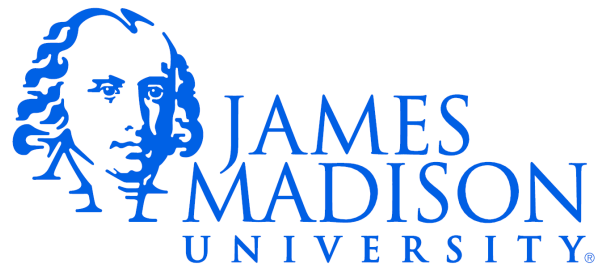


CS159



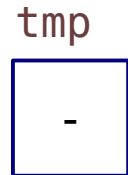
Primitive Arrays

```
int tmp;  
int[] numbers;  
  
numbers = new int[3];  
  
numbers[2] = 7;  
  
tmp = numbers[2];  
numbers[0] = tmp;  
  
System.out.println(numbers);  
  
for (int i = 0; i < numbers.length; i++)  
{  
    System.out.println(numbers[i]);  
}
```

What will be printed?

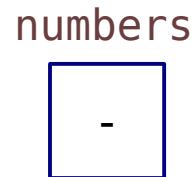
Primitive Arrays

```
int tmp;  
int[] numbers;
```



➔ numbers = new int[3];

```
numbers[2] = 7;
```



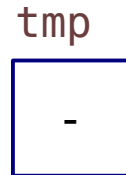
```
tmp = numbers[2];  
numbers[0] = tmp;
```

```
System.out.println(numbers);
```

```
for (int i = 0; i < numbers.length; i++)  
{  
    System.out.println(numbers[i]);  
}
```

Primitive Arrays

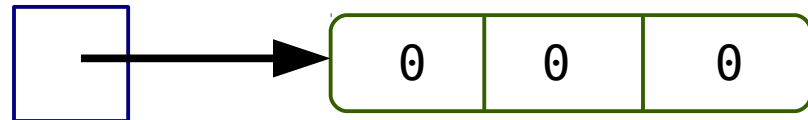
```
int tmp;  
int[] numbers;  
  
numbers = new int[3];
```



→ numbers[2] = 7;

```
tmp = numbers[2];  
numbers[0] = tmp;
```

numbers

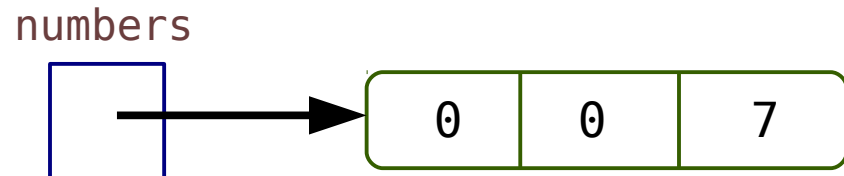
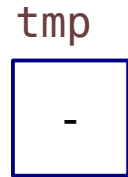


```
System.out.println(numbers);
```

```
for (int i = 0; i < numbers.length; i++)  
{  
    System.out.println(numbers[i]);  
}
```

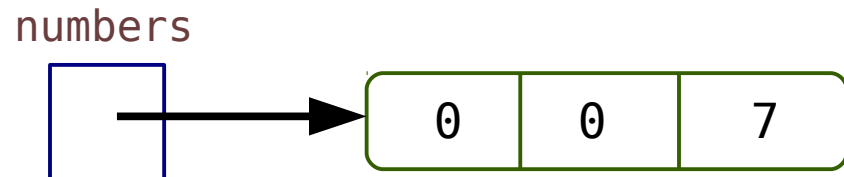
Primitive Arrays

```
int tmp;  
int[] numbers;  
  
numbers = new int[3];  
  
numbers[2] = 7;  
→ tmp = numbers[2];  
   numbers[0] = tmp;  
  
System.out.println(numbers);  
  
for (int i = 0; i < numbers.length; i++)  
{  
    System.out.println(numbers[i]);  
}
```



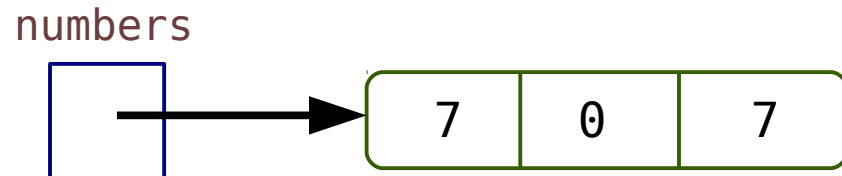
Primitive Arrays

```
int tmp;  
int[] numbers;  
  
numbers = new int[3];  
  
numbers[2] = 7;  
  
tmp = numbers[2];  
numbers[0] = tmp;  
  
System.out.println(numbers);  
  
for (int i = 0; i < numbers.length; i++)  
{  
    System.out.println(numbers[i]);  
}
```



Primitive Arrays

```
int tmp;  
int[] numbers;  
  
numbers = new int[3];  
  
numbers[2] = 7;  
  
tmp = numbers[2];  
numbers[0] = tmp;
```



➔ `System.out.println(numbers);`

```
for (int i = 0; i < numbers.length; i++)  
{  
    System.out.println(numbers[i]);  
}
```

Reference Arrays

```
Point p1;
Point[] points;

p1 = new Point(1.0, 1.0);
points = new Point[3];

for (int i = 0; i < points.length; i++)
{
    points[i] = p1;
}

p1.setX(3.0);

System.out.println(p1);

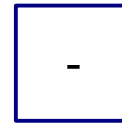
for (int i = 0; i < points.length; i++)
{
    System.out.println(points[i]);
}
```

What will be printed?

