Java: Packages

Quick summary

- Packages are a way to organize classes into groups.
- Packages are directories for your classes.
- Putting classes into packages helps you avoid naming conflicts with classes.
- Packages are lower case by convention, with no spaces or underscores.
- You can choose the package when creating a new class.
- If you leave the Package field blank when creating a class, it puts the class into the default package.

Eclipse: packages

When you create a class in a package other than the default, the package name is listed at the top:

```
package warehouse;


public class Facility {


}
```

In the class with your main method, if you want to use a class that's not in the default package, you have to import the package like this:

```
import warehouse.Facility;
```

Here I'm importing the Facility class from the warehouse package. To use any class from a package, use this syntax:

```
import warehouse.*;
```

If you click Command + Shift + o, or right click and choose Source > Organize Imports, Eclipse will automatically import only the classes you need based on the code you're writing, so you don't have to worry about this much.

Packages within packages

Just as you can have folders within folders, you can have packages within packages. to denote packages within packages, you use dots (instead of slashes with folders).

import warehouse.location.Kansas;

Here we have a package named `warehouse` that contains a package named `location`. Inside location, we're using the `Kansas` class.

Example

Here's an example of a package from Java 7 for Absolute Beginners:

package com.bryantcs.examples.syntaxExample;

Here's the explanation from the same book:

By convention, the package declaration follows a particular format. Although not strictly Java syntax (a program can work without it), most Java developers think ill of you if you don't follow the format. It's easiest to unravel this format by working from right to left:

- syntaxExample indicates the local name of the package. AverageTest is a simple application and so had just the one package.
- examples indicates the parent package for the local package. We keep all of the examples for this book in the examples package.
- com.bryantcs is the identifier for all of the applications, to keep them separate from those of other people when they get out on the Internet. After all, many people are likely to have an examples package, and someone else might even have an examples.syntaxExample package. Package declarations often look like reverse URLS. You might expect to see something like bryantcs.com/examples/syntaxExample (but don't visit that site; it doesn't exist). As we see next, the domain portion (com.bryantcs) provides the final bit of insurance that our class is unique.

– Java 7 for Absolute Beginners:

Now, suppose we want to use other classes within the same package. We can then use a wildcard character (*) to specify any class within the package, thus:

```
    import java.text.*;
```

– Java 7 for Absolute Beginners:

Syntax traditions

Traditionally, you flip around the package to fit a reverse URL: com….

You can have child classes that have parent classes in different packages.