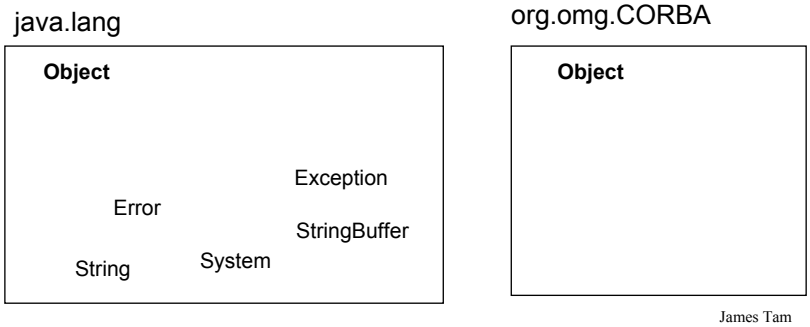
Java Packages

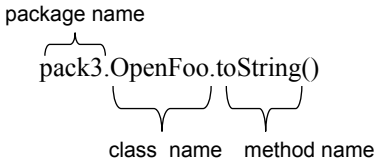
Packages, a method of subdividing a Java program and grouping classes

Packages

- A collection of related classes that are bundled together
- Used to avoid naming conflicts for classes
- Also it allows for only some implementation details to be exposed to other classes in the package (only some classes can be instantiated outside of the package)



Fully Qualified Names



Importing Packages

Importing all classes from a package

Format

import <*package name*>.*;

Example

import java.util.*;

Importing a single class from a package

Format

import <package name>.<class name>;

Example

import java.util.Vector;

Importing Packages (2)

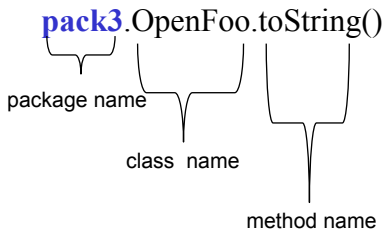
When you do not need an import statement:

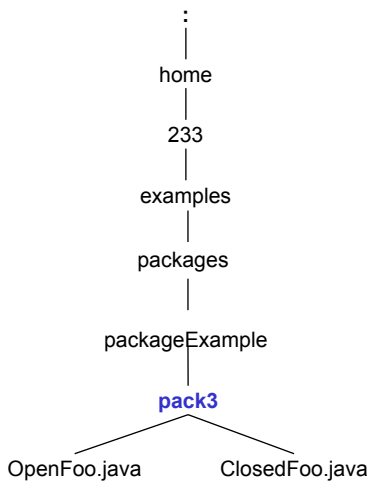
- When you are using the classes in the java.lang package.
- You do not need an import statement in order to use classes which are part of the same package

Default Package

- If you do not use a package statement then the class implicitly becomes part of a default package
- All classes which reside in the same directory are part of the default package for that program.

Fully Qualified Names: Matches Directory Structure





Where To Match Classes To Packages

1. In directory structure: The classes that belong to a package must reside in the directory with the same name as the package (previous slide).
2. In the classes' source code: At the top class definition you must indicate the package that the class belongs to.

Format:

package <*package name*>;
<*visibility – public or package*> class <*class name*>
{

}

Matching Classes To Packages (2)

Example

```
package pack3;
public class OpenFoo
{
    :
}

package pack3;
    class ClosedFoo
{
    :
}
```


Matching Classes To Packages (2)

Example

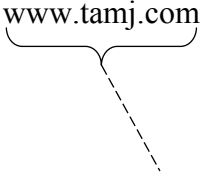
```
package pack3;
public class OpenFoo
{
    :
}

package pack3;
    class ClosedFoo
{
    :
}
```

Public access: Class can be instantiated by classes that aren't a part of package pack3

Package access (default): Class can only be instantiated by classes that are a part of package pack3

Sun's Naming Conventions For Packages

Based on Internet domains (registered web addresses)
e.g., www.tamj.com



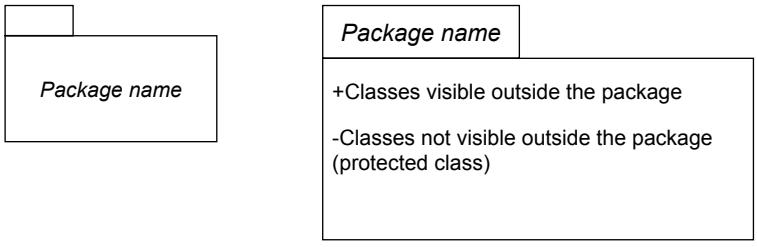
com.tamj .games
.productivity

Sun's Naming Conventions For Packages

Alternatively it could be based on your email address
e.g., tamj@cpsc.ucalgary.ca

ca.ucalgary.cpsc.tamj .games
.productivity

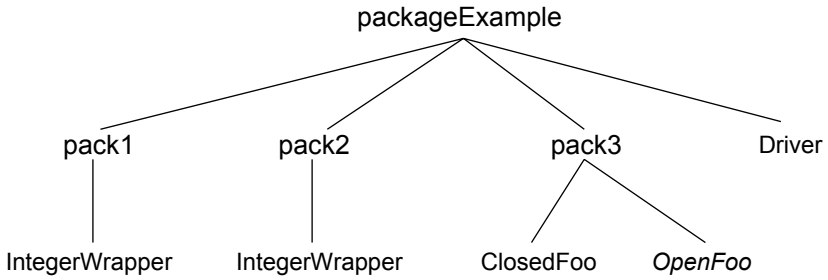
Graphically Representing Packages In UML



