Lab 01: Java from Command Line; Packages

CS 0445: Data Structures

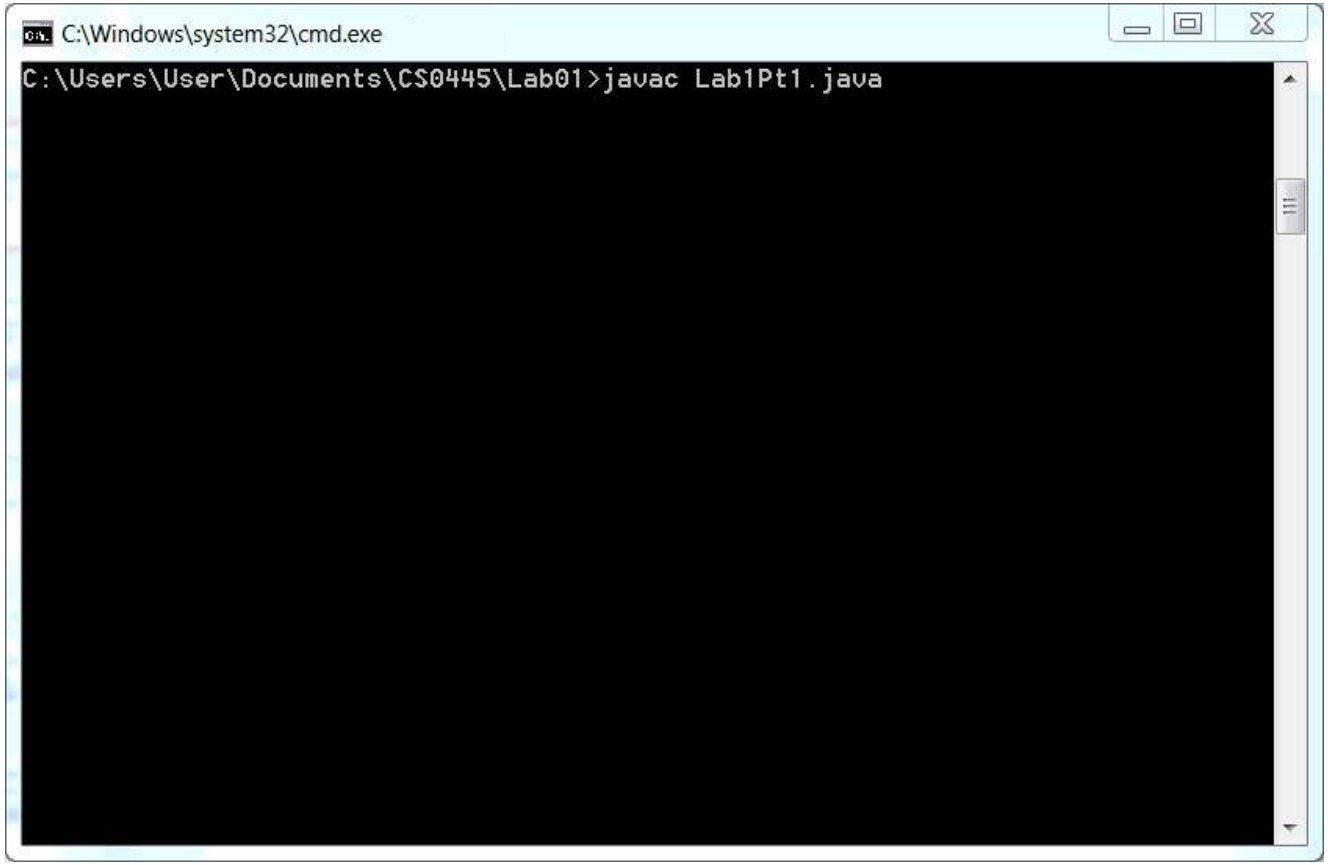
TAs: Jon Rutkauskas
Brian Nixon

http://db.cs.pitt.edu/courses/cs0445/current.term/

Sep 9, 2019
University of Pittsburgh, Pittsburgh, PA

Java From Command Line

- We'll be compiling and running our Java programs from the command line in this lab