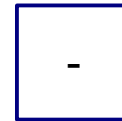
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
  
for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```

p1

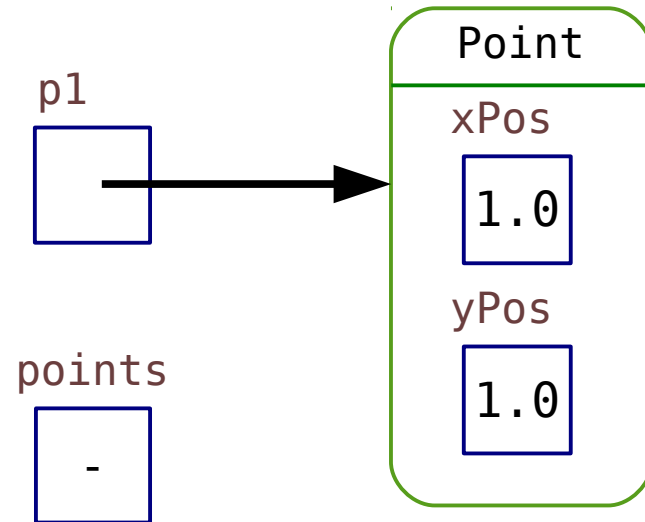


points



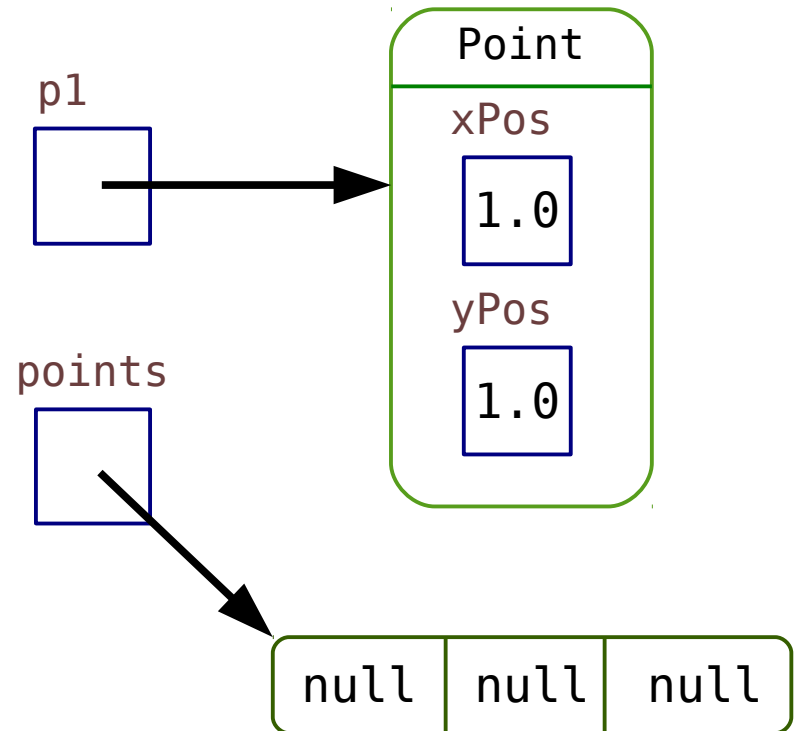
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
  
