MODULE M · OCR J277 2.1.2, 2.2.1-2.2.3, 2.3.1 · CAMBRIDGE 0478 8.1-8.3

Programming

Variables, **data types**, **operators**, **selection**, **loops**, **nesting**, **strings**, **arrays**, **files**, **sub programs**, **scope**, **random numbers**, **records** and **maintainable code**. **Cambridge 0478: every card except M27**, without the GCSE-only rows. OCR code is Exam Reference Language (counts from 0); Cambridge code is pseudocode (usually counts from 1). Spec versions as checked in November 2026.

01 Variables, constants, I/O (M1) · both boards

Variable Named memory location; value can change while the program runs.

Constant Named value that cannot change; easier to read and to update in one place.

Assignment Right side worked out, then stored on the left; overwrites the old value.

GCSE only `const maxSeats = 52`; `name = input("Name")`; `print(...)`; no declarations.

IGCSE only `DECLARE Name : STRING`; `CONSTANT MaxSeats ← 52` (literal only); `OUTPUT "Name"` then `INPUT Name`.

03 Data types, casting (M3) · both boards

Types Integer 52; real 3.75; Boolean TRUE/FALSE; character 'M'; string "07700 900123".

Choose Phone number = string: keeps the leading 0, no arithmetic.

GCSE only Casting: `int("3")`, `str(345)`, `float()`, `real()`, `bool()`.

IGCSE only INTEGER, REAL, CHAR, STRING, BOOLEAN; REAL as 4.0; CHAR in ' '; no casting.

05 Arithmetic operators (M5) · both boards

Operators + - * / ^. 10 / 4 = 2.5; 2 ^ 3 = 8. * and / before + and -.

MOD, DIV Remainder and whole-number quotient: 17 MOD 5 = 2; 17 DIV 5 = 3.

Uses x MOD 2 = 0 means even. 200 min = 200 DIV 60 = 3 h, 200 MOD 60 = 20 min.

Notation GCSE only `17 MOD 5` IGCSE only `MOD(17, 5)`, `DIV(17, 5)`

06 Comparison operators (M6) · both boards

Result Always a Boolean, TRUE or FALSE. Used in IF and loop conditions.

Both `< <= > >=`. `>= 50` includes 50; `> 50` does not.

GCSE only Equal `==`, not equal `!=` (single = assigns).

IGCSE only Equal `=`, not equal `<>` (← assigns). Called relational operators.

07 AND, OR, NOT (M7) · both boards

AND TRUE only when both are TRUE.

OR TRUE when at least one is TRUE.

NOT Reverses: NOT TRUE = FALSE.

Range In: `r >= 1 AND r <= 10`. Out: `r < 1 OR r > 10`. Full comparison each side.

08 IF, ELSE (M8) · both boards

Selection A condition decides which block runs; only the first TRUE branch runs.

Order Test the narrowest range first: `<= 2` before `<= 10`.

GCSE only `if ... then` / `elseif ... then` / `else` / `endif`

IGCSE only `IF` / `THEN` / `ELSE` / `ENDIF`; no ELSE IF: nest an IF in the ELSE.

09 CASE / switch (M9) · both boards

Use One variable against fixed values; first match runs, then it is finished.

Default default / OTHERWISE comes last; runs if nothing matches.

GCSE only `switch c: case "A": default: endswitch`

IGCSE only `CASE OF C 'A' : OUTPUT ... OTHERWISE ... ENDCASE`

10 FOR loops (M10) · both boards

Count-controlled Repeats a set number of times, known before the loop starts.

Inclusive 0 to 9 and 1 to 10 both run 10 times. 3 to 15 step 4: 3, 7, 11, 15.

GCSE only `for i = 10 to 0 step -1` … `next i`

IGCSE only `FOR I ← 1 TO 30` … `NEXT I`; optional STEP

11 WHILE loops (M11) · both boards

Checked first Repeats while TRUE; can run zero times.

Ending The body must change the condition, or it is an infinite loop. Give variables a value first.

GCSE only `while x >= 300` … `endwhile`; "condition-controlled".

IGCSE only `WHILE X >= 300 DO` … `ENDWHILE`; "pre-condition".

12 Post-condition loops (M12) · both boards

Checked last Body always runs at least once; stops when the condition is TRUE.

Choose Known repeats: FOR. May run 0 times: WHILE. At least once (validation): post-condition.

GCSE only `do` … `until x >= 1 AND x <= 8`

IGCSE only `REPEAT` … `UNTIL X >= 1 AND X <= 8`

14 Nesting (M14) · both boards

Nested IF Inner condition tested only when the outer branch is taken.

Nested loops Inner loop runs fully for each outer pass: 3 × 4 = 12 runs.