Opening Command Line

- Opening
  - Windows
    - Windows key or Winkey + R, type “cmd”, press Enter
      - (Or you can use Powershell)
    - Or from File Explorer, Shift+Left Click to open in current directory
  - MacOS
    - CMD+space, type “terminal”, press Enter

Using the Command Line

- “cd” to **c**hange **d**irectory
  - “cd cs445” will go into the cs445 folder if it exists
  - “cd ..” will go up one directory
  - You can do multiple changes at once
    - For example “cd cs445/lab1” or “cd ../..”
- Lost?
  - Print current directory
    - MacOS and Linux use “pwd” (**p**rint **w**orking **d**irectory)
    - Windows just use “cd” with no arguments
  - Display contents of current directory
    - MacOS and Linux use “ls”, “ls -al”
    - Windows use “dir”

Compiling and running

- “javac” takes a source file (.java) and compiles it into a class file (.class)
  - javac Example.java
- “java” takes a class name
  - Put any *command line arguments* after the class name
    - java Example “hello world!”

Compiling and running

- "javac" takes a source file (.java) and compiles it into a class file (.class)
  - javac Example.java
- "java" takes a class name
  - Put any *command line arguments* after the class name
    - java Example "hello world!"

```
public static void main(String[] args) {
      System.out.println(args[0]);
}
```

Package Organization

Workspace (e.g. Desktop, Pitt folder, etc)

cs445

lab1

Lab1.java

This file starts with...

```
package cs445.lab1;
```

Package Organization

Workspace (e.g. Desktop, Pitt folder, etc)

cs445

lab1

Lab1.java

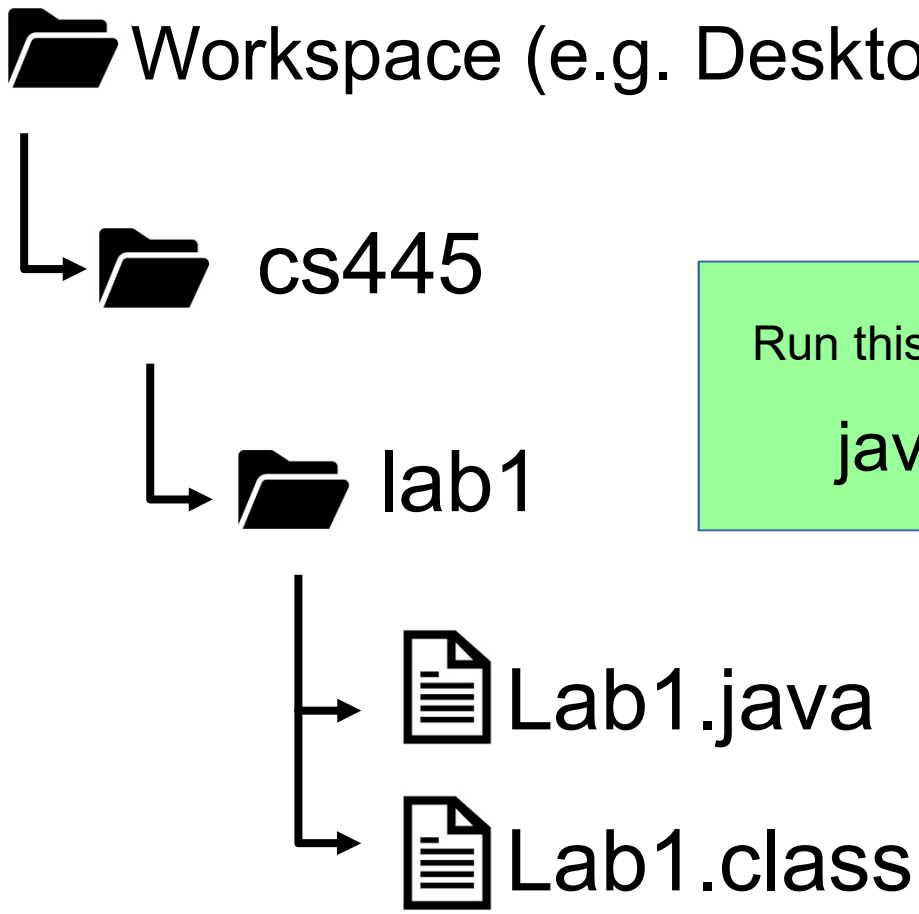
Compile from your **Workspace** directory!

