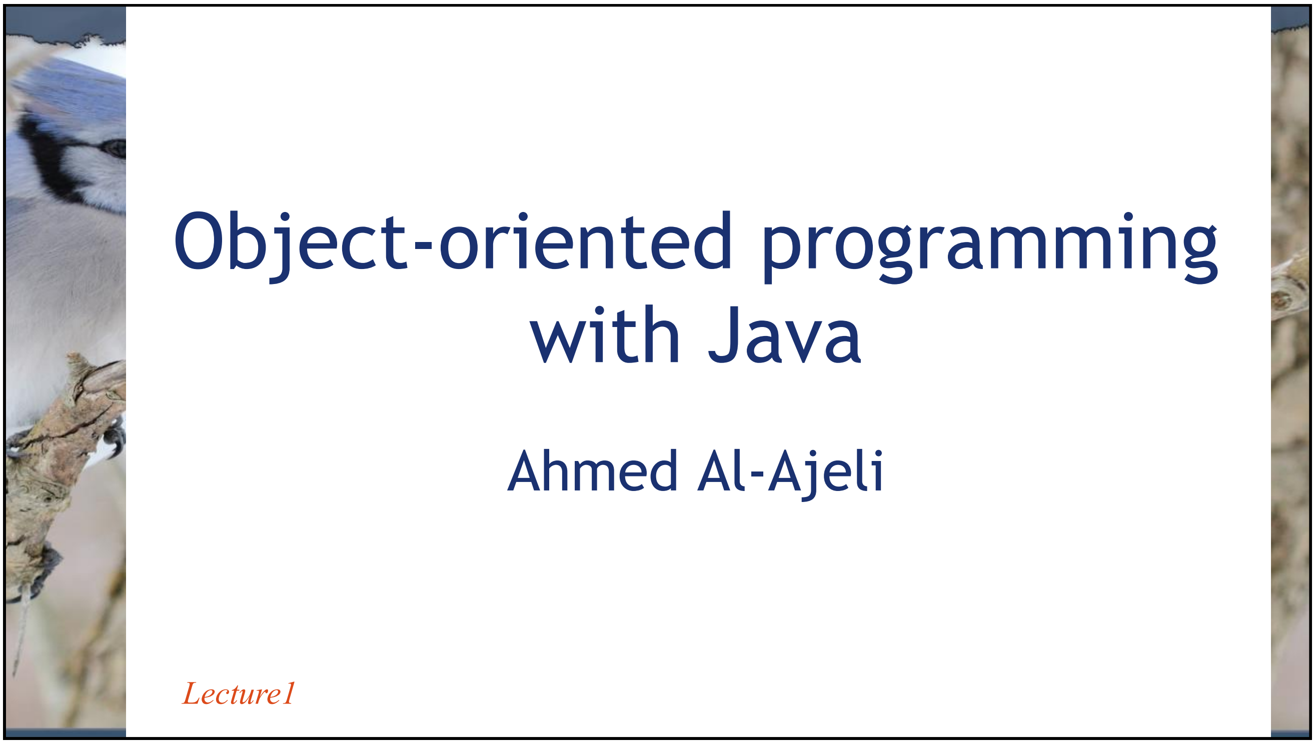
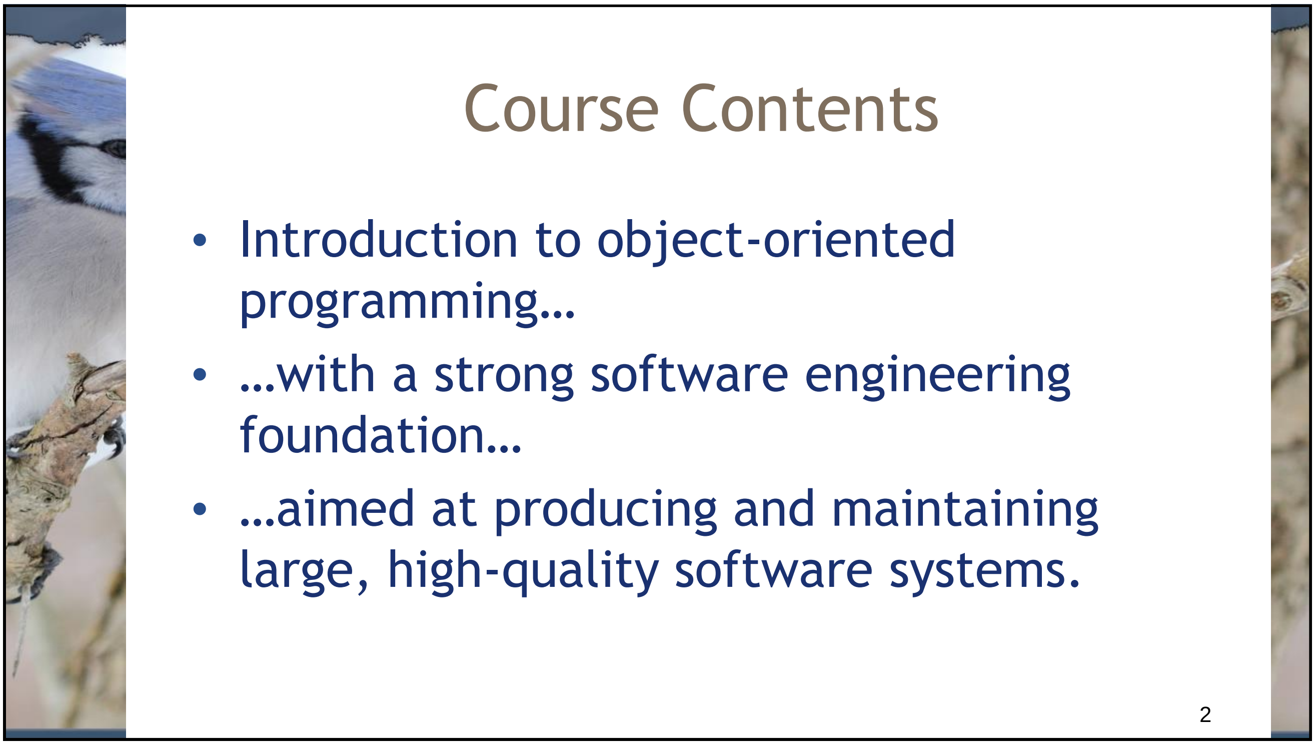