for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



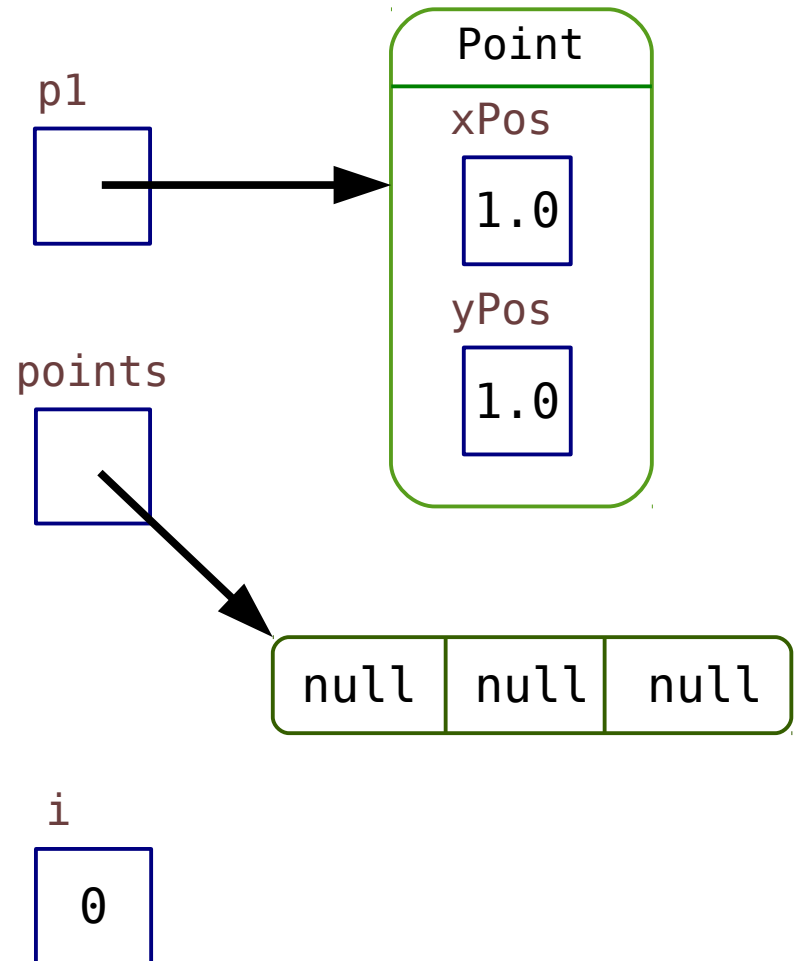
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
→ for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



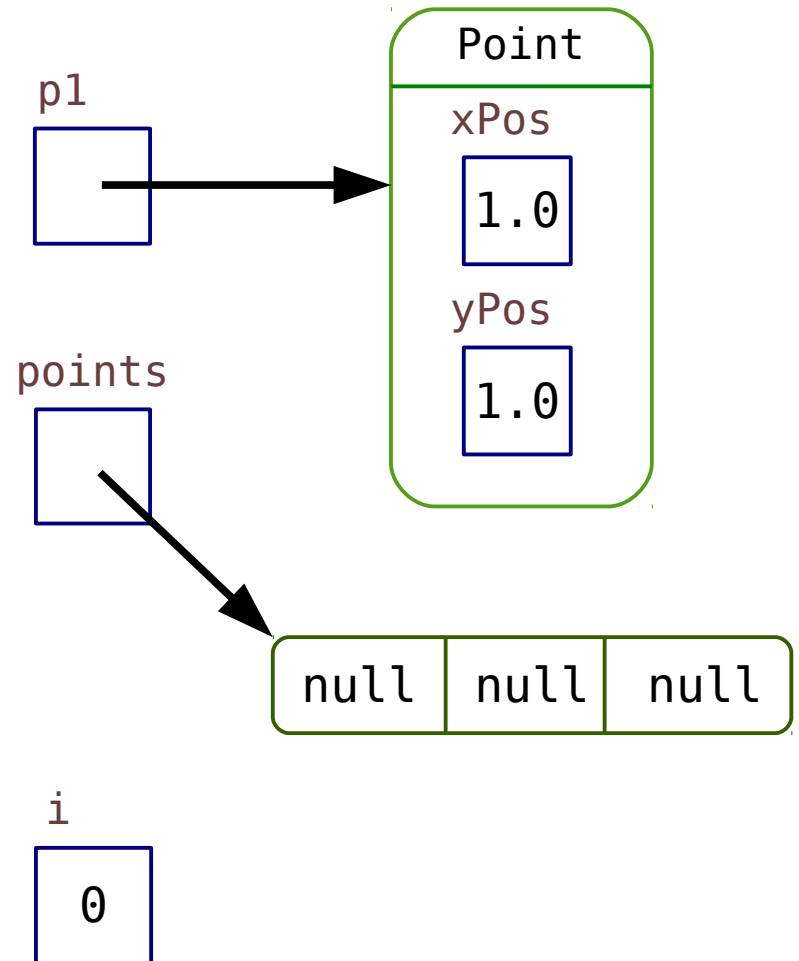
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
→ for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