If you're currently in the **package** directory (cs445/lab1) move back up to *outside* the cs445 folder (cd ../..)

javac cs445/lab1/Lab1.java

NOT javac Lab1.java

Package Organization



Run this from your Workspace directory!

```
java cs445.lab1.Lab1
```

We use dots here and no file extension

Package Organization

Workspace (e.g. Desktop, Pitt folder, etc)

- cs445
  - lab1
    - Lab1.java
    - Lab1.class

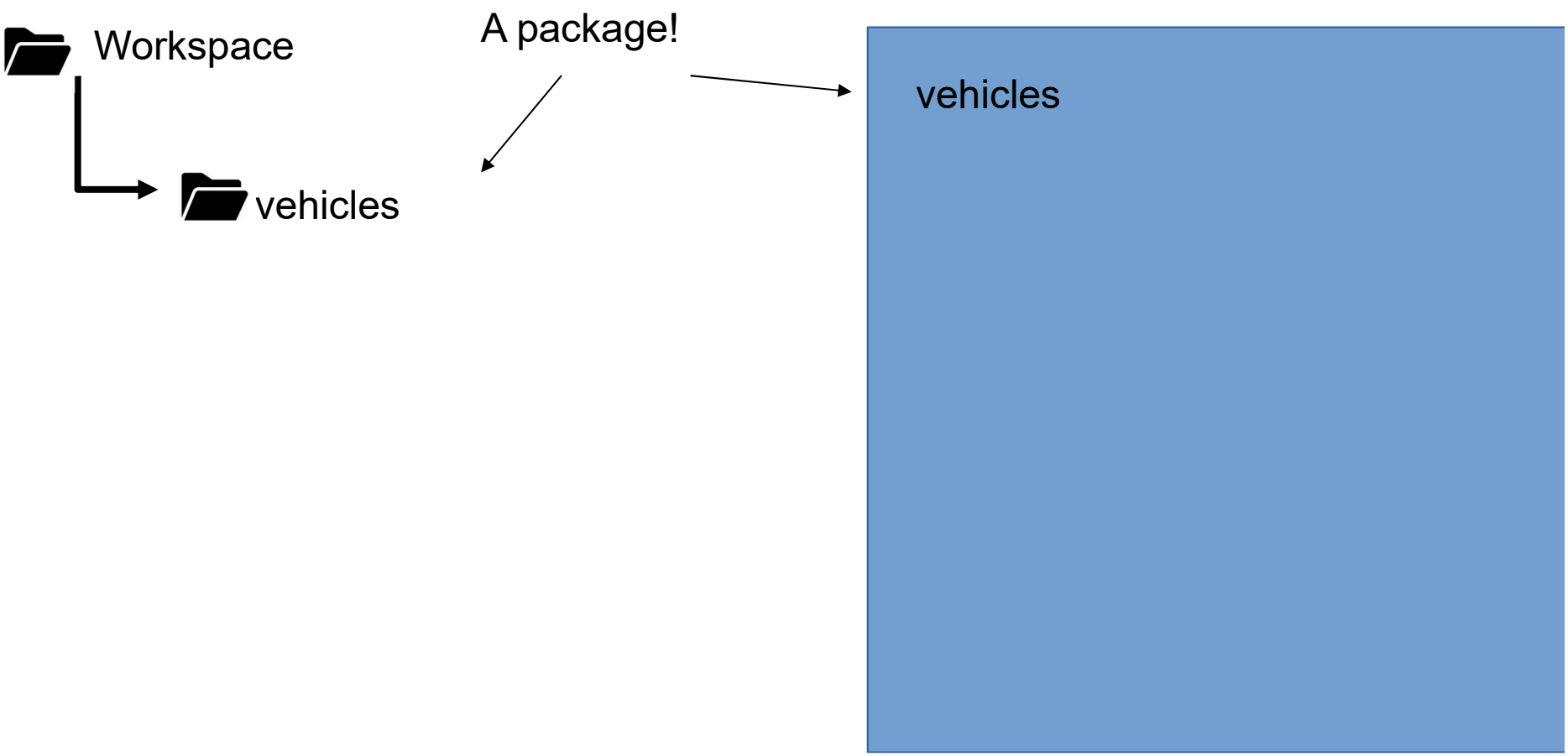
Run this from your Workspace directory!

```
java cs445.lab1.Lab1
```

This tells java to expect The Lab1 class in the cs445.lab1 package!