Object-oriented programming with Java

Ahmed Al-Ajeli

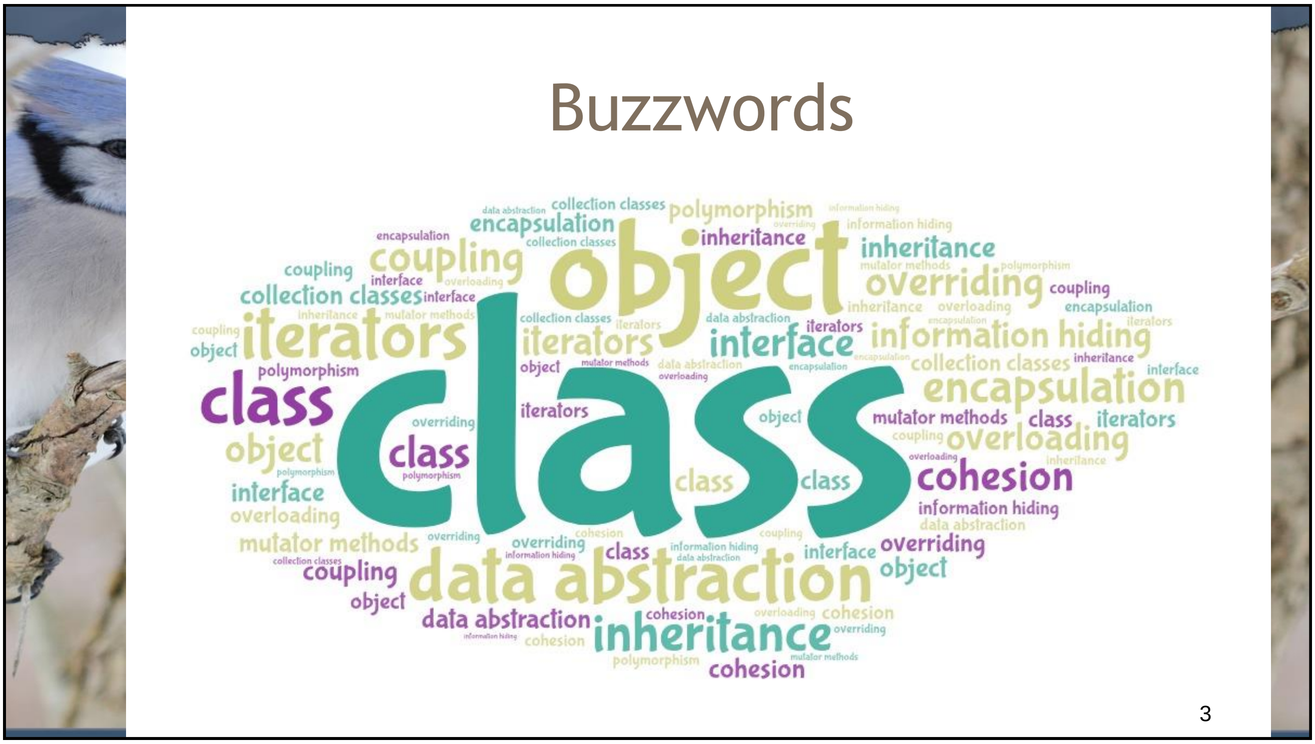
Lecture1



Course Contents

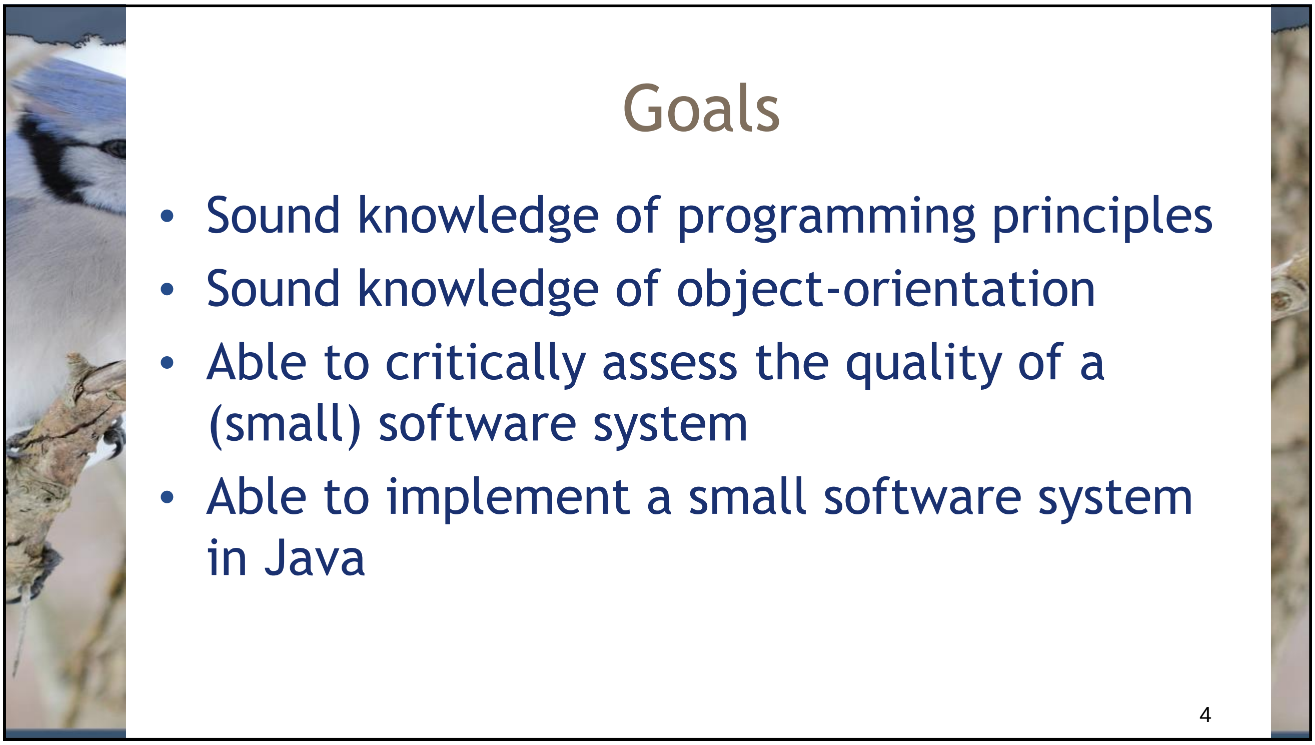
- Introduction to object-oriented programming…
- …with a strong software engineering foundation…
- …aimed at producing and maintaining large, high-quality software systems.

2



Buzzwords

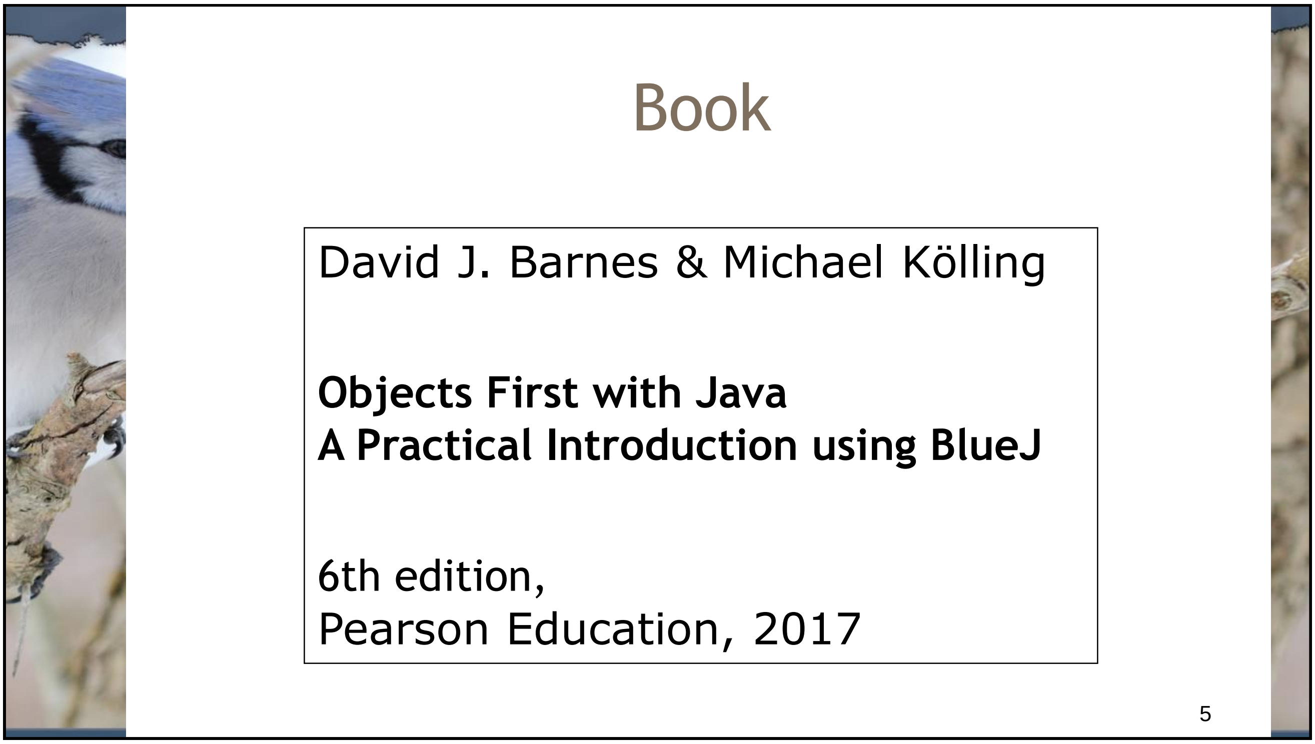
3



Goals

- Sound knowledge of programming principles
- Sound knowledge of object-orientation
- Able to critically assess the quality of a (small) software system
- Able to implement a small software system in Java

