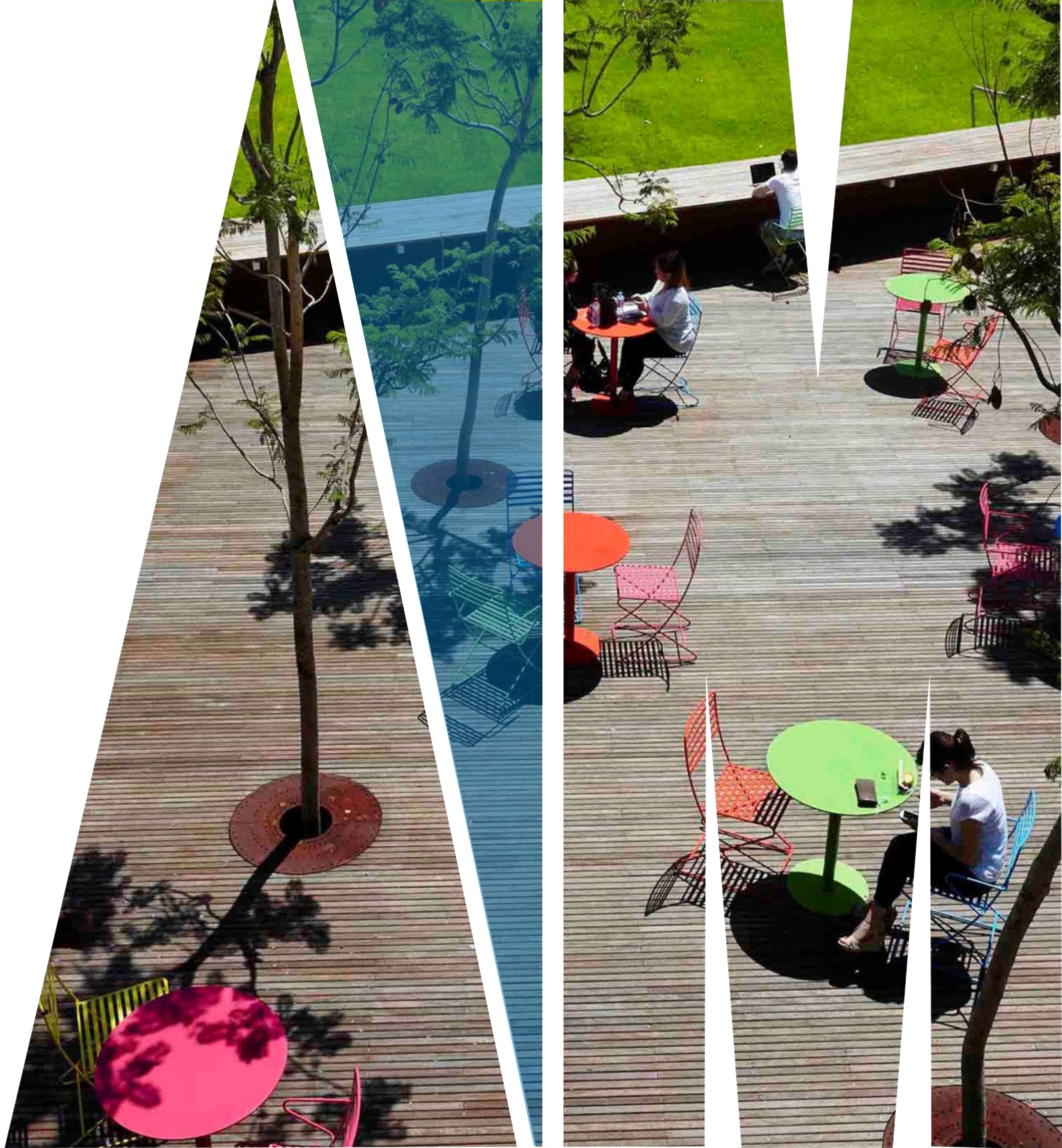
FIT2099 Object-Oriented Design and Implementation

Abstract classes



Outline

Abstract classes

- Real world examples

Polymorphism

Concrete classes versus abstract classes

Abstract methods

UML representation

ABSTRACT CLASSES

REAL-WORLD REPRESENTATION

A definition of animals:

THIS IS AN ABSTRACT IDEA CREATED BY HUMANS

Animals have several characteristics that set them apart from other living things. Animals are eukaryotic and multicellular. Unlike plants and algae, which produce their own nutrients animals are heterotrophic, feeding on organic material and digesting it internally. With very few exceptions, animals respire aerobically….. etc..