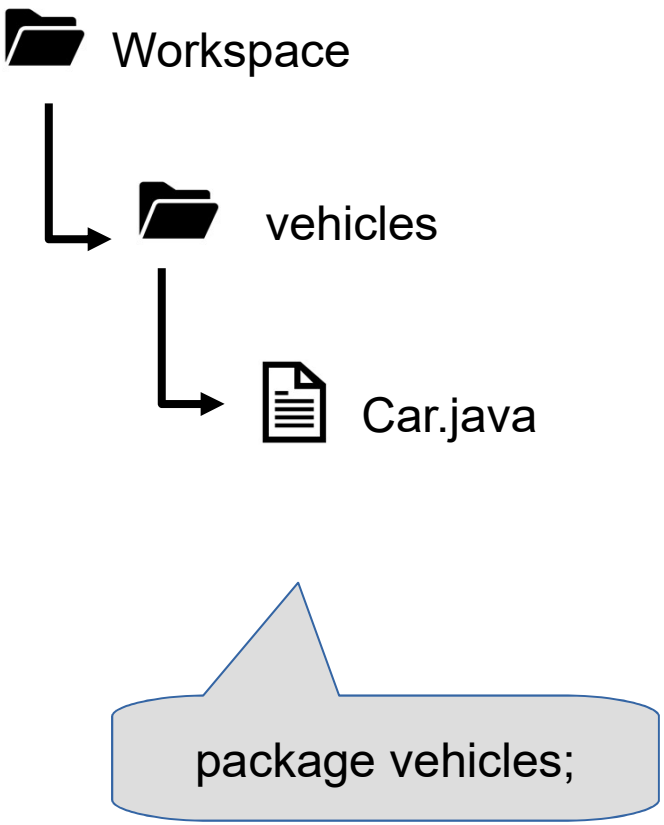
Why use Packages?

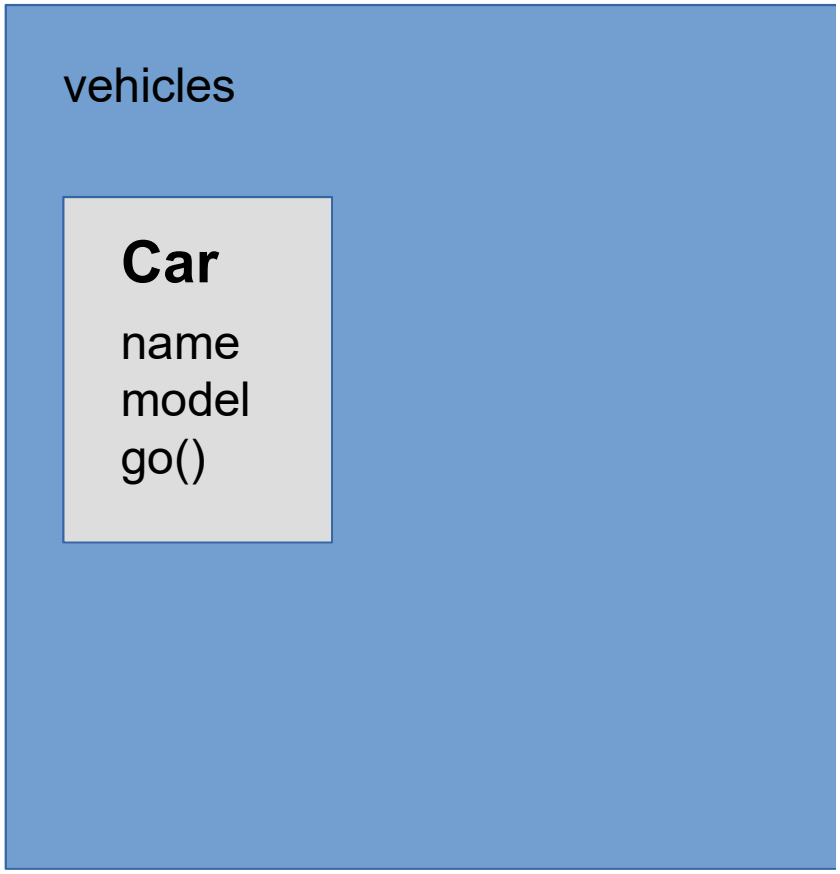
- Easy, intuitive grouping of multiple classes



Why use Packages?

