

1.14 Calling Instance Methods

Name: _____ Class: _____ Date: _____

Total: 9 marks

KEY WORDS instance method 实例方法 • nullpointerexception 空指针异常 • instance 实例
• class 类 • object 对象

Objective

Build the skills to answer exam questions on **calling instance methods**.

You must be able to:

- call an **instance method** 实例方法 with `objectName.methodName(arguments)`
- understand that it acts on the **state** of that object
- explain why calling a method on a **null** reference causes a **NullPointerException** 空指针异常
- distinguish calling an instance method from a class method

1 Worked examples

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

■ Instance method call

Call it on an object: `objectName.methodName(arguments)`. It acts on the **state** of the object it is called on, and its return value can be used in a larger expression.

■ The null trap

Calling a method on a reference that is **null** causes a **NullPointerException**.

■ Instance vs class method

An **instance** method is called on an **object**; a **class** (**static**) method is called on the **class**.

2 Practice

2.1 State the syntax to call an instance method.

[1]

2.2 State what happens if you call a method on a `null` reference. [1]

2.3 State the difference between an instance method and a class method. [2]

3 Exam-style questions

3.1 An instance method is called on [1]

- | | |
|---|---|
| A | B |
| C | D |
- A** the class
 - B** an object
 - C** a primitive value
 - D** the compiler

3.2 Calling a method on a `null` reference throws a [1]

- | | |
|---|---|
| A | B |
| C | D |
- A** syntax error
 - B** `NullPointerException`
 - C** overflow error
 - D** nothing at all

3.3 `String s = "hi";`

- (a) State the value of `s.length()`. [1]
- (b) State whether `length` is an instance or a class method. [1]
- (c) State what `s = null; s.length();` causes. [1]