Variable Scope

By the end of this video you will be able to…

- Describe the notion of variable scope
- Explain the basic rules of scope for Java
- Draw memory models that incorporate scope
- Trace code using Java's rules for variable scope

```
public class SimpleLocation
{
    public double latitude;
    public double longitude;

    public SimpleLocation(double latIn,
                          double lonIn)
    {
        this.latitude = latIn;
        this.longitude = lonIn;
    }
    // More code here
}
```

There are 6 variables in this code. Can you find them?

```
public class LocationTester
{
  public static void main(String[] args)
  {
    SimpleLocation lima =
       new SimpleLocation(-12.0, -77.0);
  }
}
```

```
public class SimpleLocation
{
    public double latitude
    public double longitude;

    public SimpleLocation(double latIn,
                          double lonIn)
    {
        this.latitude = latIn;
        this.longitude = lonIn;
    }
    // More code here
}
```

There are 6 variables in this code. Can you find them?

```
public class LocationTester
{
  public static void main(String[] args)
  {
    SimpleLocation lima =
         new SimpleLocation(-12.0, -77.0);
  }
}
```

```
public class LocationTester
{
  public static void main(String[] args)
  {
    SimpleLocation lima =
       new SimpleLocation(-12.0, -77.0);
    latitude = 12.04;
  }
}
```

ERROR. Variable not defined here

```
public class LocationTester
{
  public static void main(String[] args)
  {
    SimpleLocation lima =
       new SimpleLocation(-12.0, -77.0);
    latitude = 12.04;
  }
}
```

The *scope* of a variable is the area where it is defined to have a value

```
public class LocationTester
{
  public static void main(String[] args)
  {
    SimpleLocation lima =
       new SimpleLocation(-12.0, -77.0);
  }
}
```

Local variables are declared inside a method

```
public class SimpleLocation
{
    public double latitude;
    public double longitude;

    public SimpleLocation double latIn,
                          double lonIn)
    {
        this.latitude = latIn;
        this.longitude = lonIn;
    }
    // More code here
}
```

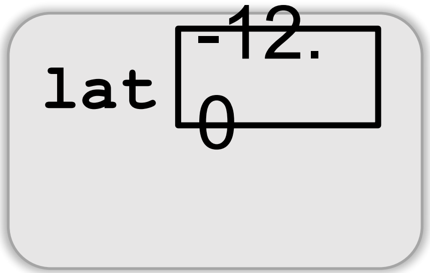
Parameters behave like local variables

```
public class SimpleLocation
{
    public double latitude;
    public double longitude;

    public SimpleLocation(double latIn,
                          double lonIn)
    {
        this.latitude = latIn;
        this.longitude = lonIn;
    }
    // More code here
}
```

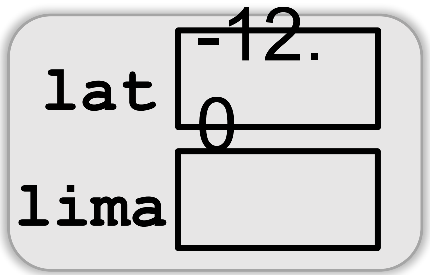
Member variables are declared outside any method

```
public class LocationTester
{
  public static void main(String[] args)
  {
    double lat = -12.0;
    SimpleLocation lima = new SimpleLocation(lat, -77.0);
  }
}
```