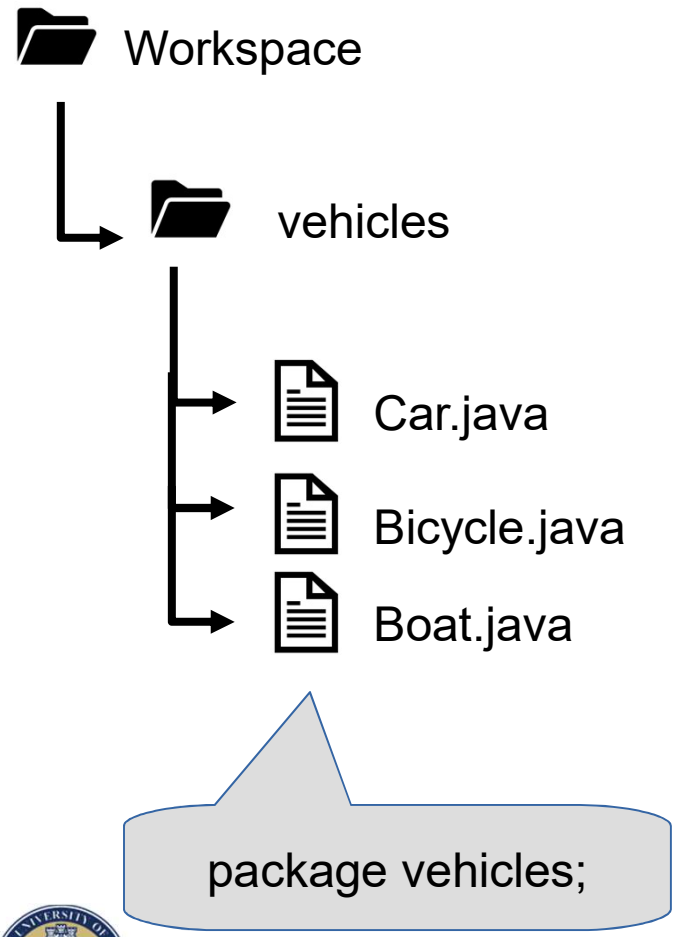
- Easy, intuitive grouping of multiple classes

