Al-Mustaqbal University
College of Engineering & Technology
Biomedical Engineering Department

Computer

Lecture 2

Algorithms Development – Pseudo-code and Flowchart

PhD. BEng. Ahmed Hasan Janabi

PhD in Cybersecurity

Email: Ahmed.Janabi@uomus.edu.iq

What is an algorithm?

- In computer programming, an algorithm is a set of well-defined instructions to solve a particular problem.
- It takes a set of input and produces a desired output.

Step 1: Start
Step 2: Define the variables
Step 3: Read values of variables
Step 4: Process
  - math operations
  - logic operations
  - comparisons
Step 5: Display result
Step 6: End

1. بداية
2. تعريف المتغيرات
3. قراءة قيمة كل متغير
4. معالجة
   - عمليات رياضية
   - عمليات منطقية
   - مقارنات
5. عرض/طباعة الناتج
6. نهاية

Algorithm to add two numbers

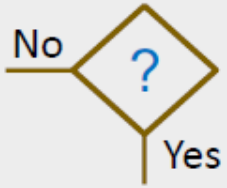
- We need to first write the steps (sequence of actins) that lead to performing the task.
- For example, an algorithm to add two numbers:
  - Take two number inputs
  - Add numbers using the + operator
  - Display the result

Step #	Description
Step1	Start
Step2	Declare or define variables num1, num2, sum;
Step3	Read values num1 and num2
Step4	Add num1 and num2 and assign the results to sum sum ← num1 + num2;
Step5	Print sum
Step6	End

Flowchart Symbols

- A Flowchart is a graphical representation that shows the behavior (workflow) of an algorithm.

Flowcharts use standard shapes including the following:

Symbol	Name	Description
	Start / End	An oval represents a ***start*** or ***end*** point
	Arrows	A line is a connector that shows flow direction between the representative shapes
	Input /Output	A parallelogram represents ***input*** or ***output***
	Process	A rectangular represents a ***process*** **(calculation)**
	Decision	A diamond indicates a ***decision*** **(comparison)**
	Connector	A circle is used to combine one part of the flowchart with another part

Adding two numbers

✓ Write the pseudo-code for algorithm to sum two numbers. Then draw the equivalent flowchart.

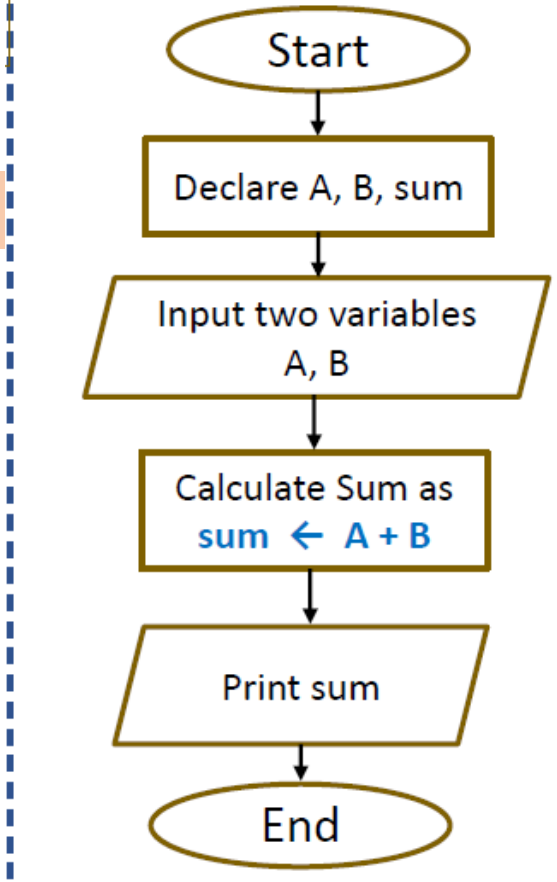
➢ Flowcharts use standard shapes including the following:

Ans:

Pseudo-code Algorithm

```
Step 1: START
Step 2: DECLARE variables num1, num2, sum;
Step 3: READ variables num1, num2;
Step 4: CALCULATE sum
            sum = num1 + num2;
Step 5: PRINT sum
Step 6: END
```