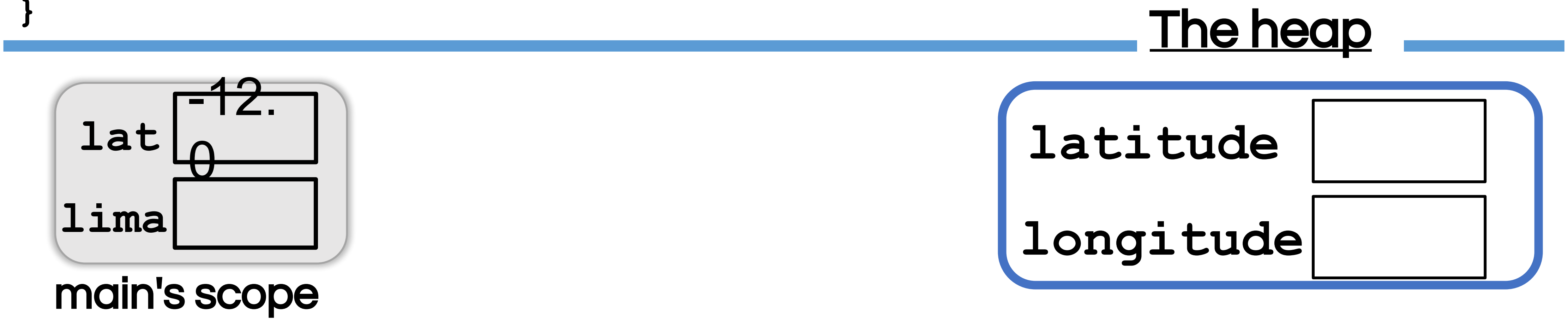
main's scope

```
public class LocationTester
{
  public static void main(String[] args)
  {
    double lat = -12.0;
    SimpleLocation lima = new SimpleLocation(lat, -77.0);
  }
}
```



main's scope

```
public class LocationTester
{
   public static void main(String[] args)
   {
      double lat = -12.0;
      SimpleLocation lima = new SimpleLocation(lat, -77.0);
   }
}
```



