ICS 141
Programming with Objects

Jessica Maistrovich

Metropolitan State University

Talk to the computer

What does it understand?

0’s and 1’s

- What can be represented with a sequence of 0’s and 1’s? (Keeping in mind that you have a limited amount of space)

Primitive Data Types can be represented with 0’s and 1’s.

Integers

- byte
- short
- int
- long

Decimal Numbers

- float
- double

Letters

- char – ‘ ‘ not “ “

Boolean values

- boolean – true, false

Things?

If you were stranded on a desert island and
could only bring one thing,
what would it be?

What did you bring?

- A number?
- A character?
- A true or false value?

What if you want to represent something that is not a primitive data type?

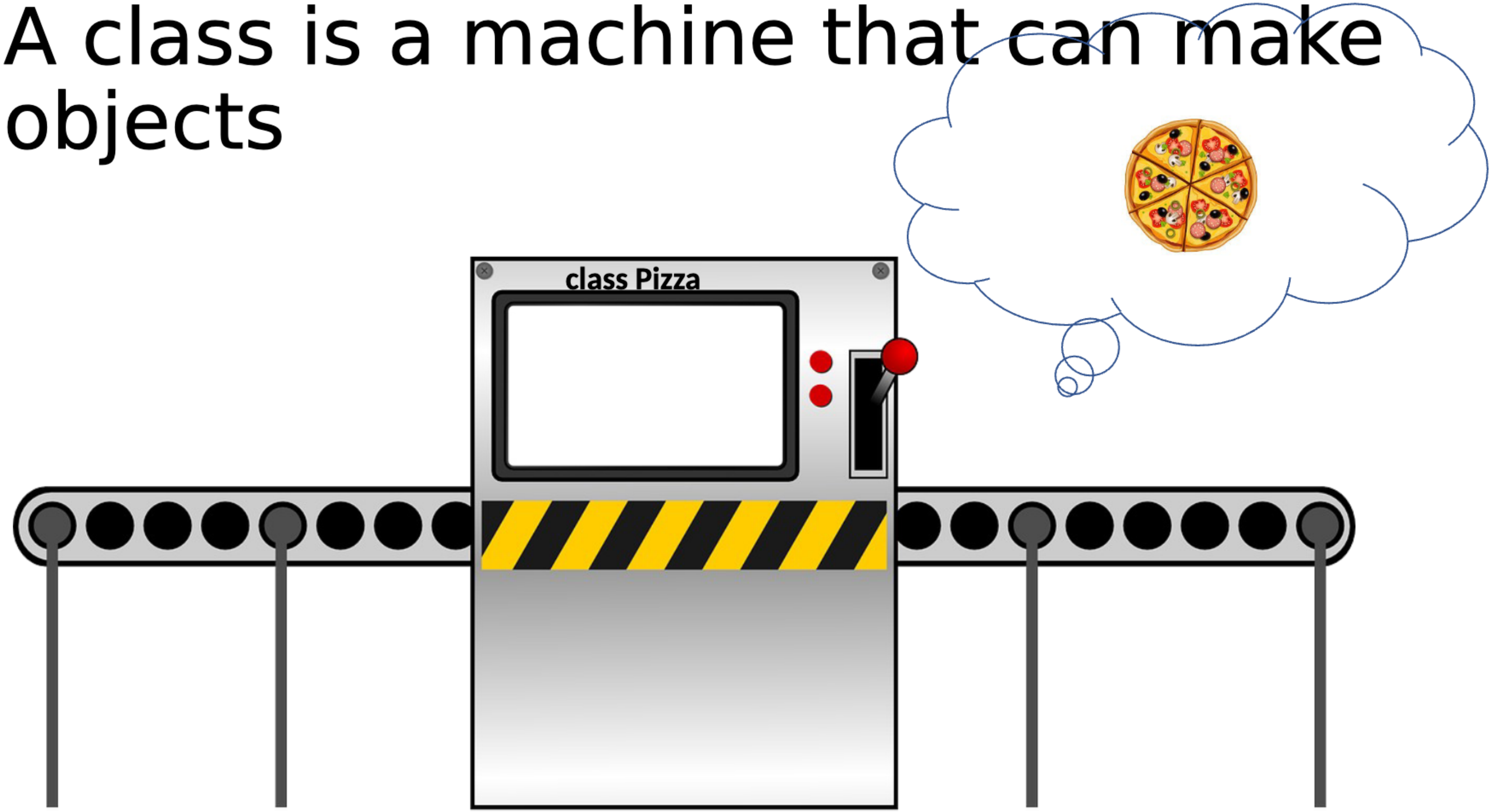
Abstract Data Type

also known as Object

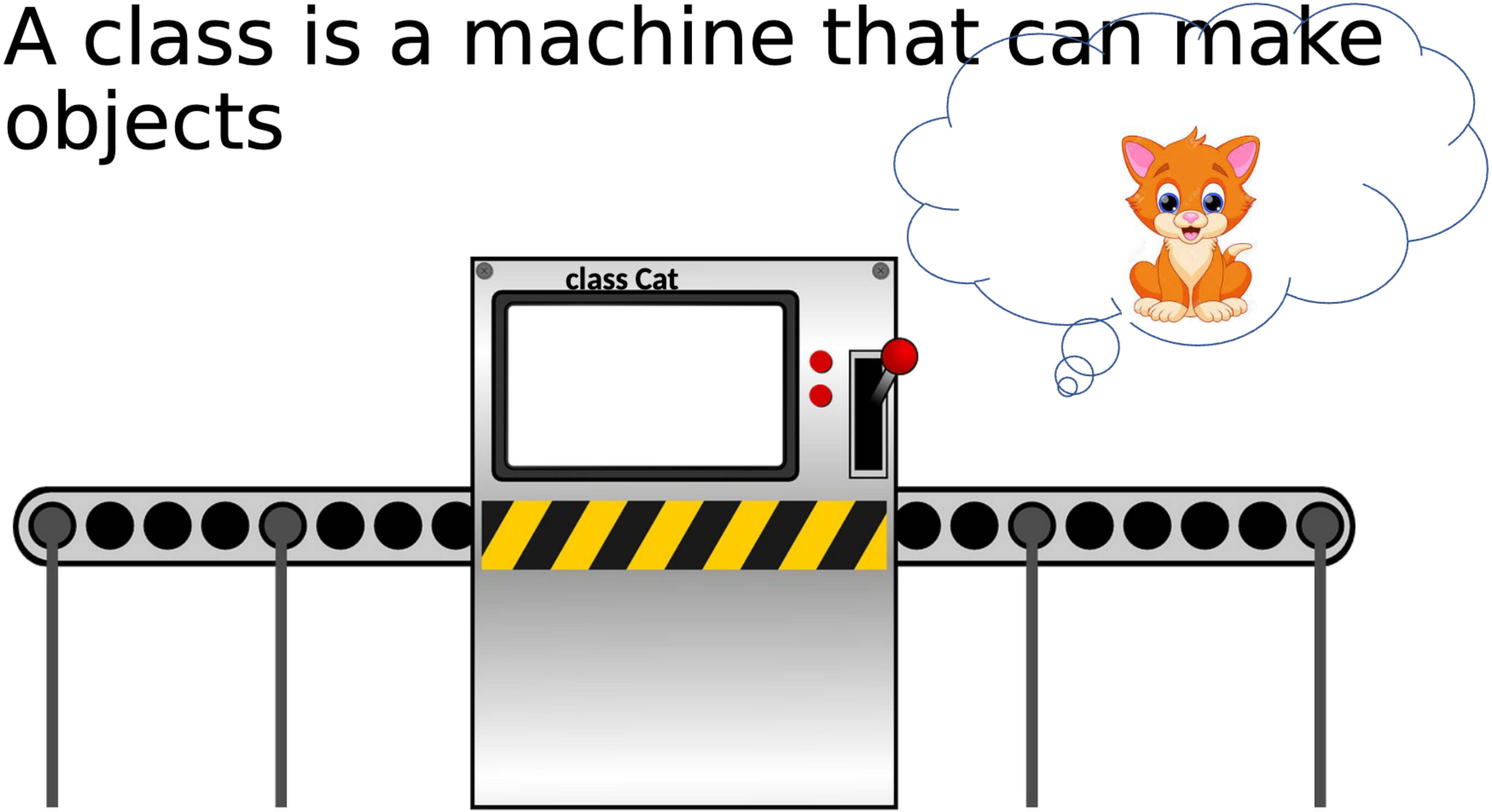
Classes – Part 1

How can we represent ADT’s in the computer?

