

# PSEUDOCODE



e-චක්‍රය  
අධ්‍යාපන

# PSEUDOCODE



## Pseudo Codes

- ▶ It is an abbreviated version of actual computer code (that's why it is called Pseudocode)
- ▶ Once pseudocode is created, it is simple to translate into real programming code.

# PSEUDOCODE



## Advantages of Pseudo Codes

- ▶ Can plan the program.
- ▶ Can use pseudo code to describe the program to non-technical user.
- ▶ Can provide guidelines to a programmer to write the program.
- ▶ Opportunity to detect any logic error prior to actual coding.

# PSEUDOCODE



## How to write Pseudo-Code?

- \* What can your computer do?
  - \* Read, Calculate, Print, Convert....
- \* How do you instruct the computer to do what you intend?
  - \* Keywords – Words that are predefined and reserved to a computer to execute certain instructions.
    - \* Example: Read, Print, Compute ...
  - \* Common words – English words to complete the instructions.
    - \* Examples: Variables, Constants, file names....

# PSEUDOCODE



## Basic Computer Operations

- \* A computer can receive data.
- \* A computer can put out information.
- \* A computer can perform arithmetic operation.
- \* A computer can assign a value to a variable or memory location.

# PSEUDOCODE



## Receive Data

- \* When computer wants data Input from a particular source.
- \* Keyboard, Disk, CD, File

# PSEUDOCODE



## Receive Data

Instruction to computer	Pseudocode writing Methods
Commands: Read Get Input	Example: Read name Get num1 Input marks

# PSEUDOCODE



## Put out Information

- \* When computer is requested to supply information output information to a device.
  - \* Printer, File, Screen



# PSEUDOCODE



## Put out Information

Computer Instructions	Pseudocode writing Methods
<p>Commands:</p> <ul style="list-style-type: none"><li>Print</li><li>Write</li><li>Put</li><li>Output</li><li>Display</li></ul>	<p>Example:</p> <ul style="list-style-type: none"><li>Print “Command Print”</li><li>Write student record to master file</li><li>Put name, address</li><li>Output grade</li><li>Display “Error code. Please re-enter...”</li></ul>

# PSEUDOCODE



## Arithmetic & Logical Operations

- \* A program can contain arithmetic operations and computer has to perform those.

Computer Instructions	Pseudocode Writing Methods
<p>Commands:</p> <ul style="list-style-type: none"><li>+ Add</li><li>- Subtract</li><li>* Multiply</li><li>/ Divide</li><li>() Parentheses</li><li>Compute</li></ul>	<p>Example:</p> <p>Add score to total_score total_score = total_score + score Divide total_marks by no_of_subjects average = total_marks/no_of_subjects Compute C= (F-32) * 5/9</p>

# PSEUDOCODE



## Assign a value to a variable or a memory location

- \* Give an initial value to a data
- \* Assign a value as a result after doing some processing
- \* Keep some information for later use

# PSEUDOCODE



## Assign a value to a variable or a memory location

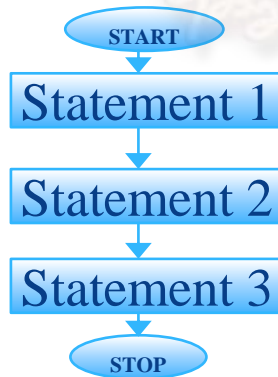
Computer Instructions	Pseudocode Writing Methods
Commands Initialize Set Store	Example: Initialize total_score to 0: total_score = 0 Set student_count to 0: student_count = 0 total_score = total_score + score total_student = total_student + 1 Store average in class_average

# PSEUDOCODE



## Sequence

- \* In a sequence control structure, statements are executed in the same order as they are written.



## Pseudocode;

**statement 1**

**statement 2**

**statement 3**

# PSEUDOCODE



## Sequence

- \* This sequence control structure can be used to represent four basic computer operations:
- \* receive Data
- \* put out information
- \* perform arithmetic
- \* Assign values

# PSEUDOCODE



## Exercises

- \* Write a pseudo code that inputs two numbers (a and b) and calculates the sum of the numbers and output the sum.

*Comment – This Pseudo code finds the sum of two given numbers*

**BEGIN**

**INPUT a**

**INPUT b**

**sum = a + b**

**OUTPUT sum**

**END**

# PSEUDOCODE



## Repetition

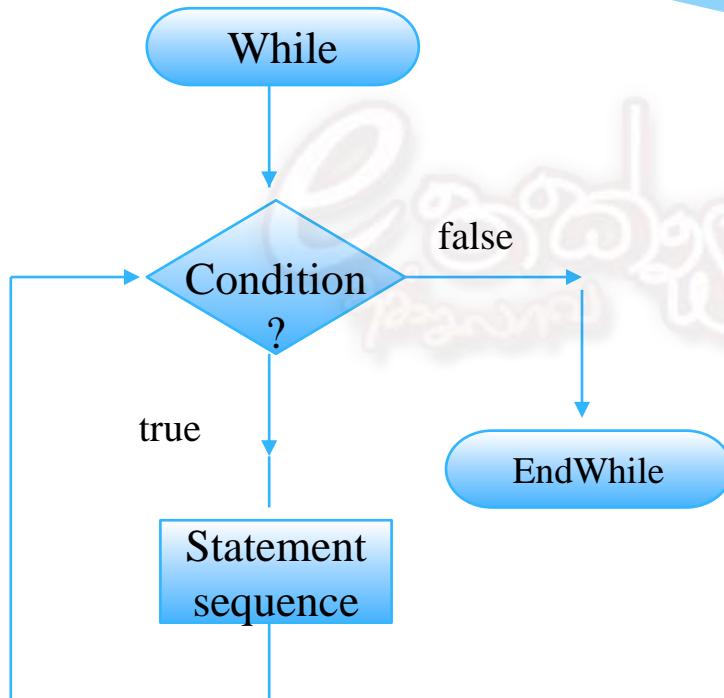
- \* Same sequence of operations need to be repeated.
- \* Use of Loop Types
  - \* WHILE ... ENDWHILE
  - \* DO WHILE ... ENDDO
  - \* FOR ... ENDFOR



