

Exercise walk-through

Data Types and Records ■ 10.1

eXercise

You must be able to

- choose the most appropriate **data type** for a value
- explain the purpose of a **record** structure
- write pseudocode to **define** a record and to **read/write** its fields

Method — Choosing a data type

Pick the **smallest precise type** that fits the value:

To store	Best type	Why
a count, an index, an ID	INTEGER	whole number
a price, a measurement	REAL	has a decimal part
a name, a sentence	STRING	text
a single grade letter	CHAR	exactly one character
“is it paid?”	BOOLEAN	a TRUE/FALSE flag

Method — Choosing a data type

Exam method — “give the data type of this expression”. Look at the **operation**, not the inputs:

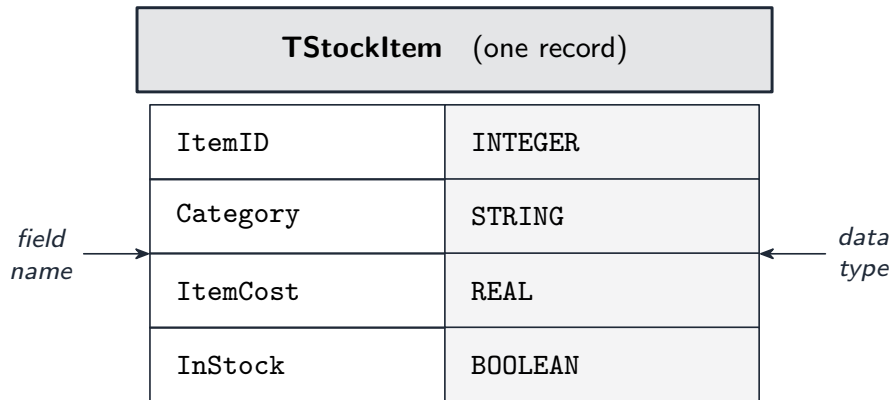
- one character — `LEFT(Name,1)` or `'A'` — gives a **CHAR**
- text in `"..."`, or joined with `&` — gives a **STRING**
- `+` `-` `*` on whole numbers gives an **INTEGER**; any `/` or a decimal value gives a **REAL**
- a comparison (`>=`, `=`) or `AND/OR/NOT` gives a **BOOLEAN**

Method — Defining a record

A **record** groups several fields of **different types** under one type name:

```
TYPE TStockItem
  DECLARE ItemID    : INTEGER
  DECLARE Category  : STRING
  DECLARE ItemCost  : REAL
  DECLARE InStock   : BOOLEAN
ENDTYPE
```

Method — Defining a record



Method — Using a record

Declare a variable of that type (or a whole **array of records**), then use **dot notation** to reach each field:

```
DECLARE Item1 : TStockItem
```

```
DECLARE Items : ARRAY[1:100] OF TStockItem
```

```
Item1.Category ← "Fruit"
```

```
Item1.ItemCost ← 2.50
```

```
OUTPUT Item1.Category, " costs ", Item1.ItemCost
```

Practice 2.1 ■ 3 marks

Give the most appropriate data type for each value: (a) a student's age; (b) a price in dollars; (c) whether a light is on.

Solution 2.1

Method: Choosing a data type

Worked steps

(a) INTEGER; (b) REAL; (c) BOOLEAN **B1** *each*.

Practice 2.2 ■ 1 mark

State the purpose of a **record** structure.

Solution 2.2

Worked steps

It groups **several fields of different data types** under **one identifier**, so values that belong together are stored as one unit **B1**.

Practice 2.3 ■ 3 marks

A record type TBook must store a Title (text), Pages (whole number) and InLibrary (a flag). Write pseudocode to **define** it.

Solution 2.3

Method: Choosing a data type

Worked steps

```
TYPE TBook
  DECLARE Title      : STRING
  DECLARE Pages      : INTEGER
  DECLARE InLibrary  : BOOLEAN
ENDTYPE
```

Solution 2.3

M1 **A1** **A1** *'TYPE...ENDTYPE', three fields, correct types*

Practice 2.4 ■ 1 mark

Book1 is a variable of type TBook. Write one pseudocode line to set its Title to "Maths".

Solution 2.4

Method: Using a record

Worked steps

`Book1.Title ← "Maths"` **B1** — *dot notation + assignment.*

Exam-style 3.1 ■ 4 marks

The table shows four pseudocode assignment statements. Give an appropriate **data type** to declare each variable.