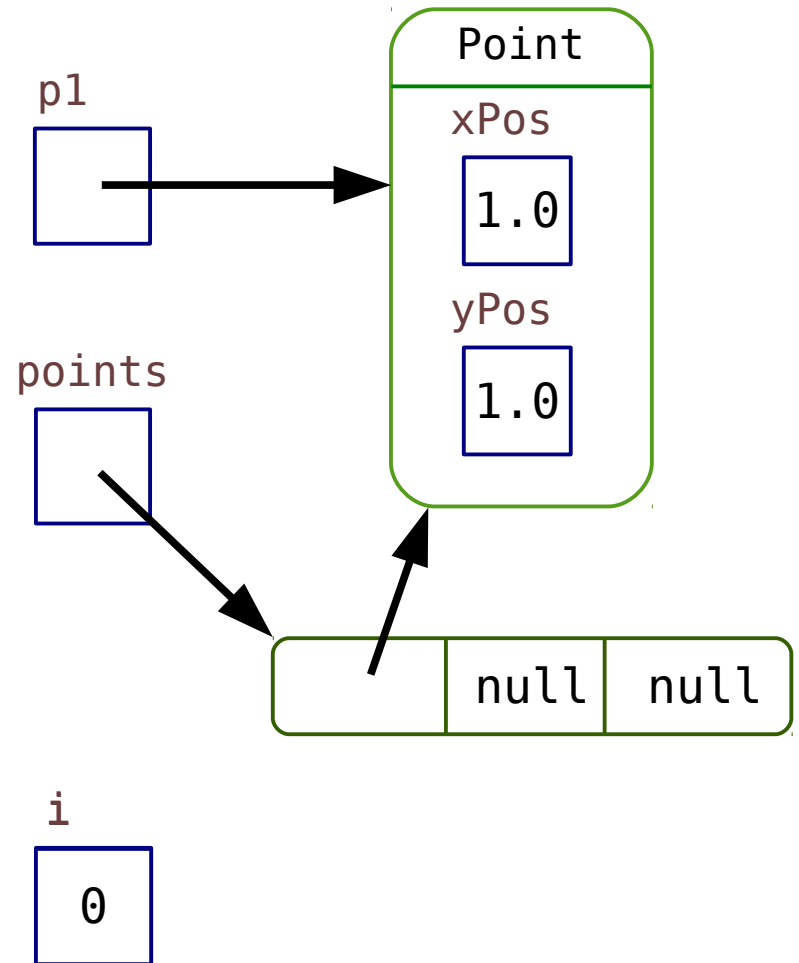
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
  
for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



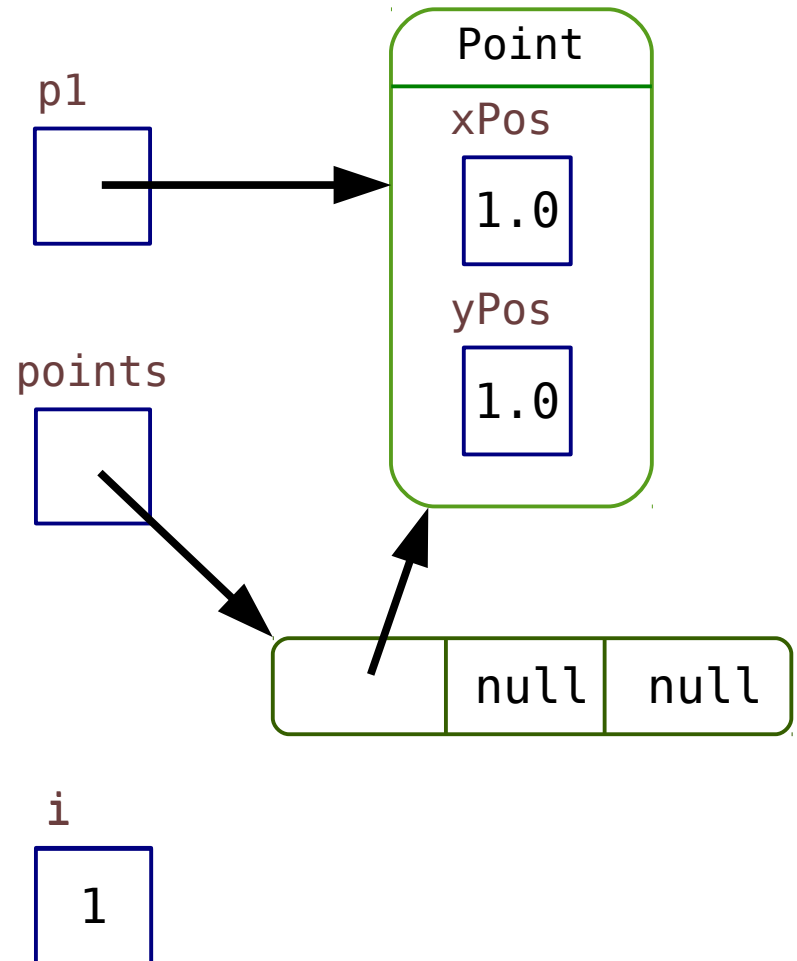
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
→ for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



