OBJECT ORIENTED ANALYSIS & DESIGN
DATA STRUCTURES & ALGORITHMS

Packages


PACKAGES

Why Packages?

- Programmers can easily determine that these classes are related.
- Programmers know where to find files of similar types.
- The names won't conflict.
- You can have define access of the types within the packages.

What is a Package?

- A Java Package is a mechanism for organizing.
- Java classes into namespaces.
- Programmers use package to organize classes belonging to the same category.
- Classes in the same package can access each other's package –access members.

Naming Convention of a Package.

- Packages names are written in all lower case. (It is not mandatory. However, it is standard convention that is followed)
- Companies use their reversed internet domain name to begin their package names.
- For example: com.example.mypackage for a named mypackage created by a programmer at example.com

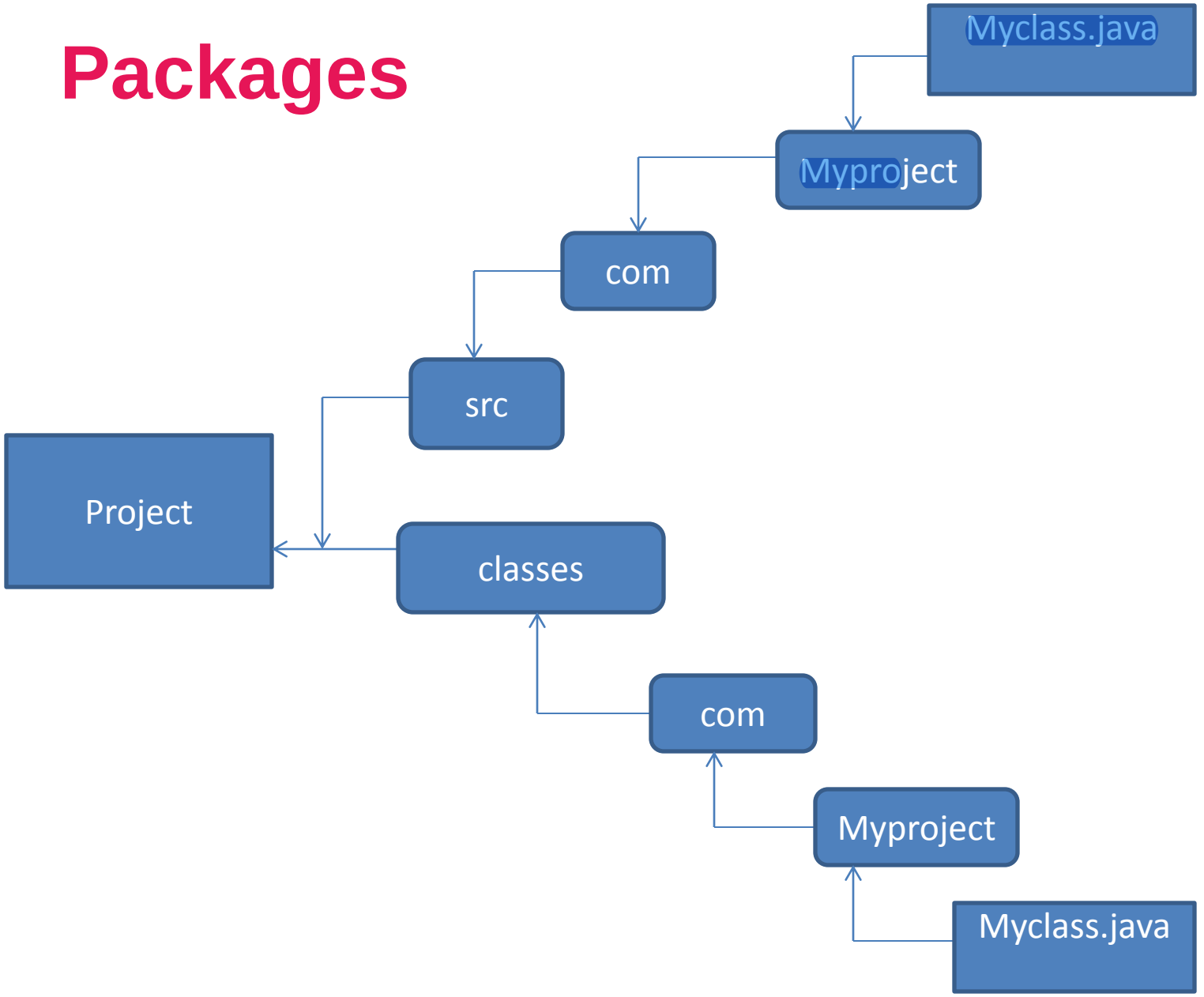
Naming Convention of a Package

If the domain name contains :

- a hypen or a special character.
- If the package name begins with a digit, illegal character reserved Java keyword such as “int”.
- In this event, the suggested convention is to add an underscore as follows:

Legalizing Package Names	
Domain Name	Package Name Prefix
hyphenated-name.example.org	Org.example.hyphenated_name
Example.int	int_.example
123name.example.com	com.example_123name

