Section 11.3 - Structured Programming

Layer 6: High-Order Language

Syllabus Content Section 11: Programming

S11.3.1 Define and use a procedure

> Defining and calling procedures

```
// A procedure with no parameters
PROCEDURE <identifier>
    <statement(s)>
ENDPROCEDURE

// A procedure with parameters
PROCEDURE <identifier>(<param1> : <data type>, <param2> : <data
type>...)
    <statement(s)>
ENDPROCEDURE

// call
CALL <identifier>
CALL <identifier>(Value1, Value2, ...)
```

S11.3.2 Explain where in the construction of an algorithm it would be appropriate to use a procedure

---

Procedures means of giving a group of statements a name. When we want to program a procedure we need to define it before the main program. We call it in the main program when we want the statements in the procedure body to be executed.

S11.3.3 Use parameters

- A procedure may have none, one or more parameters
- A parameter can be passed by reference or by value

> Example - definition and use of procedures with and withour parameters

```
PROCEDURE DefaultSquare
        CALL Square(100)
ENDPROCEDURE

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

IF Size = Default THEN
        CALL DefaultSquare
ELSE
        CALL Square(Size)
ENDIF
```

S11.3.4 Define and use a function

> Defining and calling functions

```
// A function with no parameters
FUNCTION <identifier> RETURNS <data type>
    <statement(s)>
ENDFUNCTION
// A function with parameters
FUNCTION <identifier>(<param1> : <data type>,<param2> : <data
type>…) RETURNS <data type>
```

```
    <statement(s)>
ENDFUNCTION
```

S11.3.5 Explain where in the construction of an algorithm it is appropriate to use a function

- A function is used in an expression, e.g. the return value replaces the call

---

> Example – definition and use of a function

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

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

S11.3.6 Use the terminology associated with procedures and functions

- including Procedure /function header, procedure / function interface, parameter, argument, return value

---

```
PROCEDURE <procedureIdentifier> // this is the procedure header
        <statement(s)> // these statements are the procedure body
ENDPROCEDURE

FUNCTION <functionIdentifier> RETURNS <dataType> // function header
        <statement(s)> // function body
        RETURN <value>
ENDFUNCTION
```

`Return value`: the value replacing the function call used in the expression

S11.3.7 Write efficient pseudocode

> Example – passing parameters by reference

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