W4 Tutorial

Object Oriented programming

OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH

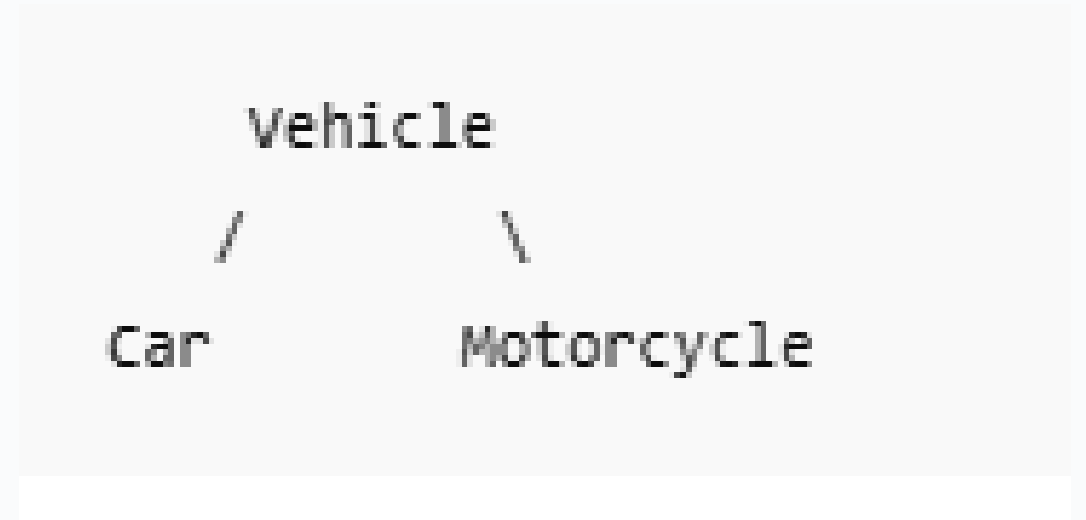TU DUBLIN

TECHNOLOGICAL
UNIVERSITY DUBLIN

Introducing Inheritance with Car & Motorcycle

Slide 2 – Step 1: Start with Two Classes

- Car.java

- Motorcyle.java

Step 2: The "is-a" relationship

- A **Car is a Vehicle**

- A **Motorcycle is a Vehicle**

- Both share core behaviour.

```
          Vehicle
         /       \
      Car        Motorcycle
```

Step 3 – Create Vehicle Superclass

Shared attributes
ownerName, registration, engineOn, currentSpeed, maxSpeed

Shared methods
startEngine(), stopEngine(), accelerate(), brake(), toString()

Common code → move to Vehicle.java.

Step 4 – Extend Vehicle

public class Car extends Vehicle {

    private String colour;

    private boolean automatic;

    ...

}

public class Motorcycle extends Vehicle {

    private boolean hasSidecar;

    ...

}

Use super(...) in constructors to call the Vehicle constructor.

Step 5: Overriding Methods

- Add a honk() method in Vehicle → "honk honk!"
- Override in subclasses:
- Car → "Beep beep!"
- Motorcycle → "Vroooom!"

Subclasses can change behaviour defined in the parent.

Step 6: Polymorphism in Action

```
vehicle v1 = new Car(...);

vehicle v2 = new Motorcycle(...);



v1.honk(); // Beep beep!

v2.honk(); // Vroooom!
```

Reflection

What changed after adding inheritance?
Why is this structure better?
Difference between **overriding** and **overloading**?