Packages


Myclass.java
Myproject
com
src
Project
classes
com
Myproject
Myclass.java

Program on Package

Company URL-> auribises.com

Package name-> com.auribises.db

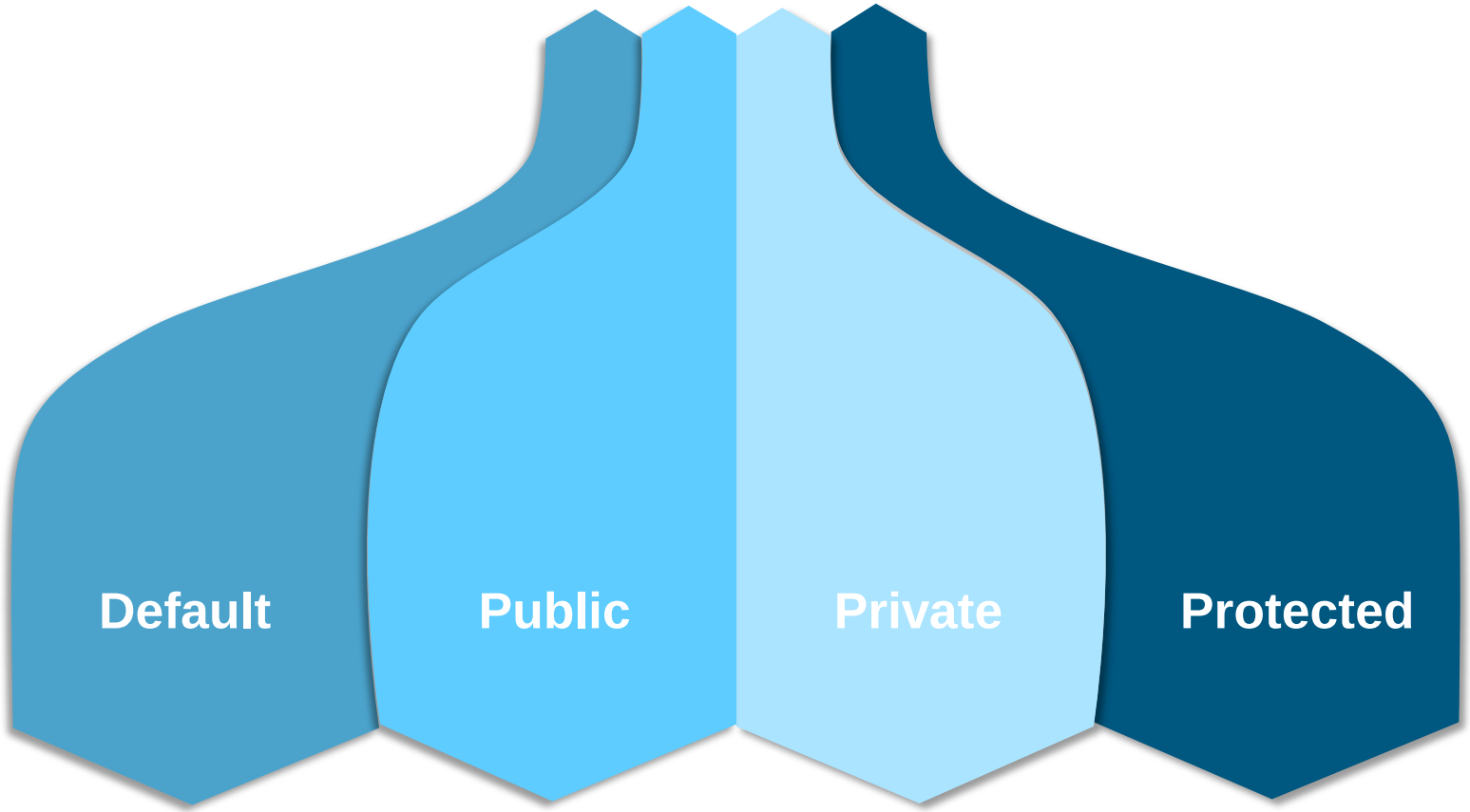
Package and Import

- A package can have many classes and each class can have many methods.
- These methods can be used by another class in another package by using the keyword “keyword”.
- Syntax is: import <package name>.<class name>
- or import <package name>.*; ->This loads all the classes in the given package.
