More on Abstraction in Java

Recap of abstract classes and methods

Produced by: Dr. Siobhán Drohan
Mairead Meagher

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/

Abstract vs Concrete

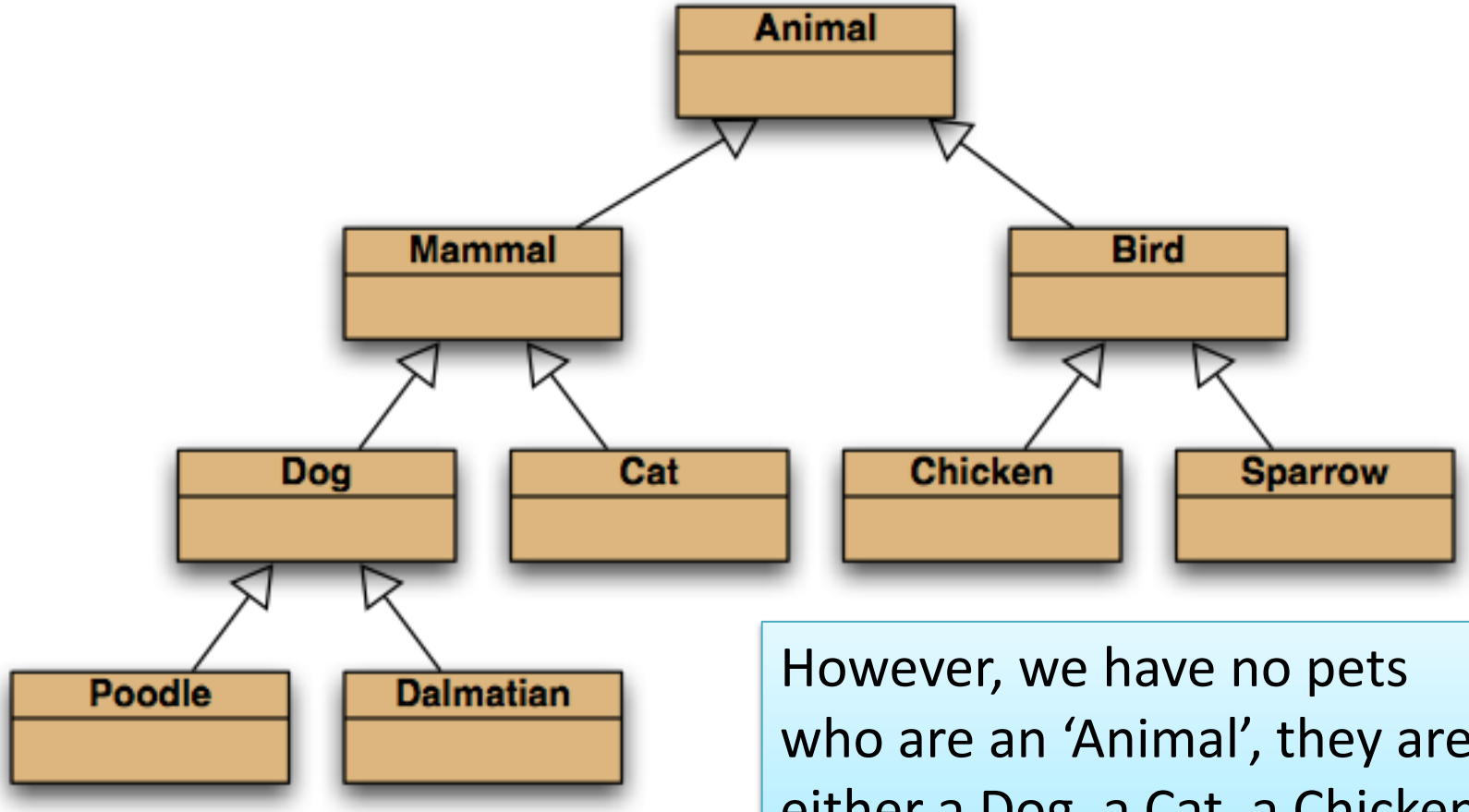
- Abstract
  - Implementation delayed

    → abstract method has no code

    → cannot instantiate an abstract class (it has, by definition “unfinished” methods)

- Concrete
  - Ready to go.
  - Everything up to now has been concrete.

Inheritance hierarchies



However, we have no pets who are an ‘Animal’, they are either a Dog, a Cat, a Chicken, etc.