A class is a machine that can make objects



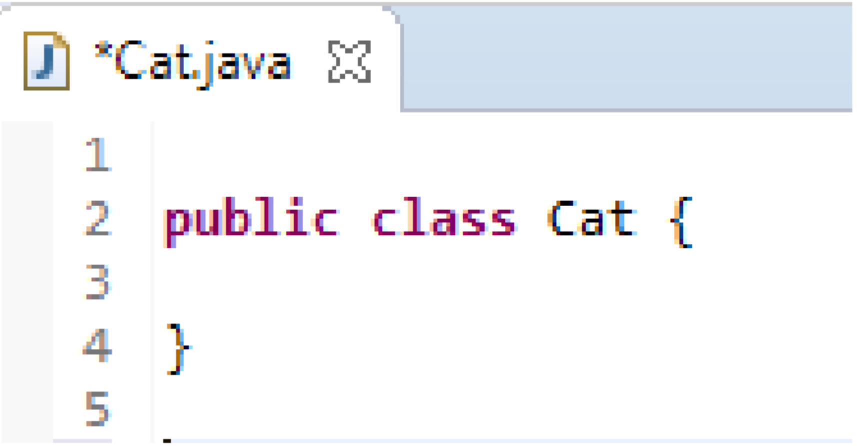
A class is a machine that can make objects



Let’s make a class in Java

```
public class ______________{


}
```

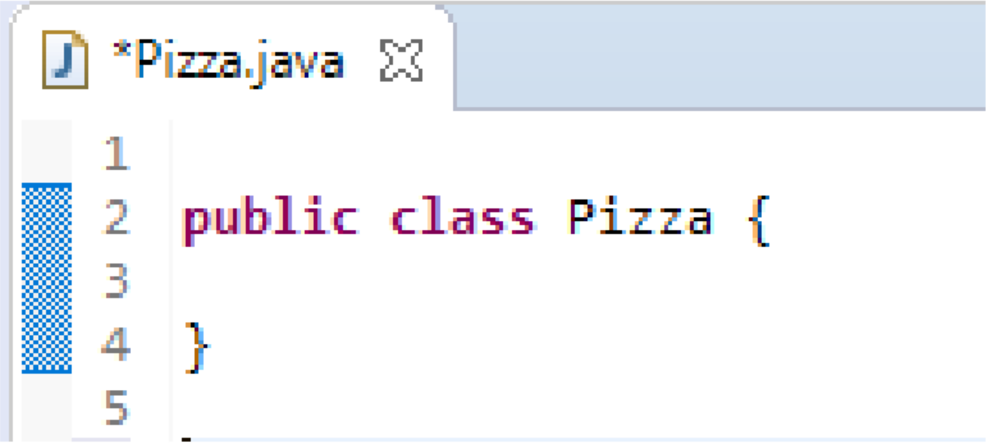


```java

public class Cat {

}

```



```java

public class Pizza {

}

```

Try it!

- Create a project called Practice

- Create a class to represent the following:
  - Taco
  - Student
  - Pool
  - Television

Customization

We can add the ability to customize

Pizza
size number of toppings gluten free or not

Cat
name age lives outside or not

What questions might you ask about a pizza? A cat?

Choose one object and come up with three fields. For now, choose fields that are able to be represented as a letter, number, or true/false (primitive data types). Please format in pseudo-UML.