- We can also import static members .For eg:PI,cos etc.

```
import static java.lang.Math.PI;
import static java.lang.Math.*;
```

Access Modifiers

Access modifiers helps to restrict the scope of a class, constructor , variable , method or data member



Access Modifier

Access Modifier specifies the scope/accessibility of a variable or a method or a class from the same class or from a different class or from a different package.

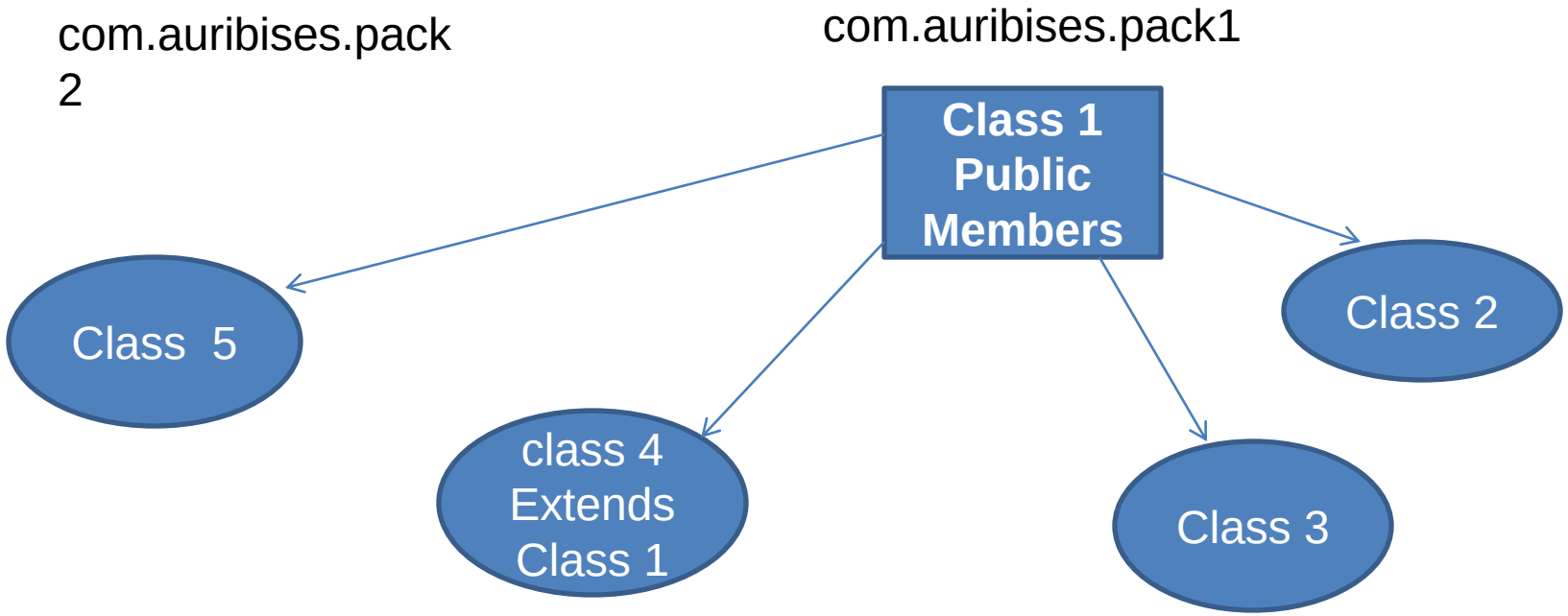
Use of Access Modifier:

- Data abstraction/hiding is one of the concept of object Oriented Programming.
- This means, client will not know the implementation details.
- This can be achieved through Access Modifier.

For Example: If an attribute is made private then it can be accessed only in the class which defined it.

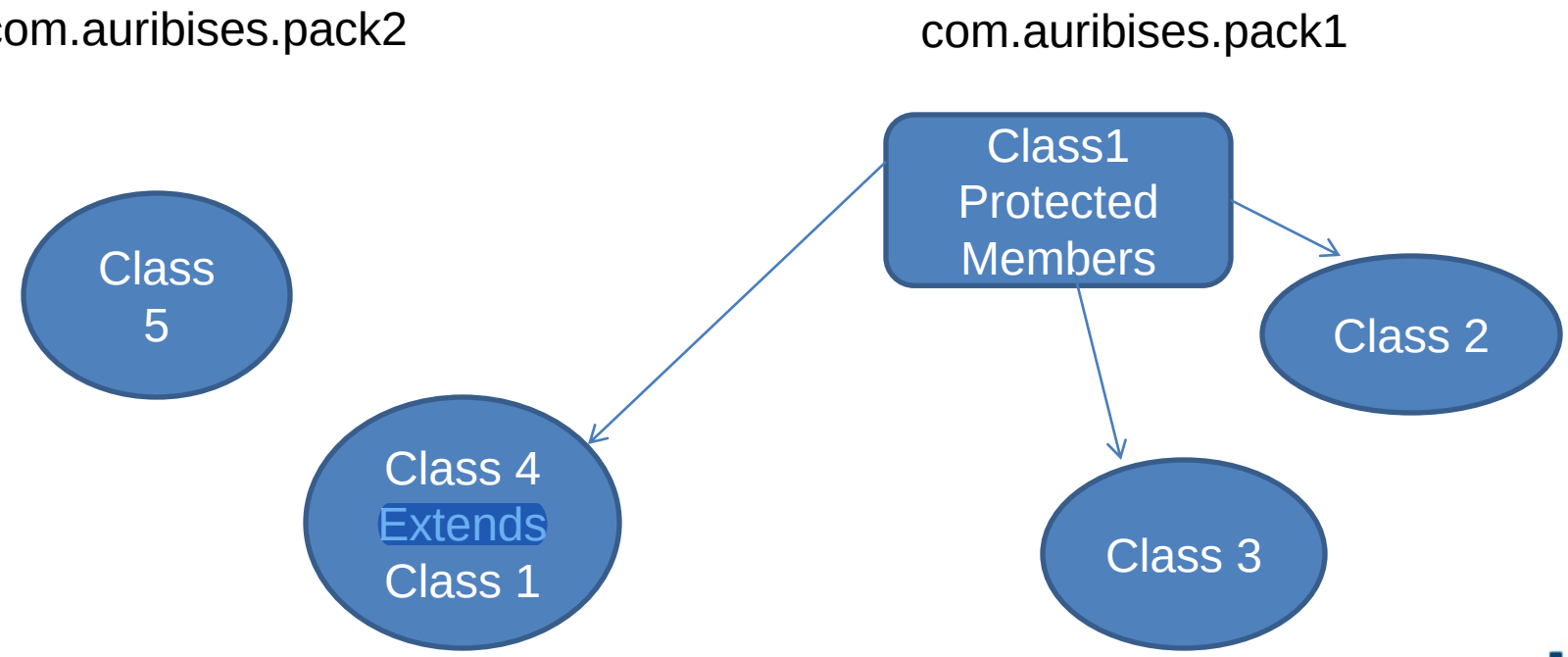
Access Modifier – Public