Abstract Methods

- Abstract methods have `abstract` in the signature.
- Abstract methods have no body.
  - ‘We promise to write this later. Every (concrete) subclass of this class will have this implemented in the subclass.’
- Abstract methods make the class abstract.
  - Think about why this is?

Abstract Classes

- An abstract class is a class that contains zero or more abstract methods.
- An class that has an abstract method must be declared abstract.
- Abstract classes cannot be instantiated.
- Abstract classes function as a “base” for subclasses.

  → abstract classes can be subclassed.
- Concrete subclasses complete the implementation.

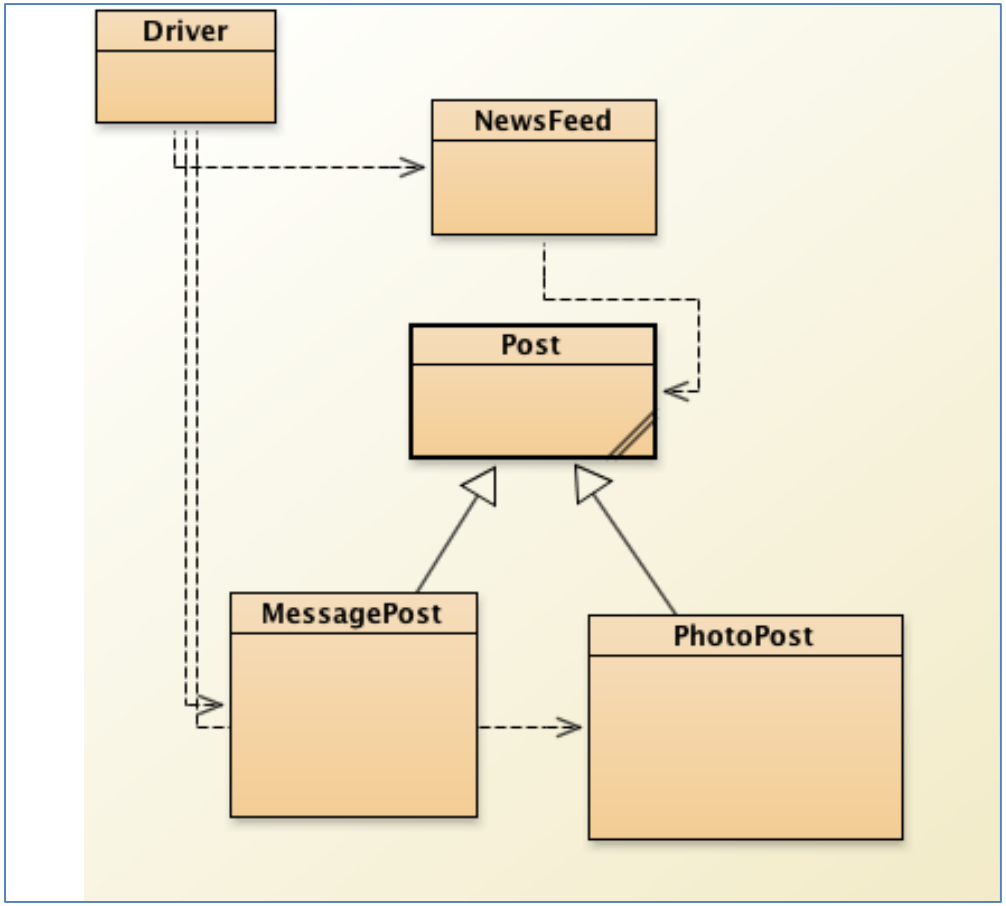
Network-V4 (no abstraction)

```
Options

Posts
1) Add a Text Post
2) Add a Photo Post
3) List all Posts
0) Exit
==>>
```

Our news feed displays either MessagePost or PhotoPost objects.

NOTE: we never create a Post object but our ArrayList is of Post.

