



Name: _____ Index: _____ Reg.no.: _____ Class: _____ Date: _____

PROCEDURES AND FUNCTIONS

- Functions and procedures are a type of **sub program**, a **sequence of instructions** that **perform a specific task or set of tasks**.
- Procedures and functions are **defined at the start of the code**.
- Sub programs are often used to **simplify** a program by **breaking it into smaller, more manageable parts**.
- Sub programs can be used to:
 - **Avoid duplicating code** and can be reused throughout a program.
 - **Improve the readability and maintainability** of code.
 - Perform **calculations**, to **retrieve data**, or to **make decisions** based on input.

Parameters are the variables that store the values of the arguments passed to a procedure or function. Some but not all procedures and functions will have parameters.

Procedures

A **procedure** is a set of programming statements grouped together under a single name that can be called to perform a task at any point in a program.

Procedures are defined using the **PROCEDURE** keyword in pseudocode.

A procedure can be written as:

➤ **WITHOUT PARAMETERS:**

```
PROCEDURE <identifier>  
    <statements>  
ENDPROCEDURE
```

Example:

```
PROCEDURE Stars  
    OUTPUT "*****"  
ENDPROCEDURE
```

To call a procedure, it can be written as:

```
CALL <identifier>
```

Example:

```
CALL Stars
```

➤ **WITH PARAMETERS are being used:**

```
PROCEDURE <identifier>(<param1>:<data type>, <param2>:<data type>...)
    <statements>
ENDPROCEDURE
```

Example:

```
PROCEDURE Area(Length: INTEGER, Width: INTEGER)
    Area ← Length * Width
    OUTPUT "The area is ", Area
ENDPROCEDURE
```

To call a procedure, it can be written as:

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

Example:

```
CALL Area (5,3)
```

Or

```
Num1 ← 5
```

```
Num2 ← 3
```

```
CALL Area (Num1,Num2)
```

Functions

A **function** is a set of programming statements grouped together under a single name that can be called to perform a task at any point in a program. In contrast to a procedure, a function will return a value back to the main program.

- Functions are defined using the **FUNCTION** keyword in pseudocode.
- Functions **always return a value** so they do not use the CALL function, instead they are called **within an expression**.

A function can be written as:

➤ **WITHOUT PARAMETERS:**

```
FUNCTION <identifier> RETURNS <data type>
    <statements>
ENDFUNCTION
```



Example:

```
FUNCTION AddNumbers() RETURNS INTEGER
  DECLARE Num1, Num2, Result : INTEGER
  OUTPUT "Enter a number 1"
  INPUT Num1
  OUTPUT "Enter a number 2"
  INPUT Num2
  Result ← Num1 + Num2
  RETURN Result
ENDFUNCTION
```

Output the value returned from the function:

Because a function returns a value, it can be called by assigning the return value directly into a variable as follows:

Method 1:

```
Total ← AddNumbers()
OUTPUT "Total is ", Total
```

Method 2:

```
OUTPUT "Total is ", AddNumbers()
```

➤ WITH PARAMETERS are being used:

```
FUNCTION <identifier>(<param1>:<data type>, <param2>:<data type>...) RETURNS <data type>
  <statements>
ENDFUNCTION
```

Example:

```
FUNCTION Same (A:INTEGER, B:REAL) RETURNS BOOLEAN
  IF (A = ROUND(B,0))
    THEN
      RETURN TRUE
    ELSE
      RETURN FALSE
  ENDIF
ENDFUNCTION
```

Output the value returned from the function:

Because a function returns a value, it can be called by assigning the return value directly into a variable as follows:

Method 1:

Z ← Same(5, 3.5)

OUTPUT Z

Method 2:

DECLARE Num1 : INTEGER

DECLARE Num2 : REAL

DECLARE Result : BOOLEAN

OUTPUT "Enter a number 1"

INPUT Num1

OUTPUT "Enter a number 2"

INPUT Num2

Result ← Same(Num1, Num2)

OUTPUT Result

Outline the role of procedures and functions in creating a program that is easier to maintain.

- Procedures / functions divide the program into smaller manageable segments making it easier to understand.
- Procedures / functions with meaningful names help to provide documentation for the program.
- Procedures and functions may be re-used to run from a single line.
- Procedures and functions can reduce repeated code.

Explain why a programmer would use procedures and parameters when writing a program.

Procedures are used

- to enable the programmer to write a collection of programming statements under a single identifier.
- to allow procedures to be re-used within the program or in other programs.
- to make program creation faster because procedures can be re-used.
- to make programs shorter without using duplication of code.
- to make programs easier to maintain due to being shorter.

Parameters are used

- to pass values from the main program to a procedure / function.
- to allow the procedure / function to be re-used with different data.



