SCHULICH School of Engineering

ENSF 593/594

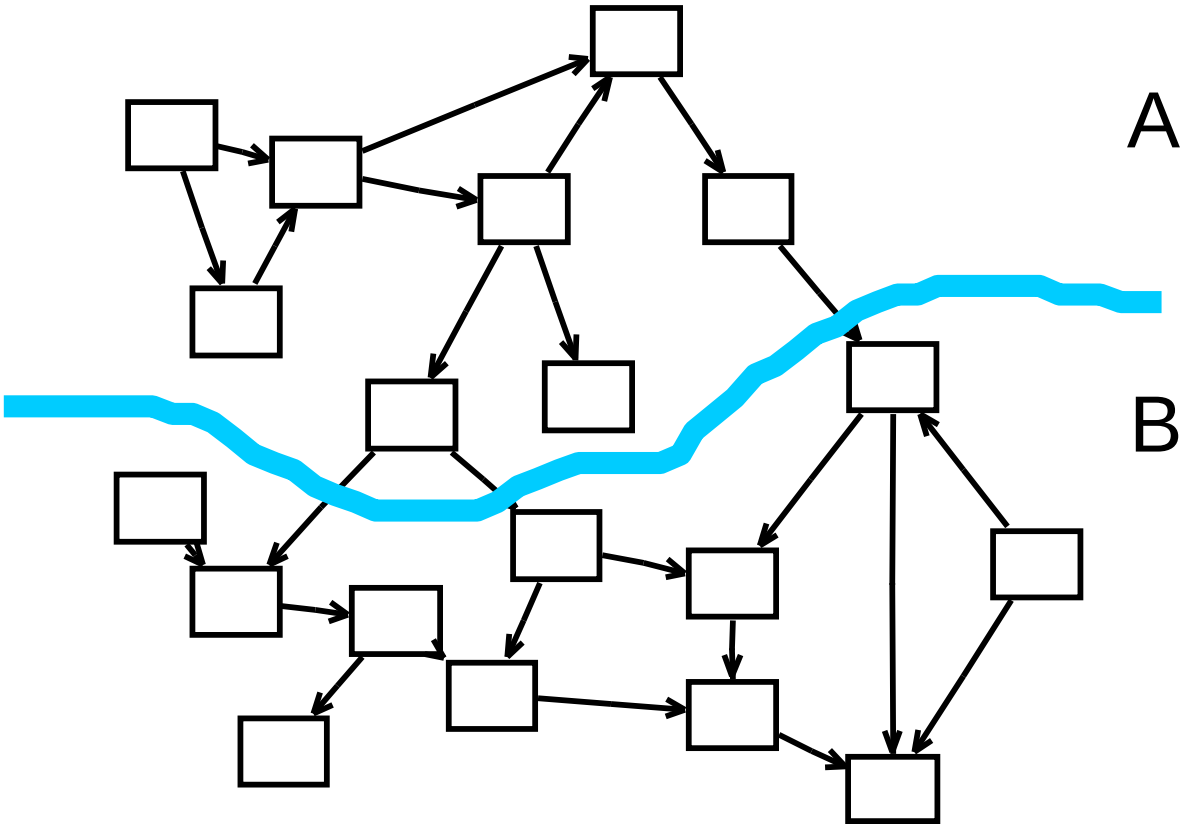
12 - Java Packages

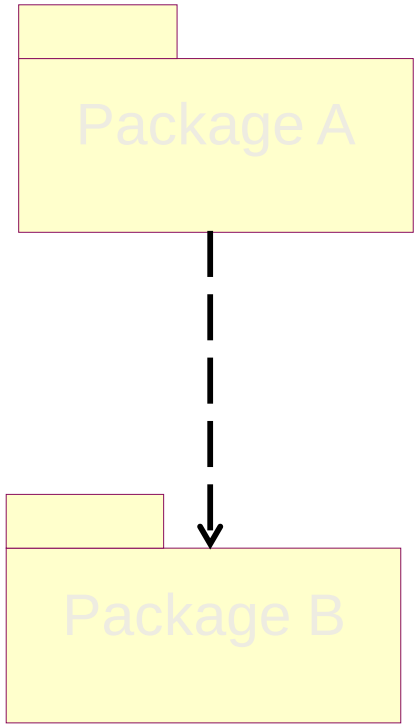
Partitioning and Packaging

Logical View

- Provides a static picture of the primary classes and their relationships
- The logical view is captured in class diagrams which contain the packages, classes, and relationships that represent the key abstractions of the system under development

