

Programming

AS Computer Science ■ Topic 11

A-Level Computer Science



Programming

What we will cover

- 1 Programming basics
- 2 Selection
- 3 Iteration
- 4 Procedures & functions
- 5 Efficient code
- 6 Recap

```
1 //age;
2 public Hotel (Integer capacity) { this.capacity = capacity; }
3 //age;
4 public synchronized boolean checkIn(AccommodationRequest request) {
5     if (guests.size() < capacity) {
6         guests.add(request);
7         return true;
8     }
9     else {
10        return false;
11    }
12 }
```

1

Basics

From design to code

Turn a **flowchart** 流程图 or structured English into **pseudocode** 伪代码, then into a language.

input or output box	→ INPUT / OUTPUT
decision diamond	→ IF...ELSE...ENDIF
loop arrow	→ WHILE / FOR / REPEAT
process box	→ assignment or calculation

Then trace a small input to check. Find the variables and their types first.

Design, then code

flowchart symbol	pseudocode
input / output	INPUT / OUTPUT
decision	IF...THEN or CASE
process	x = expression
loop arrow	WHILE / FOR / REPEAT

Structure diagram, then pseudocode, then a language — each step commits to a little more detail, and only the last one is language-specific.

Variables, operators, functions

- **constant** 常量 fixed; **variable** 变量 changes
- assign with ←; DIV, MOD
- **precedence** 优先级: NOT > × / > + - > compare > AND > OR

```

s = 1 2 3 4 5 6 7 8
   C O M P U T E R
LENGTH(s) = 8    RIGHT(s, 2) = "ER"
LEFT(s, 3) = "COM"  CASE(s) = "COMPUTER"
MID(s, 4, 3) = "PUT"  CASE(s) = "computer"

```

library routines 库例程: LEFT, MID, LENGTH

Assignment is not equality

`count = count + 1`

`count = 7`

variable
(can change)

`Pi = 3.14159`

constant
(fixed)

The right-hand side is **evaluated first**, then stored in the variable named on the left — which is why `x ← x + 1` makes sense.

Worked example: build a username

```
Name ← "TURING"  
Year ← "1912"  
User ← LEFT(Name,3)  
       & RIGHT(Year,2)
```

What ends up in User?

`LEFT("TURING",3) = "TUR"`

`RIGHT("1912",2) = "12"`

joined by &

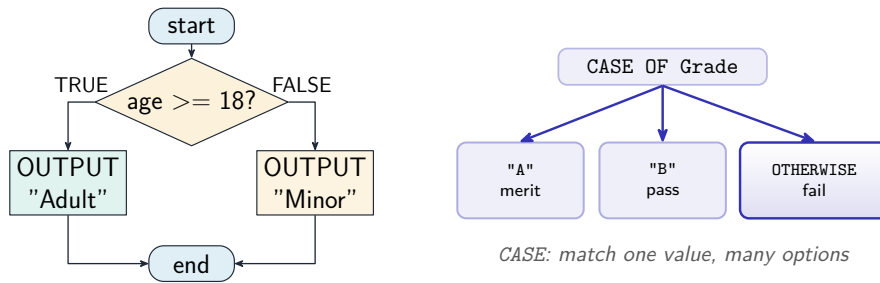
`User = "TUR12"`

String positions count from 1. LEFT/RIGHT/MID cut pieces; & joins them.

2

Selection

Selection



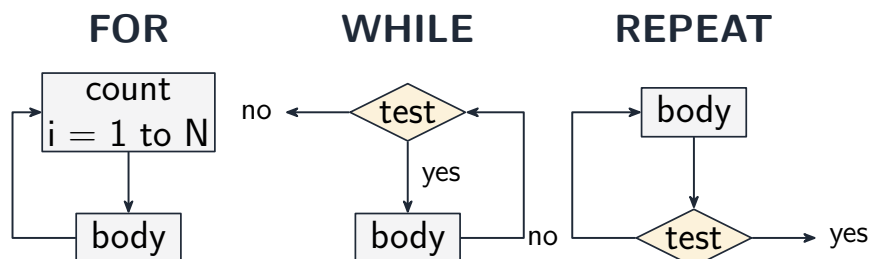
IF/ELSE: one branch runs

Use CASE instead of deep **nested** 嵌套 IFs when testing one value against many.