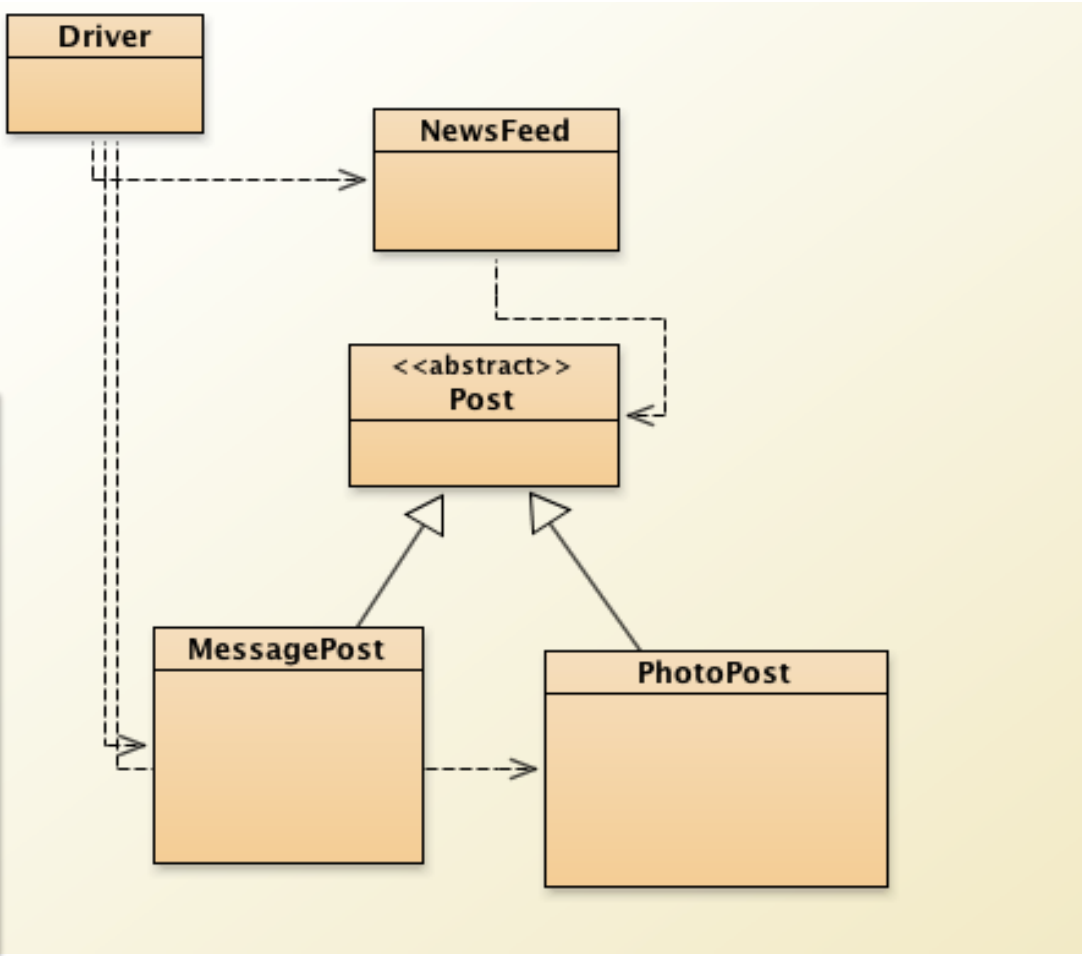


Network-V5 (Post as an abstract class)

```
Options
Posts
1) Add a Text Post
2) Add a Photo Post
3) List all Posts
0) Exit
==>>
```

So, because we never create a Post object but our ArrayList is of Post…

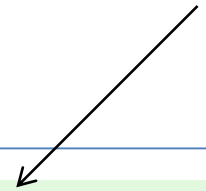
We can make Post abstract!



Network-V5 (Post as an abstract class)

- We can never create a ‘post’ object
  - We cannot instantiate one because Post is abstract.
- In Post, we define fields and methods that can be used later for all subclasses (using super)
  - e.g. display(), constructor.

Syntax for abstract classes



```
public abstract class Post
{
    private String username;  // username of the post's author
    private long timestamp;
    private int likes;
    private ArrayList<String> comments;
```

displayExcerpt() as an abstract method

- If you wish all subclasses of a class to implement a particular method as part of its code, simply write an abstract method heading in superclass.

- Each subclass must have this method fully coded OR the class must be declared as abstract.

displayExcerpt() as an abstract method

```
abstract String displayExtract();
```

Post

```
/**
 * return a short extract of the post message
 * @return A string containing the first 10 chars of the post
 */
String displayExtract()
{
    return "Message extract "+ message.substring(0,10) + "....";
}
```

MessagePost

```
/**
 * return a short extract of the photo caption
 * @return A string containing the first 10 chars of the caption
 */
String displayExtract()
{
    return "Photo caption: " + caption.substring(0,10) + "....";
}
```

PhotoPost

Any Questions?

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/