Flowchart

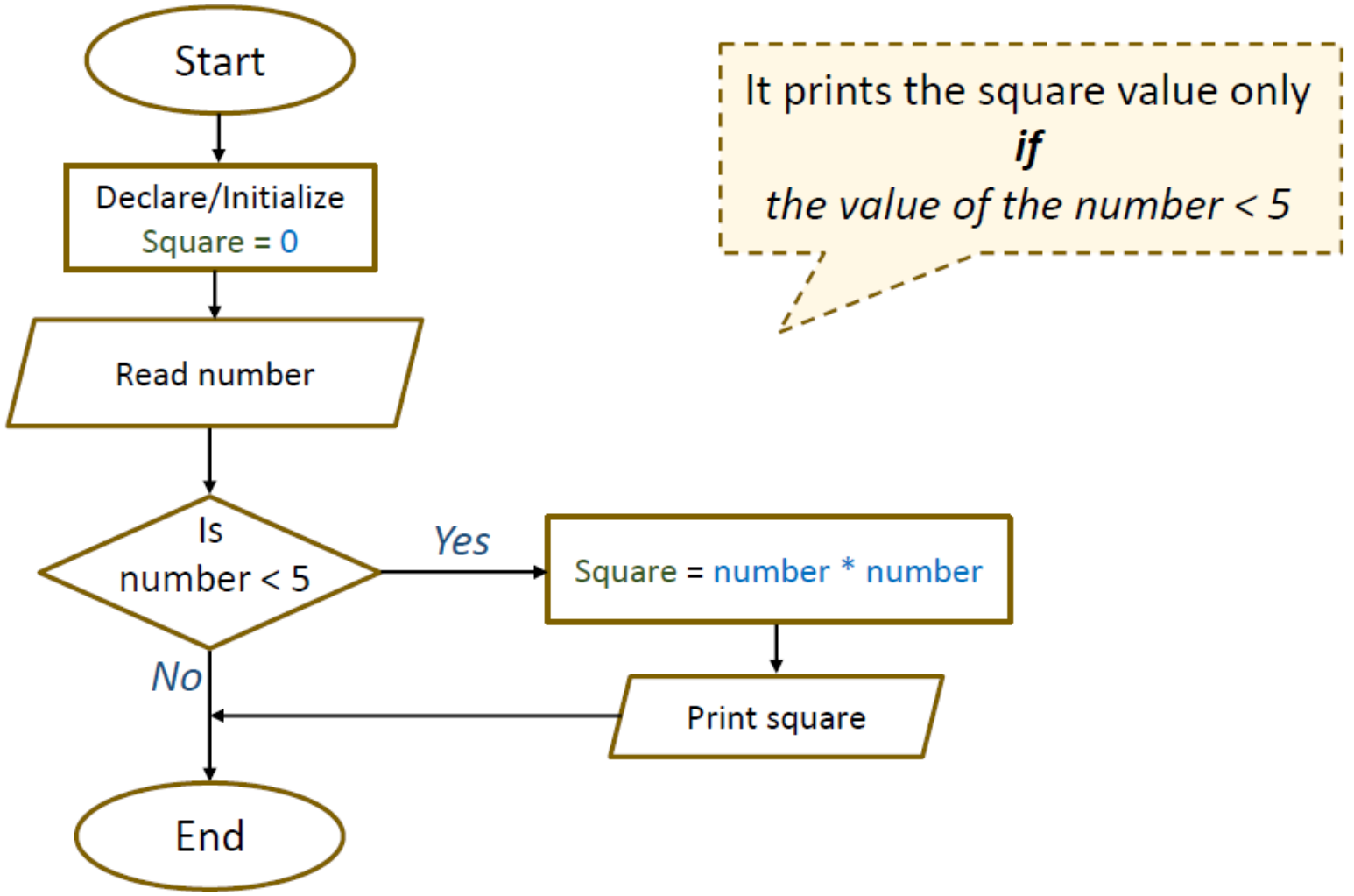


Note:

- The ***pseudo-code*** and ***flowchart*** of an algorithm is not written for a particular programming language.
- They can be used to plan a solution before coding it.

Print if the condition met

- Draw the Flowchart to print the square of a number if the number is less than 5.



Note:

- The lines and arrows in it show the sequence of steps and the relationship among them.

Difference between pseudo-code and flowchart

Main differences between Pseudo-code and Flowchart

- The pseudo-code is a high-level description of an algorithm while the flowchart is a graphical representation of an algorithm.
- An algorithm is a set of instructions for solving a problem or accomplishing a task.
- Every computerized device uses algorithms, which cut the time required to do things manually.