The heap

lat -12.0

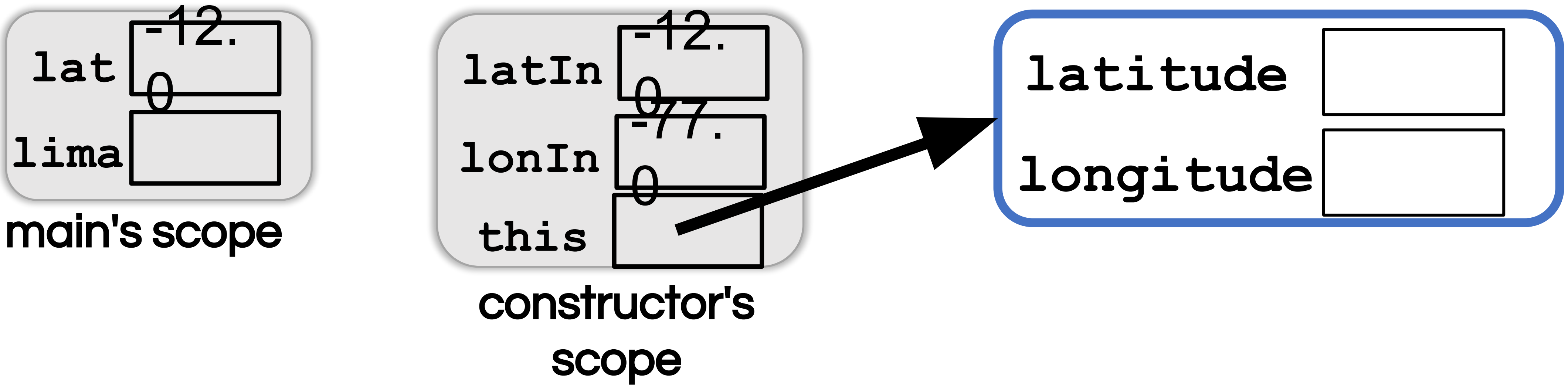
lima

main's scope

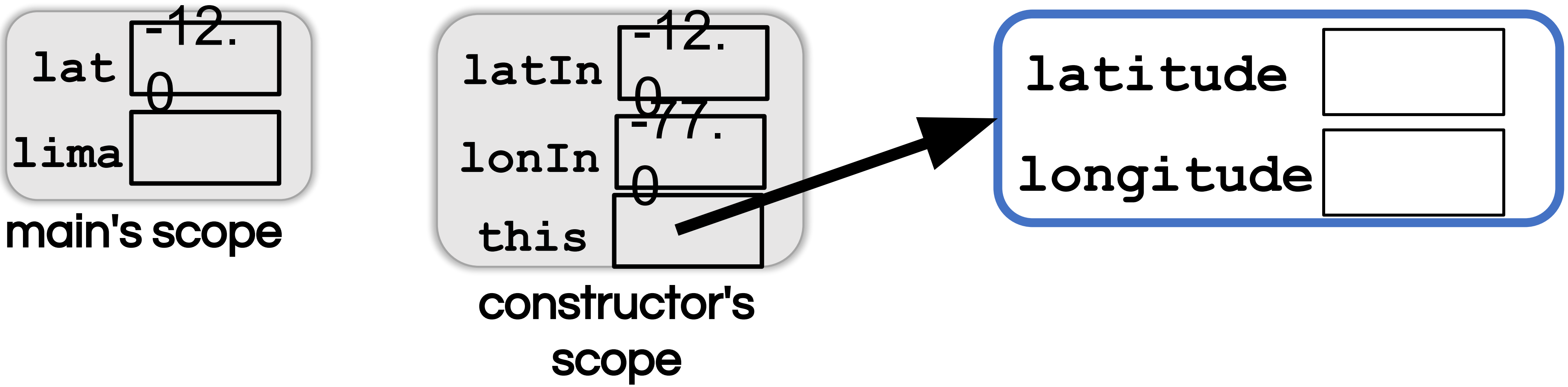
latitude

longitude

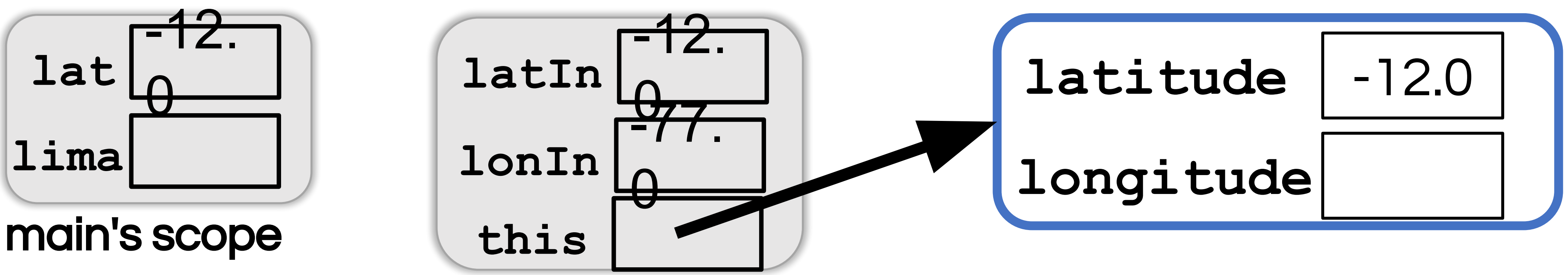
```
public SimpleLocation(double latIn,
                      double lonIn)
{
      this.latitude = latIn;
      this.longitude = lonIn;
}
```



```
public SimpleLocation(double latIn,
                      double lonIn)
{
    this.latitude = latIn;
    this.longitude = lonIn;
}
```