Public: When an attribute of method is declared as public then it can be accessed anywhere . Any package, any class accessibility is available.



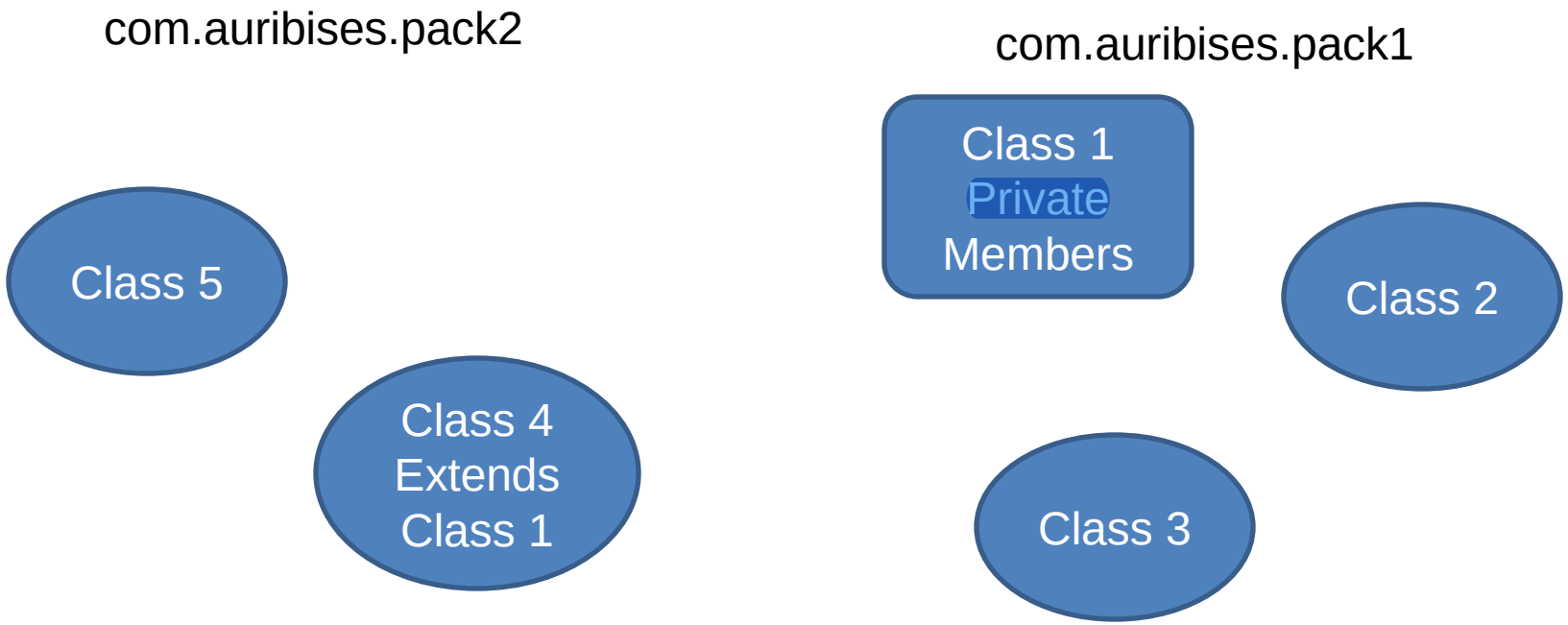
Access Modifier-Protected

Protected: When an attribute of a method is declared as protected then it is visible to all the classes in the same package and all subclasses in different package.