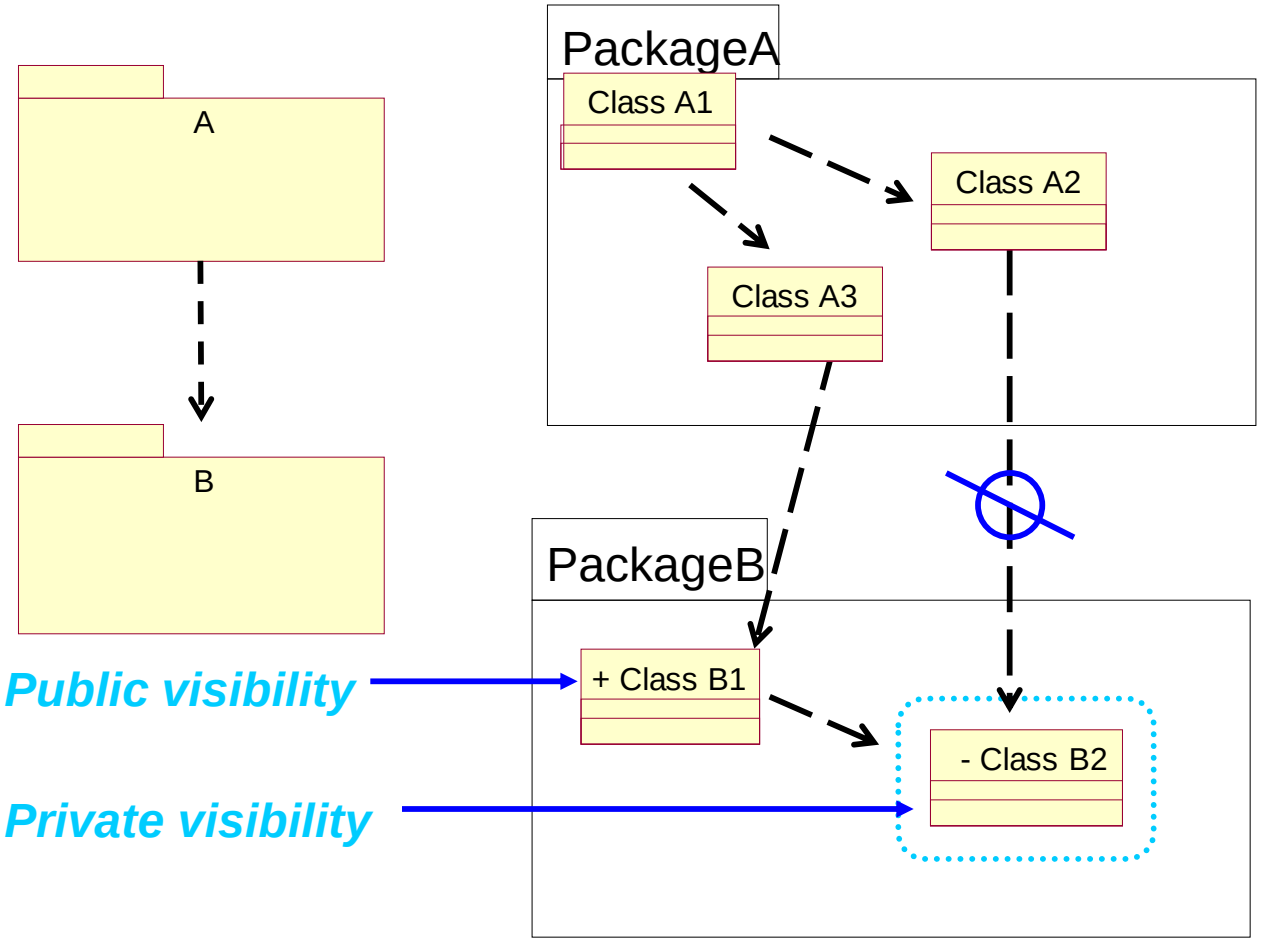
Example: Partitioning





Maximum cohesion within packages, minimal coupling between packages. The dependencies between packages (right) reflect/support the dependencies between the classes contained within the packages (left).