4

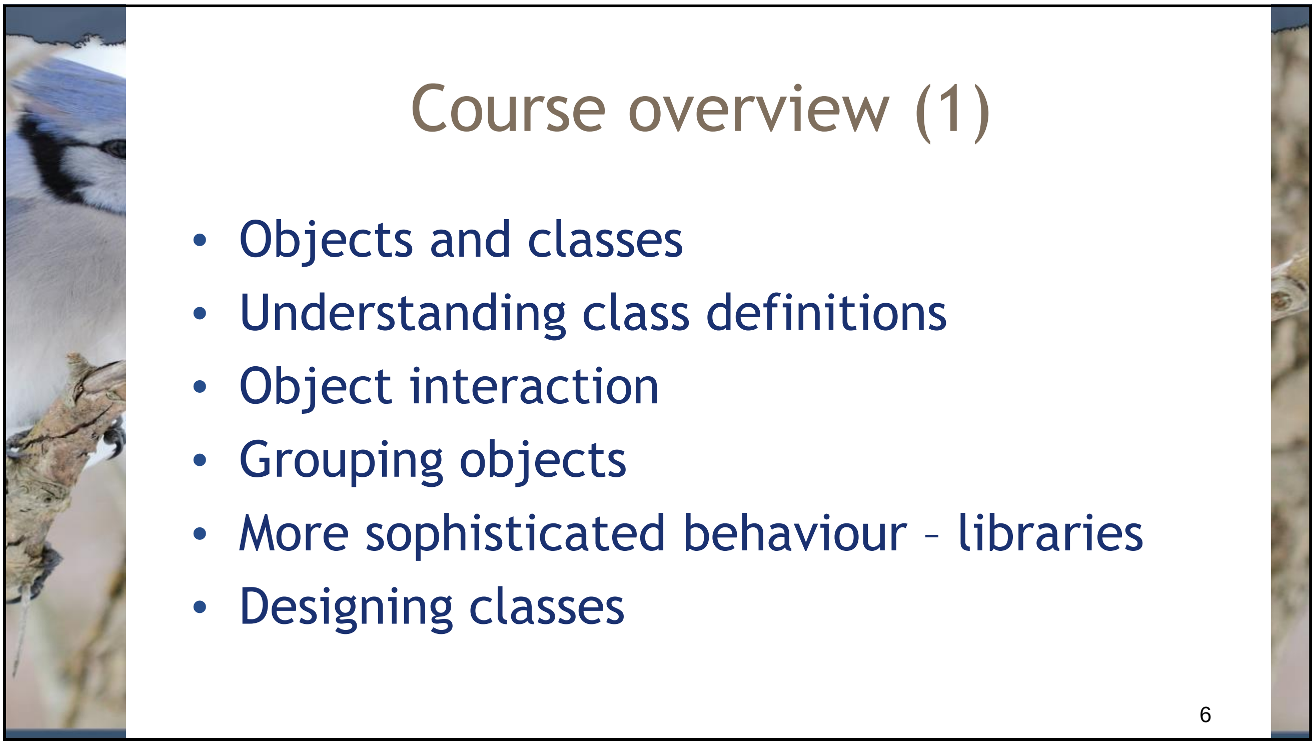


Book

David J. Barnes & Michael Kölling

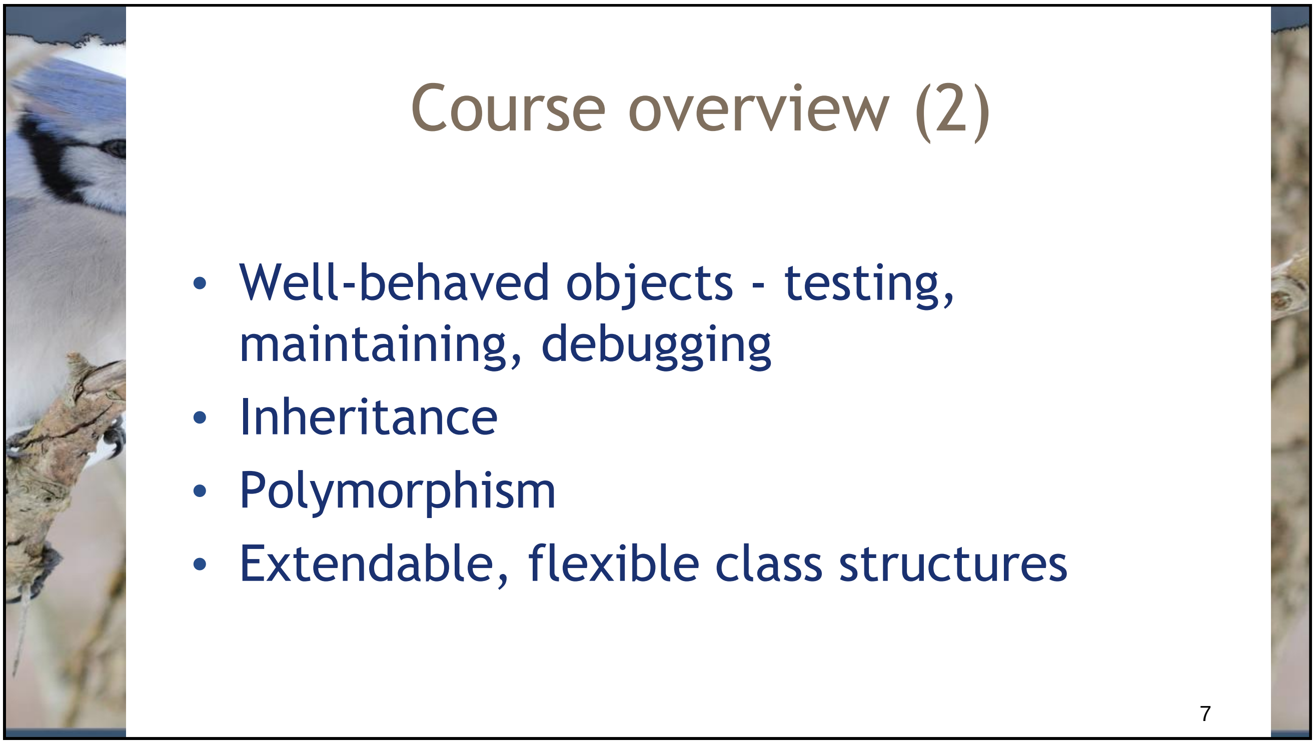
Objects First with Java
A Practical Introduction using BlueJ

6th edition,
Pearson Education, 2017



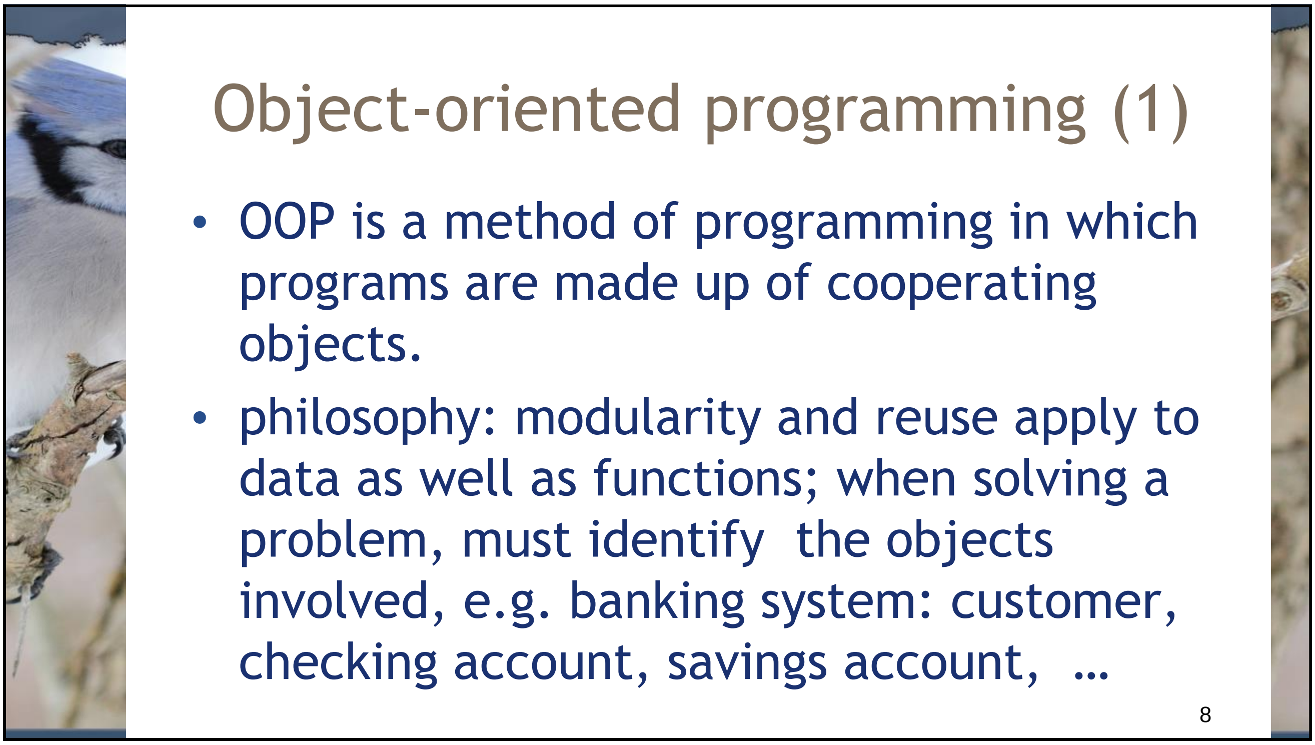
Course overview (1)

- Objects and classes
- Understanding class definitions
- Object interaction
- Grouping objects
- More sophisticated behaviour – libraries
- Designing classes