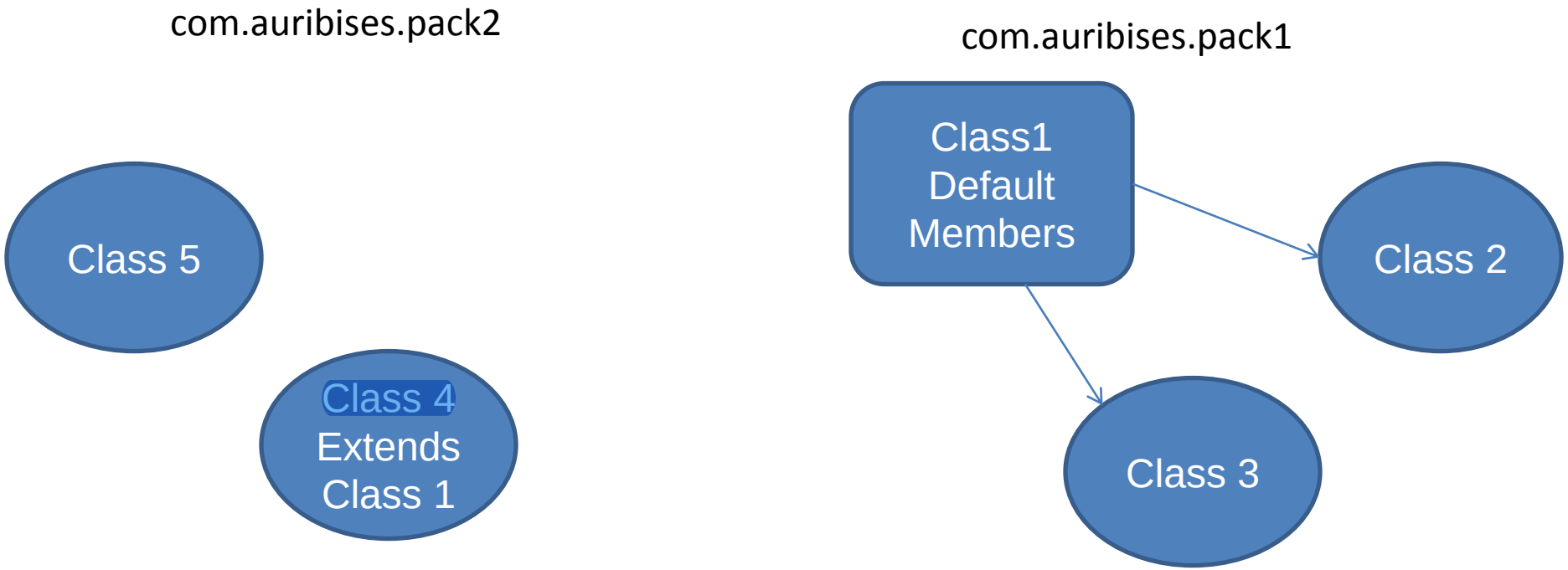
Access Modifier- Private

Private: If a method , variable or constructor is defined as private then it can only be accessed within the declared class itself. Access is not available outside the class.