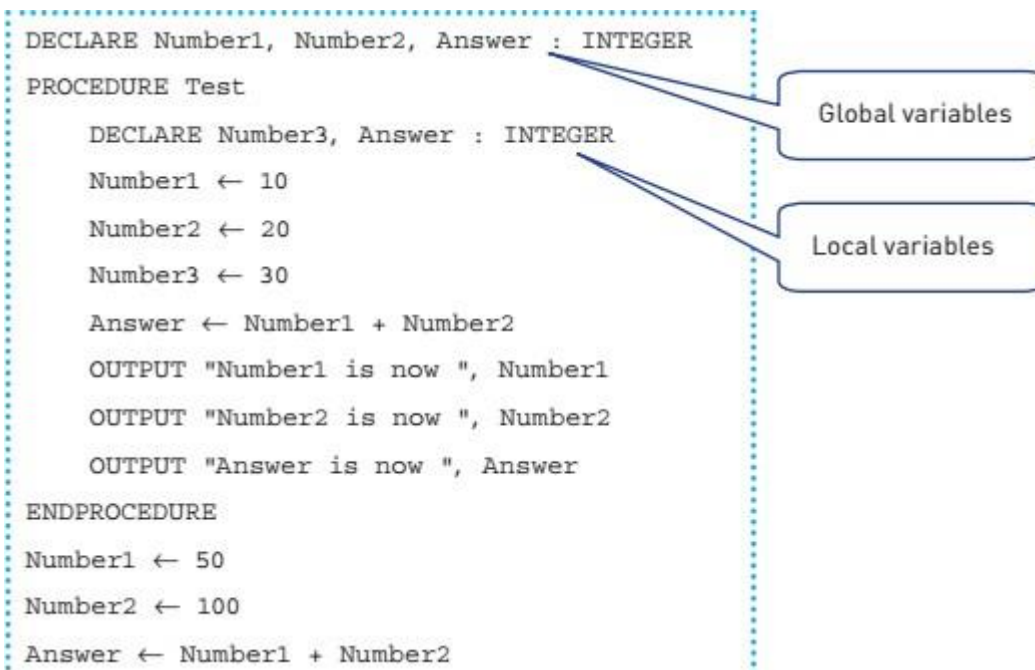
Local and Global variables

What is a local variable?

- A local variable is a **variable declared within a specific scope**, such as a function or a code block.
- Local variables are accessible **only within the block in which they are defined**, and their lifetime is limited to that particular block.
- Once the execution of the block ends, **the local variable is destroyed**, and its memory is released.

What is a global variable?

- A global variable is a **variable declared at the outermost level of a program**.
- They are declared **outside any modules** such as functions or procedures.
- Global variables have a **global scope**, which means they can be **accessed and modified** from **any part of the program**.



Some past paper questions

1. **Four** descriptions of programming concepts and **five** programming concepts are shown.

(a) Draw **one** line to link each description to the most appropriate programming concept.

Not all programming concepts will be used.

Description	Programming concept
a subroutine that may not return a value	function
a value that is declared and used within a specific procedure	procedure
a value that a procedure expects you to supply when it is called	parameter
a subroutine that will always return a value	global variable
	local variable

[4]

(b) Write the pseudocode to use a procedure named **Average** that passes the values **25** and **50** to the procedure.

.....
 [2]

2. Describe **two** differences between local variables and global variables.

Difference 1

 Difference 2

 [4]

3. Write a pseudocode program that uses a **procedure** to display a greeting message, "Hello, [name]!", where [name] is entered by the user.

.....

.....

.....

.....

.....

.....

.....

.....

4. Write a pseudocode program that defines a **function** to add two numbers. The program should input two numbers, call the function, and display the result.

.....

.....

.....

.....

.....

.....

.....

.....

10 A function is declared using pseudocode.

```
FUNCTION ConvertToCm(Inches: REAL) RETURNS REAL  
    RETURN Inches * 2.4  
ENDFUNCTION
```

Tick (✓) **one** box which accurately describes the use of the variable `Inches`

- A** answer ☐
- B** call ☐
- C** parameter ☐
- D** response ☐

[1]

5. The function **IsDivisible(A, B)** returns **TRUE** if **A** is evenly divisible by **B** (i.e., the remainder when **A** is divided by **B** is 0) and **FALSE** otherwise.

Write pseudocode statements to:

- Define the function
- Call the function with **M** and **N** and store the return value in **Result**

[illegible]

6. The function **MaxValue(A, B, C)** returns the largest value among A, B, and C. All the values are whole numbers.

Write pseudocode statements to:

- Define the function
- Call the function with **X**, **Y**, and **Z** and store the return value in **MaxNum**

Function definition

[6]

- Write a procedure **GreetUser** that takes a string parameter **Name** and **displays a greeting message** in the format:

"Hello, [Name]! Welcome to the program."

Then, call the procedure with the name "**Alice**".

- Write a procedure **CalculatePerimeter** that takes **two integer parameters, Length and Width**, and **calculates and displays** the perimeter of a rectangle.

(The formula for the perimeter is: $\text{Perimeter} = 2 * (\text{Length} + \text{Width})$)

Then, call the procedure with Length = **8** and Width = **6**.