

# Exercise walk-through

Abstraction and Program Design ■ 3.1

eXercise

## You must be able to

- describe **abstraction** 抽象 as hiding details to focus on the main idea
- explain how **procedural abstraction** 过程抽象 lets code use a method by what it does
- break a large problem into smaller behaviours using **method decomposition** 方法分解

## Method — Abstraction

**Abstraction** hides details so you can focus on the main idea of a problem.

## Method — Procedural abstraction

Lets you use a method by **what it does**, not **how** it does it — you can call `Math.sqrt` without knowing its internal algorithm.

## Method — Method decomposition

Break a large problem into smaller, well-named methods, each handling one behaviour.

## Practice 2.1 ■ 1 mark

Define abstraction.

## Solution 2.1

### Worked steps

hiding details to focus on the main idea of a problem **B1**.

## Practice 2.2 ■ 1 mark

State what procedural abstraction lets you do.

## Solution 2.2

Method: Procedural abstraction

### Worked steps

use a method by what it does, without knowing how it works inside **B1**.

## Practice 2.3 ■ 1 mark

State what method decomposition means.

## Solution 2.3

Method: Method decomposition

### Worked steps

breaking a large problem into smaller methods, each handling one behaviour **B1**.

## Exam-style 3.1 ■ 1 mark

Abstraction means

- A** adding more detail
- B** hiding detail to focus on the main idea
- C** running faster
- D** a syntax error

## Solution 3.1

- A adding more detail
- B hiding detail to focus on the main idea** 
- C running faster
- D a syntax error

### Worked steps

B.

## Exam-style 3.2 ■ 1 mark

Using a method by what it does, not how, is

- A** overloading
- B** procedural abstraction
- C** casting
- D** recursion

## Solution 3.2

Method: Procedural abstraction

A overloading

B procedural abstraction



C casting

D recursion

### Worked steps

B.

## Exam-style 3.3 ■ 1 mark

A large program is split into small methods, each doing one task.

- (a) Name this approach.
- (b) State one benefit.
- (c) Name the idea of hiding detail.

## Solution 3.3

Method: Method decomposition

### Worked steps

- (a) method decomposition **B1**. (b) easier to read, reuse, or test (any one) **B1**.  
(c) abstraction **B1**.