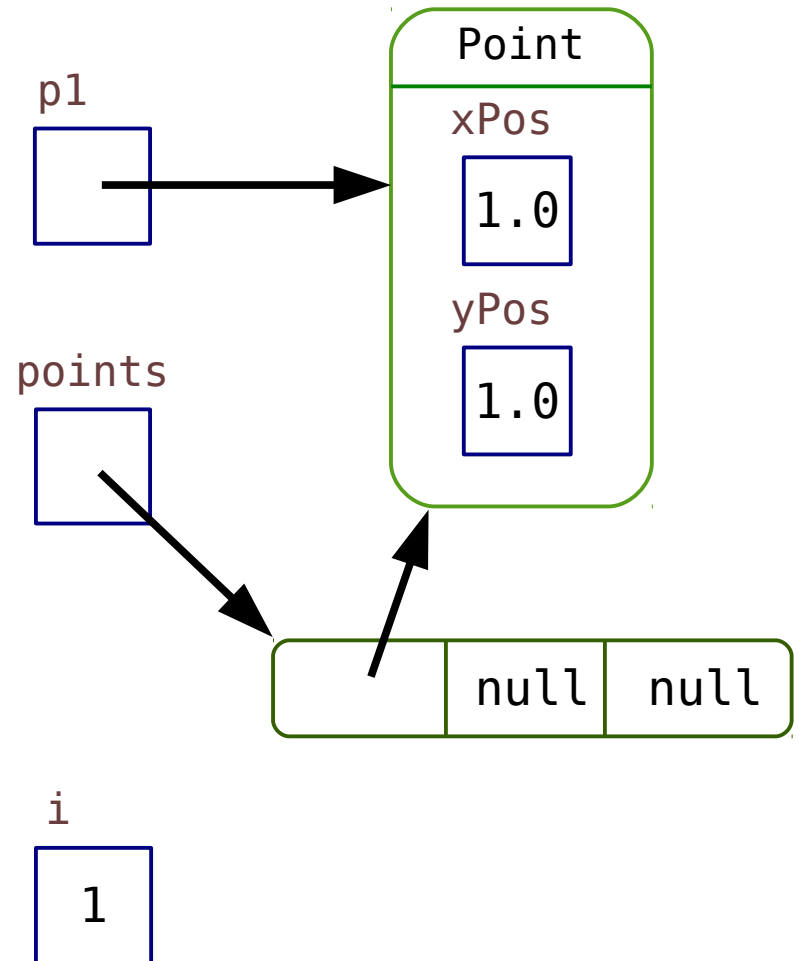
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
→ for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



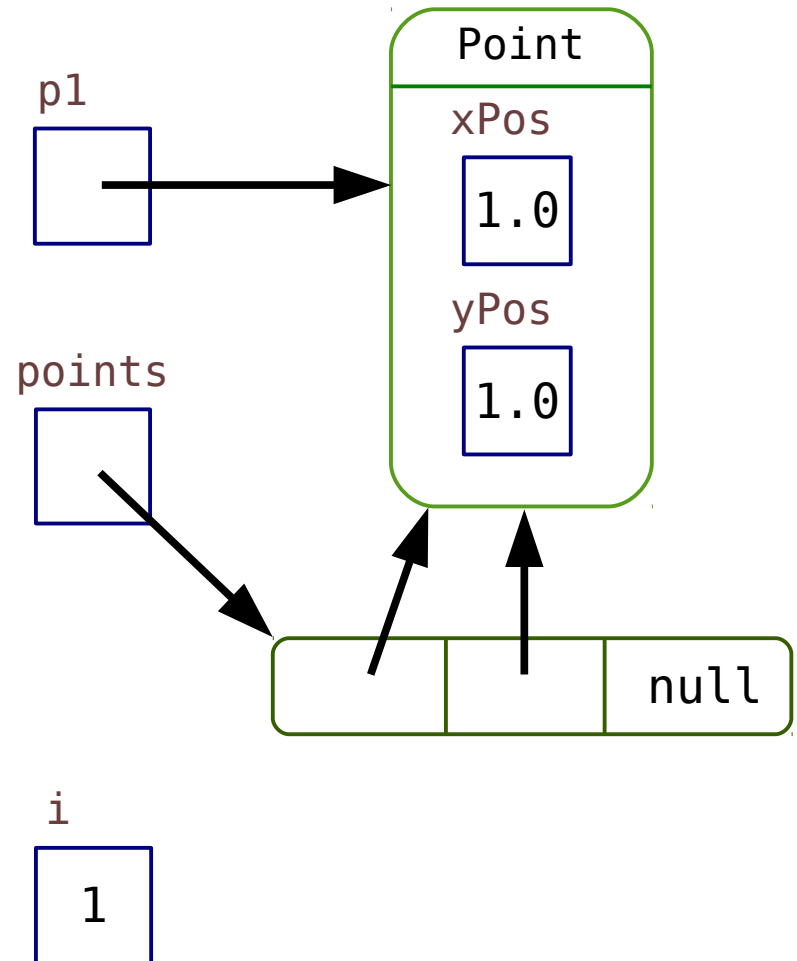
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
  