3 Iteration

Three loops

- FOR – count known up front
- WHILE – test first, may run **zero** times
- REPEAT – test after, runs **at least once**



4

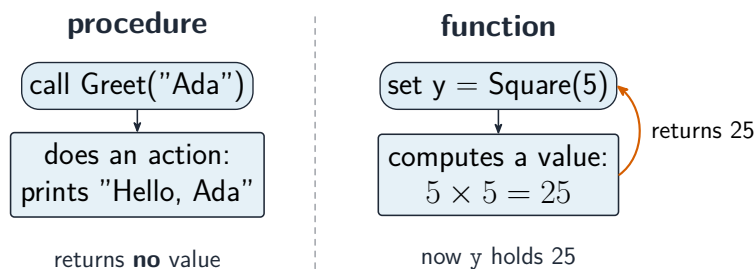
Procedures and functions

Programming
└ Procedures and functions

Procedures and functions

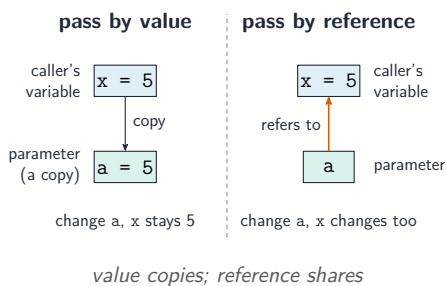
- **procedure** 过程 – does an action, returns nothing
- **function** 函数 – returns a value for an expression

Named **subroutines** 子程序 make **structured programming** 结构化编程.



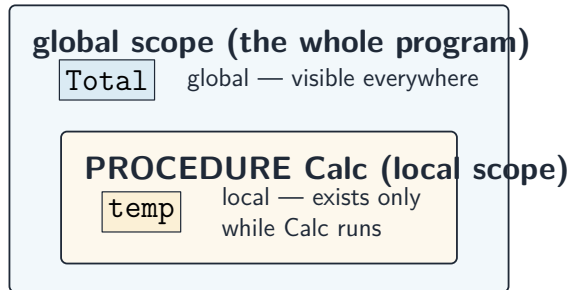
Programming
└ Procedures and functions

Parameters and scope



- **by value** 传值 – a copy; caller unchanged
- **by reference** 传引用 – can change the caller's variable
- prefer **local** 局部变量 over **global** 全局变量 (scope 作用域)

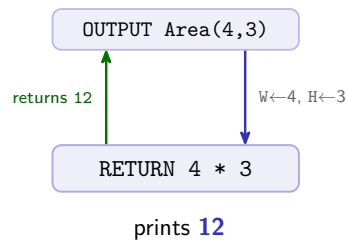
Where a name is visible



A **local** 局部 name exists only inside its routine; a **global** 全局 one is visible everywhere, which is exactly why it is easy to break by accident.

Worked example: a function that returns

```
FUNCTION Area(W : INTEGER,  
             H : INTEGER) RETURNS INTEGER  
  RETURN W * H  
ENDFUNCTION  
OUTPUT Area(4, 3)
```



A function call sits **inside an expression** because it returns a value; a procedure call is a statement on its own (CALL).

5 Efficiency

Writing efficient pseudocode

slower

```
FOR i = 1 TO n
  x = slow() // every pass
  use(x, i)
NEXT i
```

faster

```
x = slow() // once
FOR i = 1 TO n
  use(x, i)
NEXT i
```

- move **invariants** 不变量 out of loops
- exit a search **early** once found
- meaningful names; initialise before use
- comment the **intent**, not the mechanics

Work that does not belong in the loop

slower

```
FOR i = 1 TO n
  x = slow() // every
  pass
  use(x, i)
NEXT i
```

faster

```
x = slow() // once
FOR i = 1 TO n
  use(x, i)
NEXT i
```

Anything whose value cannot change should be computed **once**, **before** the loop.

6

Recap

Topic 11 – key takeaways

1 Selection

IF/ELSE; CASE for one value,
many options

2 Iteration

FOR / WHILE (zero) / REPEAT
(at least one)

3 Subroutines

procedure acts; function returns a
value

4 Parameters

by value copies; by reference
shares

Check yourself

- 1 Which loop may run zero times?
- 2 Procedure or function: which returns a value?
- 3 Which parameter passing changes the caller's variable?
- 4 $7 \text{ MOD } 2 = ?$

```
{  
_____  
_____  
_____  
}
```

Answers 1. WHILE 2. a function 3. by reference 4. 1