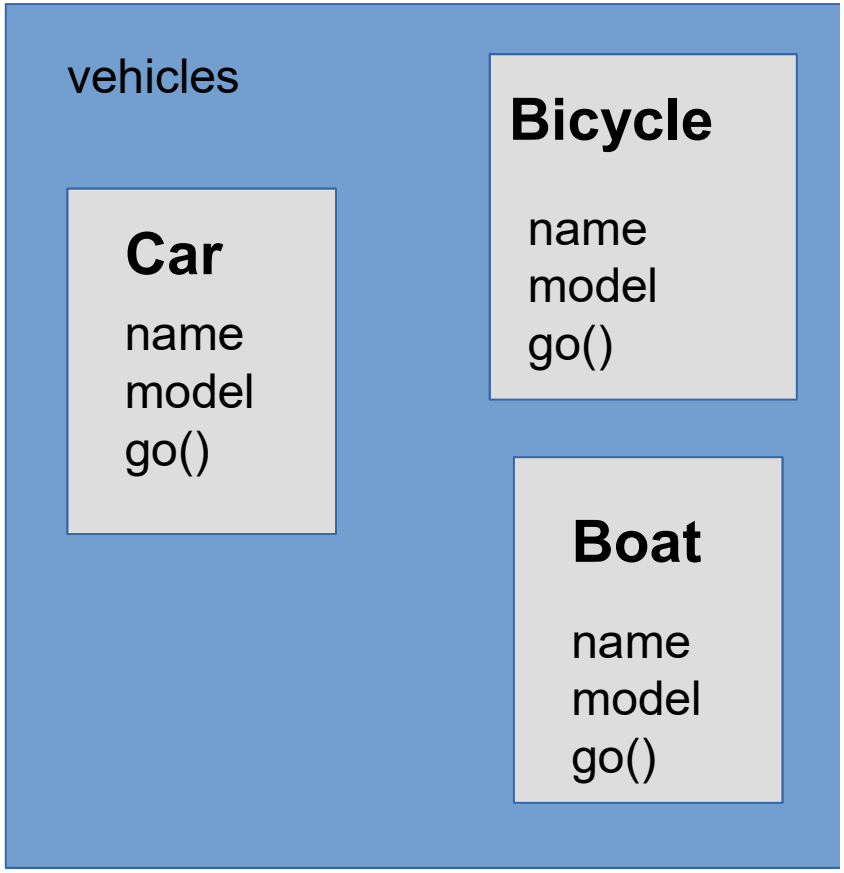




Why use Packages?

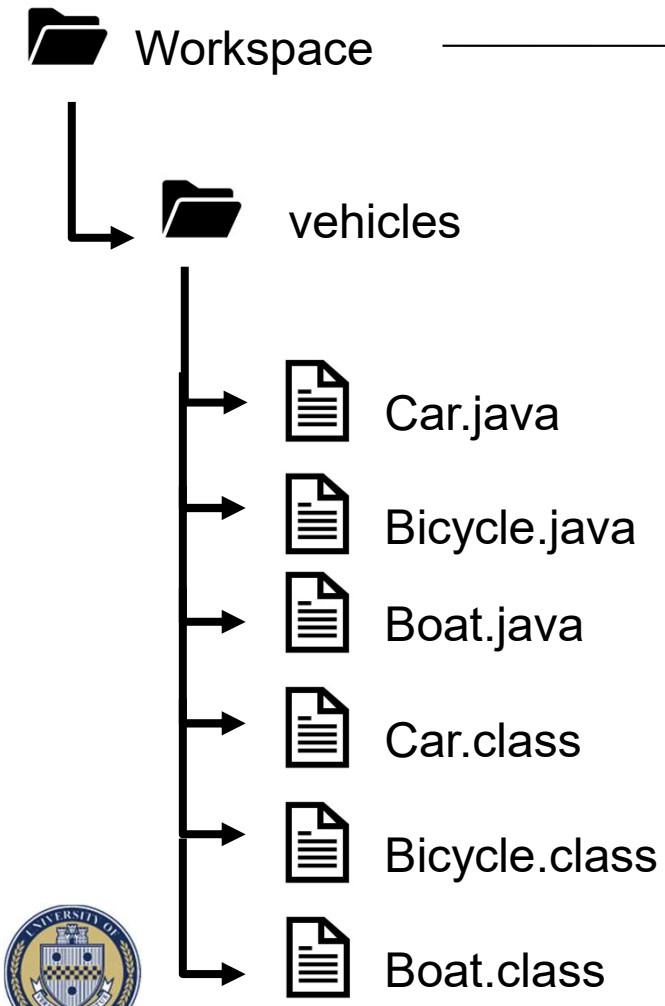
- Easy, intuitive grouping of multiple classes





Why use Packages?