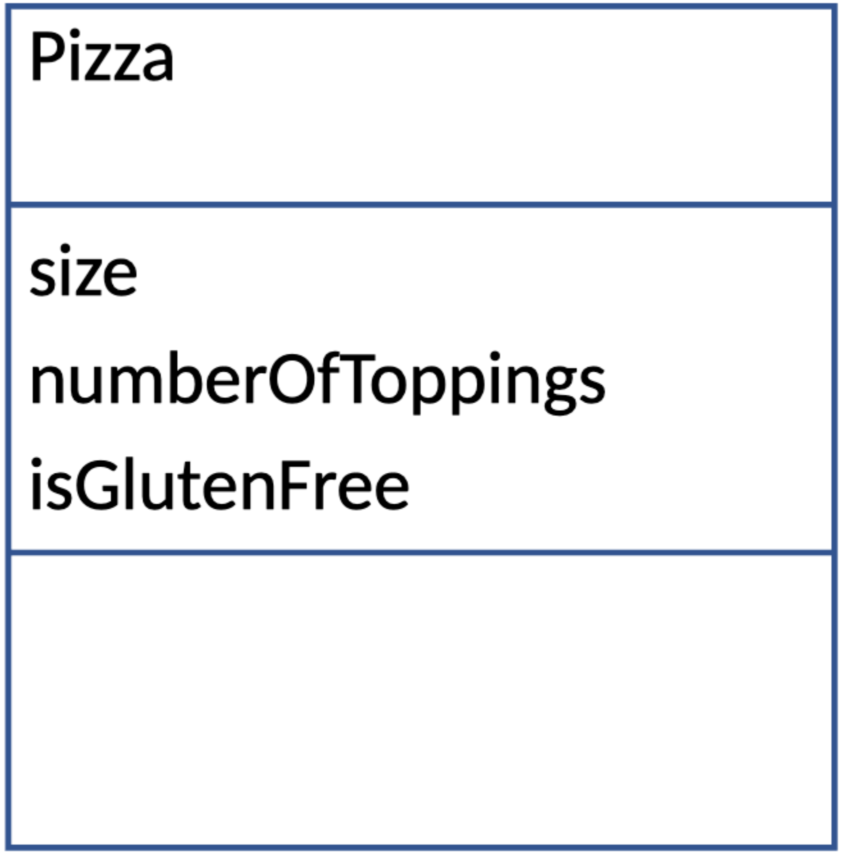
****Must describe an individual thing, not a set. For example: shoe size vs number of shoes

- Shoe size is a description of one pair of shoes
- Number of shoes is a description of a collection of shoes

Those descriptions can be turned into variable names.

- Naming conventions are almost the same as Python
  - Letters and numbers
    - Can use $ and _ (but don’t)
  - Start with letter
  - Use full words (size instead of s)
  - No keywords or reserved words
  - Capitalize constants
- Different:
  - numberOfToppings instead of number_of_toppings
  - Can use _ when multiple word constant NUM_TOPPINGS