Statement	Data type
A ← LEFT(Name, 1)	
B ← Price * Quantity	
C ← Count + 1	
D ← (Score >= 50)	

Solution 3.1

Worked steps

A → **CHAR** (or **STRING** — one character from **LEFT**); B → **REAL** (a price/decimal);
C → **INTEGER** (whole-number add); D → **BOOLEAN** (a comparison) **B1** *each*.

Exam-style 3.2 ■ 4 marks

A factory stores data about each component in a record type Component:

Field	Example	Meaning
Item_Num	123478	a numeric ID
Reject	FALSE	rejected?
Stage	'B'	a stage letter
Limit_1	13.5	a value 0–100

Exam-style 3.2 ■ 4 marks

- (i) Write pseudocode to **declare the record structure** Component.
- (ii) A 1D array Item of **2000** elements will store all components. Write pseudocode to **declare** the Item array.

Solution 3.2

Method: Using a record

Worked steps

(i)

TYPE Component

 DECLARE Item_Num : INTEGER

 DECLARE Reject : BOOLEAN

 DECLARE Stage : CHAR

 DECLARE Limit_1 : REAL

ENDTYPE

Solution 3.2

M1 **A1** **A1** **A1** 'TYPE...ENDTYPE', INTEGER + BOOLEAN, CHAR for a single letter, REAL for 'Limit_1'

(ii) DECLARE Item : ARRAY[1:2000] OF Component **M1** **A1** 'ARRAY[...] OF Component', bounds '1:2000'.

Exam-style 3.3 ■ 2 marks

State **two** benefits of using an **array of records** instead of several separate arrays.

Solution 3.3

Method: Using a record

Worked steps

Any two: one identifier for all the data; easy to process with a **loop** and an index; all fields for one item stay together; easy to pass to a procedure **B1** **B1**.

Exam-style 3.4 ■ 3 marks

A bicycle-hire company stores one record for each hire. Each record holds a hire reference (up to 8 characters), the bicycle number, the date of hire, the number of hours hired, the cost per hour and whether the bicycle has been returned.

(a) **Complete** the table by giving an appropriate data type for each field.

field	example value	data type
HireRef	"HR004821"	_____
HoursHired	3	_____
CostPerHour	4.50	_____
Returned	TRUE	_____

Exam-style 3.4 ■ 3 marks

- (b) **Write** pseudocode to declare the record structure HireRecord, including a field for the date.
- (c) A 1D array Hire of 500 elements will hold all the records. **Write** the pseudocode declaration for this array.
- (d) **Write** pseudocode to assign the value 4.50 to the cost per hour of the **first** record in the array.
- (e) **Outline** three benefits of storing the data as an array of records rather than as six separate parallel arrays.
- (f) **Evaluate** the expression `Hire[1].HoursHired * Hire[1].CostPerHour`, and **state** the data type of the result.

Solution 3.4

Method: Using a record

Worked steps

(a) HireRef — STRING **B1**; HoursHired — INTEGER; CostPerHour — REAL **B1**; Returned — BOOLEAN **B1**.

(b)

```
TYPE HireRecord
  DECLARE HireRef : STRING
  DECLARE BicycleNumber : INTEGER
  DECLARE HireDate : DATE
  DECLARE HoursHired : INTEGER
  DECLARE CostPerHour : REAL
  DECLARE Returned : BOOLEAN
ENDTYPE
```

Solution 3.4

B1 B1 B1 B1 B1 'TYPE'/'ENDTYPE'; all six fields present; correct types; 'DATE' used for the date; correct pseudocode syntax throughout

(c) DECLARE Hire : ARRAY[1:500] OF HireRecord **B1 B1**.

(d) Hire[1].CostPerHour ← 4.50 **B1 B1** — the record element is selected with '[' and the field with a dot.

(e) Any three: the fields belonging to one hire are kept **together**, so a record can be passed to a procedure as a single item **B1**; there is no risk of the arrays drifting **out of step** with one another **B1**; adding a new field means changing one type definition instead of creating another array **B1**. Also acceptable: the code reads more clearly; the whole record can be written to a file in one operation.

(f) $3 \times 4.50 = 13.50$ **B1**; the result is REAL, because multiplying an INTEGER by a REAL gives a REAL **B1**.