

8.1 Programming concepts

Name: _____ Class: _____ Date: _____

Total: 36 marks

KEY WORDS variable 变量 • output 输出 • input 输入 • function 函数 • procedure 过程

Objective

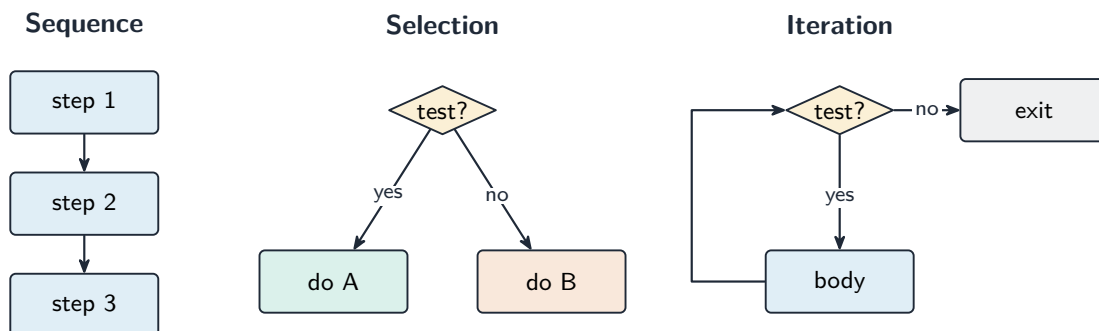
Build the skills to answer exam questions on **programming concepts** 编程概念 — variables, data types, the three control structures (**sequence**, **selection**, **iteration** 顺序、选择、迭代), operators, and procedures/functions.

You must be able to:

- declare variables/constants and use the five data types
- use sequence, selection (IF/CASE) and iteration (FOR/WHILE/REPEAT)
- use operators (MOD, DIV) and write procedures/functions

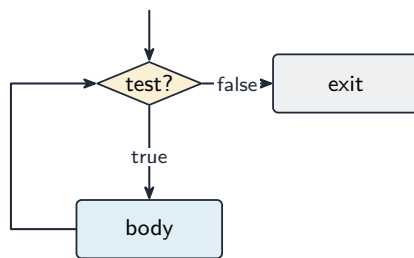
1 Worked examples

■ Control structures



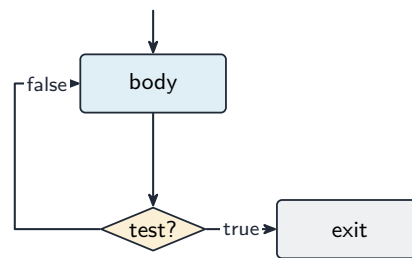
Sequence runs steps in order; selection chooses a branch; iteration repeats a body

WHILE (pre-condition)



tested first $\Rightarrow$ may
run **zero** times

REPEAT (post-condition)



tested last $\Rightarrow$ runs
at least once

A WHILE loop may run 0 times; a REPEAT loop runs at least once

- $17 \text{ MOD } 5 = 2$ (remainder); $17 \text{ DIV } 5 = 3$ (whole part).

■ Declaring, choosing and repeating

Declarations. A **variable** 变量 has a value that can change while the program runs; a **constant** 常量 is set once and never changes. In Cambridge pseudocode:

```
DECLARE Total : INTEGER
DECLARE Name : STRING
CONSTANT VAT = 0.20
```

The five data types are **INTEGER** (whole numbers), **REAL** (numbers with a decimal part), **CHAR** (one character), **STRING** (text) and **BOOLEAN** (**TRUE** or **FALSE**).

Selection. **IF ... THEN ... ELSE ... ENDIF** chooses between two paths; **CASE OF ... OTHERWISE ... ENDCASE** is neater when one variable is tested against many values.

Iteration, and how to choose:

Loop	When to use it	Runs at least once?
FOR ... TO ... NEXT	the number of repeats is known (count-controlled)	no, if the range is empty
WHILE ... DO ... ENDWHILE	the condition is tested before the body	no
REPEAT ... UNTIL	the condition is tested after the body	yes

Worked example (count-controlled total).

```

DECLARE Total : INTEGER
DECLARE Mark  : INTEGER
DECLARE Count : INTEGER
Total ← 0
FOR Count ← 1 TO 10
    OUTPUT "Enter mark"
    INPUT Mark
    Total ← Total + Mark
NEXT Count
OUTPUT "The total is ", Total

```

Every mark in a “write pseudocode” question is earned by one of these: declaring and initialising, a correctly formed loop with the right count, the input inside the loop, the running total updated correctly, and a sensible output after the loop.