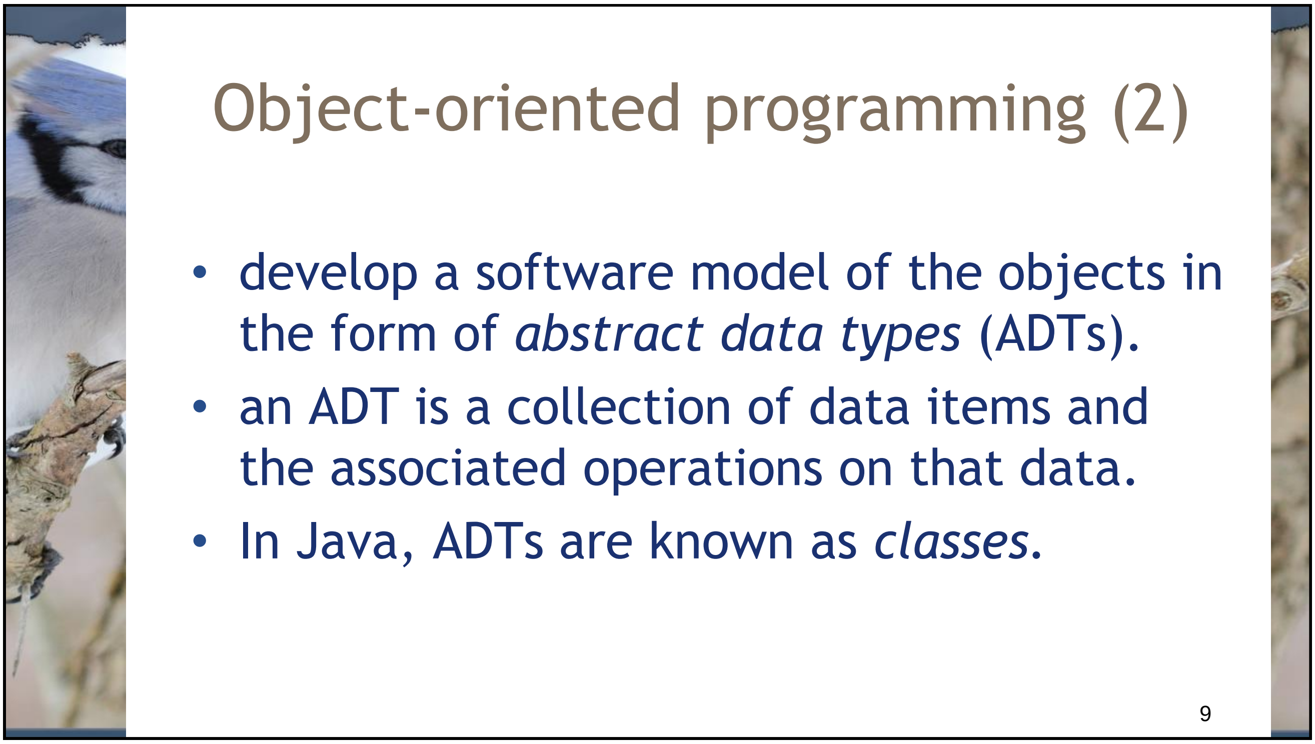
Course overview (2)

- Well-behaved objects - testing, maintaining, debugging
- Inheritance
- Polymorphism
- Extendable, flexible class structures



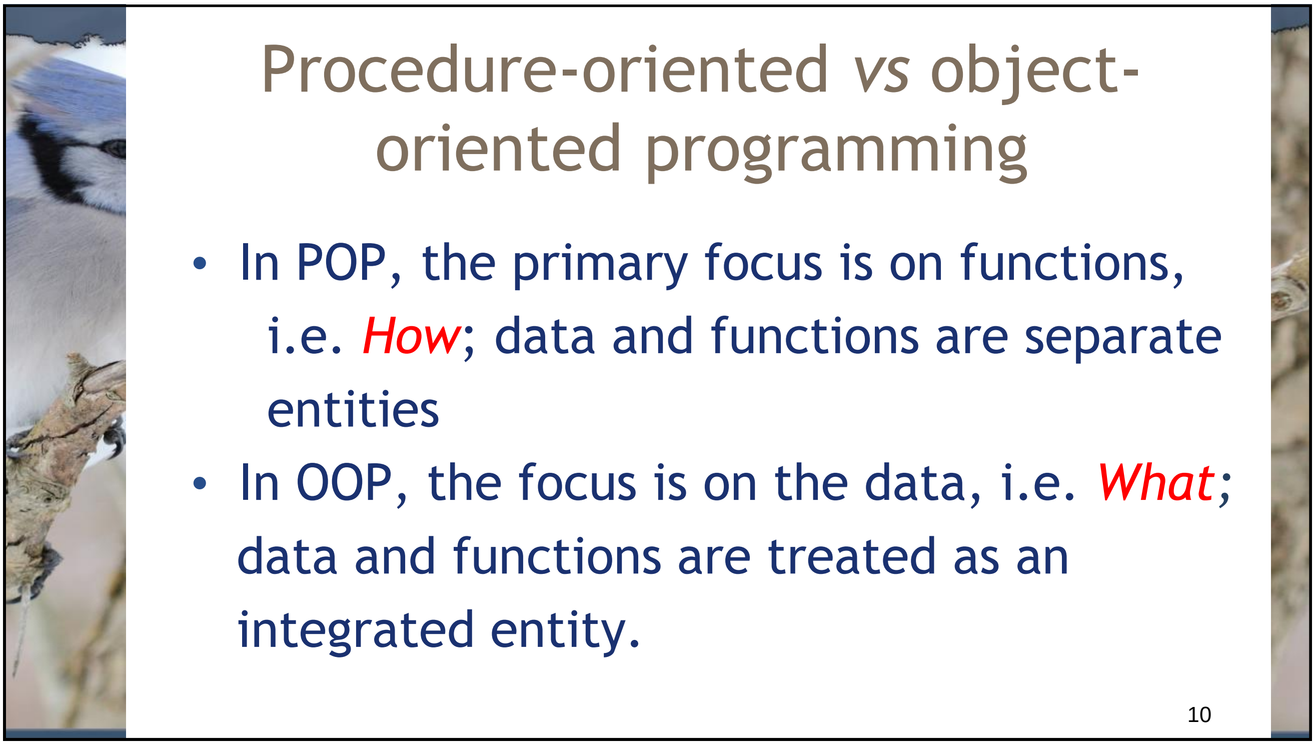
Object-oriented programming (1)

- OOP is a method of programming in which programs are made up of cooperating objects.
- philosophy: modularity and reuse apply to data as well as functions; when solving a problem, must identify the objects involved, e.g. banking system: customer, checking account, savings account, …



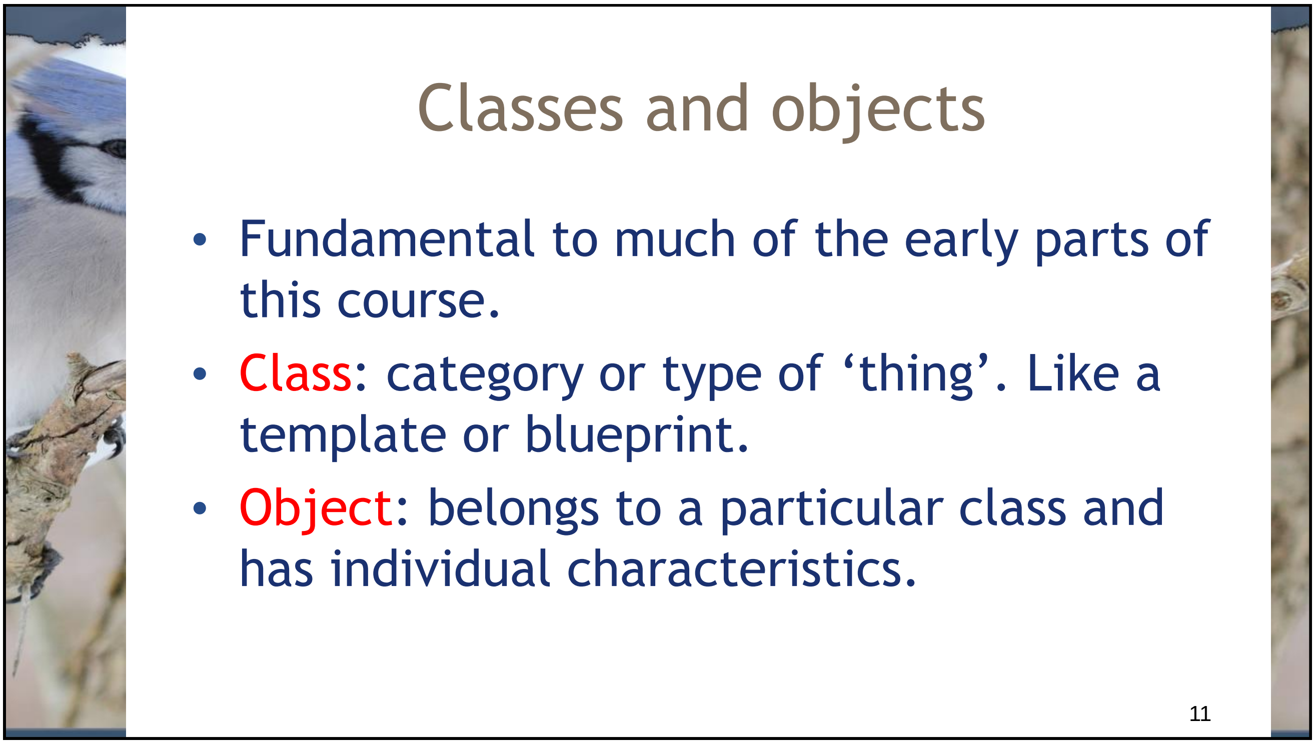
Object-oriented programming (2)

- develop a software model of the objects in the form of *abstract data types* (ADTs).
- an ADT is a collection of data items and the associated operations on that data.
- In Java, ADTs are known as *classes*.