- There are three main ways or scenarios used in the programming. These are:

1) ***Sequencing*** – putting things in the right order
2) ***Conditions*** – performing different things depending on some rules
3) ***Iteration (Looping)*** – repeating operations

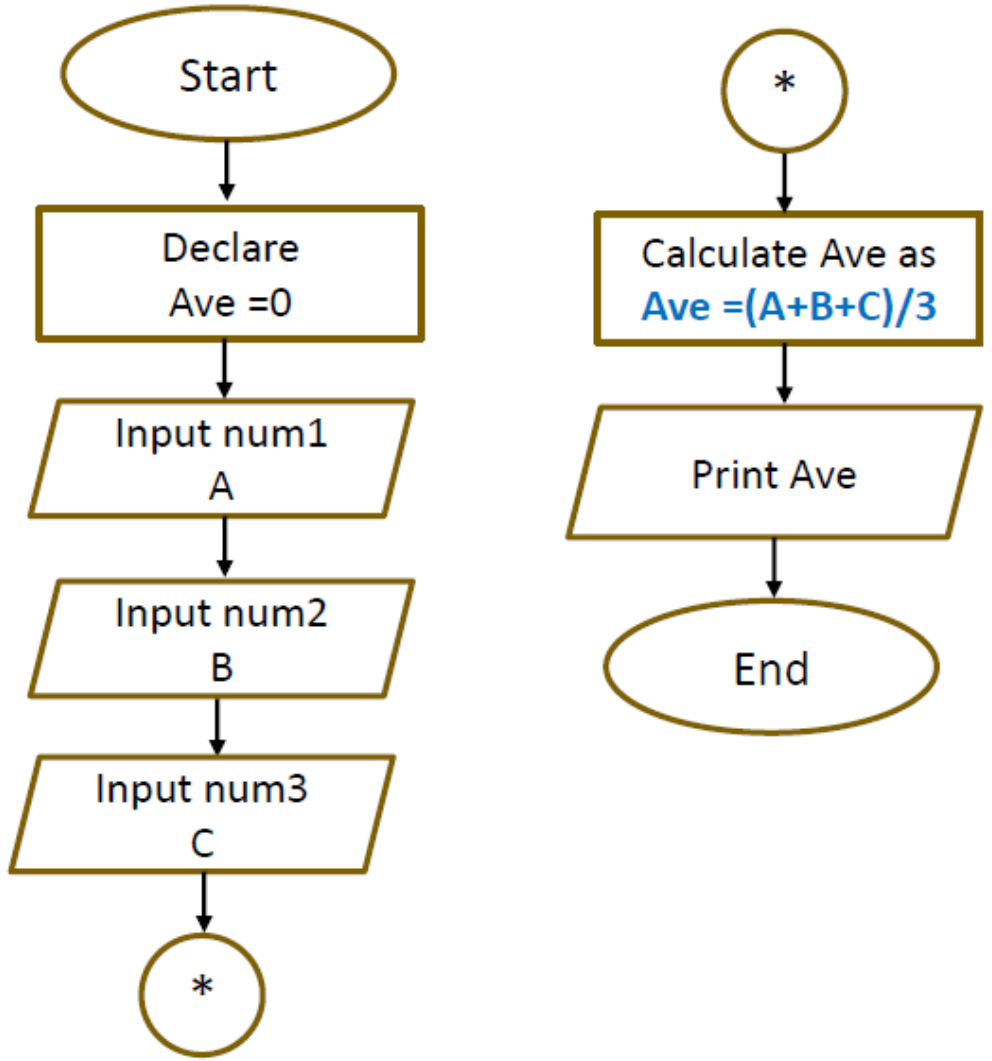
Sequencing

- Calculate the average of three input numbers.

Pseudo-code Algorithm

Step 1: Start
Step 2: Declare variables Ave;
Step 3: Read variables num1, num2, num3;
Step 4: Calculate Ave
Ave = (num1 + num2 + num3)/3;
Step 5: Print Ave
Step 6: End

Flowchart



Conditions

- Conditions are very important to understand the flow-control of the code being executed.
- Conditions are used to make a decision (choose between two or more alternatives based on the condition)

  - For example, conditions become very essential to use when facing different directions, so based on the condition should choose the thing to be executed.

