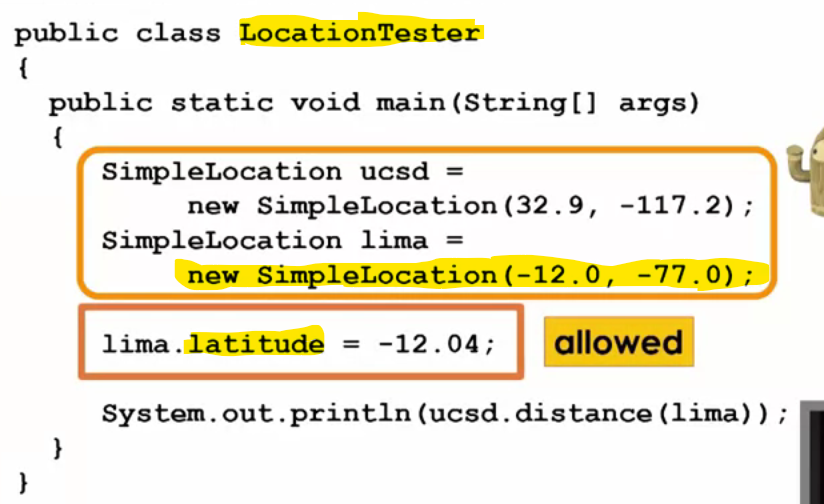
## Member variables , data the object needs to store, exist through out the class. Declared outside any methods but inside declaration of class

## 

**Any other class can access them**

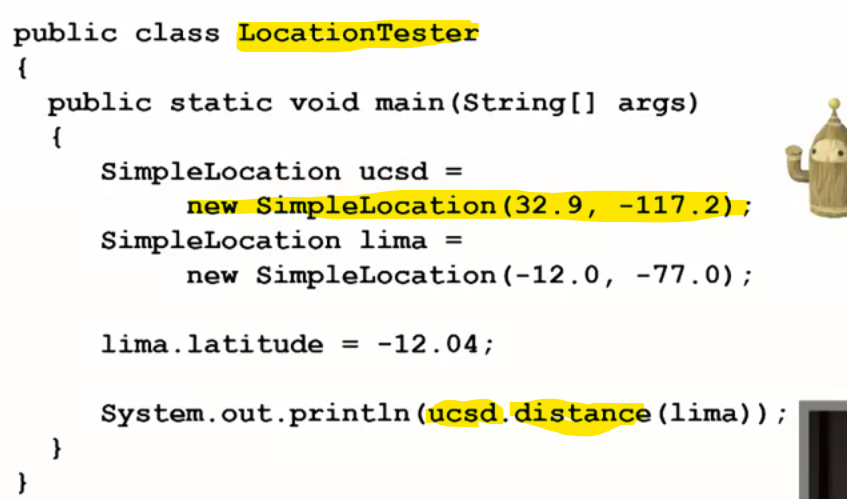
****

**Similarly Methods can be declared public**

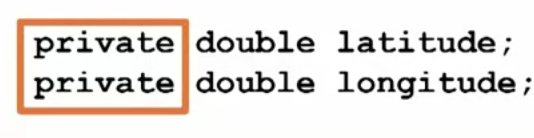
**A close-up of words

Description automatically generated**

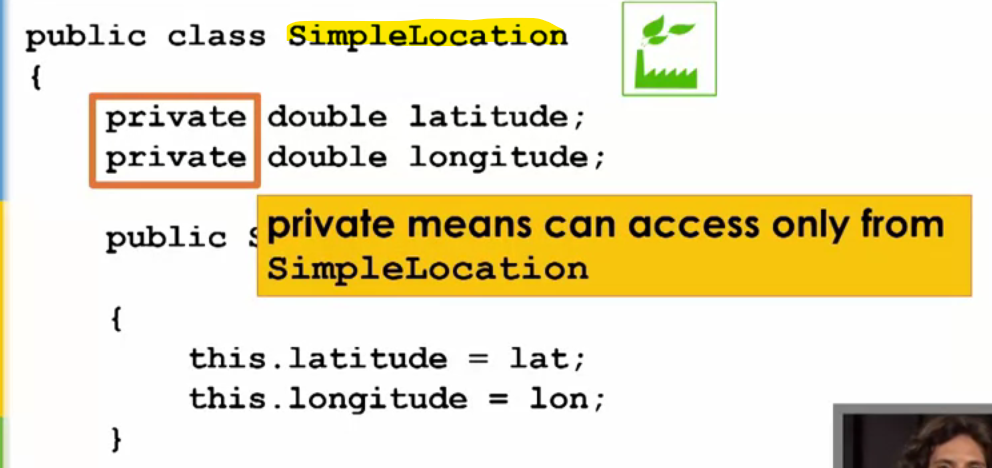
**Any other class can access distance method**

