Week 5: Topics

- `this` Keyword in Java

Variable Hiding

- We cannot create two instances/local variables with the same name
- However, it is legal to create

  One instance variable and one local variable or method parameter with the same name

- In such a scenario

  The local variable will hide the instance variable

    This is called "variable hiding"

Variable Hiding

Blackboard: Week5/VariableHiding

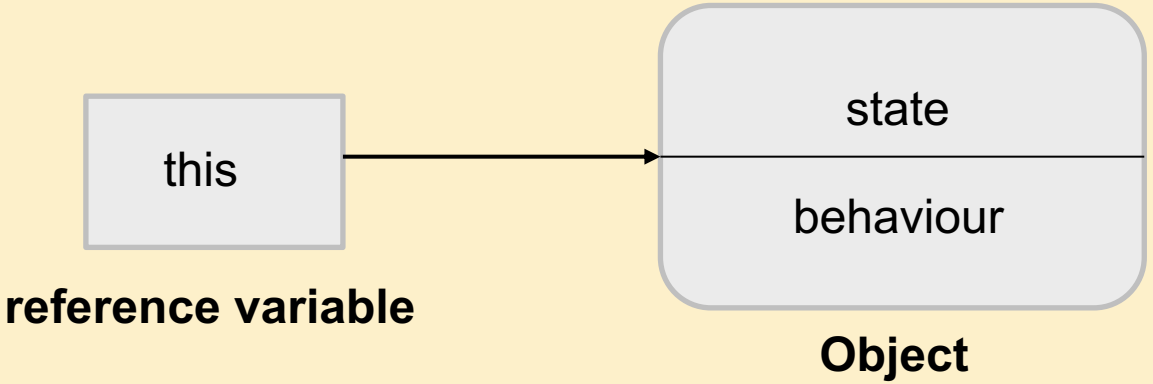
A quick tutorial:

Blackboard: Week5/Account_1

`this` Keyword in Java

- Within an instance method or a constructor, `this` is a reference to the **current object**

  The object whose method or constructor is being called



- You can refer to any member of the current object

  From within an instance method or a constructor by using t h i s

`this:` To refer to the current class instance variable

- The `this` keyword can be used to refer to the current class instance variable
- In the following example, the parameters and instance variables are same

  We use the t h i s keyword to distinguish between local and instance variables

`this:` To refer to the current class instance variable

Blackboard: Week5/This_1/Student1.java

Blackboard: Week5/This_1/Student2.java

Problem:

Suppose you are smart enough to choose different names for your instance variable and methods arguments.

Create a class `Account` with a `setData` method that takes parameters and sets the value of the instance variables.

Create two objects of the class, each calling the `setData` method.

Blackboard: Week5/Account_2/Account1.java

Blackboard: Week5/Account_2/Account2.java

Blackboard: Week5/Account_2/Account3.java

`this`: To invoke current class method

- You may invoke the method of the current class by using the `this` keyword
- If you do not use the `this` keyword

  Compiler automatically adds t h i s keyword while invoking the method

`this:` To invoke the current class method

Blackboard: Week5/This_2/A.java

`this:` To invoke the current class method

Blackboard: Week5/This_2/TestThis.java

`this()`: To invoke current class constructor

- The `this()` constructor call can be used to invoke the current class constructor
- It is used to reuse the constructor

  In other words, it is used for constructor chaining

`this()`: To invoke current class constructor

Blackboard: Week5/This_3/A.java

`this()`: to invoke current class constructor

Blackboard: Week5/This_3/A.java

Calling a Parameterized Constructor from Default Constructor

Blackboard: Week5/This_4/A.java

Calling a Parameterized Constructor from Default Constructor

Blackboard: Week5/This_4/TestThis.java

Constructor Chaining

Blackboard: Week5/This_5/Student.java

Constructor Chaining

Blackboard: Week5/This_5/TestThis.java

`this`: Passed as an argument in a method call

Blackboard: Week5/This_6/Student.java

`this`: Passed as an argument in a constructor call

Blackboard: Week5/This_7/A.java

`this`: Used to return current class instance

Blackboard: Week5/This_8/A.java