MONASH University

ABSTRACT CLASSES
REAL-WORLD REPRESENTATION

Animal

You see many animals in real life, but there are only **kinds of animals**.

That is, you never look at something brown and furry and say "**that is an animal and there is no more specific way of defining it**".

ABSTRACT CLASSES

REAL-WORLD REPRESENTATION

Animal

Speak() *How does a (specific animal) sound?*

Oink oink!

Miau miau!

Woof woof!

Quack quack!

POLYMORPHISM

Polymorphism in OOP can be broadly described as **the ability of a <u>message</u> to be displayed in more than one form**.

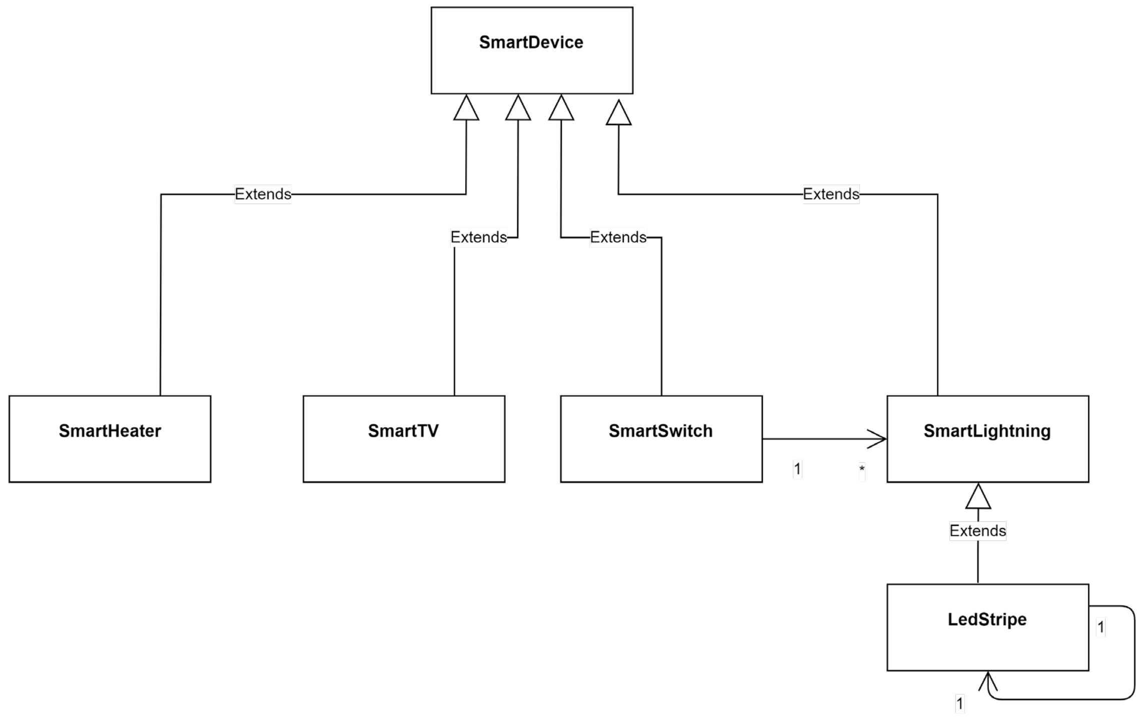
Another way to define it:
The ability of **performing a single action in different ways.**

In practical (aka. coding) terms:
The ability of **defining one interface** and have **multiple implementations.**

POLYMORPHISM WITH CONCRETE CLASSES?

All classes we have defined so far are **concrete classes** because they provide implementations of every method they declare.

We used @override to change the implementation of a couple of base class methods in one child class.