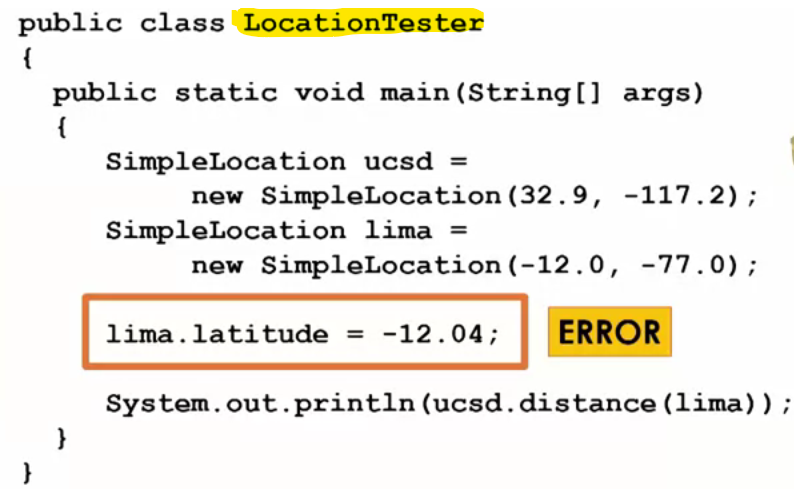
****

**Member variables declared private**

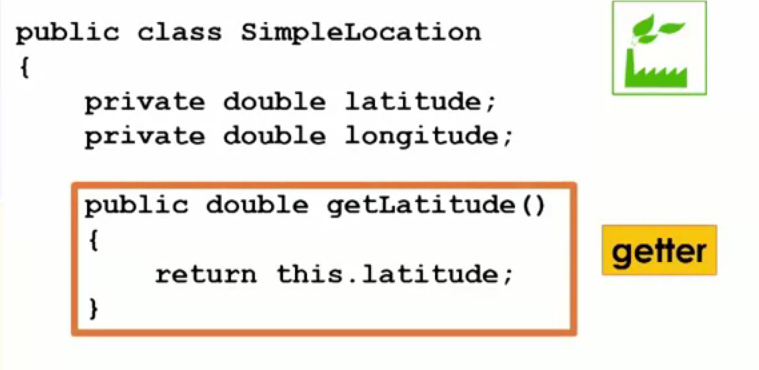
****

**These member variables are accessible only in the class where they are declared.**

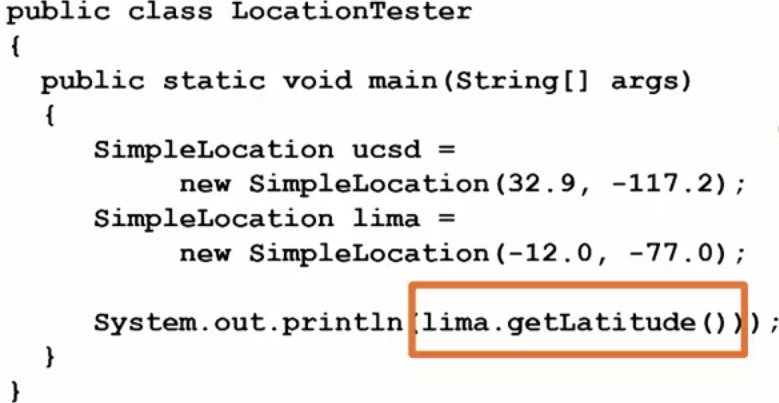
****

****

****

****



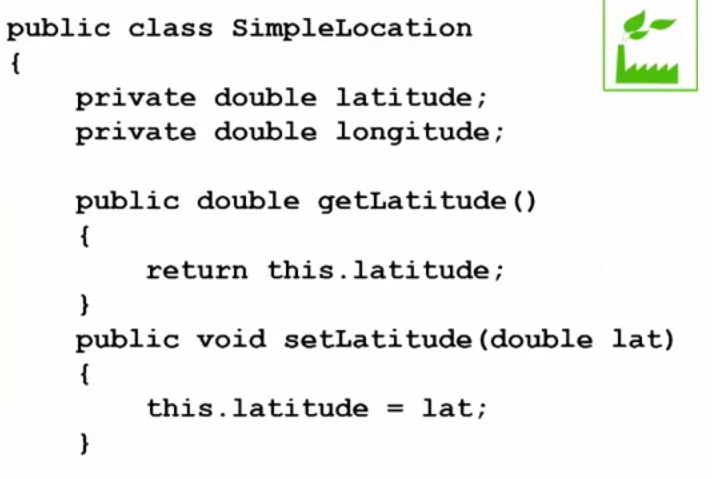