Turn into variable names

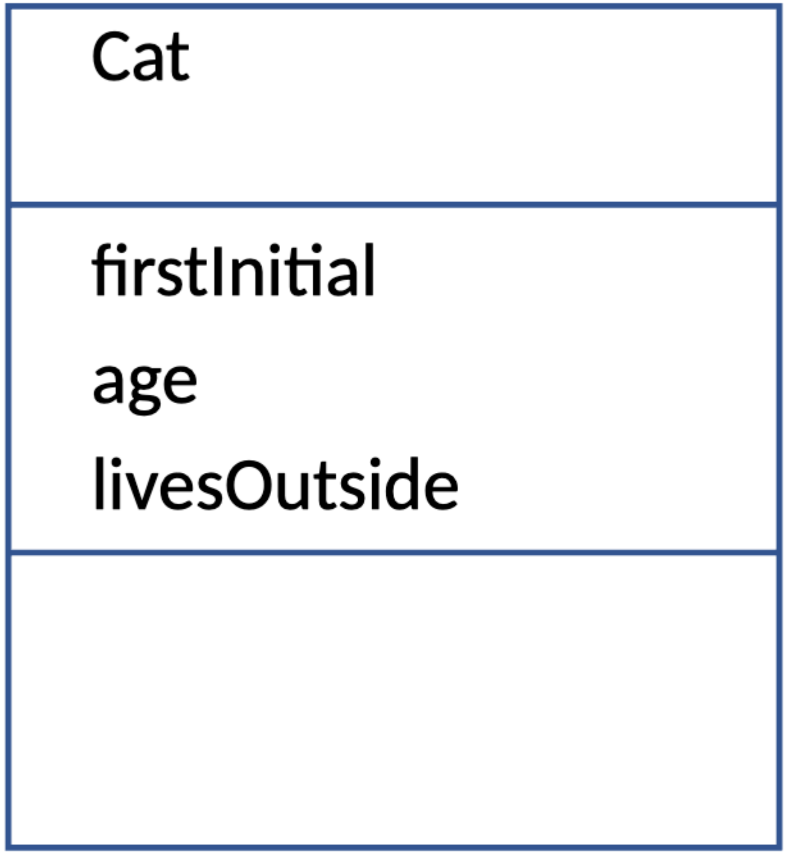


Pizza

size

numberOfToppings

isGlutenFree



Cat

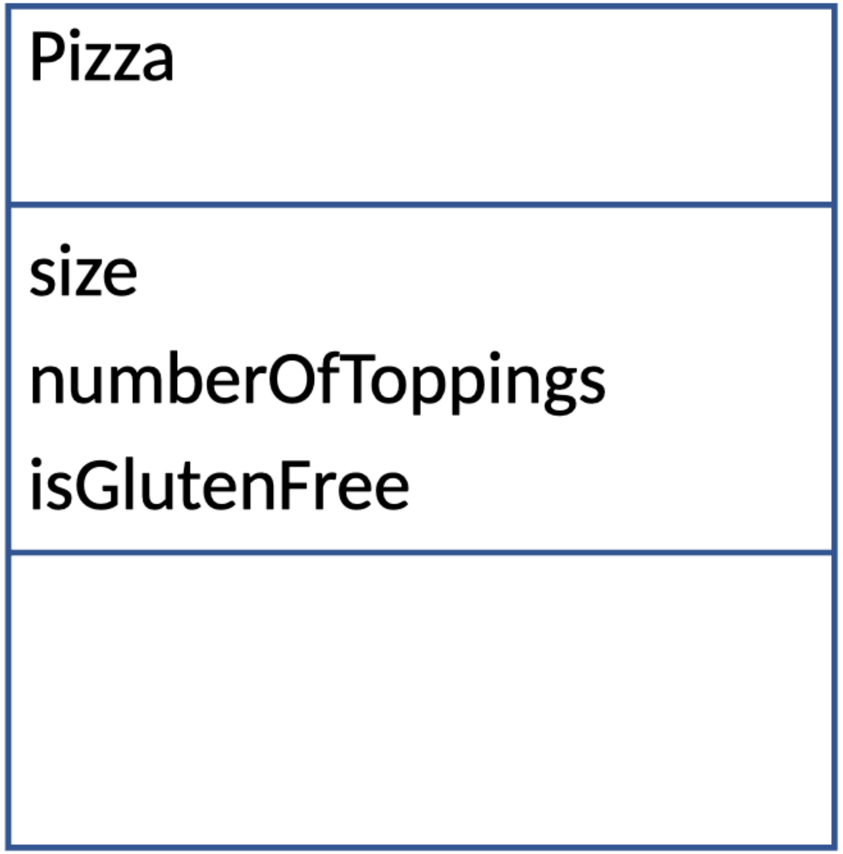
firstInitial

age

livesOutside

* * Officially “Fields” – Specifically Instance Variables

