Northumbria University
NEWCASTLE

KF6012 - Web Application Integration

2019-20

The Four Pillars of Object Oriented Programming

Lecture Overview

- The four pillars of Object Oriented Programming

The four pillars of OO Programming

Encapsulation

Abstraction

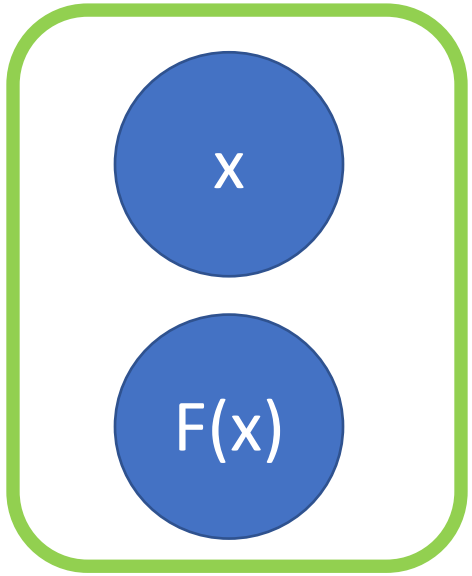
Inheritance

Polymorphism

Encapsulation

Encapsulation

- In object oriented programming, related variables and functions are combined together into classes/objects



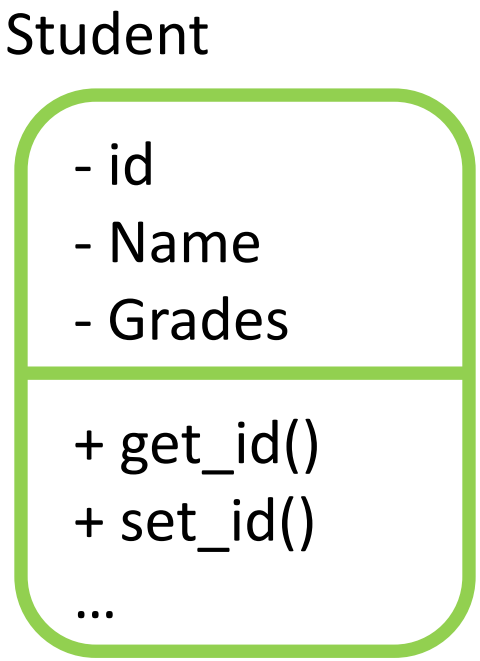
Variables are called properties

Functions are called methods

- Combining these is known as “encapsulation”

Encapsulation

- For example we might define a “student” class like this:



Variables are called properties

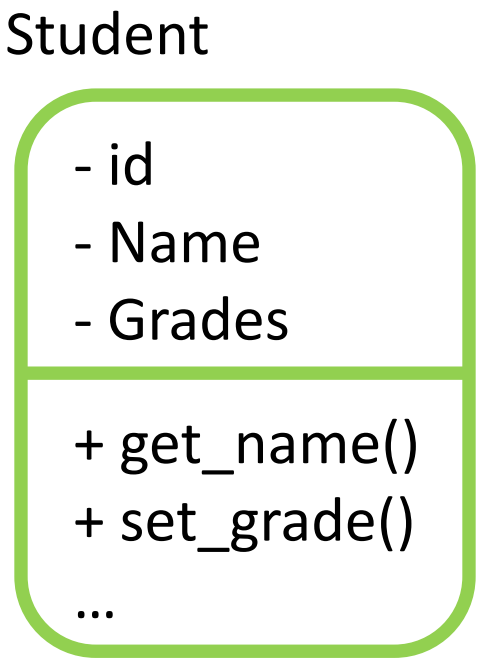
Functions are called methods

Abstraction

Abstraction

- Abstraction in OO means that we provide simpler ways for programs to interface with methods and properties
  - Some methods and properties are exposed to code outside the class
  - Some methods and properties are kept hidden
- Abstraction is helpful because
  - It makes the code simpler, there are only a few things we need to worry about when interacting with an object
  - It reduces the impact of changes to code outside of a class