****

**Setter**

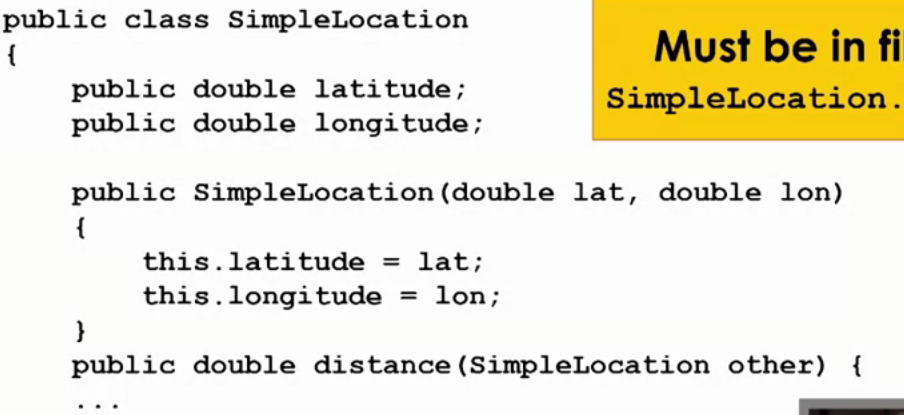
**A close-up of a computer code

Description automatically generated**



****

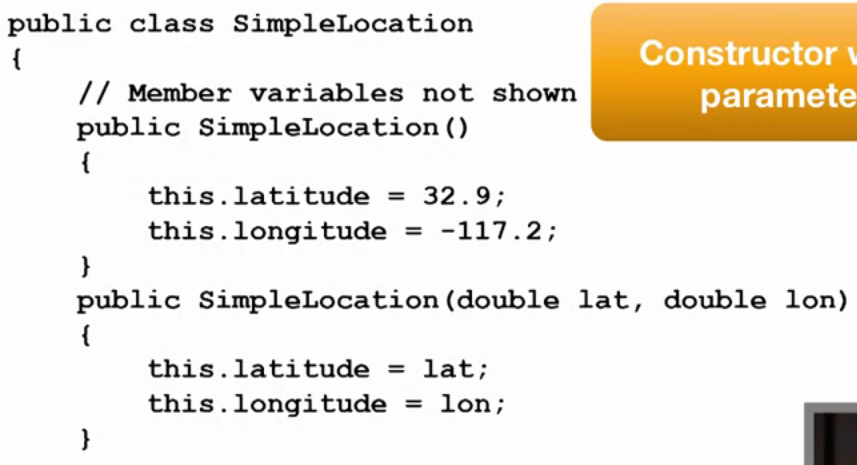
**Parameterized Constructor**

****



**Default Constructor**

Create a new constructor that takes in no arguments.



