

20 Programming paradigms – Part 1

Name: _____ Class: _____ Date: _____

Vocabulary 词汇

Write each word three times. 把每个单词抄写三遍。

English	中文	Write it 3 times (English)		
programming paradigm	编程范式	_____	_____	_____
imperative programming	命令式编程	_____	_____	_____
object-oriented programming	面向对象编程	_____	_____	_____
object	对象	_____	_____	_____
instances	实例	_____	_____	_____
class	类	_____	_____	_____

Recall

A **programming paradigm** 编程范式 is a style of programming —a way of structuring a program.

New ideas

Read these first. Each idea has a short worked example. Then do the Practice.

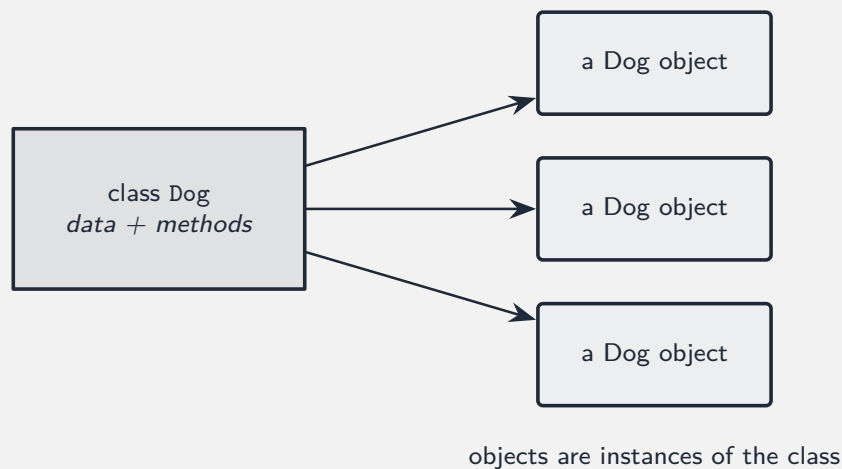
■ 1 Four paradigms



- **imperative programming** 命令式编程—a sequence of commands that change state;
- **object-oriented programming** 面向对象编程—programs built from objects;
- plus low-level and declarative styles.

Worked example. To add up a list, an imperative program writes a loop that adds each item to a running total, step by step. A declarative style instead just says "give me the sum of the list" and lets the language work out how.

■ 2 Classes and objects



In OOP, an **object** 对象 combines data and methods. Objects are **instances** 实例 of a **class** 类 (the template).

Worked example. A Dog class is a template with data (name, age) and a method (**bark**). **rex** and **bella** are two objects made from it —each with its own name and age, but the same methods.

Watch out: a class is only the *template* (the blueprint); an object is a specific *instance* built from it. One class can create many objects.

Practice

Try these in order.

20.1 What is a *programming paradigm*? [1]

20.2 What does *imperative* programming use? [1]

20.3 What is an *object* made of? [1]

20.4 What is a *class*? [1]

20.5 What is the relationship between an object and a class? [1]

20.6 A program needs 100 students, each with a name and grade. Explain why one `Student` class with 100 objects is better than writing 100 separate sets of variables. [2]

20.7 Challenge. Describe a `Car` class (its data and one method) and name two objects you could create from it, each with different data. [2]
