

Programming

AS Computer Science ■ Topic 11

A-Level Computer Science

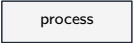


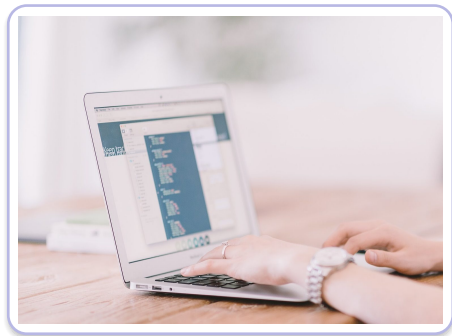
What we will cover

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

```
1 usage
public Hotel (Integer capacity) { this.capacity = capacity; }
1 usage
public synchronized boolean checkIn(AccommodationRequest request) {
    if (guests.size() < capacity) {
        guests.add(request);
        return true;
    }
    else {
        return false;
    }
}
```

From design to code

flowchart symbol		pseudocode
	→	INPUT / OUTPUT
	→	IF...THEN or CASE
	→	x = expression
	→	WHILE / FOR / REPEAT



programmer

Variables, operators, functions

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

s =

1	2	3	4	5	6	7	8
C	O	M	P	U	T	E	R

LENGTH(s) = 8

LEFT(s, 3) = "COM"

MID(s, 4, 3) = "PUT"

RIGHT(s, 2) = "ER"

UCASE(s) = "COMPUTER"

LCASE(s) = "computer"

library routines 库例程: LEFT, MID,
LENGTH

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"

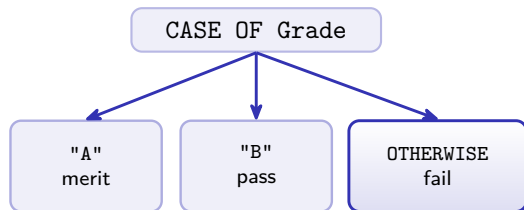
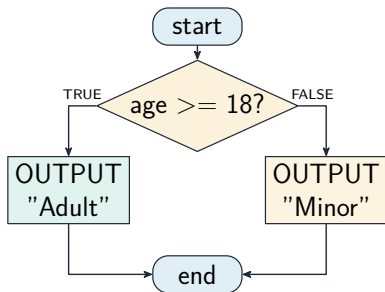
RIGHT("1912",2) = "12"

joined by &

User = "TUR12"

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

Selection

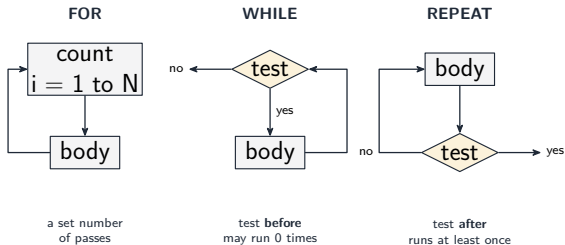


CASE: match one value, many options

IF/ELSE: one branch runs

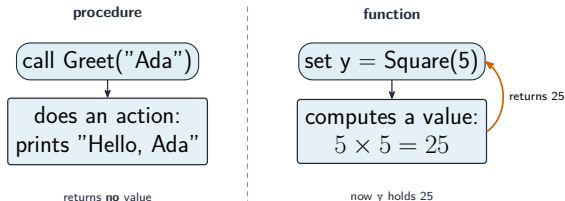
Use CASE instead of deep **nested** 嵌套 IFs when testing one value against many.

Three loops



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

Procedures and functions

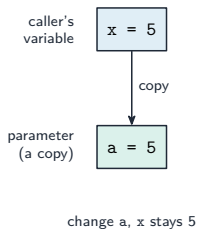


- **procedure** 过程 – does an action, returns nothing
- **function** 函数 – returns a value for an expression

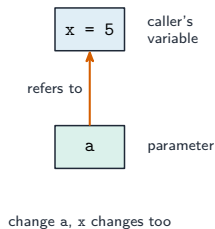
Named **subroutines** 子程序 make **structured programming** 结构化编程.

Parameters and scope

pass by value



pass by reference

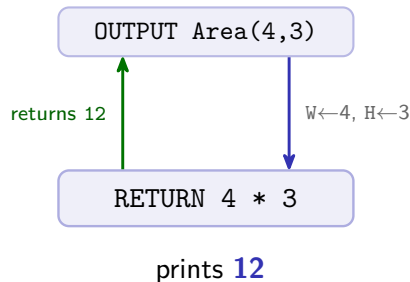


value copies; reference shares

- **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)

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).

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

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