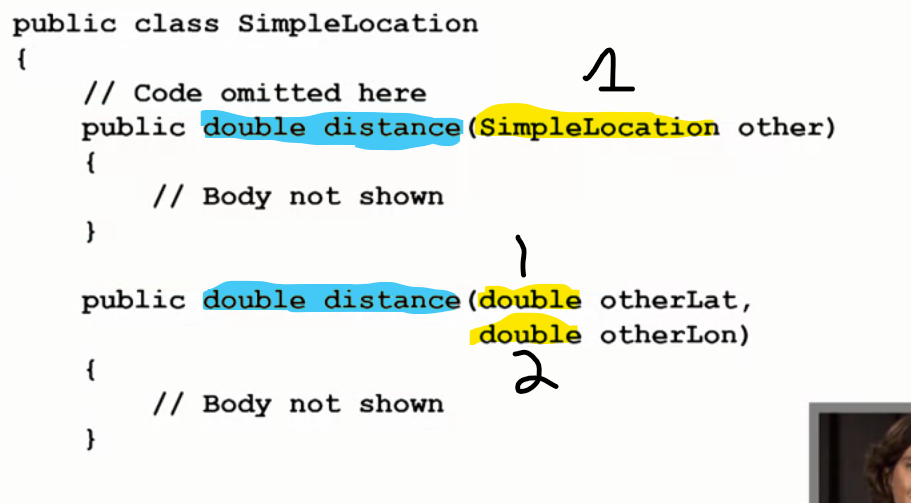


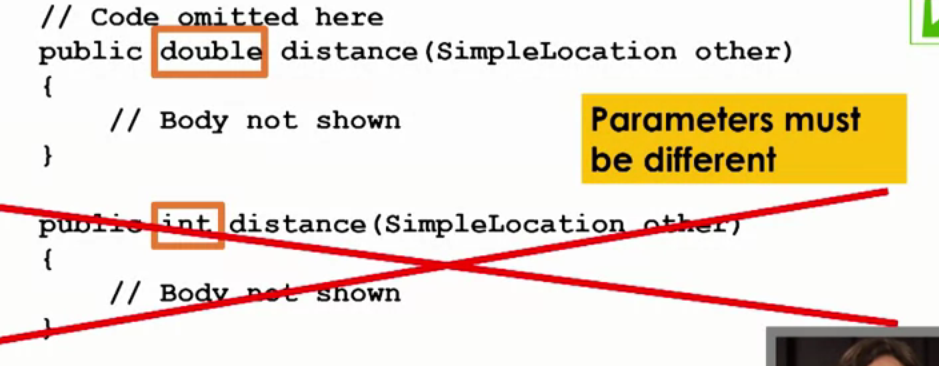
Inside which we have given default values

**Overload Constructor**

Two different copies of constructor that take different numbers and types of arguments.

**Overload Methods**

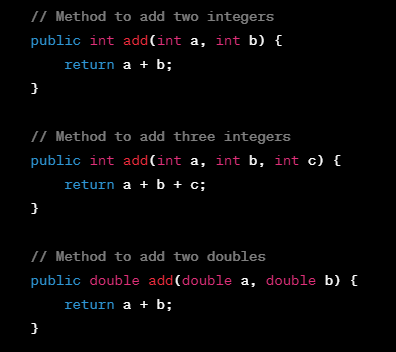
****

****



**Same Name , Same Parameter List(Number and Type) but different return type not allowed.**

**Following is allowed**

****

<http://unfoldingmaps.org/javadoc>

public class MyClass

{

  public int a;

  public double b;

  public MyClass(int first, double second)

  {

    this.a = first;

    this.b = second;

  }

  public boolean same(MyClass other)

  { 30 30 29.7===29.7

    return other.a == this.a && other.b == this.b;

  }

}

public class MyClassTester

{

  public static void main(String[] args)

  {

    MyClass c1 = new MyClass(30, 123.9);

    MyClass c2 = new MyClass(30, 29.7);

(30,29.7)

    MyClass c3 = new MyClass(c1.a, c2.b);

    System.out.println(c2.same(c3));

  }

}

public class MyClass3

{

  private int a;

  public MyClass3(int first)

  {

    this.a = first;

  }

}

public class MyClassTester

{

  public static void main(String[] args)

  {

    MyClass3 c1 = new MyClass3(30);

    System.out.println(c1.a);

  }

}

public class MyClass

{

  public int a;

  public double b;

  public MyClass(int first, double second)

  {

    this.a = first;

    this.b = second;

  }

  public boolean same(MyClass other)

  {

    return other.a == this.a && other.b == this.b;

  }

}

public class MyClassTester

{

  public static void main(String[] args)

  {

    MyClass c1 = new MyClass(30, 123.9);

    MyClass c2 = new MyClass(30, 29.7);

    MyClass c3 = new MyClass(c1.a, c2.b);

    System.out.println(c2.same(c3));

  }

}

We have created four objects