for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



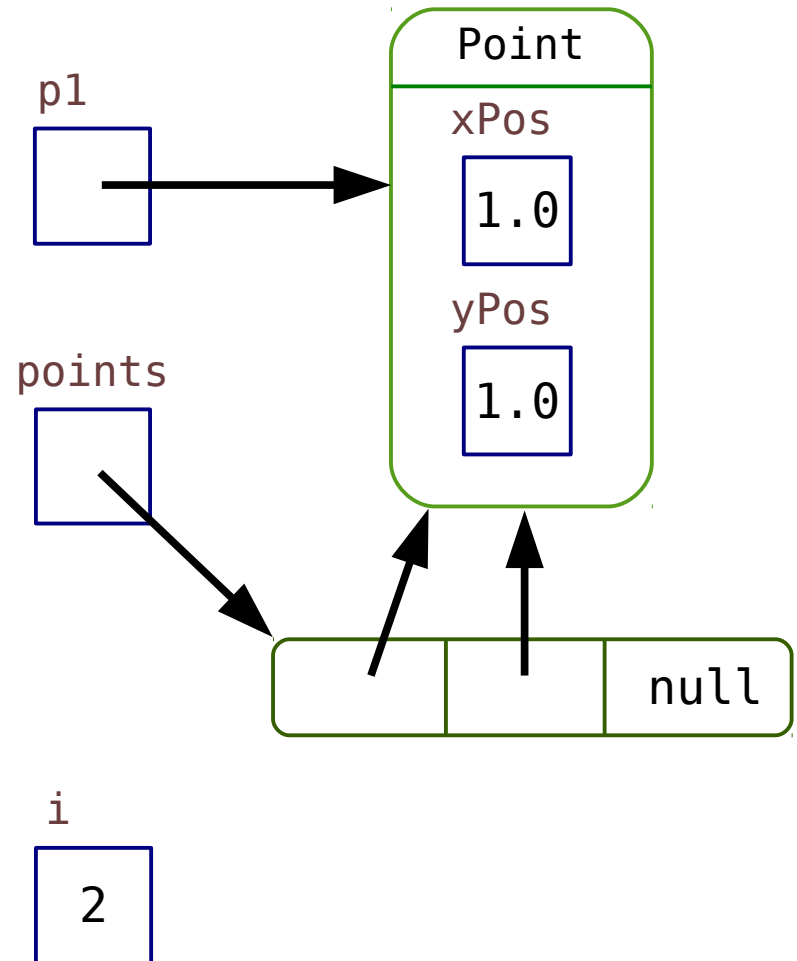
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
→ for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



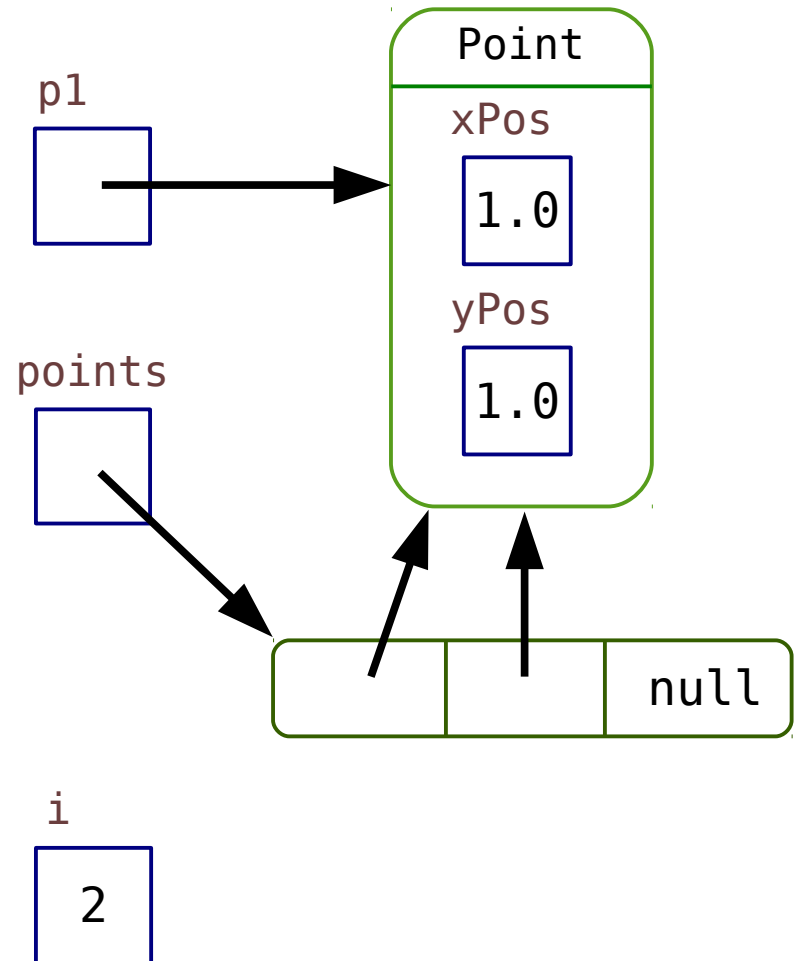
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
→ for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