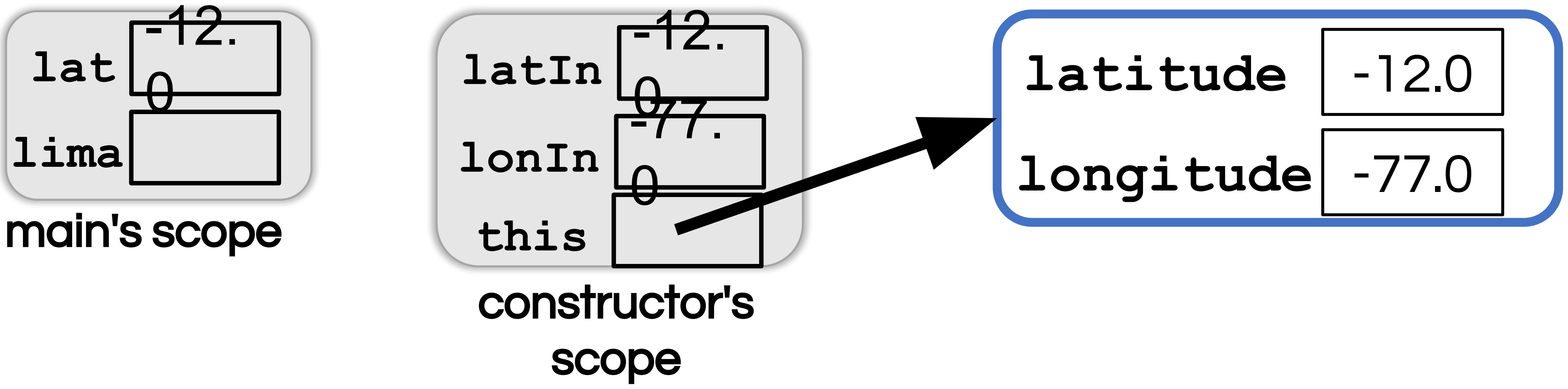
```
public SimpleLocation(double latIn,
                      double lonIn)
{
    this.latitude = latIn;
    this.longitude = lonIn
}
```



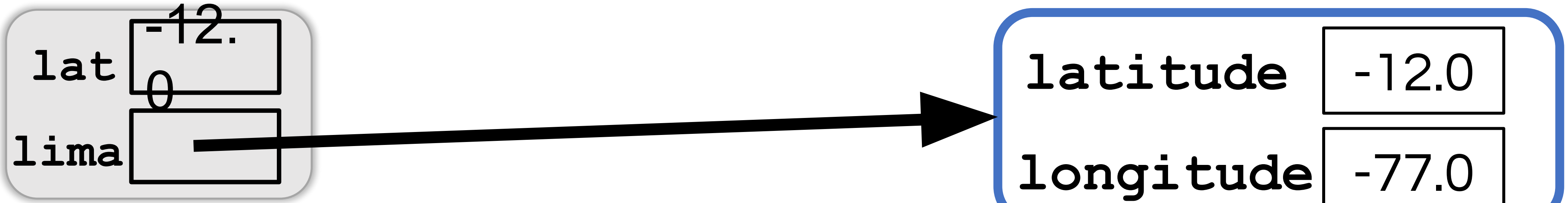
main's scope

constructor's scope

```
public SimpleLocation(double latIn,
                      double lonIn)
{
    this.latitude = latIn;
    this.longitude = lonIn;
}
```



```
public class LocationTester
{
  public static void main(String[] args)
  {
    double lat = -12.0;
    SimpleLocation lima = new SimpleLocation(lat, -77.0);
  }
}
```