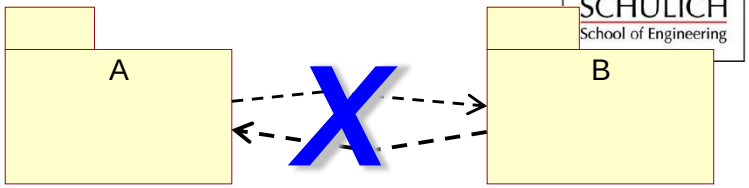
Package Dependencies: Package Element Visibility



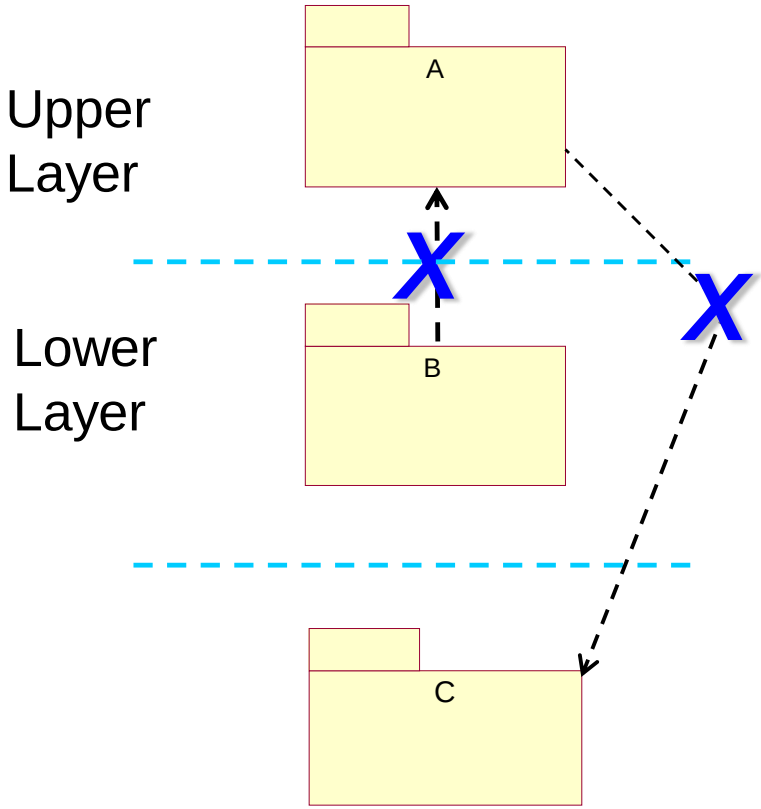
Only public classes can be referenced outside of the owning package

OO Principle: Encapsulation

Package Coupling: Tips



- Packages should not be cross-coupled
- Packages in lower layers should not be dependent upon packages in upper layers
- In general, dependencies should not skip layers