Abstraction



The ‘–’ means these are private to the object

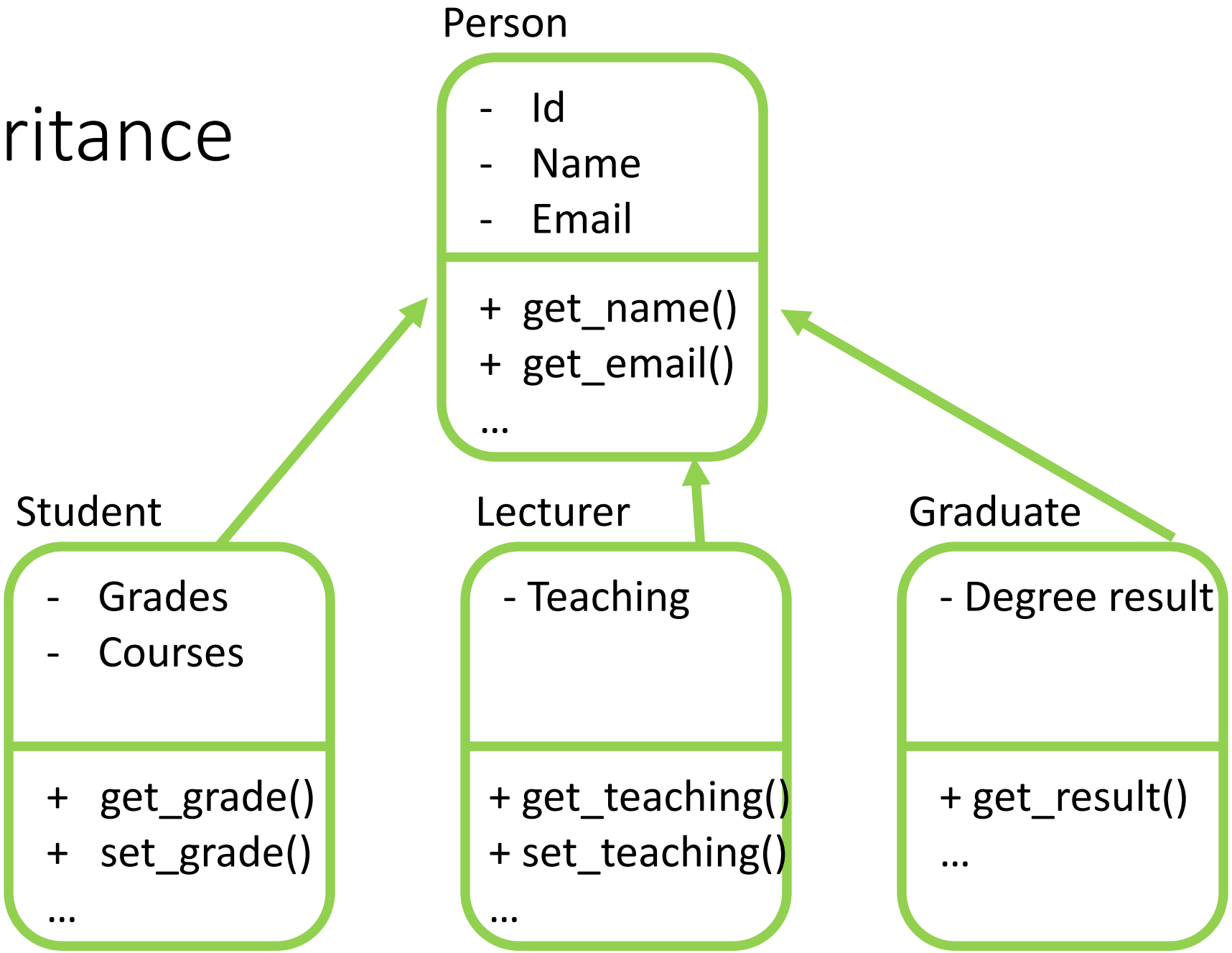
The ‘+’ means these are public outside the object