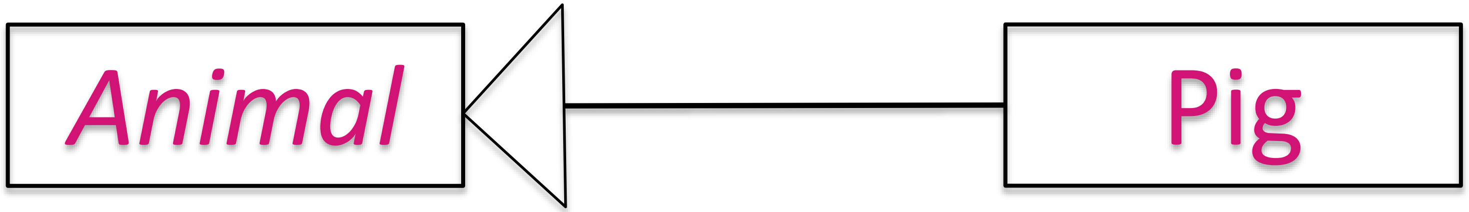


MONASH University

THE ABSTRACT CLASS

They are **incomplete** classes for which you **never intend to create objects…what?**

Subclasses must declare the **missing pieces** to become **concrete classes**, from which you can instantiate objects; otherwise, these subclasses, too, will be abstract.



Abstract class

Concrete class

DECLARING
ABSTRACT CLASS

You make a class abstract by declaring it with keyword **`abstract`**

An abstract class normally contains one or more **abstract methods**, which are declared with the keyword **`abstract`** and provides **no implementations**.

```java
public abstract class Animal {

    public abstract void speak();

    // ...
}
```

No brackets! {}

MONASH University

CONCRETE CLASSES V/S ABSTRACT CLASS

Concrete classes
Allow to create an object using the “new” keyword
Have all their methods **implemented**
Programmers **can** create objects with concrete classes
They can contain attributes

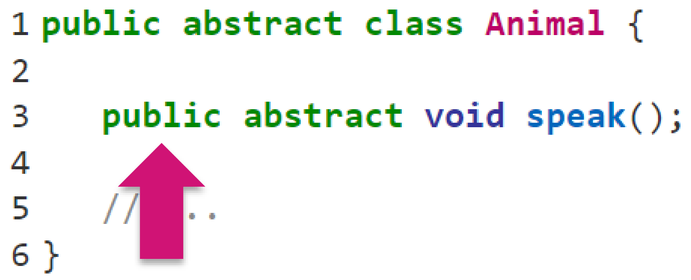
CONCRETE CLASSES V/S ABSTRACT CLASS

Concrete classes	Abstract classes
Allow to create an object using the "new" keyword	Are generally **base** classes
Have all their methods **implemented**	Have a collection of **implemented** and **non-implemented** (abstract) methods
Programmers **can** create objects with concrete classes	Programmers **cannot** create objects with abstract classes
They can contain attributes	They can contain attributes

MONASH University

THE ABSTRACT METHODS

Abstract methods have the same visibility rules as regular methods, except that **they cannot be private** (it makes little sense). Similarly, **constructors** and **static** methods cannot be declared abstract



```java
public abstract class Animal {

    public abstract void speak();

    // ...
}
```

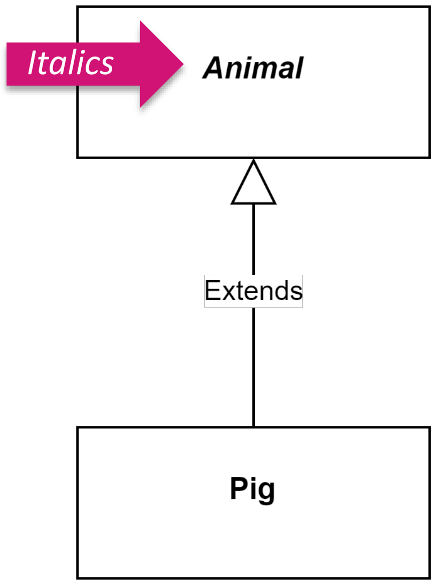
Abstract methods have **no implementations** because the abstract classes are too general (they only specify the **common interfaces** of the subclasses).

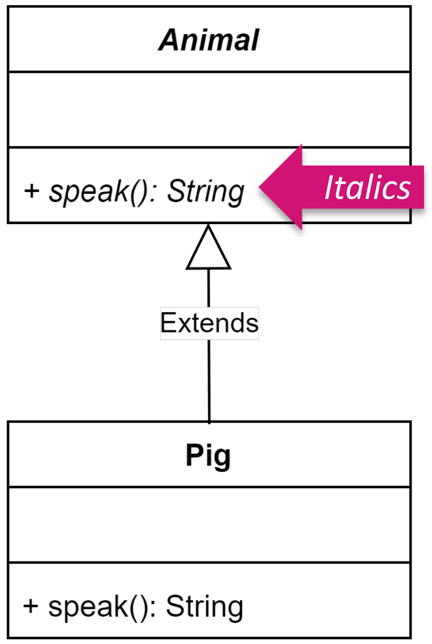
If a subclass **does not implement all abstract methods** it inherits from an abstract class, the subclass must also be **abstract.**