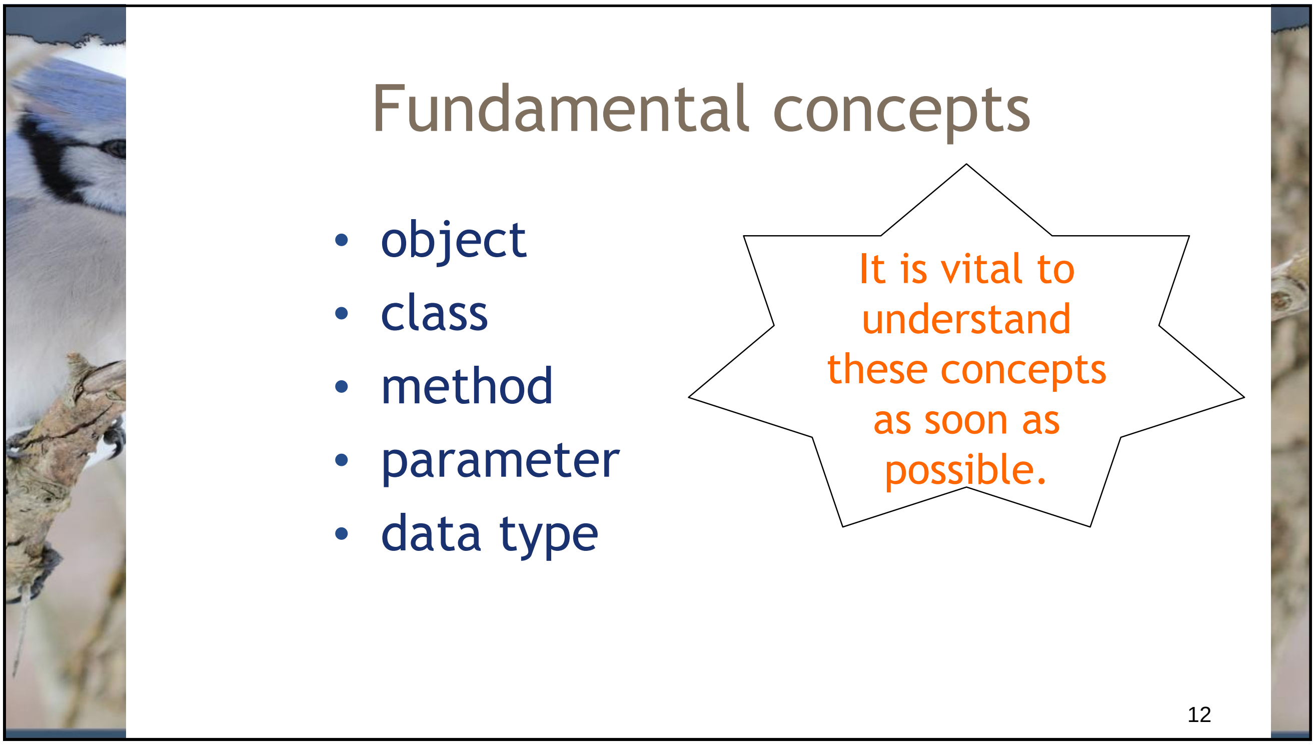
Procedure-oriented *vs* object-oriented programming

- In POP, the primary focus is on functions, i.e. *How*; data and functions are separate entities
- In OOP, the focus is on the data, i.e. *What*; data and functions are treated as an integrated entity.



Classes and objects

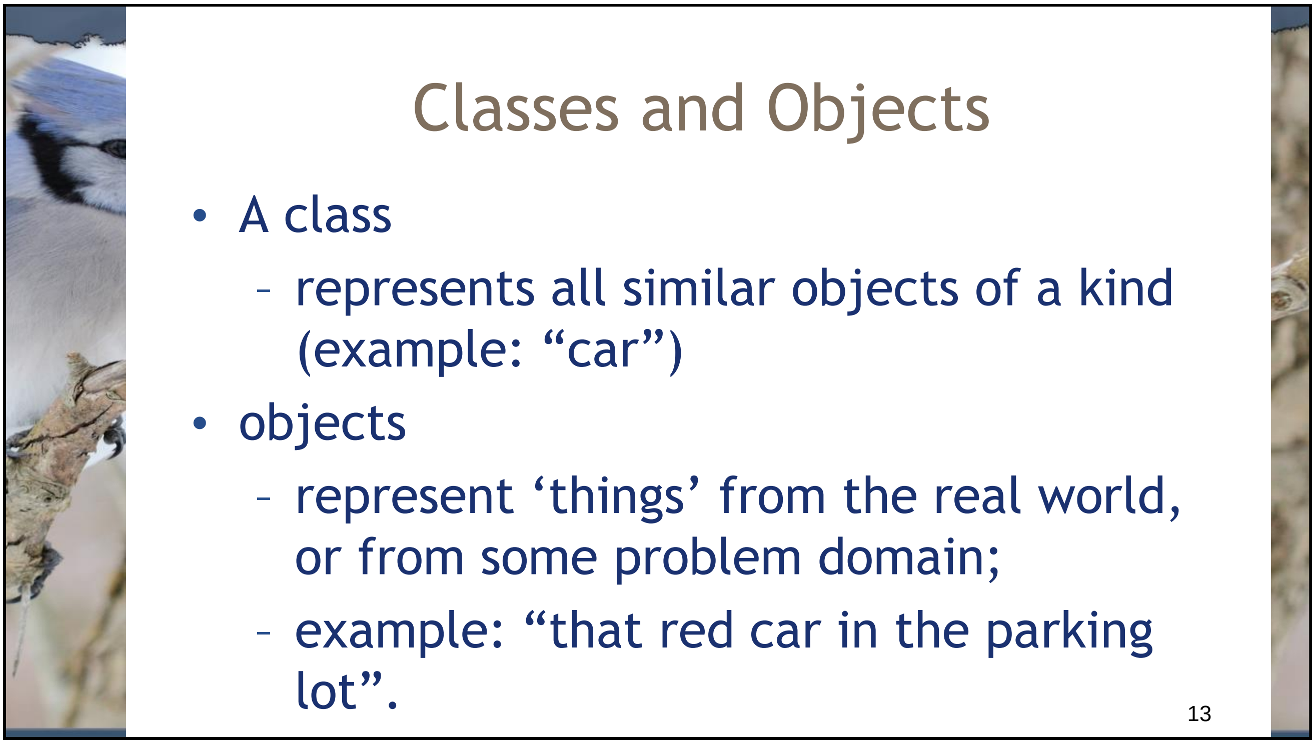
- Fundamental to much of the early parts of this course.
- Class: category or type of ‘thing’. Like a template or blueprint.
- Object: belongs to a particular class and has individual characteristics.