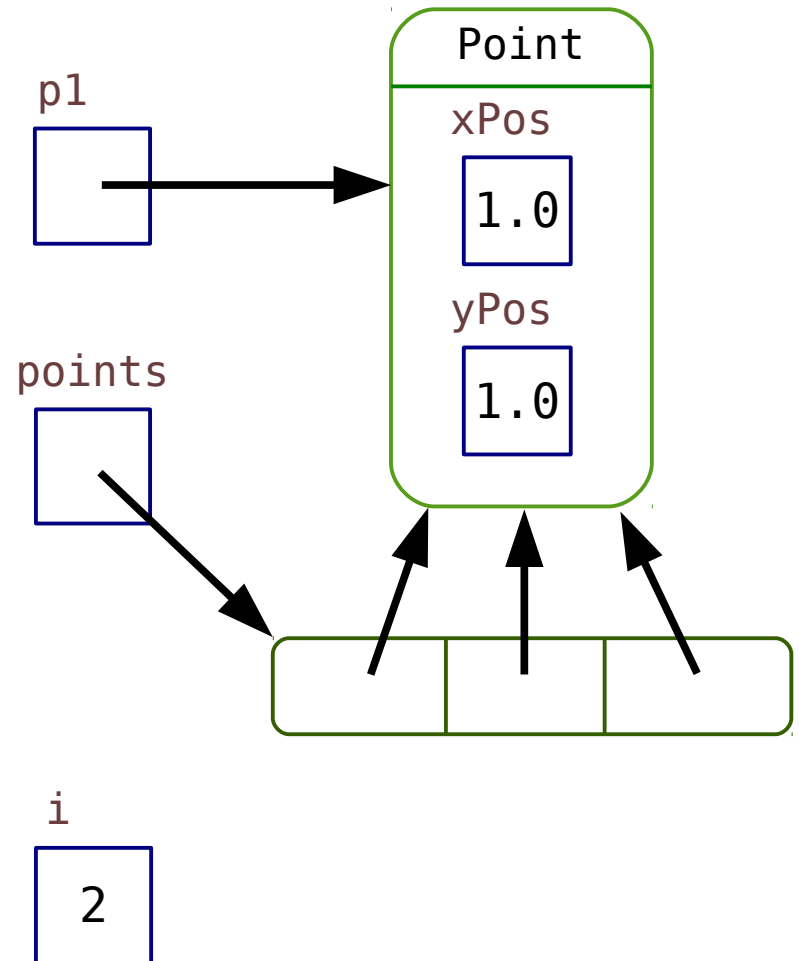
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
  
for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



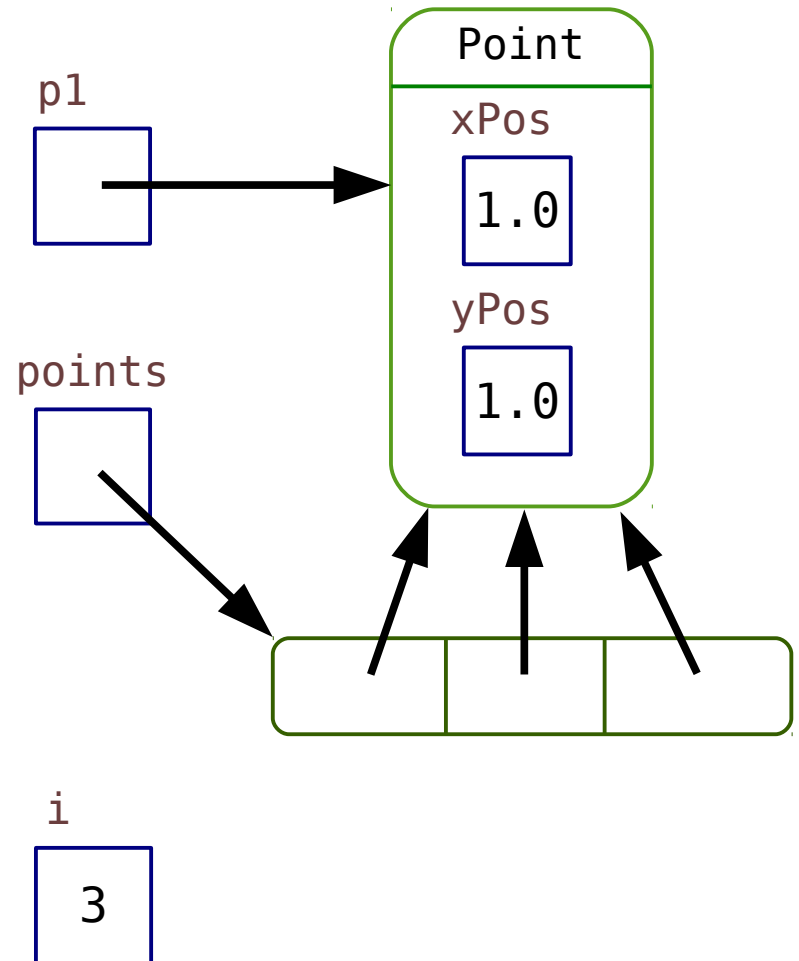
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
→ for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



