

A-Level 9618: what changes

Ref Pseudocode

Layout and operators

Cambridge publishes **two** pseudocode specifications, and they disagree. Everything in topics 1-10 is **IGCSE 0478**. A-Level **9618** changes the details below — and a student who studies both loses marks by writing one in the other's paper.

	IGCSE 0478	A-Level 9618
THEN	on its own line , indented 2	on the IF line
constants	CONSTANT Pi $\leftarrow$ 3.142	CONSTANT Pi = 3.142
DIV / MOD	functions: DIV(10, 3)	operators: 10 DIV 3
joining strings	commas in OUTPUT	the & operator
identifiers	must start with a capital	mixed case allowed
NEXT	must name its counter	good practice
ROUND, RANDOM()	both published	neither exists

This is the 0478 form, and it runs here:

```
DECLARE Total : REAL
CONSTANT Rate  $\leftarrow$  0.5
Total  $\leftarrow$  10
IF MOD(7, 2) = 1
    THEN
        OUTPUT "Odd, half rate ", Total * Rate
ENDIF
```

The 9618 form of the same algorithm is written like this — note THEN on the IF line, = in the CONSTANT, and MOD between its two values:

```
DECLARE Total : REAL
CONSTANT Rate = 0.5
Total  $\leftarrow$  10
IF 7 MOD 2 = 1 THEN
    OUTPUT "Odd, half rate " & STR(Total * Rate)
ENDIF
```

△ **Which one is right depends only on which paper you are sitting.** Neither is a mistake; each is a mistake in the other's exam. The playground on this page runs the **0478** rules, so a 9618 construct is refused with a message naming the syllabus it belongs to — which is a quick way to check your own habits.

Common mistakes

- Writing THEN on the IF line in an 0478 answer, because that is what a textbook or a website showed. It is the single commonest cross-syllabus slip.
- 10 MOD 3 in 0478. There it is MOD(10, 3).
- ROUND(3.14159, 2) in a 9618 answer. 9618 has no ROUND at all.

Parameters: BYVAL and BYREF

0478 passes **every** parameter by value (topic 9). 9618 lets you choose, and says so in the header.

- BYVAL — the procedure gets a **copy**. Changing it changes nothing outside. This is the default when neither word is written.
- BYREF — the procedure works on the caller's **own variable**, so a change is visible after the call.

This is 9618 only. It does not run here, and it is not accepted in an 0478 answer:

```
PROCEDURE Increase(BYREF Value : INTEGER)
    Value ← Value + 1
ENDPROCEDURE

DECLARE Count : INTEGER
Count ← 5
CALL Increase(Count)
OUTPUT Count      // prints 6
```

The 0478 way to get a value back

Use a FUNCTION and assign what it returns. This is the whole of the 0478 answer, and it runs:

```
FUNCTION Increased(Value : INTEGER) RETURNS INTEGER
    RETURN Value + 1
ENDFUNCTION

DECLARE Count : INTEGER
Count ← 5
Count ← Increased(Count)
OUTPUT Count
```

Common mistakes

- Writing BYREF in an 0478 answer. The keyword is not in that syllabus.
- Expecting a plain 9618 parameter to behave like BYREF. Without the word it is BYVAL.

Records and other types

9618 has data structures that 0478 does not. You will meet them only at A-Level, and none of them runs under the 0478 rules of this page.

A **record** groups fields of different types under one name (9618 section 4):

```
TYPE Student
    DECLARE Name : STRING
    DECLARE Mark : INTEGER
ENDTYPE

DECLARE Candidate : Student
Candidate.Name ← "Mei"
Candidate.Mark ← 91
OUTPUT Candidate.Name, " scored ", Candidate.Mark
```

An **enumerated type** lists the only values a variable may take:

```
TYPE Vehicle = (Car, Bus, Taxi)
DECLARE MyRide : Vehicle
MyRide ← Taxi
```

9618 also adds a DATE data type, random-access files (OPENFILE ... FOR RANDOM), pointers, and classes with CLASS ... ENDCLASS for its object-oriented paper.

△ 0478 has **five** data types and arrays, and nothing else (topic 2). A 0478 answer that declares a record is answering a different syllabus.

Common mistakes

- Using a record in an 0478 answer. Use parallel arrays instead — one array per field, sharing an index.
- Thinking TYPE means the same as DECLARE. TYPE defines a **new type**; DECLARE creates a **variable** of an existing one.