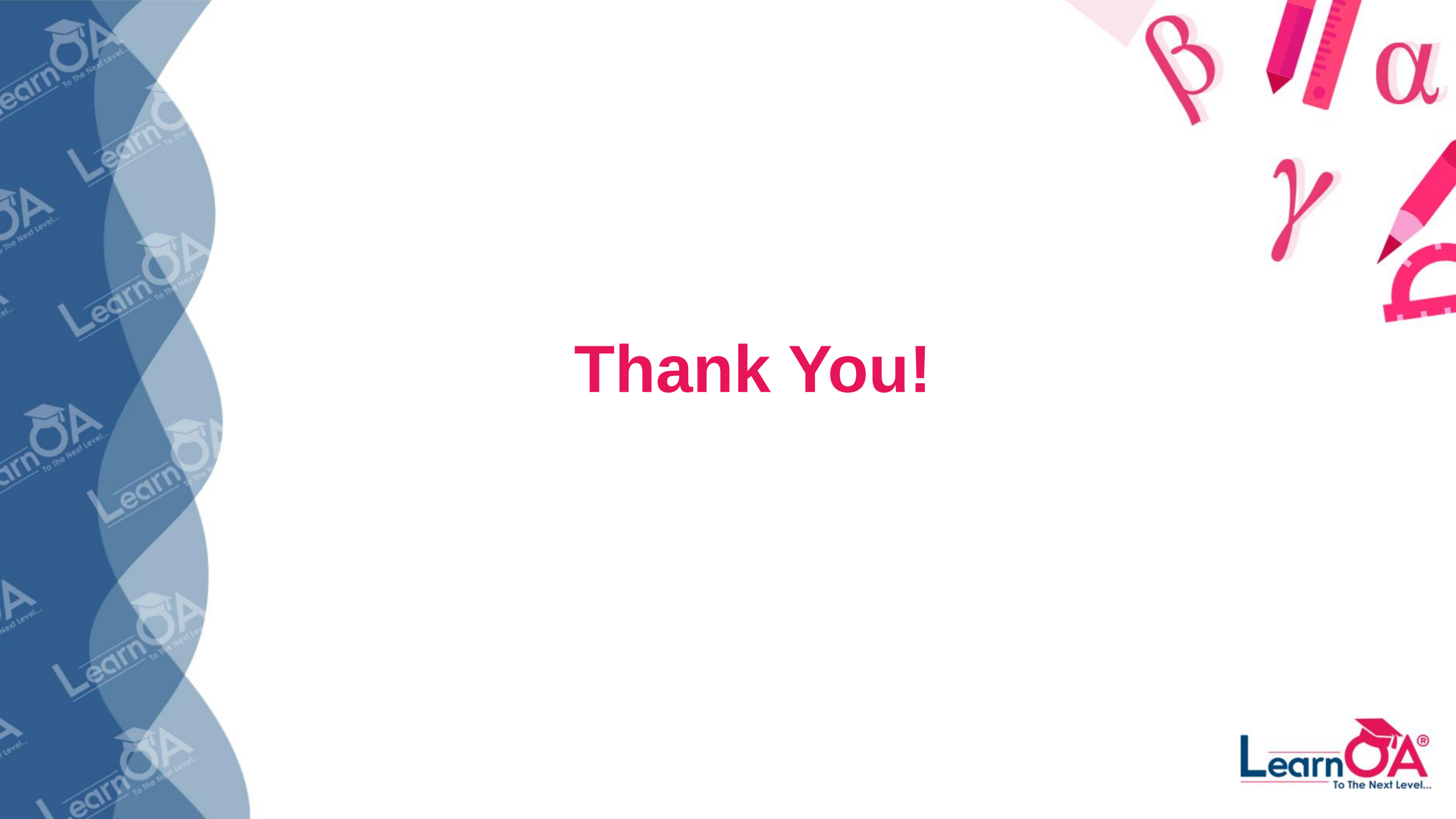
Access Modifier- Default

Default : When no access modifier is defined then it is to have default access modifier. This attribute/method is used only in the given package . It is not accessible outside the package.



Access Modifiers

	Default	Public	Private	Protected
Same class	Yes	Yes	Yes	Yes
Same Package subclass	Yes	No	Yes	Yes
Same Package non-subclass	Yes	No	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes


Thank You!