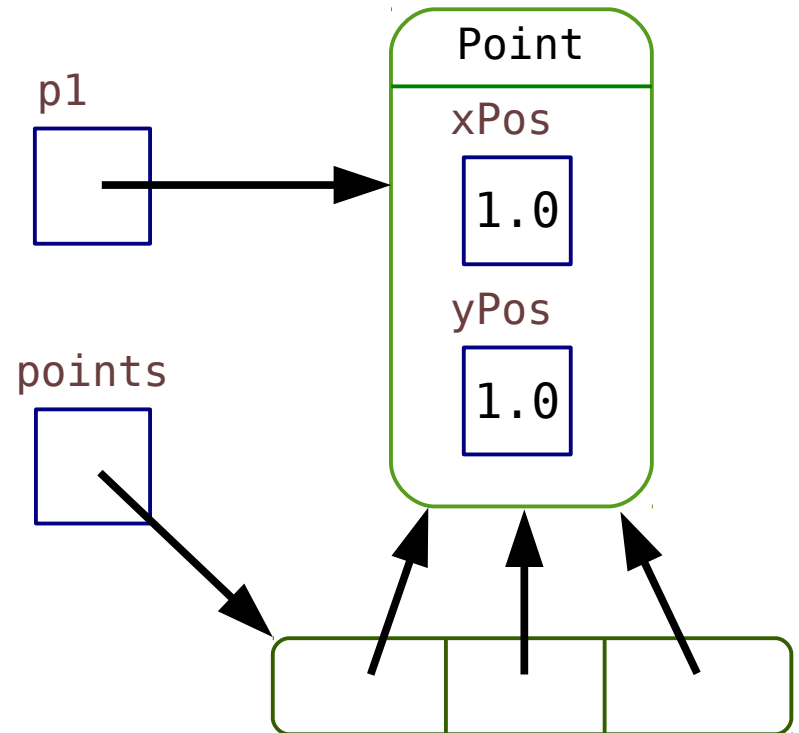
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
→ for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



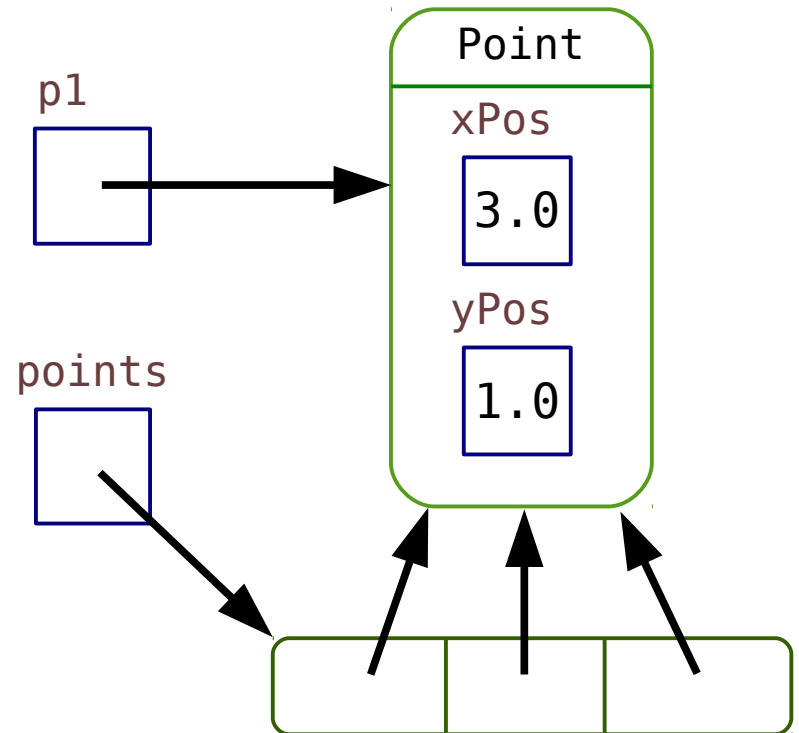
Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
  
for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
→ p1.setX(3.0);  
  
System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



Reference Arrays

```
Point p1;  
Point[] points;  
  
p1 = new Point(1.0, 1.0);  
points = new Point[3];  
  
for (int i = 0; i < points.length; i++)  
{  
    points[i] = p1;  
}  
  
p1.setX(3.0);  
  
→ System.out.println(p1);  
  
for (int i = 0; i < points.length; i++)  
{  
    System.out.println(points[i]);  
}
```



Array Issues

- ==
- Assigning array variables
- Passing arrays as arguments
- Final arrays
- Cloning arrays
 - Shallow vs. deep copies
- The [Arrays](#) Class

Class Syntax Reminder

```
public class Person
{
    private static int count = 0;

    private String name;
    private final int id;

    public Person(String name)
    {
        this.name = name;
        id = count;
        count++;
    }

    public String getName()
    {
        return name;
    }

    public void setName(String name)
    {
        this.name = name;
    }

    public int getID()
    {
        return id;
    }

    public static void resetPersonCount()
    {
        count = 0;
    }
}
```

```

public class Person
{
    private static int count = 0;

    private String name;
    private final int id;

    public Person(String name)
    {
        this.name = name;
        id = count;
        count++;
    }

    public String getName()
    {
        return name;
    }

    public void setName(String name)
    {
        this.name = name;
    }

    public int getID()
    {
        return id;
    }

    public static void resetPersonCount()
    {
        count = 0;
    }
}

```

Driver:

```

Person alice, bob, eve;

alice = new Person("Alice");
bob = new Person("Bob");

System.out.println(alice.getID());
System.out.println(bob.getID());

Person.resetPersonCount();

eve = new Person("Eve");

System.out.println(eve.getID());

```

Syntax OK?

What will be printed?