

Exercise walk-through

Method Signatures ■ 1.9

eXercise

You must be able to

- describe the parts of a **method signature** 方法签名: the name and parameter list
- distinguish a **parameter** 形参 from an **argument** 实参
- understand that arguments must match the number, order, and types of the parameters
- explain **method overloading** 方法重载

Method — Method signature

A signature is the method's **name** plus its **parameter list**, e.g. `max(int a, int b)`.

Method — Parameter vs argument

A **parameter** appears in the method **header** (`int a`); an **argument** is the value passed in the **call** (`max(3, 5)`). Arguments must match the parameters in number, order, and type.

Method — Overloading

Overloading lets several methods share a name with **different parameter lists**.

Practice 2.1 ■ 2 marks

State the two parts of a method signature.

Solution 2.1

Method: Method signature

Worked steps

the method name and its parameter list **B1** **B1**.

Practice 2.2 ■ 2 marks

State the difference between a parameter and an argument.

Solution 2.2

Method: Parameter vs argument

Worked steps

a parameter is named in the method header; an argument is the value supplied in the call `B1 B1`.

Practice 2.3 ■ 1 mark

In the call `sum(3, 5)`, state whether 3 is a parameter or an argument.

Solution 2.3

Method: Parameter vs argument

Worked steps

an argument **B1**.

Exam-style 3.1 ■ 1 mark

A method signature consists of the method name and its

- A** return value
- B** parameter list
- C** body
- D** comments

Solution 3.1

Method: Method signature

A return value

B parameter list



C body

D comments

Worked steps

B.

Exam-style 3.2 ■ 1 mark

Two methods with the same name but different parameter lists is called

A overriding

B overloading

C casting

D recursion

Solution 3.2

Method: Overloading

A overriding

B overloading 

C casting

D recursion

Worked steps

B.

Exam-style 3.3 ■ 1 mark

A method `area(int w, int h)` is called with `area(4, 6)`.

- (a) Name `w` and `h` in the header.
- (b) Name 4 and 6 in the call.
- (c) State whether `area(4)` is a valid call.

Solution 3.3

Method: Parameter vs argument

Worked steps

(a) parameters **B1**. (b) arguments **B1**. (c) no — it has the wrong number of arguments **B1**.