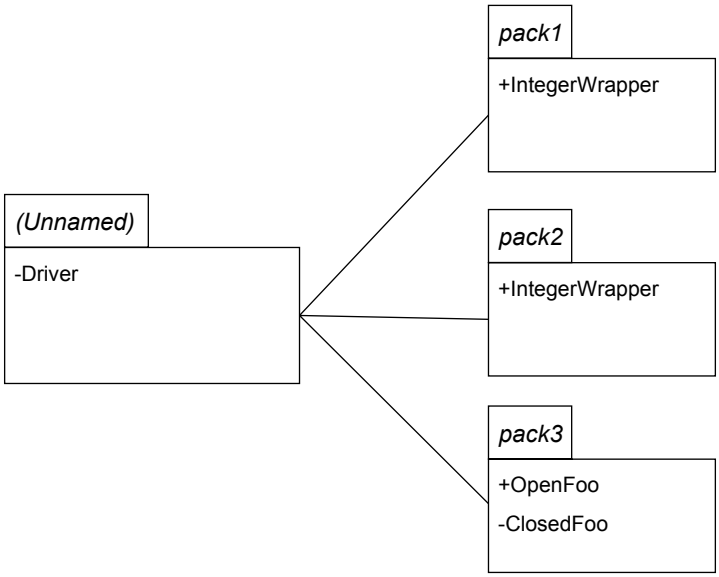
Packages An Example

The complete example can be found in the directory:
/home/233/examples/packages/packageExample

(But you should have guessed the path from the package name)



Graphical Representation Of The Example



Package Example: The Driver Class

```
import pack3.*;
class Driver
{
   public static void main (String [] argv)
   {
      pack1.IntegerWrapper iw1 = new pack1.IntegerWrapper ();
      pack2.IntegerWrapper iw2 = new pack2.IntegerWrapper ();
      System.out.println(iw1);
      System.out.println(iw2);

      OpenFoo of = new OpenFoo ();
      System.out.println(of);
      of.manipulateFoo();
   }
}
```

Package Example: Package Pack1, Class IntegerWrapper

```
package pack1;
public class IntegerWrapper
{
   private int num;

   public IntegerWrapper ()
   {
      num = (int) (Math.random() * 10);
   }
   public IntegerWrapper (int newValue)
   {
      num = newValue;
   }
   public void setNum (int newValue)
   {
      num = newValue;
   }
```


Package Example: Package Pack1, Class IntegerWrapper (2)

```
   public int getNum ()
   {
      return num;
   }

   public String toString ()
   {
      String s = new String ();
      s = s + num;
      return s;
   }
}
```

Package Example: Package Pack2, Class IntegerWrapper

```
package pack2;
public class IntegerWrapper
{
   private int num;

   public IntegerWrapper ()
   {
      num = (int) (Math.random() * 100);
   }
   public IntegerWrapper (int newValue)
   {
      num = newValue;
   }
   public void setNum (int newValue)
   {
      num = newValue;
   }
```


Package Example: Package Pack2, Class IntegerWrapper (2)

```
    public int getNum ()
   {
      return num;
   }

   public String toString ()
   {
      String s = new String ();
      s = s + num;
      return s;
   }
}
```

Package Example: Package Pack3, Class OpenFoo

```
package pack3;
public class OpenFoo
{
   private boolean bool;
   public OpenFoo () { bool = true; }
   public void manipulateFoo ()
   {
      ClosedFoo cf = new ClosedFoo ();
      System.out.println(cf);
   }
   public boolean getBool () { return bool; }
   public void setBool (boolean newValue) { bool = newValue; }
   public String toString ()
   {
      String s = new String ();
      s = s + bool;
      return s;
   }
}
```


Package Example: Package Pack3, Class ClosedFoo

```
package pack3;
class ClosedFoo
{
   private boolean bool;

   public ClosedFoo () { bool = false; }
   public boolean getBool () { return bool; }

   public void setBool (boolean newValue) { bool = newValue; }

   public String toString ()
   {
      String s = new String ();
      s = s + bool;
      return s;
   }
}
```

Updated Levels Of Access Permissions: Attributes And Methods

Private "-"

- Can only access the attribute/method in the methods of the class where it's originally defined.

Protected "#"

- Can access the attribute/method in the methods of the class where it's originally defined or the subclasses of that class.

Package - no UML symbol for this permission level

- Can access the attribute/method from the methods of the classes within the same package
- *If the level of access is unspecified in a class definition this is the default level of access*

Public "+"

- Can access attribute/method anywhere in the program

James Tam

Updated Levels Of Access Permissions

Access level	Accessible to			
	Same class	Class in same package	Subclass in a different package	Not a subclass, different package
Public	Yes	Yes	Yes	Yes
Protected	Yes	Yes	Yes	No
Package	Yes	Yes	No	No
Private	Yes	No	No	No

James Tam

You Should Now Know

- How packages work in Java
  - How to utilize the code in pre-defined packages
  - How to create your own packages
- How the 4 levels of access permission work