Layout Each structure has its own end keyword; indent each level.

IGCSE only You write at most three levels of nesting.

16 String handling (M16) · both boards

Both Length (spaces count), substring (start, number of characters), upper, lower.

GCSE only From 0: `s.length`, `s.substring(5, 5)`, `s.left(3)`, `s.right(3)`, `s.upper`, `+`, `ASC("A")` = 65, `CHR(97)` = "a".

IGCSE only Generally from 1: `LENGTH(S)`, `SUBSTRING(S, 6, 5)`, `UCASE(S)`, `LCASE(S)`.

Example "Lighthouse": length 10; OCR substring(5, 5) = Cambridge SUBSTRING(..., 6, 5) = "house".

18 1D arrays (M18) · both boards

Array Fixed length; one data type; one identifier; elements reached by index.

Loop Use a FOR loop with the index as the loop variable to read or write every element.

GCSE only `array temps[5]`: indexes 0 to 4. `array c = ["Blue", "Pink"]`.

IGCSE only `DECLARE T : ARRAY[1:5] OF INTEGER`; first index generally 1.

19 2D arrays (M19) · both boards

2D array Rows and columns; two indexes, usually [row, column].

Traverse Nested loops: rows outside, columns inside. One row or column: one loop.

GCSE only `array sales[3,4]`; `sales[1, 2]`. Emulates a database table (row = record).

IGCSE only `ARRAY[1:3, 1:4] OF INTEGER`; same element is `Sales[2, 3]`.

21 File handling (M21) · both boards

Steps Open, then read and/or write, then close. Data stays after the program ends.

GCSE only `f = open("v.txt")`, `f.readLine()`, `f.writeLine(...)` adds to the END, `f.endOfFile()`, `f.close()`, `newFile()`.

IGCSE only `OPENFILE "V.txt" FOR READ` / `FOR WRITE` (new file: old data lost), `READFILE`, `WRITEFILE`, `CLOSEFILE`. One mode at a time.

22 Procedures (M22) · both boards

Procedure Named sub program that does a task and returns no value.

Call Its code runs, then control returns to the line after the call.

Benefits Written once, reused; easier to read, test, debug and maintain.

Call it GCSE only `showBanner()` IGCSE only `CALL ShowBanner`

23 Functions (M23) · both boards

Function Sub program that returns one value to where it was called.

Use Inside an expression: `area = roomArea(4.5, 3.2)` stores 14.4.

GCSE only `function f(x)` … `return v` … `endfunction`

IGCSE only `FUNCTION F(X : REAL) RETURNS REAL`; never use CALL with a function.

24 Parameters (M24) · both boards

Parameter Variable in the definition that receives a value.

Argument Actual value in the call; matched by position: number, order, type.

GCSE only No types: `procedure greet(name, room)`. Arrays can be passed and returned.

IGCSE only Typed: `PROCEDURE Greet(N : STRING, R : INTEGER)`; up to three.

25 Local and global (M25) · both boards

Scope Where in the program a variable can be used.

Local Declared in a sub program; used only there, only while it runs. Self-contained; names reusable.

Global Declared in the main program; used anywhere, but anything can change it.

GCSE only `global feeRate = 0.1`; local and global constants too.

IGCSE only No keyword: DECLARE inside a procedure or function makes it local.

26 Random numbers, library routines (M26) · both boards

Both Generate a random number in a range, store it, use it (dice, games).

GCSE only `random(1, 6)`: integer 1 to 6 inclusive; `random(-1.0, 10.0)`: real.

IGCSE only Library routine: pre-written, tested, called by name. MOD, DIV, ROUND, RANDOM.

IGCSE only `RANDOM()`: 0 to 1. `ROUND(RANDOM() * 5, 0) + 1`: 1 to 6. `ROUND(7.456, 2)` = 7.46.

27 Records (M27) GCSE only

Record Groups related fields about one item, e.g. one book.

Field One item of data in a record, with its own name and data type.

vs array Fields can be different types; an array holds one type.

Access Record and field: `book.pages`. ERL has no record keyword.

28 Maintainable code (M28) · both boards

Names Meaningful identifiers for variables, constants, arrays, sub programs.

Comments `//` explain what and why; ignored by the translator.

Sub programs Named, reusable parts that can be tested separately.

GCSE only Indentation and naming conventions (camelCase) are listed too.

01
OCR compares with `==`; Cambridge assigns with ← and compares with `=`.

02
Each side of AND/OR needs a **full comparison**. "Or more" means `>=`.

03
WHILE runs **while** TRUE; UNTIL stops **when** TRUE. Validate with a **loop**, not one IF.

04
OCR arrays and strings start at **0**; Cambridge generally at **1**. Functions **return**, not print.