Operators. MOD gives the remainder and DIV the whole number of divisions: 17 MOD 5 is 2 and 17 DIV 5 is 3. Together they split a value, for example turning 137 seconds into 137 DIV 60 = 2 minutes and 137 MOD 60 = 17 seconds.

Procedures and functions. Both are named blocks of code that can be called from anywhere, which avoids repeating code. A **function** 函数 returns a value to the line that called it, so it is used inside an expression; a **procedure** 过程 does not return a value.

2 Practice

2.1 State the difference between a variable and a constant. [2]

2.2 Give the value of $23 \text{ MOD } 4$. [1]

2.3 Give the pseudocode declaration for a variable that stores a student's name, and a constant for a tax rate of 0.2. [2]

2.4 Identify the data type you would use for each of: a price, whether a box is ticked, a single grade letter. [3]

2.5 Explain the difference between a WHILE loop and a REPEAT loop. [2]

3 Exam-style questions

3.1 A loop that always runs **at least once** is a: [1]

- **A** FOR loop
- **B** WHILE (pre-condition) loop
- **C** REPEAT (post-condition) loop
- **D** CASE statement

3.2 A program reads 10 marks and outputs their total.

(a) State which control structure repeats the input 10 times. [1]

(b) Write pseudocode using a count-controlled loop to total 10 input marks. [4]

3.3 State the difference between a procedure and a function. [2]

3.4 A program records the times, in whole seconds, taken by 20 runners.

(a) Complete the pseudocode declarations for the total, one runner's time and the loop counter. [3]

(b) Write pseudocode that inputs the 20 times, adds them to a total, and outputs the average. [6]

(c) The program must also output the fastest time. Describe, in words, the extra steps needed inside the loop. [3]

(d) A time of 137 seconds must be shown as minutes and seconds. Give the two expressions, using MOD and DIV, and state the values they produce. [3]

(e) The conversion in part (d) is needed in three places in the program. Explain why writing it as a function is better than repeating the code, and state what a function does that a procedure does not. [3]

Solutions

2.1 a variable's value can change while the program runs; a constant's value is fixed.

2.2 3.

3.1 C. A REPEAT (post-condition) loop runs at least once.

3.2 (a) iteration (a count-controlled / FOR loop).

(b) `total ← 0; FOR i ← 1 TO 10; INPUT mark then total ← total + mark; NEXT i then OUTPUT total.`

3.3 a function returns a value to where it was called; a procedure does a task but returns no value.

2.3 `DECLARE Name : STRING; CONSTANT TaxRate = 0.2.`

2.4 `Price: REAL;` whether a box is ticked: `BOOLEAN;` a single grade letter: `CHAR.`

2.5 A WHILE loop tests the condition before the body, so the body may run zero times; a REPEAT loop tests after the body, so it always runs at least once.

3.4 (a) `DECLARE Total : INTEGER, DECLARE Time : INTEGER, DECLARE Count : INTEGER.`

(b) `Total ← 0` before the loop; `FOR Count ← 1 TO 20 with NEXT Count;` `INPUT Time` inside the loop; `Total ← Total + Time` inside the loop; `Average ← Total / 20` after the loop; `OUTPUT Average.`

(c) Set a variable, say `Fastest`, to the first time (or to a very large value) before the loop; inside the loop compare each new time with `Fastest`; if the new time is smaller, replace `Fastest` with it, and output `Fastest` after the loop.

(d) `137 DIV 60` gives 2 minutes; `137 MOD 60` gives 17 seconds; so the time is displayed as 2 minutes 17 seconds.

(e) The code is written once and called three times, so the program is shorter and a correction only has to be made in one place; it is also easier to read and test. A function returns a value to the line that called it, so it can be used inside an expression, while a procedure does not return a value.