lat -12.0

lima

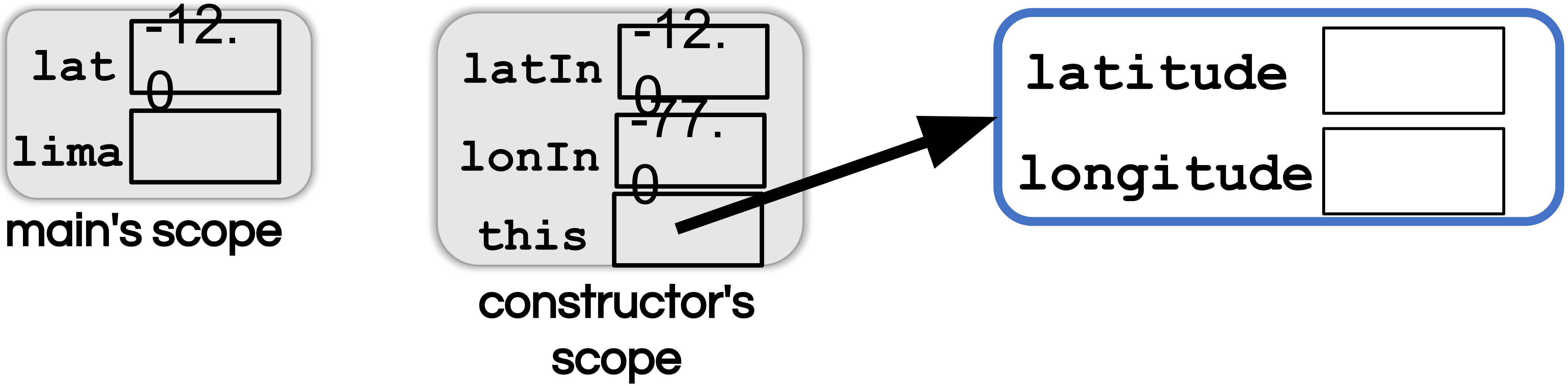
latitude -12.0

longitude -77.0

main's scope

```
public SimpleLocation(double latIn,
                      double lonIn)
{
    this.latitude = latIn;
    this.longitude = lonIn;
}
```