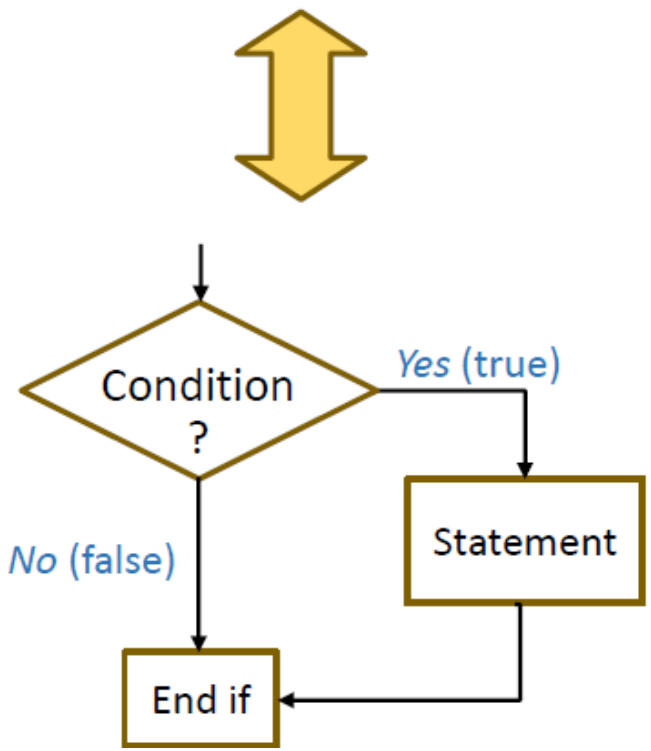
Use the *if* statement

```
If (condition) then
  run this statement;
Else
run the other statement;
```

Conditions

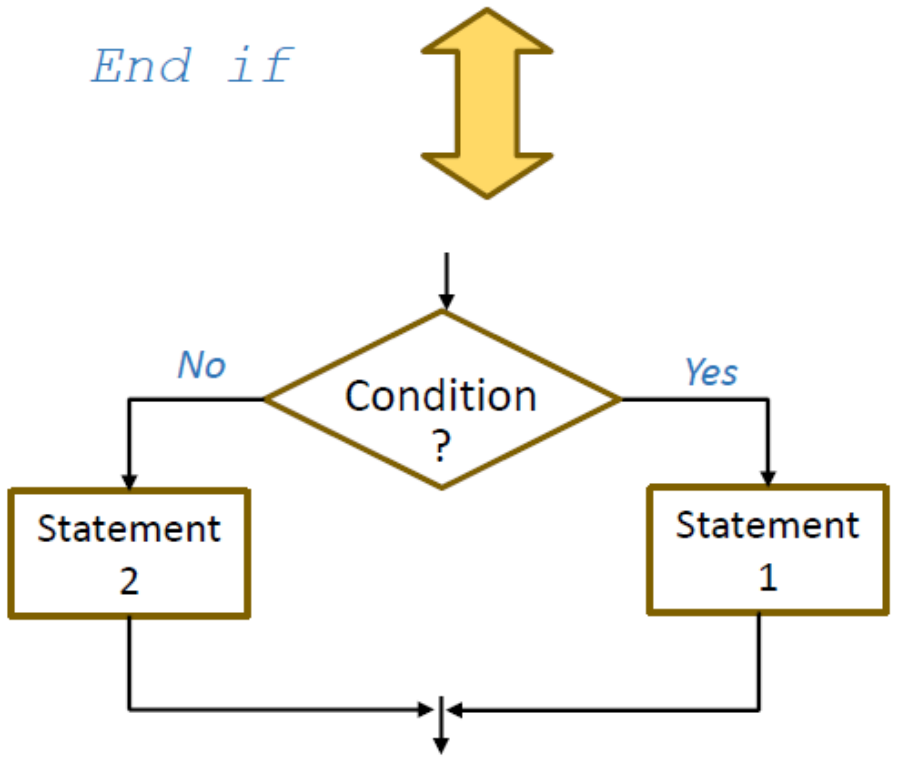
1 *if .... then*

```
If (condition) then
 statement(s)

End if
```



2 *if .... then .... else*

```
If (condition) then
 statement 1
Else
 statement 2

End if
```