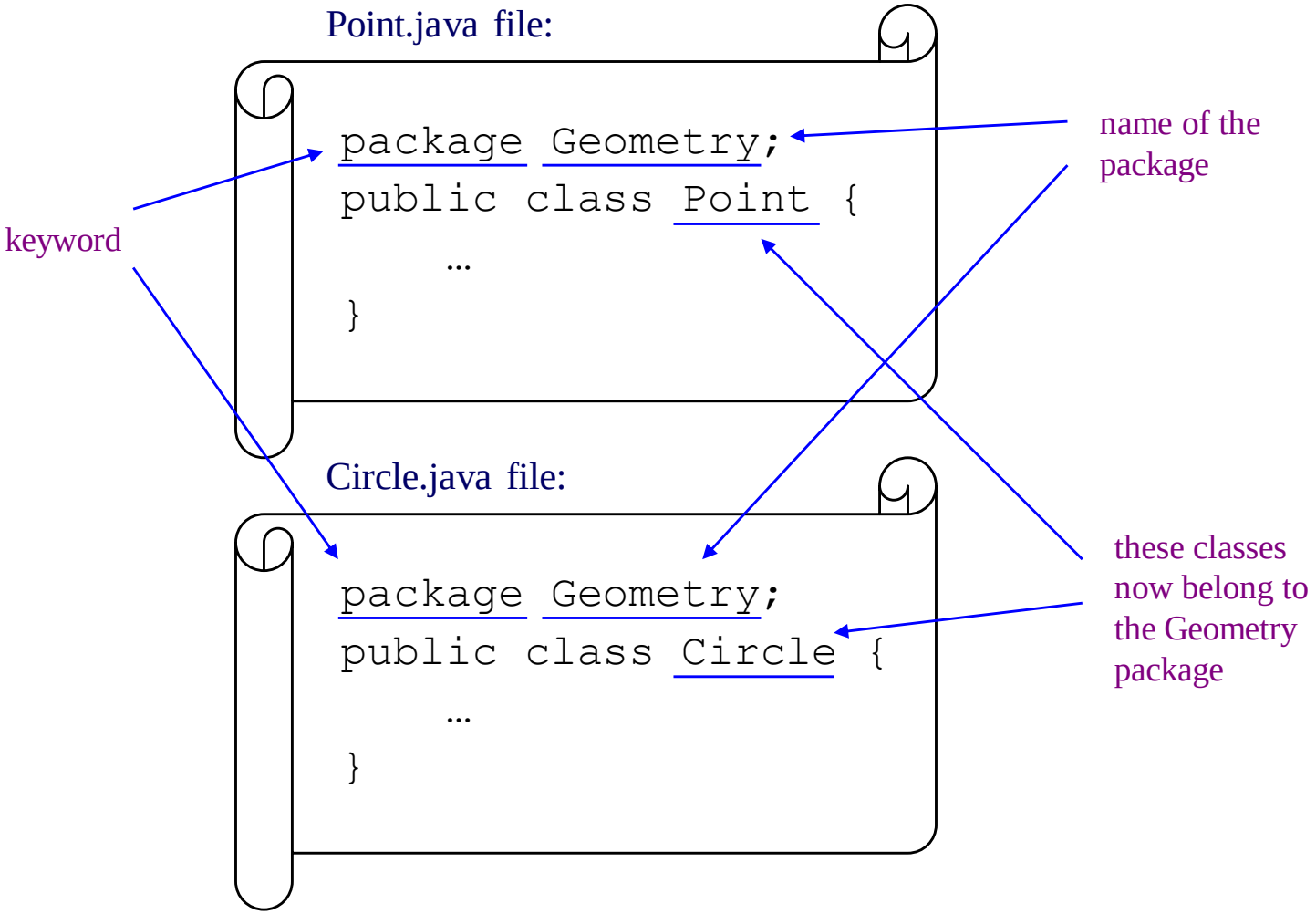
X = Coupling violation

Packages in Java

Packages

- Classes and interfaces are put into a package by using the “package” declaration before any code in the source file.
- Example:

Packages (continued)



Packages (continued)

- Normally, all the class and interface files are put into a subdirectory with the same name as the package.
- Unless declared private, methods and variables are accessible to all code in the package (i.e. have “package scope”).

Packages (continued)

- To access public classes, methods and variables in *another* package, you can:
  - Use the fully qualified name. E.g.

```
Geometry.Point myPoint;
```

Import part or all of a package, and use the simple name. E.g.

