Students studying for the AP Computer Science exam, starting in the Fall of 2014, should append this text at the end of Chapter Ten, Lesson Three.

Accessor and Mutator Methods

We just demonstrated how to write a method such as **askName**() to savely give a copy of private, internal data to some other object. This type of method is called an "accessor" method because it gives access to data.

Now, if you want to allow another object to change your internal data, you can create a "modifier" or "mutator" method as well. "Mutator" is a fancy term for any function that lets other classes change your object. Let's add a mutator method that will allow others to rename our **Knight**.

```
public class Knight
{
    // your name is private
    private String myName = "Sir Jousts-a-Lot";

    // but anyone can change your name, so long as it's valid
    public void setName(String newName)
    {
        // validate new name before updating internal data
        if ((newName != null) && (newName.length() > 0))
        {
            myName = newName; // update my internal name
        }
    }
}
```

Our **setName**() function allows others to change our **private** name, but will first validate the new name to make sure it is not **null** or an empty string. If **myName** was public then anyone could set it to an invalid value, but making it **private** gives us a chance to validate any request to change it.

Visibility of Classes to other Packages

Remember that each Java class belongs to a package such as **java.util**, or one that you define such as **my.package**, or a **default** package if you don't specify any package name. Each package can hold many objects. By default, only objects inside the same package can access each other.

Think about these three objects, **Widget1**, **Widget2**, and **Widget3**, each in a different package.

- **Widget1** (default package)
- **my.package.Widget2**
- **my.otherpackage.Widget3**

By default, none of these objects can see or use each other. However, we can apply the **public** keyword to an entire class. That will make the class visible to other packages.

```
public class Widget2
{
}
```

Now the **Widget2** object can be used from anywhere, including **Widget1** in the default package and **Widget3** in **my.otherpackage**.

If you do not declare any visibility for your class, then by default it gets "package-private", which means it can be used by other classes in the package but is not visible outside the package. So **Widget2** still cannot use **Widget1** or **Widget3**. However, any other classes in **my.package** can use **Widget2**, and other classes in **my.otherpackage** can use **Widget3**.