Fundamental concepts

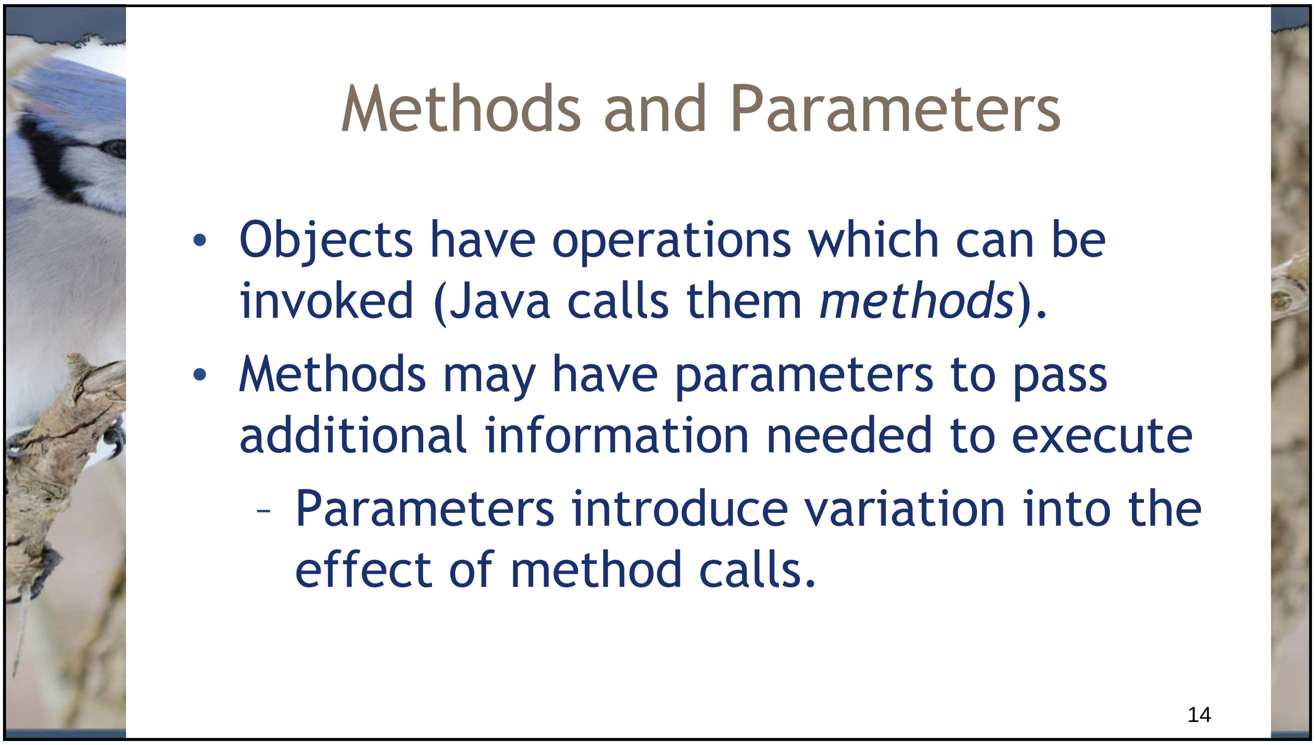
- object
- class
- method
- parameter
- data type

It is vital to understand these concepts as soon as possible.



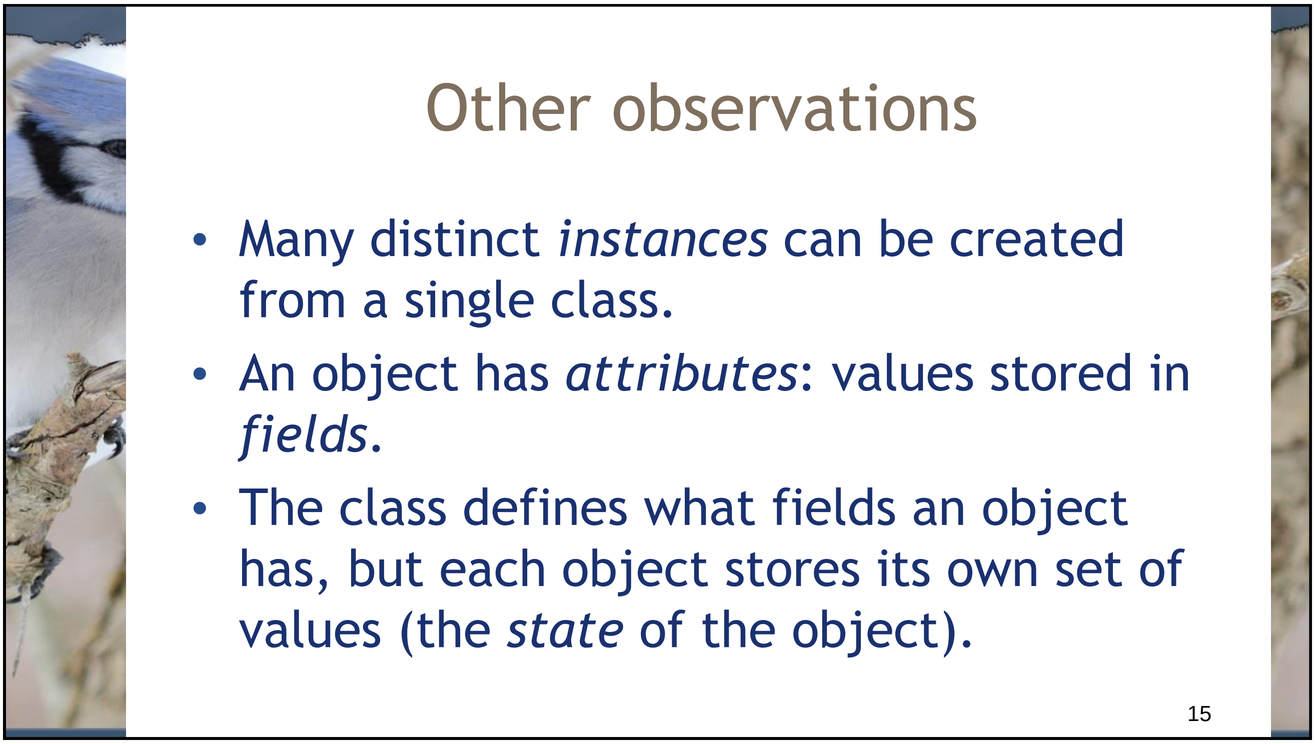
Classes and Objects

- A class
  - represents all similar objects of a kind (example: “car”)
- objects
  - represent ‘things’ from the real world, or from some problem domain;
  - example: “that red car in the parking lot”.



Methods and Parameters

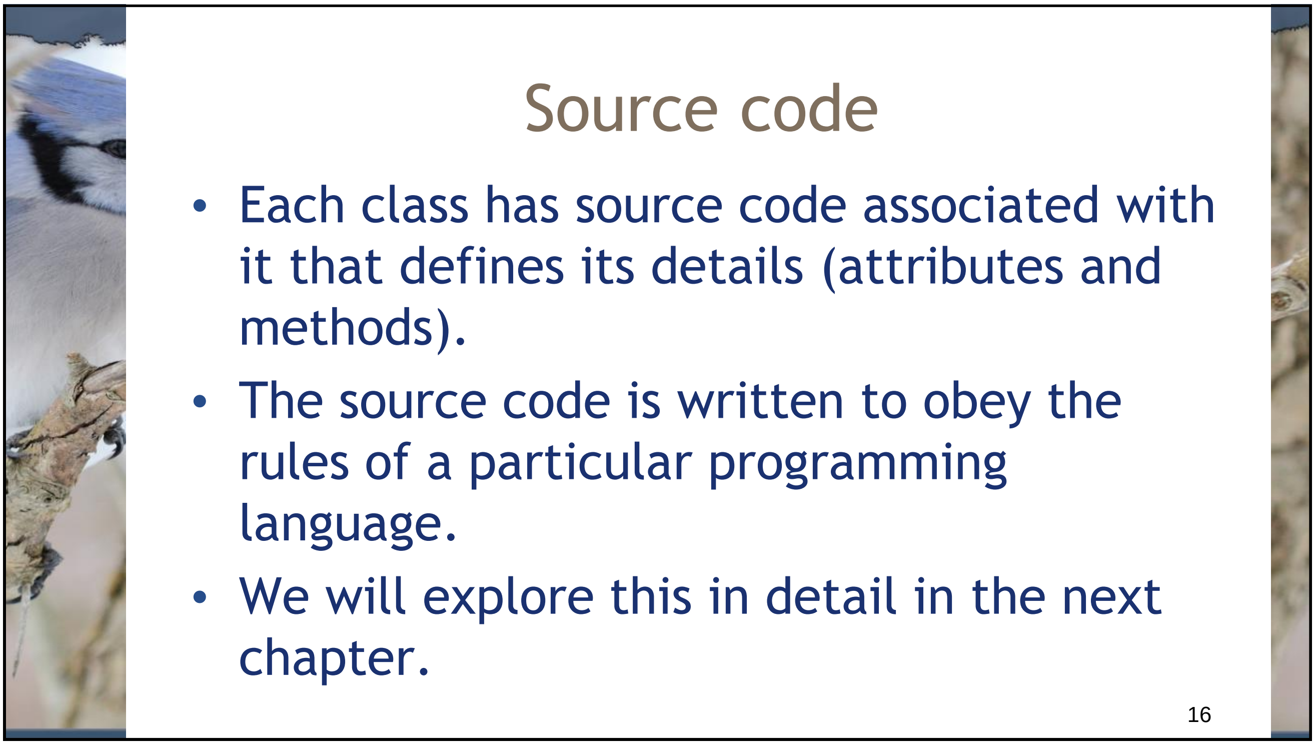
- Objects have operations which can be invoked (Java calls them *methods*).
- Methods may have parameters to pass additional information needed to execute
  - Parameters introduce variation into the effect of method calls.