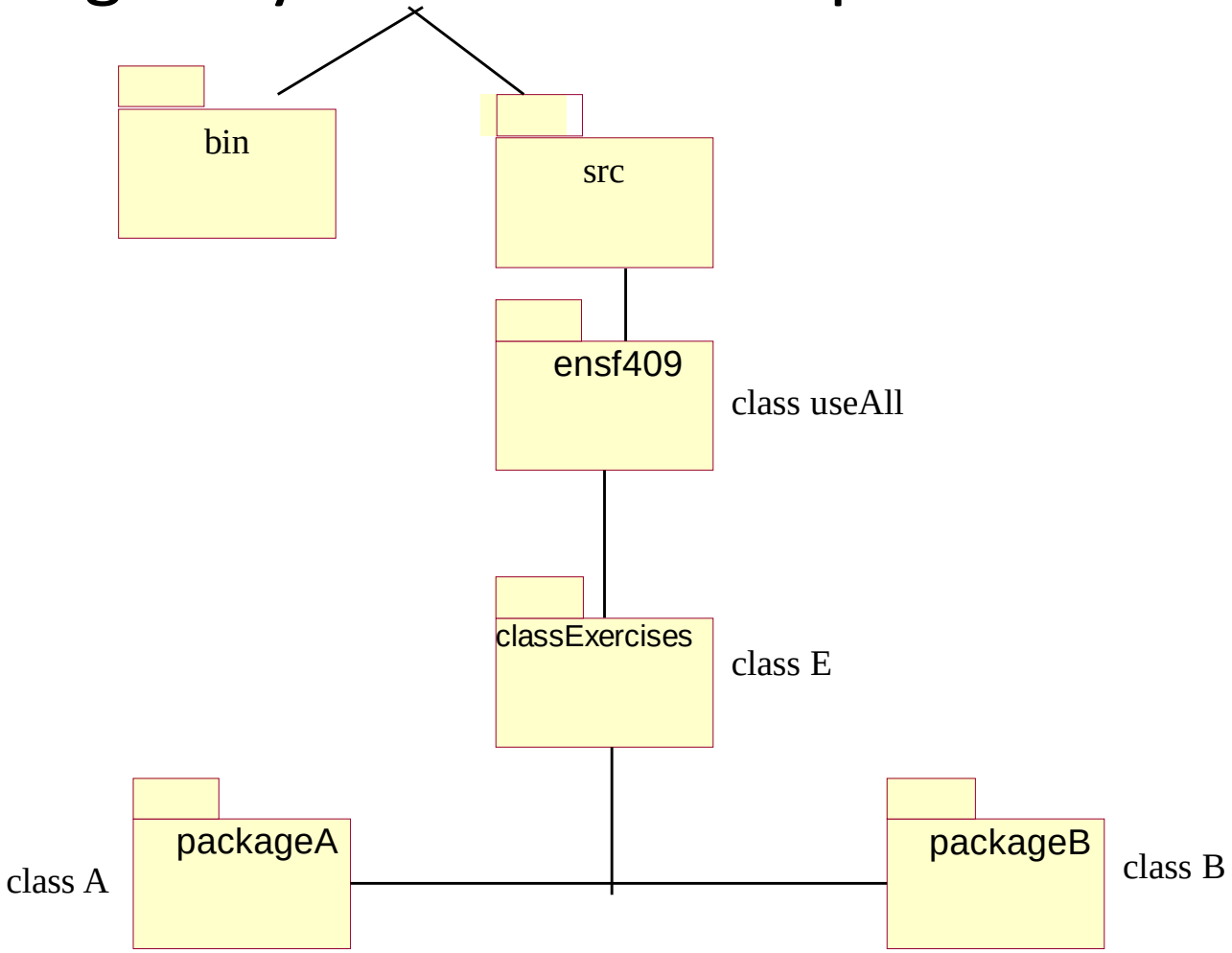
```
import Geometry.*;  // imports all
…                   // classes in package
Point myPoint;
```

Class Exercise

Class Exercise

- Create a working directory called src that contains a class called Demo.
- Under the src directory, we would like to create java package called ensf409.calssExercises.packageA, With two classes A1, and A2.
- Under the src directory, we would like to create java package called ensf409.classExercises.packageB, With two classes B1, and B2.

Package Physical Relationship



javac *.java classExercises/*.java classExercises/packageA/*.java classExercises/packageB/*.java

javac *.java classExercises/*.java classExercises/packageA/*.java classExercises/packageB/*.java -d ../../bin