Nested if structure

3 ***Nested structure of if .... then***

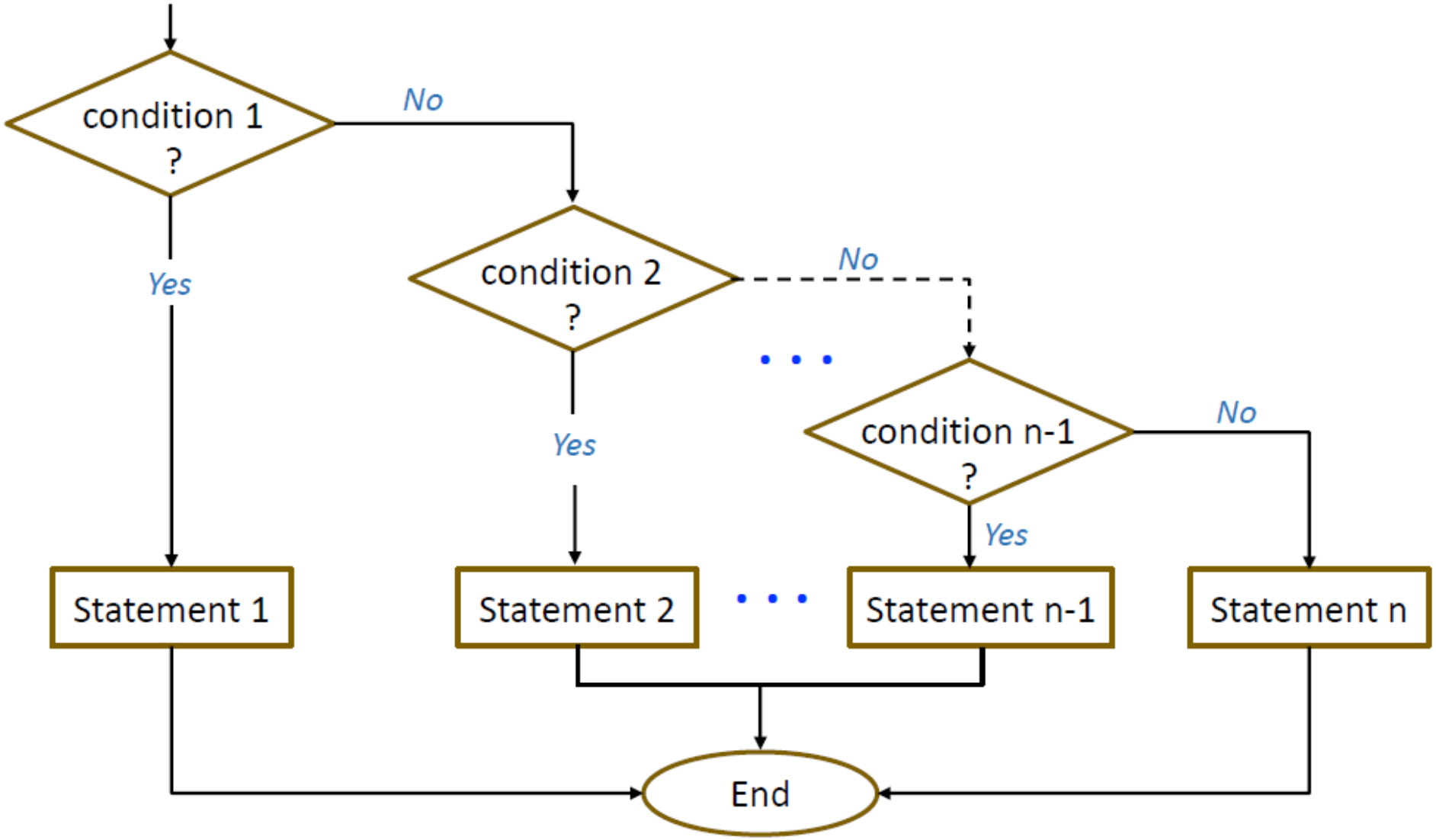
```
If (condition 1) then
 statement 1

If (condition 2) then
 statement 2

If (condition 3) then
 statement 3

Else
  statement 4

End if
End if
End if
```

if .... then .... elseif

```
If (condition 1) then
 statement 1

Elseif (condition 2) then
 statement 2

Elseif (condition 3) then
 statement 3

Else
  statement 4

End if
```