Turn into variable names

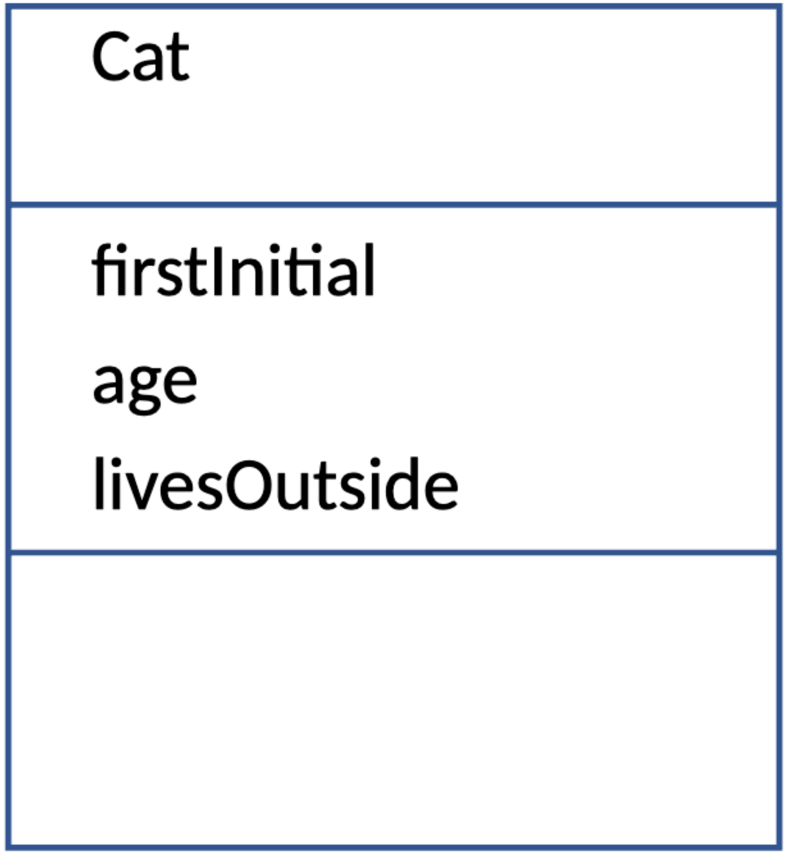


Pizza

size

numberOfToppings

isGlutenFree



Cat

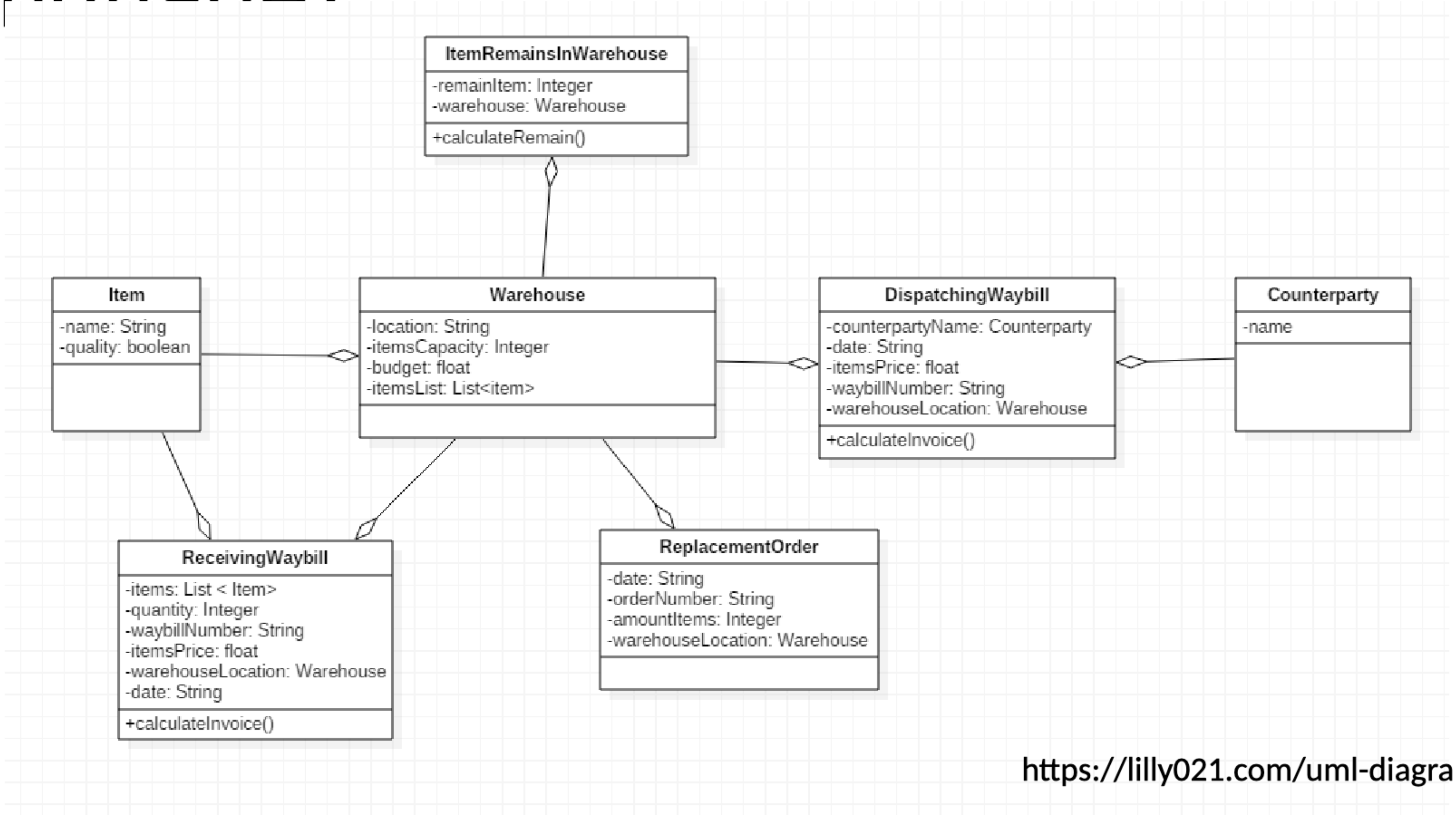
firstInitial

age

livesOutside

Why the weird boxes?

