

Exercise walk-through

Developing Procedures ■ 3.13

eXercise

You must be able to

- **define** a procedure with a name, parameters, and body
- use **parameters** 参数 so a procedure works with different inputs
- apply **procedural abstraction** 过程抽象 to break a problem into parts
- explain how procedures reduce **duplication** 重复

Method — Defining a procedure

Give it a **name**, a list of **parameters**, and a **body** of instructions.

Method — Parameters

Parameters let one procedure work with **different input values** each time it is called.

Method — Procedural abstraction

Breaking a large problem into smaller, well-named procedures reduces **duplication** and improves readability and reuse.

Practice 2.1 ■ 2 marks

State the three parts of a procedure definition.

Solution 2.1

Worked steps

a name, parameters, and a body **B1** **B1** *for any two; for all three.*

Practice 2.2 ■ 1 mark

State the purpose of a parameter.

Solution 2.2

Worked steps

to let the procedure work with different input values **B1**.

Practice 2.3 ■ 1 mark

State one benefit of using procedures.

Solution 2.3

Worked steps

any one of: less duplication, easier reuse, better readability **B1**.

Exam-style 3.1 ■ 1 mark

Breaking a large problem into smaller procedures is

- A** iteration
- B** procedural abstraction
- C** compression
- D** an event

Solution 3.1

Method: Procedural abstraction

A iteration

B procedural abstraction



C compression

D an event

Worked steps

B.

Exam-style 3.2 ■ 1 mark

A parameter lets a procedure

- A** run only once
- B** work with different input values
- C** never return anything
- D** store metadata

Solution 3.2

- A** run only once
- B** work with different input values 
- C** never return anything
- D** store metadata

Worked steps

B.

Exam-style 3.3 ■ 1 mark

A program repeats the same five lines in three different places.

- (a) State how to avoid this duplication.
- (b) State one benefit of doing so.
- (c) Name the general idea involved.

Solution 3.3

Worked steps

(a) put the lines in a procedure and call it in each place **B1**. (b) less duplication (or easier to maintain) **B1**. (c) procedural abstraction **B1**.