if … then … elseif – Flowchart

- *if .... then .... elseif* Flowchart

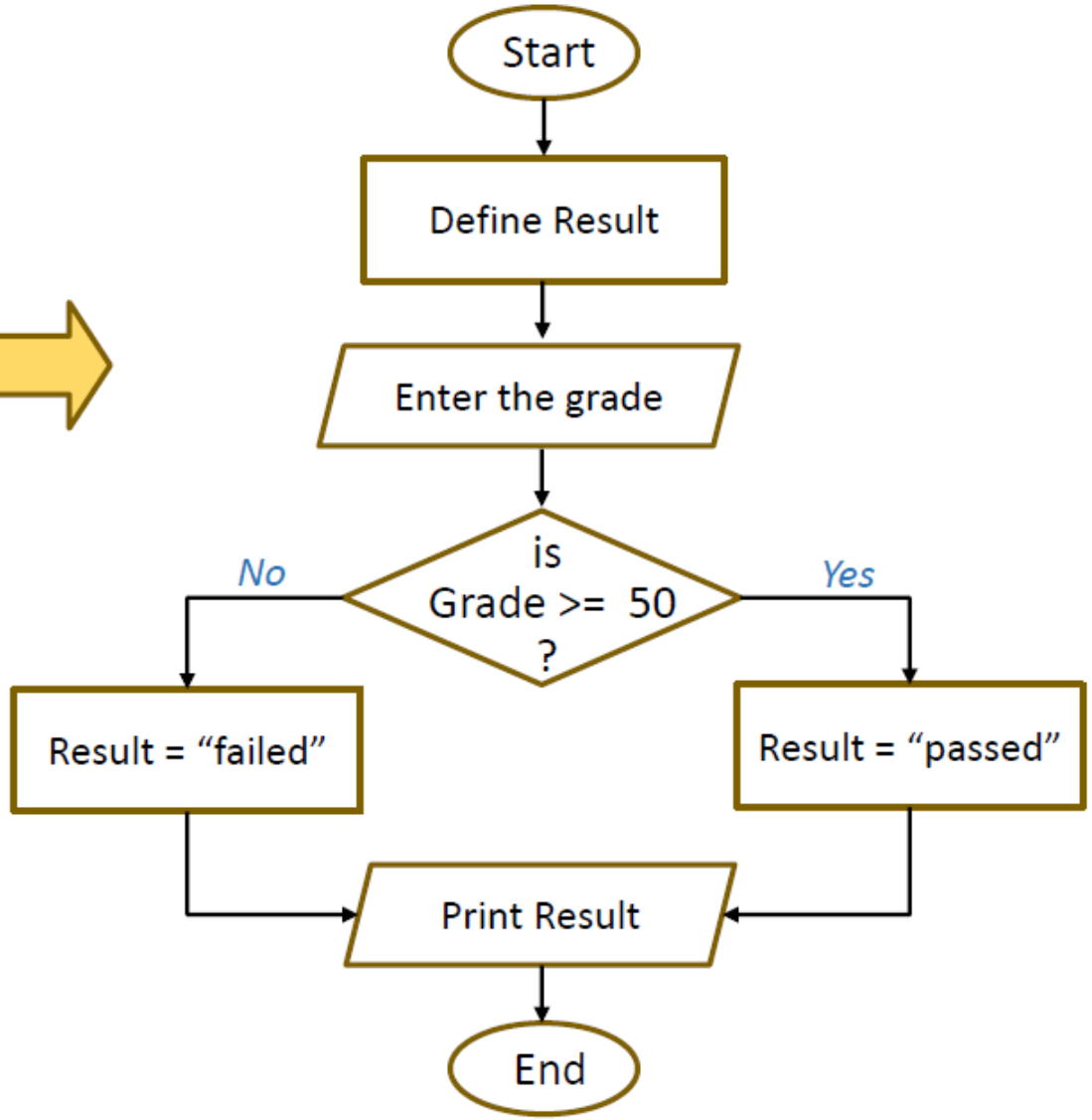


Condition – Example

✓ Write the pseudo-code to check if a student has passed the exam or failed. Then draw the equivalent flowchart?

Ans:

Step 1: Start
Step 2: Declare Result
Step 3: Read the student's grade
Step 4: *If* (Grade >= 50)
Result = "passed"
else
Result = "failed"
Step 5: Print result
Step 6: End



Classwork #1 – Student's Grade

✓ Send a message to a student depending on the obtained score?

Ans:

Step 1: Start
Step 2: Declare Result, message
Step 3: Read the student's grade
Step 4: *If* (Grade >= 50)
Result = "pass"
message = "Well Done!"
else
Result = "fail"
message = "See me!"
Step 5: Print result
Step 6: End

Draw the equivalent flowchart

Flowchart

Multiple Conditions – Example

✓ Is it possible to have multiple conditions in an algorithm?