this is optional



lat -12.0

lima

main's scope

latIn -12.0

lonIn -77.0

this

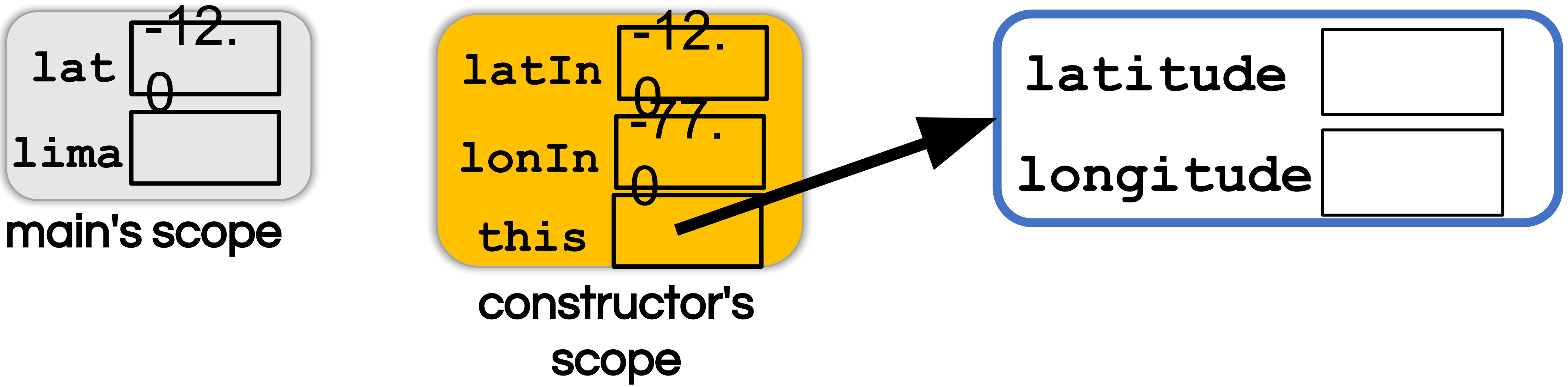
constructor's scope

latitude

longitude

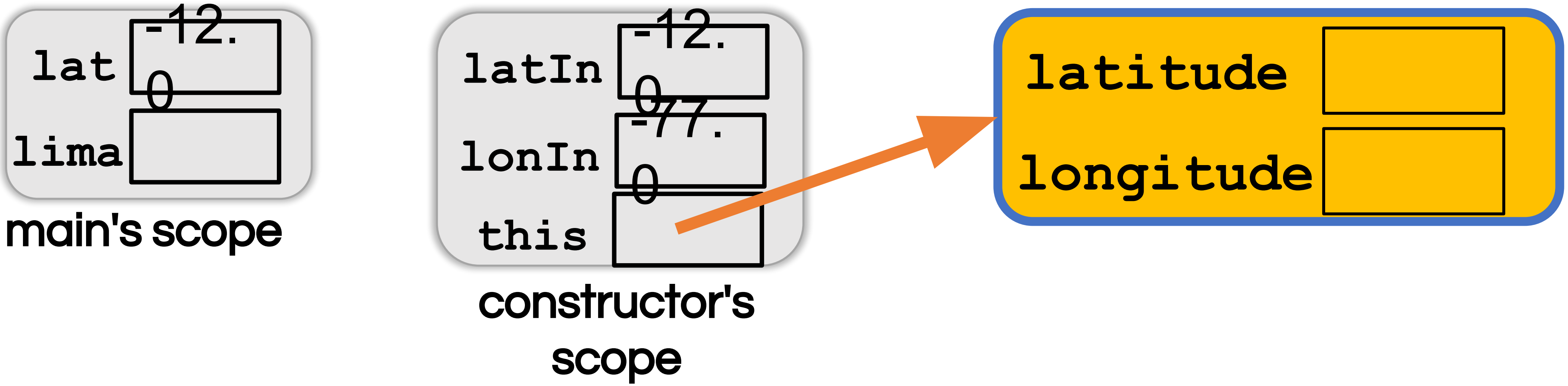
```
public SimpleLocation(double latIn,
                      double lonIn)
{
    latitude = latIn;
    longitude = lonIn;
}
```

Looks for `latitude` in the constructor's local scope



```
public SimpleLocation(double latIn,
                      double lonIn)
{
    latitude = latIn;
    longitude = lonIn;
}
```

Doesn't find it, so looks in calling object scope



lat -12.0

lima

main's scope

latIn -12.0

lonIn -77.0

this

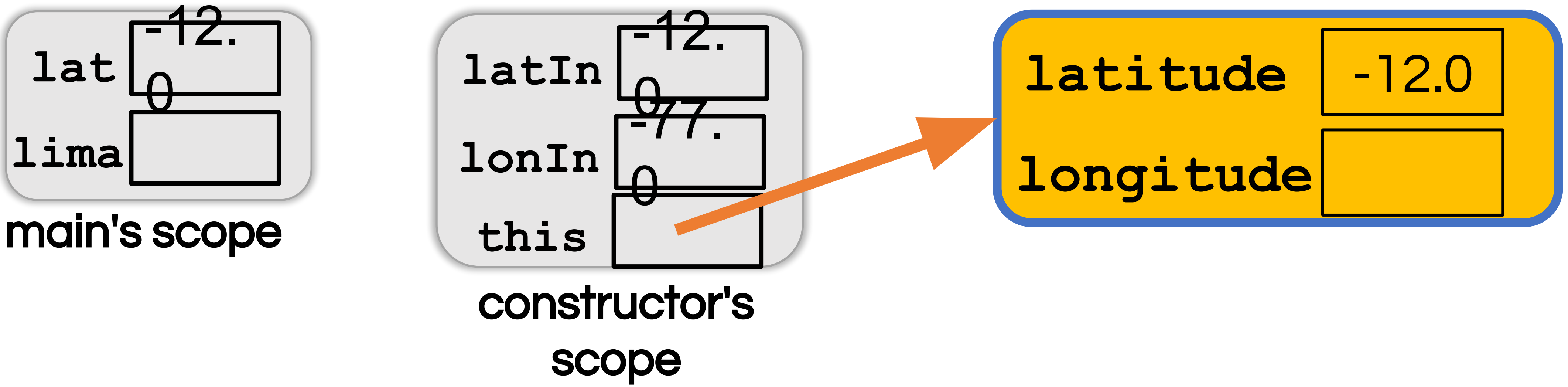
constructor's scope

latitude

longitude

```
public SimpleLocation(double latIn,
                      double lonIn)
{
    latitude = latIn;
    longitude = lonIn;
}
```

Doesn't find it, so looks in calling object scope



lat
-12.0
lima
main's scope
latIn
-12.0
lonIn
-77.0
this
constructor's scope
latitude
-12.0
longitude