UML Diagram (Unified Modeling Language)



https://lilly021.com/uml-diagrams/

Declaring variables

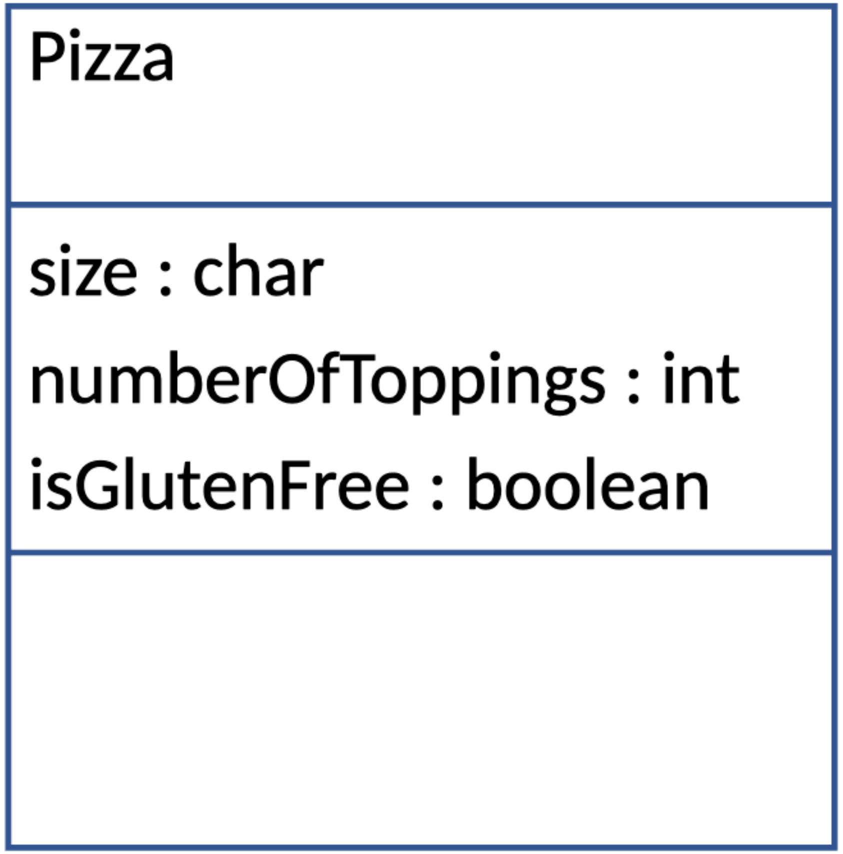
Telling Java what data type the variable is

Java is strongly typed

- Conserve space
- Can’t change – “built to order storage”
- Declaring a variable means telling java what type that variable is

- First time Java sees a variable, it must be declared!!!