- Compile and import easily



javac vehicles/*.java

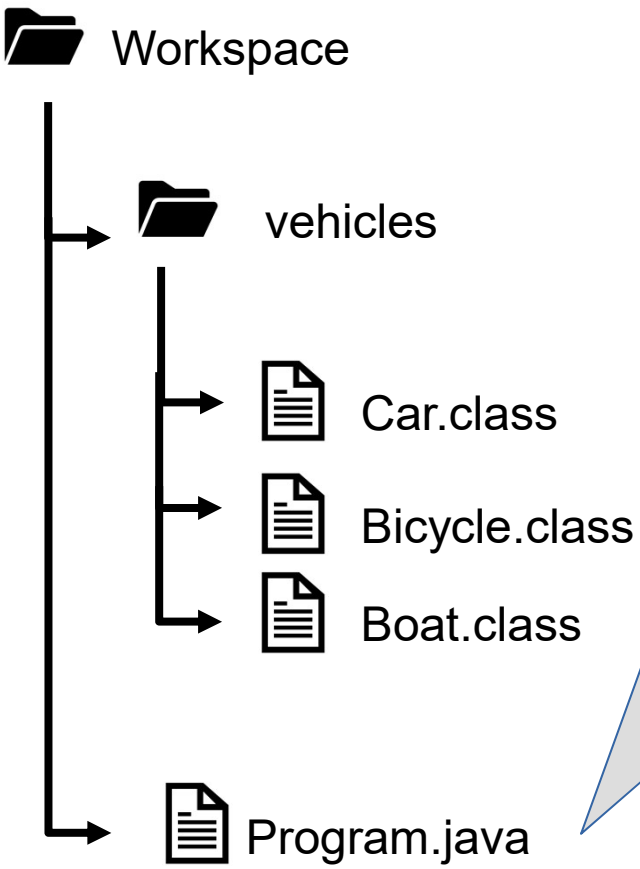
Or if you don't need everything, you can just compile one

javac vehicles/Car.java

javac will automatically compile anything else that file depends on

Why use Packages?

- Compile and import easily



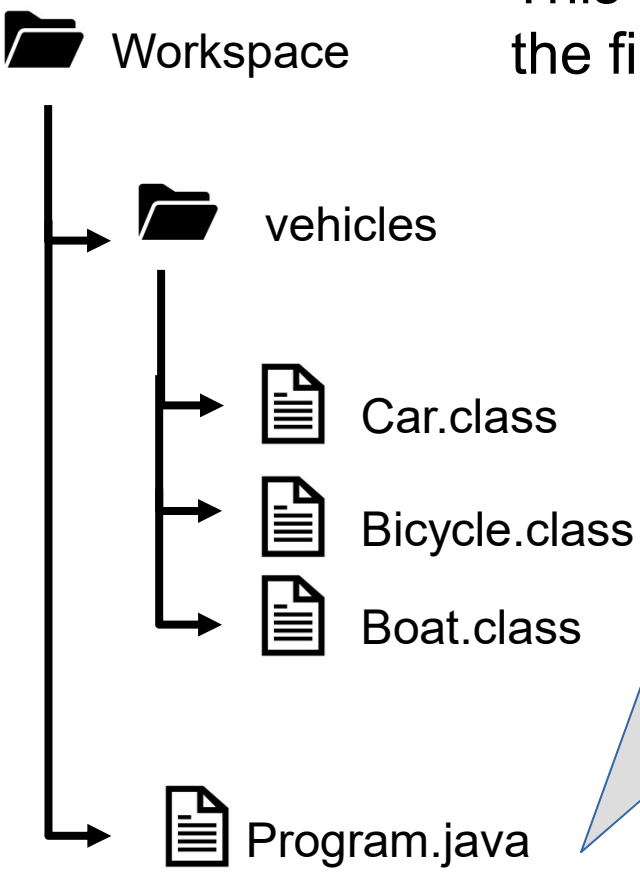

```
import vehicles.*;

public class Program {
        public static void main(String[] args) {
                Car myCar = new Car();
                Bicycle myBike = new Bicycle();
                Boat myBoat = new Boat();
        }
}
```

Why use Packages?

You can also import individually or one by one

This is the same notation you used to package the files together!




```
import vehicles.Car;
Import vehicles.Bicycle;
Import vehicles.Boat;

public class Program {
        public static void main(String[] args) {
                Car myCar = new Car();
                Bicycle myBike = new Bicycle();
                Boat myBoat = new Boat();
        }
}
```

Why use Packages?

Security:

- Having classes in the same package allows them to access each other's data
- (unless they are *private*)

Modifier	Class	Package	Subclass	Everyone else
public	Yes	Yes	Yes	Yes
protected				No
(none)			No	
private		No		

Lab 1

- Lab 1 has three parts:

Part 1 is compiling and running Java programs from the Command Line

Part 2 is using command line arguments

Part 3 is using Java packages

- Follow the instructions in the pdf

- Ask for help if you get stuck!

- We will also fix any JDK or PATH issues during this lab.
  - E.g., "`'javac' is not recognized as an internal or external command…`"