Other observations

- Many distinct *instances* can be created from a single class.
- An object has *attributes*: values stored in *fields*.
- The class defines what fields an object has, but each object stores its own set of values (the *state* of the object).

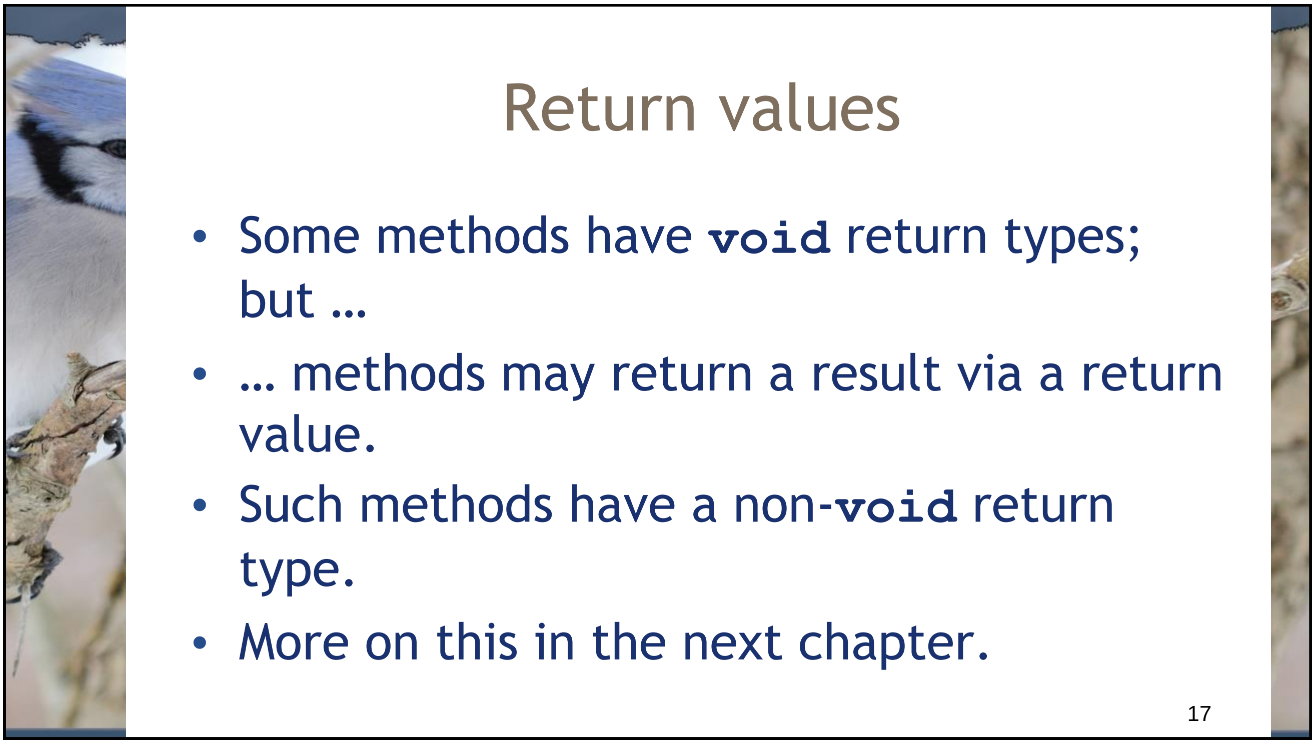
15



Source code

- Each class has source code associated with it that defines its details (attributes and methods).
- The source code is written to obey the rules of a particular programming language.
- We will explore this in detail in the next chapter.

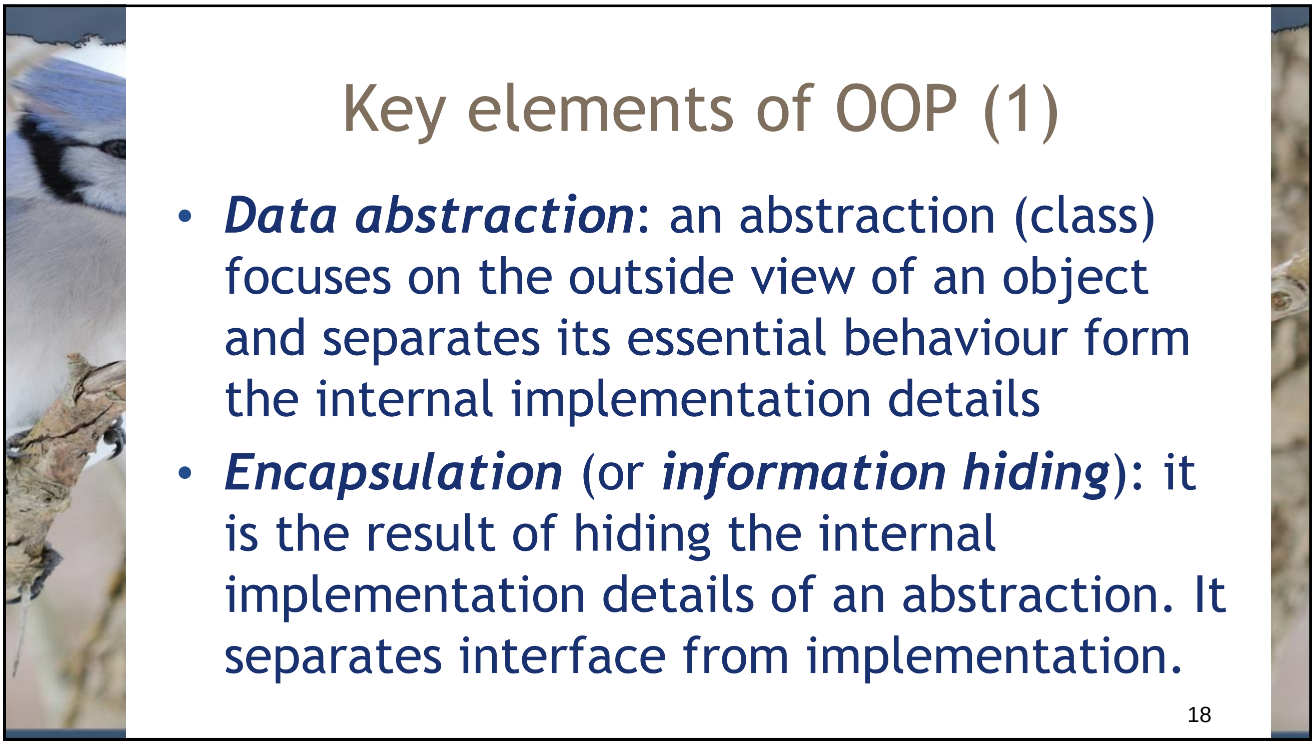
16



Return values

- Some methods have **`void`** return types; but …
- … methods may return a result via a return value.
- Such methods have a non-**`void`** return type.
- More on this in the next chapter.