Declare the type – UML style

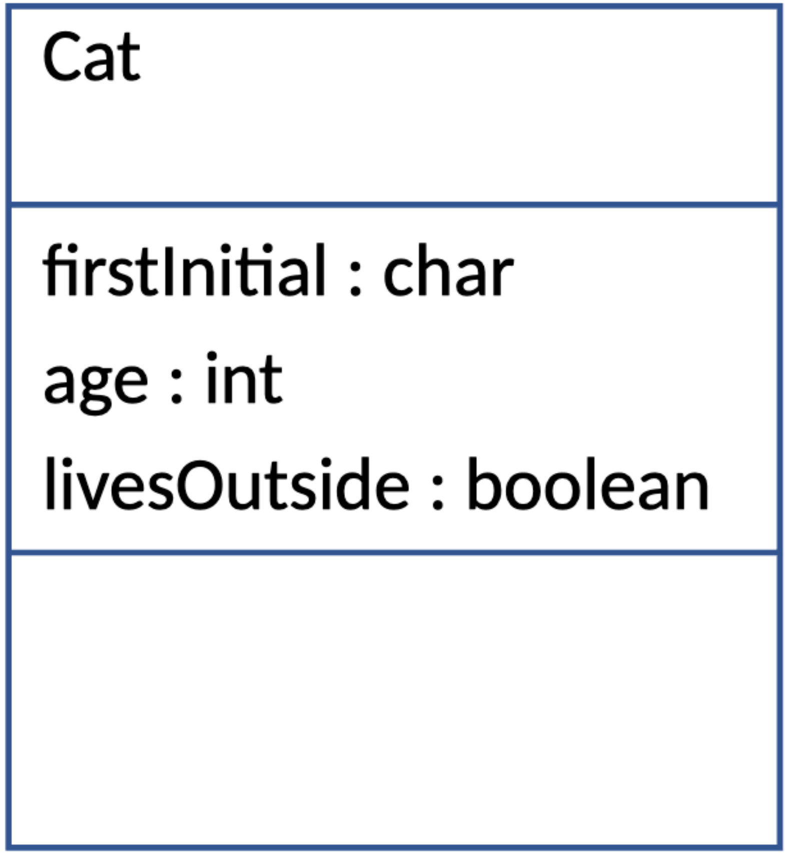


Pizza

size : char

numberOfToppings : int

isGlutenFree : boolean



Cat

firstInitial : char

age : int

livesOutside : boolean

Before we put this in Eclipse

- Remember Java is case sensitive

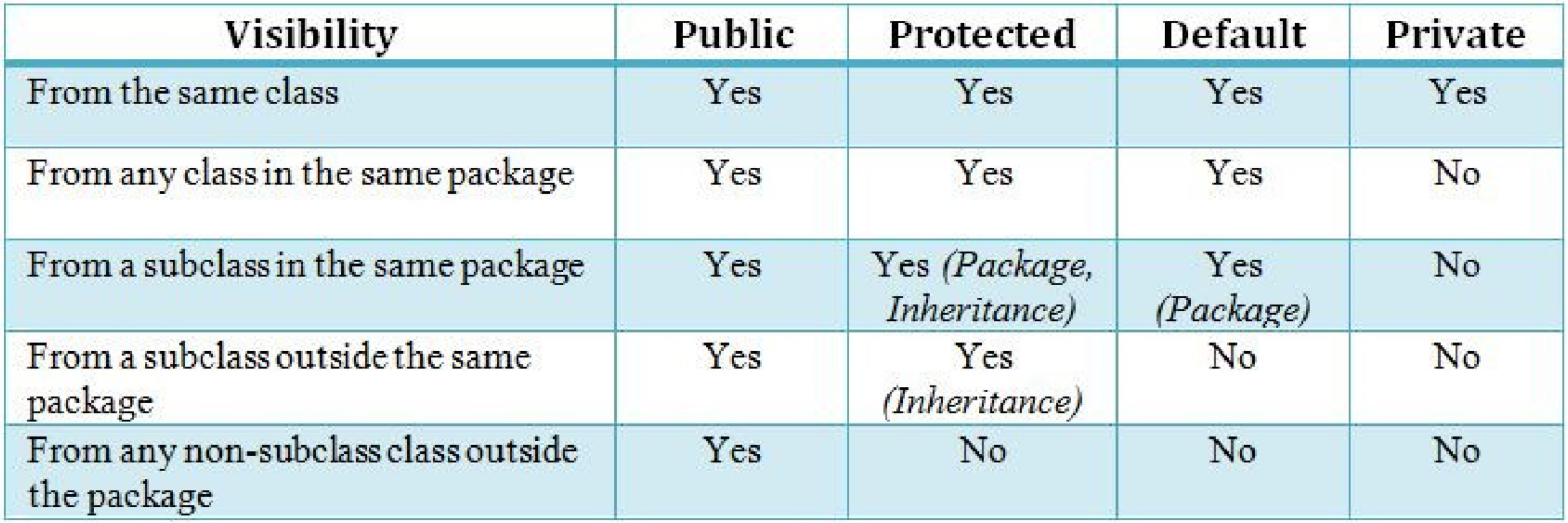
- Statements have to end in semicolon ;
  - Hint: Every line in java ends in bracket or semicolon

- Convention to indent inside {}

Visibility Modifiers

What do you want to be “hidden” from others?

Visibility Modifiers



Visibility	Public	Protected	Default	Private
From the same class	Yes	Yes	Yes	Yes
From any class in the same package	Yes	Yes	Yes	No
From a subclass in the same package	Yes	Yes *(Package, Inheritance)*	Yes *(Package)*	No
From a subclass outside the same package	Yes	Yes *(Inheritance)*	No	No
From any non-subclass class outside the package	Yes	No	No	No