

1.9 Method Signatures

Name: _____ Class: _____ Date: _____

Total: 10 marks

KEY WORDS parameter 形参 • argument 实参 • method signature 方法签名
• method overloading 方法重载 • type 类型

Objective

Build the skills to answer exam questions on **method signatures**.

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

1 Worked examples

Study these first. Each one shows the method for a question type used later.

■ Method signature

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

■ 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.

■ Overloading

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

2 Practice

2.1 State the two parts of a method signature.

[2]

2.2 State the difference between a parameter and an argument. [2]

2.3 In the call `sum(3, 5)`, state whether 3 is a parameter or an argument. [1]

3 Exam-style questions

3.1 A method signature consists of the method name and its [1]

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

3.2 Two methods with the same name but different parameter lists is called [1]

- | | |
|---|---|
| A | B |
| C | D |
- A** overriding
 - B** overloading
 - C** casting
 - D** recursion

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

- (a) Name `w` and `h` in the header. [1]
- (b) Name 4 and 6 in the call. [1]
- (c) State whether `area(4)` is a valid call. [1]