Lecture 13 Supplement

Access Modifiers

Dr. Martin O'Connor
CA166
www.computing.dcu.ie/~moconnor

Access Modifiers

- The first (left-most) modifier used lets you control what other classes have access to a member field
- For the moment, consider only public and private.
  - public modifier—the field is accessible from all classes.
  - private modifier—the field is accessible only within its own class.
- In the spirit of encapsulation, it is common to make fields private. This means that they can only be *directly* accessed from the Bicycle class
- Access from other classes is possible through *getter* and *setter* methods. E.g.: getSpeed(), getGear(), setCadence()

Controlling Access to Members of a Class

- Access level modifiers determine whether other classes can use a particular field or invoke a particular method.
- There are two levels of access control
  - At the top level—public, or *package-private* (no explicit modifier).
  - At the member level—public, private, protected, or *package-private* (no explicit modifier).
- A class may be declared with the modifier public, in which case that class is visible to all classes everywhere
- If a class has no modifier (the default, also known as *package-private*), it is visible only within its own package

Controlling Access to Members of a Class

- At the member level (within a class), you can also use the public modifier or no modifier (*package-private*) just as with top-level classes, and with the same meaning.
- For members, there are two additional access modifiers: private and protected.
- The private modifier specifies that the member can only be accessed in its own class.
- The protected modifier specifies that the member can only be accessed within its own package (as with *package-private*) and, in addition, by a subclass of its class in another package.

Controlling Access to Members of a Class

- At the member level (within a class), you can also use the public modifier or no modifier (*package-private*) just as with top-level classes, and with the same meaning.
- For members, there are two additional access modifiers: private and protected.
- The private modifier specifies that the member can only be accessed in its own class.
- The protected modifier specifies that the member can only be accessed within its own package (as with *package-private*) and, in addition, by a subclass of its class in another package.

Access Modifiers – Access Levels

The following table shows the access to members permitted by each modifier

Modifier	Class	Package	Subclass	World
`public`	Y	Y	Y	Y
`protected`	Y	Y	Y	N
`No modifier`	Y	Y	N	N
`private`	Y	N	N	N

Access Modifiers – Access Levels

- The first data column indicates whether the class itself has access to the member defined by the access level. As you can see, a class always has access to its own members.
- The second column indicates whether classes in the same package as the class (regardless of their parentage) have access to the member
- The third column indicates whether subclasses of the class declared outside this package have access to the member
- The fourth column indicates whether all classes have access to the member

Access Modifiers – Access Levels

- Access levels affect you in two ways:
- First, when you use classes that come from another source, such as the classes in the Java platform, access levels determine which members of those classes your own classes can use
- Second, when you write a class, you need to decide what access level every member variable and every method in your class should have
- Advice
  - Use the most restrictive access level that makes sense for a particular member. Use private unless you have a good reason not to
  - Avoid public fields except for constants.