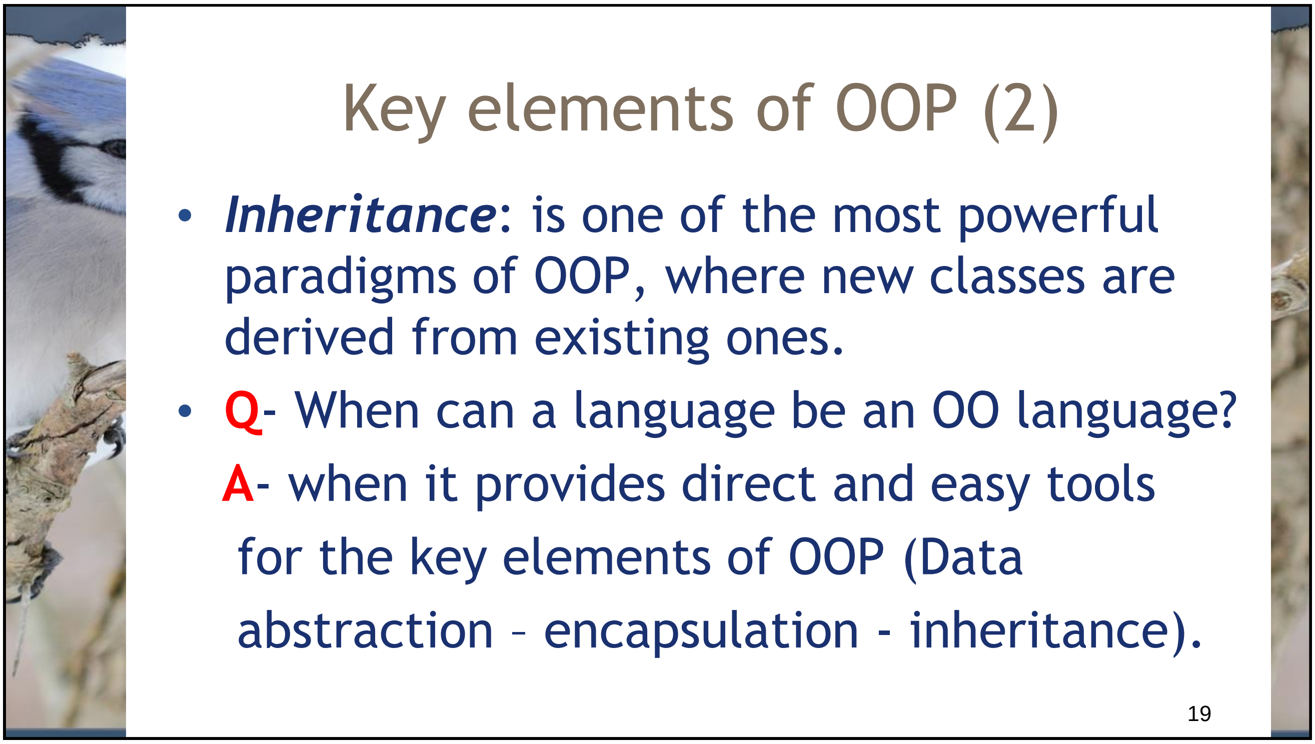
17



Key elements of OOP (1)

- ***Data abstraction***: an abstraction (class) focuses on the outside view of an object and separates its essential behaviour form the internal implementation details
- ***Encapsulation*** (or ***information hiding***): it is the result of hiding the internal implementation details of an abstraction. It separates interface from implementation.

18

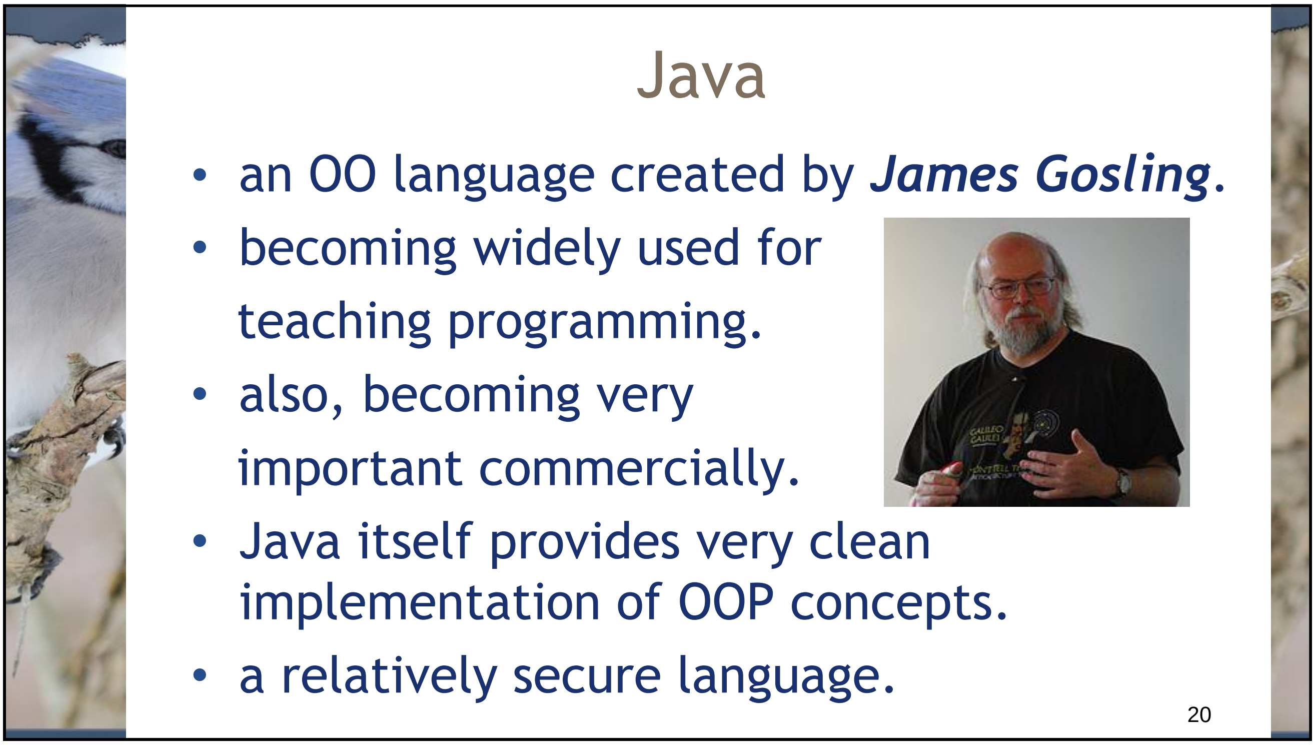


Key elements of OOP (2)

- ***Inheritance***: is one of the most powerful paradigms of OOP, where new classes are derived from existing ones.
- **Q**- When can a language be an OO language?

  A- when it provides direct and easy tools for the key elements of OOP (Data abstraction – encapsulation - inheritance).

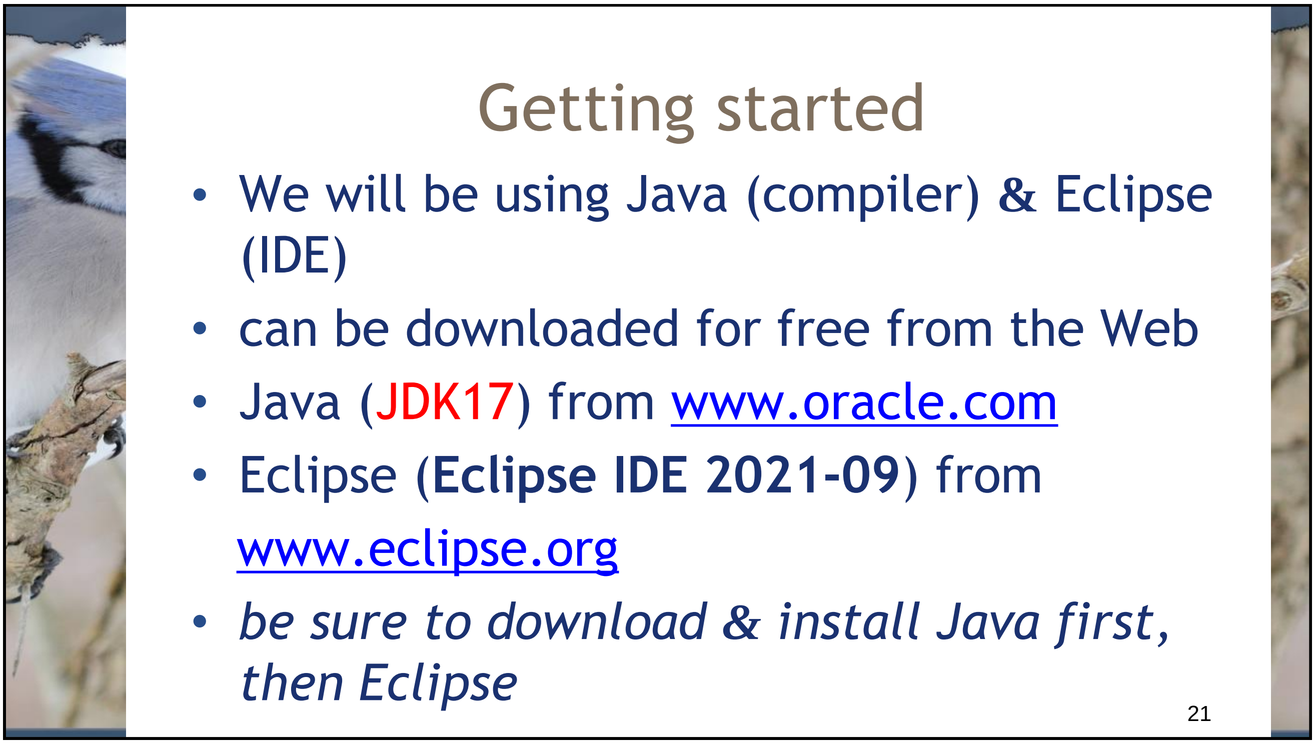
19



Java

- an OO language created by ***James Gosling***.
- becoming widely used for teaching programming.
- also, becoming very important commercially.
- Java itself provides very clean implementation of OOP concepts.
- a relatively secure language.

20



Getting started

- We will be using Java (compiler) & Eclipse (IDE)
- can be downloaded for free from the Web
- Java (JDK17) from www.oracle.com
- Eclipse (**Eclipse IDE 2021-09**) from www.eclipse.org
- *be sure to download & install Java first, then Eclipse*

21