

Cambridge International 9618 pseudocode

The rule on answers. Paper 2 answers must be in pseudocode, and variations with your programming language are unacceptable. Use only functions given in the exam insert.

Comments

```
// this procedure swaps
// values of X and Y
```

Each line of a multi-line comment starts with //.

Assignment

```
Counter ← 0
Counter ← Counter + 1
```

The assignment operator is ←.

Constants

```
CONSTANT HourlyRate = 6.50
CONSTANT DefaultText = "N/A"
```

Only literals can be used as the value of a constant.

Declaring a variable and its type

```
DECLARE Counter : INTEGER
DECLARE TotalToPay : REAL
DECLARE GameOver : BOOLEAN
```

Types: INTEGER, REAL, CHAR, STRING, BOOLEAN, DATE.
Declaring is described as good practice.

Input

```
INPUT Answer
```

Output

```
OUTPUT "You have ", Lives, " lives
left"
```

Integer division and remainder

```
10 DIV 3 // evaluates to 3
10 MOD 3 // evaluates to 1
```

The examples are from the Paper 2 insert; the guide defines DIV and MOD without examples.

Comparison

```
= <> < <= > >=
```

Joining strings

```
"Summer" & " " & "Pudding"
```

Produces "Summer Pudding".

IF... ELSE

```
IF ChallengerScore > ChampionScore THEN
    OUTPUT ChallengerName, " is the new
champion"
ELSE
    OUTPUT ChampionName, " is still the
champion"
ENDIF
```

CASE / switch

```
CASE OF Move
    'W' : Position ← Position - 10
    'S' : Position ← Position + 10
    OTHERWISE : CALL Beep
ENDCASE
```

A case can be a range: <value1> TO <value2>. Cases are tested in order and only the first match runs. OTHERWISE must be last.

Count-controlled loop

```
FOR Index ← 1 TO 30
    StudentNames[Index] ← ""
NEXT Index
```

Both bounds inclusive. Repeating the identifier after NEXT is good practice.

Pre-condition loop (WHILE)

```
WHILE Number > 9
    Number ← Number - 9
ENDWHILE
```

Post-condition loop

```
REPEAT
    OUTPUT "Please enter the password"
    INPUT Password
UNTIL Password = "Secret"
```

Declaring a 1D array

```
DECLARE StudentNames : ARRAY[1:30] OF
STRING
```

Good practice to state the lower bound; generally 1 is used.

2D arrays

```
DECLARE NoughtsAndCrosses :
ARRAY[1:3,1:3] OF CHAR
NoughtsAndCrosses[2,3] ← 'X'
```

Records

```
TYPE StudentRecord
    DECLARE LastName : STRING
    DECLARE YearGroup : INTEGER
ENDTYPE

DECLARE Pupil1 : StudentRecord
Pupil1.LastName ← "Johnson"
Form[Index].YearGroup ←
Form[Index].YearGroup + 1
```

Defining a procedure

```
PROCEDURE Square(Size : INTEGER)
    FOR Side ← 1 TO 4
        CALL MoveForward(Size)
        CALL Turn(90)
    NEXT Side
ENDPROCEDURE
```

Every parameter has a data type.

Calling a procedure

```
CALL Square(100)
CALL DefaultSquare()
```

Defining a function

```
FUNCTION Max(Number1 : INTEGER, Number2
: INTEGER) RETURNS INTEGER
    IF Number1 > Number2 THEN
        RETURN Number1
    ELSE
        RETURN Number2
    ENDIF
ENDFUNCTION
```

RETURN ends the function immediately.

Calling a function

```
OUTPUT "Penalty Fine = ", Max(10,
Distance*2)
```

The keyword CALL should not be used when calling a function.

By value and by reference

```
PROCEDURE SWAP(BYREF X : INTEGER, Y :
INTEGER)
    Temp ← X
    X ← Y
    Y ← Temp
ENDPROCEDURE
```

BYREF need not be repeated, so Y is by reference too. Parameters should not be passed by reference to a function.

String length

```
LENGTH("Happy Days") // returns 10
```

Substrings

```
MID("ABCDEFGH", 2, 3) // returns
"BCD"
```

String, start position, length. Position 1 is the first character.

Converting between types

```
INT(27.5415) // returns 27
```

The Paper 2 insert adds NUM_TO_STR(87.5) returns "87.5", STR_TO_NUM("23.45") and IS_NUM().

Opening and reading a text file

```
OPENFILE "FileA.txt" FOR READ
READFILE "FileA.txt", LineOfText
CLOSEFILE "FileA.txt"
```

Modes: READ, WRITE, APPEND. A file is open in one mode at a time.

Writing to a text file

```
OPENFILE "FileB.txt" FOR WRITE
WRITEFILE "FileB.txt", LineOfText
CLOSEFILE "FileB.txt"
```

WRITE creates a new file and loses existing data. APPEND adds after it.

Class and constructor

```
CLASS Pet
    PRIVATE Name : STRING
    PUBLIC PROCEDURE NEW(GivenName :
STRING)
        Name ← GivenName
    ENDPROCEDURE
ENDCLASS
```

Not defined in this guide

Else-if chains.

Examiners have penalised

- Joining strings with + instead of &
- Dropping the arrow from FOR, or writing LEN()
- Writing CALL in front of a function
- Reserved words as variable names
- Half a condition
- Brackets and the multiplication sign
- Python habits in file handling
- OUTPUT where the question wants RETURN
- Forgetting BYREF in a header