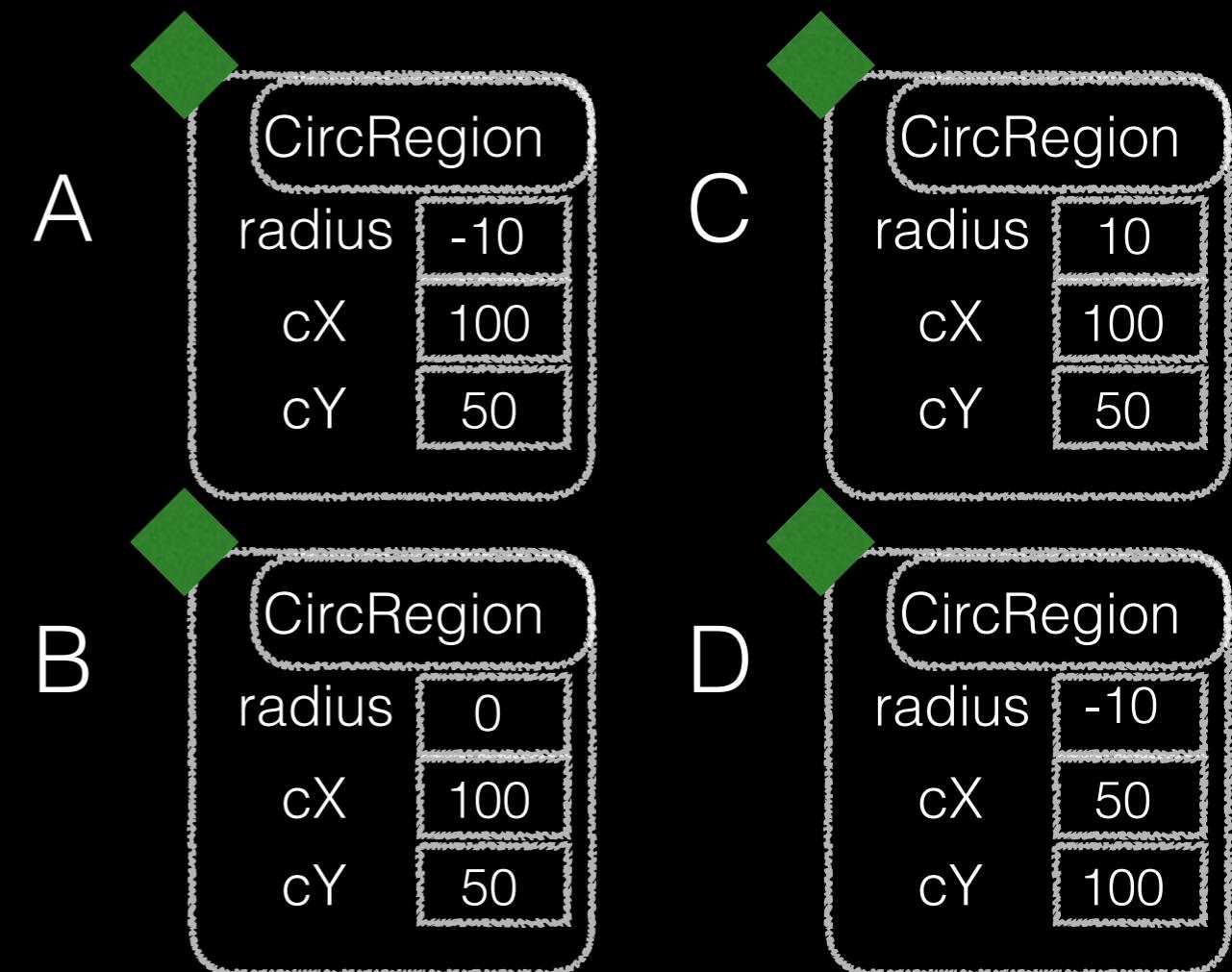
1. A fresh object, with a new reference, is created with uninitialized fields
2. The constructor with parameters that match the arguments is called
3. The whole **new** expression evaluates to the new reference



```

class CircRegion2 {
    int radius;
    int cX;
    int cY;
    CircRegion2(int radius, int cX, int cY) {
        if(radius < 0) {
            this.radius = 0;
        } else {
            this.radius = radius;
        }
        this.cX = cX;
        this.cY = cY;
    }
}
class ExamplesRegion {
    CircRegion2 cr2 = new CircRegion2(-10, 100, 50);
}

```



Constructors:

1. Are special methods, called when **new** is used
2. Are passed the newly-constructed object as **this**, and any arguments
3. Typically assign values into fields using **this.field = value**

When **new** is used

1. A fresh object, with a new reference, is created with uninitialized fields
2. The constructor with parameters that match the arguments is called
3. The whole **new** expression evaluates to the new reference