# PSEUDOCODE



## The While Loop



**WHILE**      <Condition>  
Statement- Sequence  
**END WHILE**

# PSEUDOCODE



## Exercises

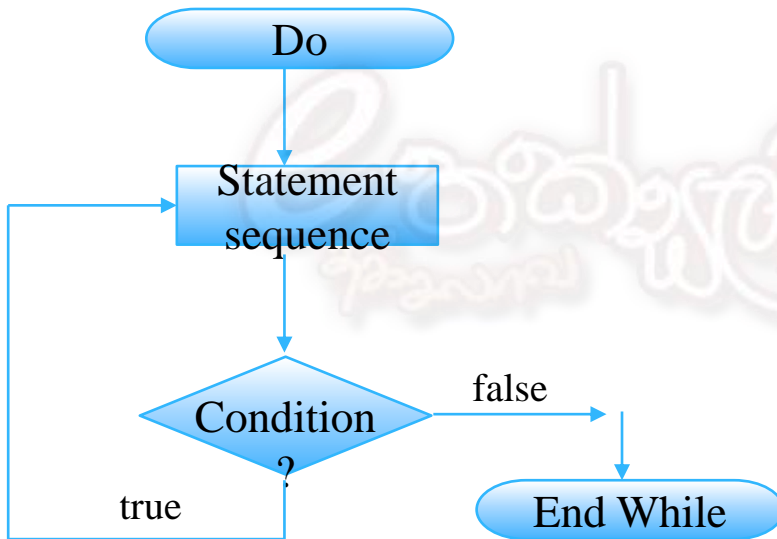
- \* Write a pseudo code that inputs ten numbers and outputs the sum and average of them.

```
BEGIN
  count = 0
  sum = 0
  WHILE count < 10
    INPUT a
    sum = sum + a
    count = count + 1
  END WHILE
  average = sum / 10
  OUTPUT sum, average
END
```

# PSEUDOCODE



## The Do-While Loop



DO  
Sequence  
WHILE <condition>

# PSEUDOCODE



## Exercises

- \* Write a pseudo code that inputs ten numbers and outputs the sum and average of them.

```
BEGIN
  count = 0
  sum = 0
  DO
    INPUT a
    sum = sum + a
    count = count + 1
  WHILE count < 10
  average = sum / count
  OUTPUT sum, average
END
```

# PSEUDOCODE



## The For Loop

- \* This loop is a specialized construct for iterating a specific number of times, often called a "counting" loop.
- \* Two keywords, FOR and ENDFOR are used
- \* The general form is:

**FOR iteration bounds**

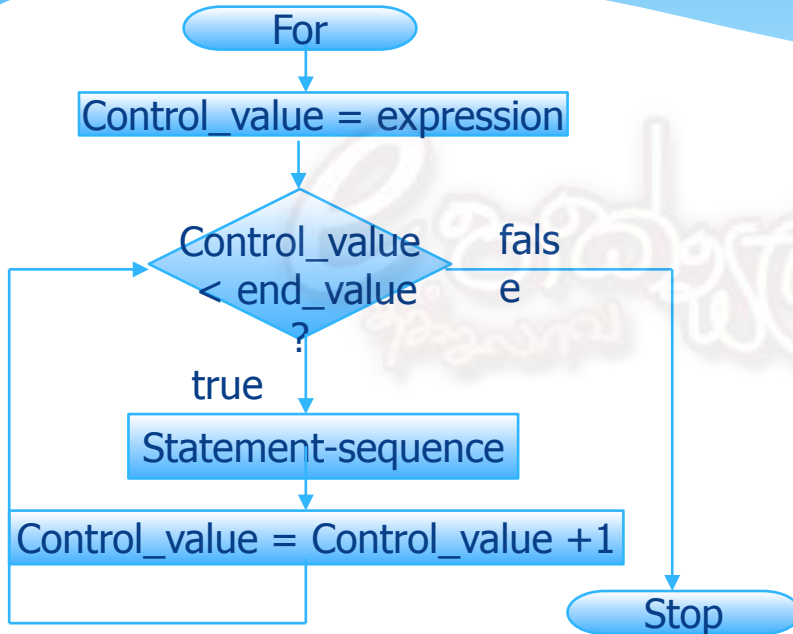
**sequence**

**ENDFOR**

# PSEUDOCODE



## The For Loop



**FOR** iteration bounds  
sequence  
**ENDFOR**

# PSEUDOCODE



## Exercises

- \* Write a pseudo code that inputs ten numbers and outputs the sum and average of them.

**BEGIN**

sum = 0

average = 0

FOR count = 1 to 10

INPUT a

sum = sum + a

END FOR

average = sum / count

OUTPUT sum, average

**END**

# PSEUDOCODE



## The Nested Loop

- \* The constructs can be embedded within each other, and this is made clear by use of indenting
- \* Nested constructs should be clearly indented from their surrounding constructs

```
SET total to zero
WHILE Temperature < zero
    INPUT Temperature
    IF Temperature < Freezing THEN
        INCREMENT total
    END IF
END WHILE
OUTPUT total
```



# PSEUDOCODE



## The End

