### Notes

#### Project01

• Sign-up for 10 minute meeting

#### Tournament

- 7pm Weds evening? April 10? Different time? A different day?
- Please LMK if you will attend, so I can arrange snacks

### Notes

#### Today

- Review "static", inheritance
- Review debugging with breakpoints

- Some people struggling to plan and write a Java program
- That is 100% fine—IF YOU GET HELP!
- Office hours, extra study problems, extra study on your own, extra programming on your own, etc.

Advanced Topics in Class Design

Member classes
Inheritance
Overriding Inherited Methods
Protected variables and methods

## The protected Modifier

The protected modifier allows a  ${\bf child}$  class to reference a variable or method in the  ${\bf parent}$  class

It provides more encapsulation than public visibility, but is not as tightly encapsulated as private visibility

A protected variable is also visible to any class in the same "package" as the parent class

## Visibility Revisited

It's important to understand one subtle issue related to inheritance and visibility

All variables and methods of a parent class, even private members, are inherited by its children

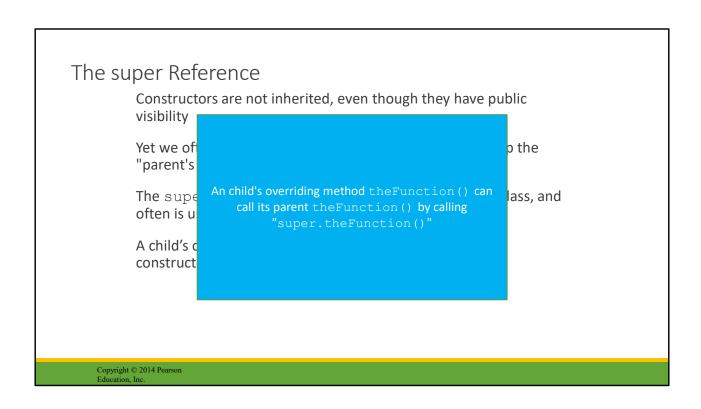
As we've mentioned, private members cannot be referenced by name in the child class

However, private members inherited by child classes exist and can be referenced indirectly

# Visibility Revisited

Because the parent can refer to the private member, the child can reference it indirectly using its parent's methods

The  ${\tt super}$  reference can be used to refer to the parent class, even if no object of the parent exists



## The super Reference

The  $\underline{\text{first line}}$  of a child's constructor should use the  $\mathtt{super}$  reference to call the parent's constructor

The  ${\tt super}$  reference can also be used to reference other variables and methods defined in the parent's class

```
Output
//******
                                                *****
// Words2.java
                 Number of pages: 1500
Number of definitions: 52500
                 Definitions per page: 35.0
public class Words2
  // Instantiates a derived class and invokes its inherited and
  // local methods.
  public static void main(String[] args)
     Dictionary2 webster = new Dictionary2(1500, 52500);
     System.out.println("Number of pages: " + webster.getPages());
     System.out.println("Number of definitions: " +
                      webster.getDefinitions());
     System.out.println("Definitions per page: " +
                      webster.computeRatio());
  }
}
```

Advanced Topics in Class Design

Member classes
Inheritance
Overriding Inherited Methods
Protected variables and methods
class Object

# The Object Class

A class called Object is defined in the java.lang package of the Java standard class library

All classes are derived from the Object class

If a class is not ex it is assumed to b

Therefore, the Ob hierarchies

Even classes you define yourself

an existing class,

of all class

# The Object Class

The  ${\tt Object}$  class contains a few useful methods, which are inherited by all classes

For example, the toString method is defined in the Object class

Every time we define the  ${\tt toString}$  method, we are actually overriding an inherited definition

## The Object Class

The equals method of the Object class returns true if two references refer to the same object (are "aliases")

We can override  ${\tt equals}$  in any class to define equality in some more appropriate way

As we've seen, the String class defines the equals method to return true if two String objects contain the same characters

Copyright © 2014 Pearson Education, Inc.

Do some examples!

Advanced Topics in Class Design

Member classes
Inheritance
Overriding Inherited Methods
Protected variables and methods
class Object
Class relationships

## Class Relationships

Classes in a software system can have various types of relationships to each other

Three of the most common relationships:

Inheritance: A is-a B
Dependency: A uses B
Aggregation: A has-a B

When you design several classes that interact, **understand the relationships** to choose whether to use inheritance, dependency, or containment.

Copyright © 2014 Pearson Education, Inc.

Student is-a Person Inventory uses Math Student has-a String (name, address, etc)