

3 Scope, methods, and references — Part 2

Name: _____ Class: _____ Date: _____

Vocabulary

Write each word three times.

English

Write it 3 times (English)

Scope

Recall

In Part A you built a class. Now two ideas about how variables and objects behave:

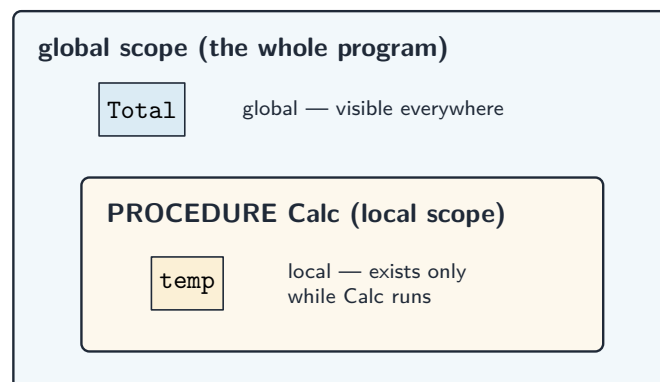
- A variable made inside a method only exists there.
- Passing an object into a method can change the original.

Each idea has a worked example before the Practice.

New ideas

■ 1 Scope

Scope is where a name can be used. A variable declared inside a method (a **local** variable) exists **only** inside that method.



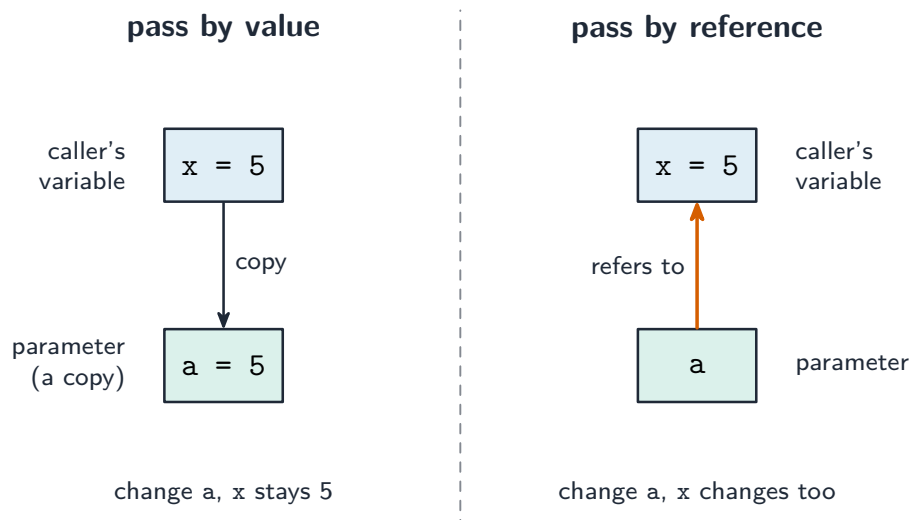
■ 2 Methods return values

A method can compute a value and **return** it:

```
public int doubleIt(int x) {  
    return x * 2;  
}
```

■ 3 Passing objects

When you pass an **object** to a method, Java passes a **reference** to it, so changes to its data are seen by the caller.



■ 4 Passing primitives

When you pass a **primitive** (like an `int`), Java passes a **copy**, so changing it inside the method does **not** affect the caller.

Watch out: changing an object's fields inside a method **does** affect the caller (it shares the object), but changing a copied `int` does **not**. Objects are shared; primitives are copied.

Practice

Try these in order. The methods are all above.

2.1 State what "scope" means. [1]

2.2 State where a local variable can be used. [1]

2.3 For `doubleIt` above, state what `doubleIt(6)` returns.

[1]

2.4 State whether changing an object's field inside a method affects the caller, and why.

[2]

2.5 State whether changing a passed `int` inside a method affects the caller, and why.

[2]