Exercise walk-through

Calling Procedures ▪ 3.12

eXercise

You must be able to

- explain how a **procedure** 过程 (function or method) is a named block of reusable code
- **call** a procedure by name to run its instructions
- pass **arguments** 实参 through its parameters
- use a **return value** 返回值 in a later expression

Method — Procedures

A **procedure** (also called a function or method) is a **named block of reusable code**. You **call** it by name to run its instructions from anywhere.

Method — Arguments and return values

Values passed in are **arguments**, received through the procedure's **parameters**. A procedure may produce a **return value** that the caller uses.

Method — Abstraction

Procedures support **abstraction** by hiding their internal detail from the caller: `area(5, 3)` returns 15 without the caller seeing how.

Practice 2.1 ▪ 1 mark

Define a procedure.

Solution 2.1

Worked steps

a named block of reusable code (a function or method) B1.

Practice 2.2 ▪ 1 mark

State how you run a procedure from elsewhere in a program.

Solution 2.2

Worked steps

call it by its name B1.

Practice 2.3 ▪ 1 mark

State what a return value is.

Solution 2.3

Worked steps

a value that a procedure produces and hands back to the caller B1.

Exam-style 3.1 ▪ 1 mark

A procedure is

A a single variable

B a named block of reusable code

C a loop

D a data type

Solution 3.1

A a single variable

B a named block of reusable code ✓

C a loop

D a data type

Worked steps

B.

Exam-style 3.2 ▪ 1 mark

Values passed into a procedure are called

A return values

B arguments

C comments

D lists

Solution 3.2

Method: Arguments and return values

A return values

B arguments ✓

C comments

D lists

Worked steps

B.

Exam-style 3.3 ▪ 1 mark

A procedure `double(n)` returns `n` $\times$ 2.
(a) State the value of `double(6)`.
(b) Name what 6 is when it is passed in.
(c) State how procedures support abstraction.

Solution 3.3

Method: Abstraction

Worked steps

(a) 12 B1. (b) an argument B1. (c) they hide the implementation detail from the caller B1.