❖ **For example, print the result of a student according to the conditions listed in the table below?**

Grade value	Printed result	التقدير
Grade < 50	Result = "Failed"	راسب
Grade < 60	Result = "Passed"	مقبول
Grade < 70	Result = "Medium"	متوسط
Grade < 80	Result = "Good"	جيد
Grade < 90	Result = "Very good"	جيد جداً
Grade > = 90	Result = "Excellent"	امتياز

Multiple Conditions – Example solution

Ans:

Step 1: Start

Step 2: Declare Grade, Result

Step 3: Read the student's grade

Step 4: *If* (Grade < 50)

Result = "failed"

else if (Grade < 60)

Result = "passed"

else if (Grade < 70)

Result = "medium"

else if (Grade < 80)

Result = "good"

else if (Grade < 90)

Result = "very good"

else

Result = "excellent"

Step 5: Print result

Step 6: End

تكملة الحل

Classwork #2 – Ticket price

✓ Write the pseudo-code algorithm to display the price of a cinema ticket as follow: (age < 12) then (price = 5K IQD), (age > 65) then (price = 7K IQD); otherwise (price = 10K IQD)?

Ans:

Step 1: Start
Step 2: Declare price
Step 3: Read the customer's Age
Step 4: *If* (Age < 12)
price = 5K
Else if (Age > 65)
price = 7K
else
price = 10K
Step 5: Display price
Step 6: End

Draw the equivalent flowchart

Flowchart

Iteration –> Repetition

- Let's say we need to print all the numbers from 1 to 1000.
- Would we do this by sequencing?
- It is much better to do tasks like that using iteration

Repeat the operation or sequence of operations until the condition is met

Sequencing

Step 1: Start

Step 2: display 1

Step 3: display 2

Step 4: display 3

•
•
•

Step 1001: display 1000

Step 1002: END

Iteration

Step 1: Start

Step 2: Declare number =1

Step 3: while the number <= 1000

{

display the number

number = number +1 (increment the number)