Inheritance

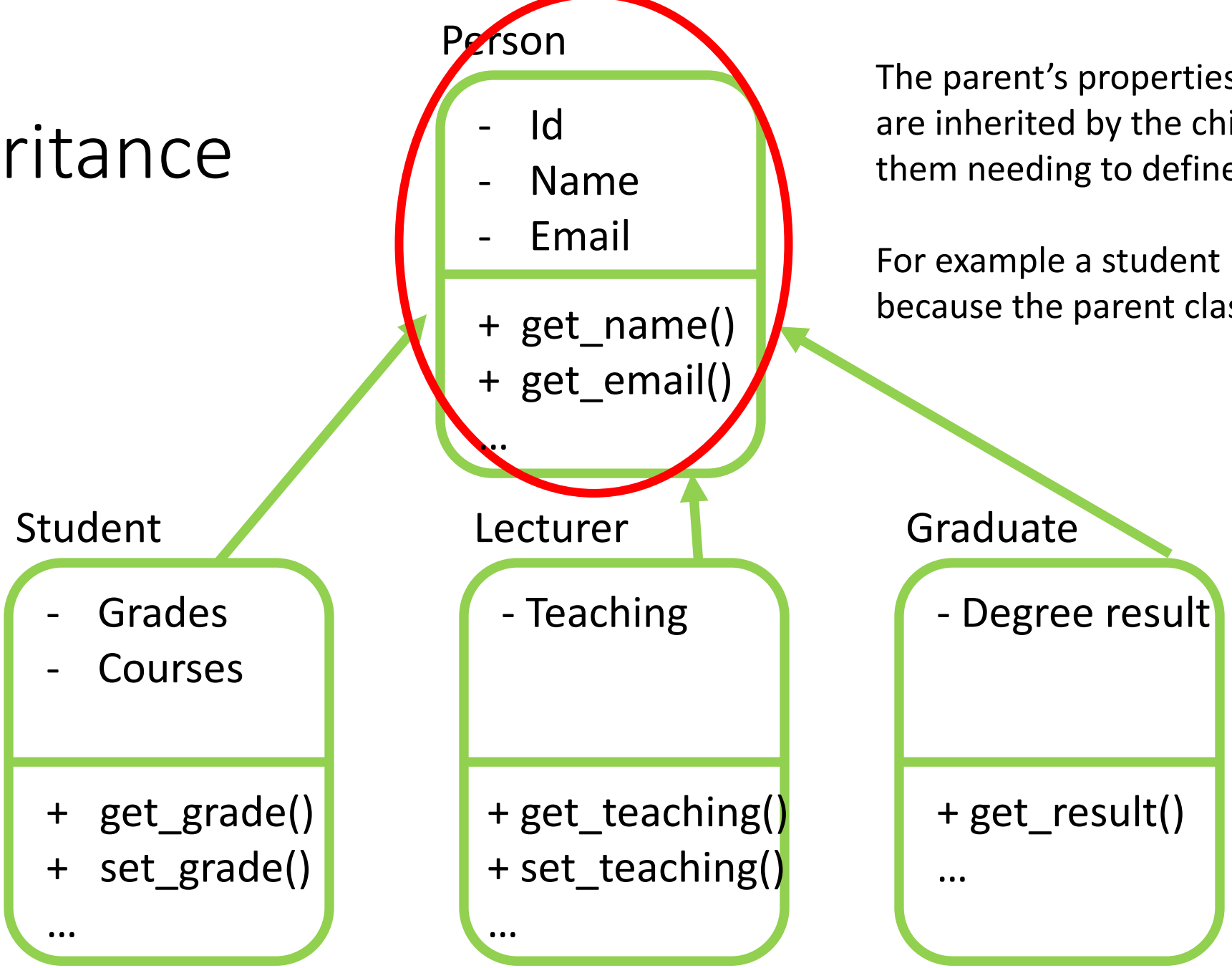
Inheritance

- Inheritance means that a class/object can ‘inherit’ properties and methods from a ‘parent’
- Multiple classes may inherit from a single parent
  - This allows us to eliminate redundant code.
  - Means we do not have to copy and paste code between functions

Inheritance



Inheritance



The parent's properties and methods are inherited by the children, without them needing to define them.

For example a student has an id, because the parent class does.

Polymorphism

Polymorphism

- This means that classes have different functionality while sharing a common interface.
  - Meaning the code using different objects/classes does not need to care about which specific class it is.
  - E.g. We can get and set the email address of a student in just the same way as we can for a lecturer.

Polymorphism

- This means that classes have different functionality while sharing a common interface.
  - Meaning the code using different objects/classes does not need to care about which specific class it is.
  - E.g. We can get and set the email address of a student in just the same way as we can for a lecturer.

- More on this is upcoming weeks

Thanks

- john.rooksby@northumbria.ac.uk