B2 PROGRAMMING · B2.1

Programming fundamentals

The building blocks of every program: **variables** and data types, **operators**, and the three constructs (sequence, selection, iteration).

01 Variables & data types

Variable	Named store; value can change.
Constant	A fixed value.
integer	Whole numbers.
real	Decimal numbers.
string	Text.
boolean	true or false.

02 Operators

Arithmetic	+ - * /
DIV	Integer division (7 DIV 2 = 3).
MOD	Remainder (7 MOD 2 = 1).
Relational	= != < > <= >=
Logical	AND OR NOT

03 The three constructs

1

Sequence

Statements run one after another, in order.

2

Selection

A condition chooses which path runs (if / else).

3

Iteration

A block repeats while a condition stays true (a loop).

04 Input, process, output

Input	Read from the user or a file.
Process	Apply constructs and operators.
Output	Display the result.
Validate	Check input is sensible first.
Assignment	Store a value, e.g. score = 0.

05 Worked example: count passes

count = 0	Start the counter (sequence).
FOR each mark	Visit every mark (iteration).
IF mark >= 50	Test the condition (selection).
count = count + 1	Add one when it passes.
Result	40, 55, 60, 30 gives count = 2.

06 Know the difference

Assignment vs comparison	Storing a value in a variable versus testing whether two values are equal.	CARE
Selection vs iteration	Choosing a path once versus repeating a block many times.	CONSTRUCTS
DIV vs MOD	The whole-number quotient versus the remainder of a division.	OPERATORS
String vs integer	Text you cannot do maths on versus a number you can ("5" is not 5).	TYPES

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Eight things that lose marks in Paper 1 if you slip on them. Skim before you walk in.

01

The three constructs: **sequence, selection, iteration**. Every program uses them.

02

Assignment stores a value; **comparison** tests equality. Don't confuse them.

03

integer = whole, **real** = decimal, **string** = text, **boolean** = true/false.

04

DIV is the whole-number quotient; **MOD** is the remainder.

05

Selection chooses a path once; **iteration** repeats a block.

06

The string **"5"** is not the number **5**; convert before doing maths.

07

Watch for **off-by-one** errors where a loop runs once too many or too few.

08

Programs follow **input, process, output**; validate input before using it.