UML SYNTAX FOR ABSTRACT CLASSES

Abstract class

Concrete class





Summary

Abstract classes

  Real world examples

Polymorphism

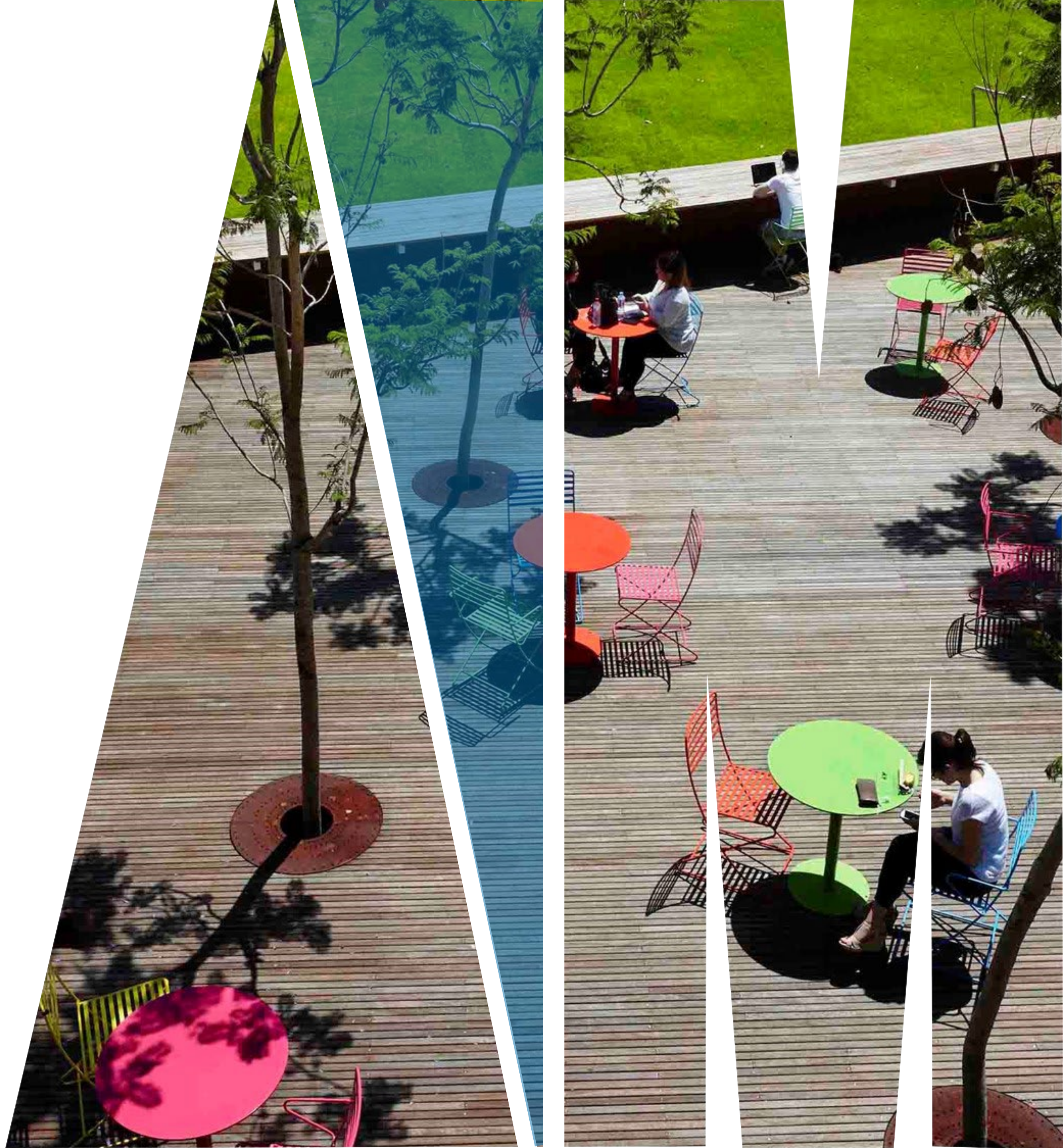
Concrete classes versus abstract classes

Abstract methods

UML representation

MONASH University

Thanks



MONASH University