}

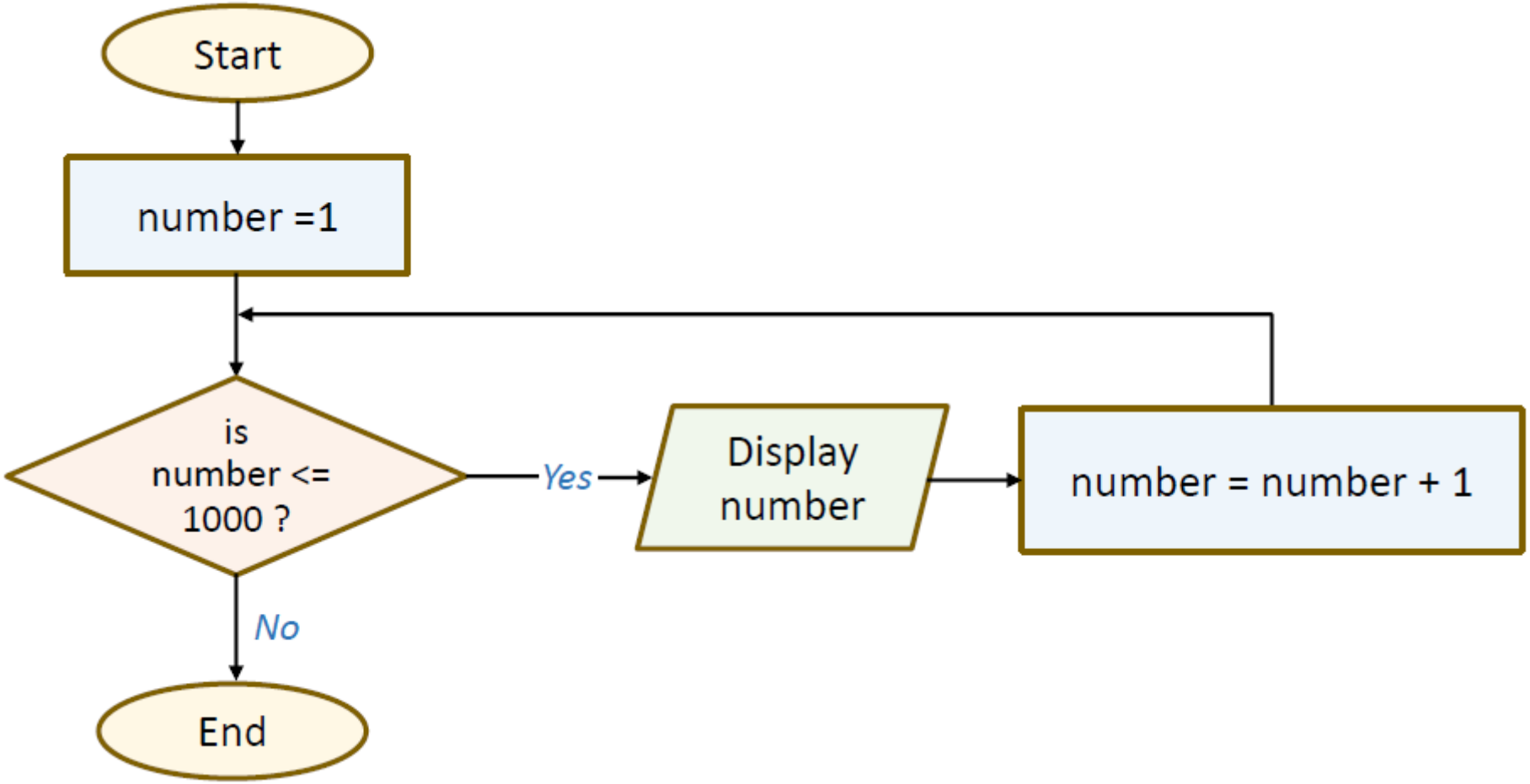
Step 4: End

go to step 3 if count <= 1000

Loop

Iteration – printing the numbers from 1 to 1000

- Flowchart
  - print the numbers from 1 to 1000

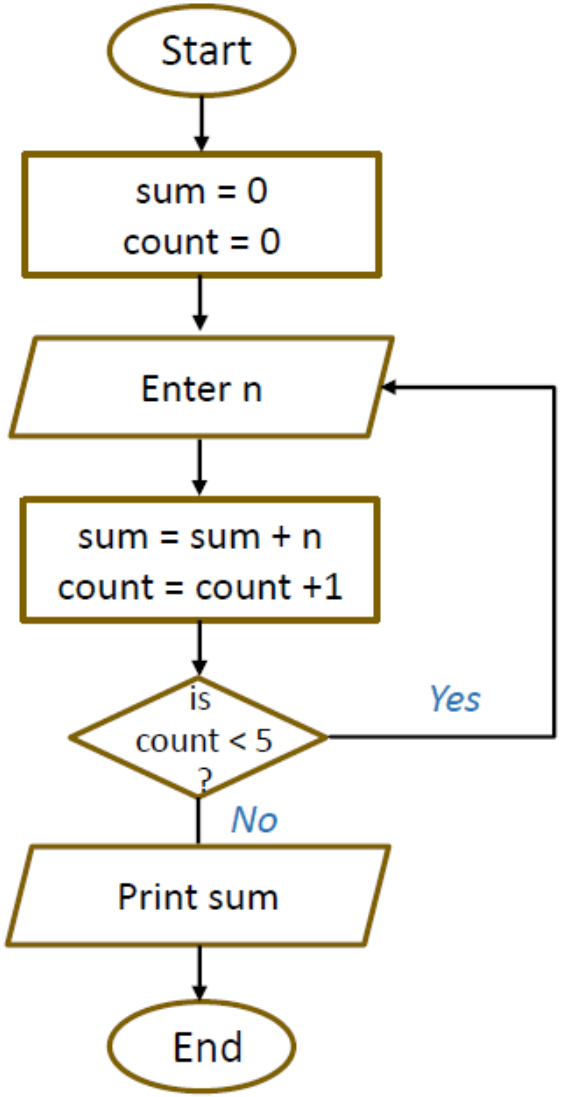


Iteration – Summing five numbers

✓ Write the pseudo-code for algorithm to sum five numbers. Then draw the equivalent flowchart?

Ans:

Step 1: Start
Step 2: Initialization
Initialize: sum = 0 and count = 0
Step 3: Enter $\boldsymbol{n}$ numbers
Step 4: Calculate sum and increment count
sum = sum + $\boldsymbol{n}$
count = count +1
Step 5: *while* (count < 5)
go to step 3
Step 6: Print sum
Step 7: End



Average of ten numbers – Example

✏️ Write a pseudo-code algorithm to find the sum and average of ten numbers entered by the user. Then draw the equivalent flowchart.

Ans:

Algorithm

Step 1: Start

Step 2: Initialization

Initialize: sum = 0, Ave = 0, and count = 0

Step 3: while count < **10**

{

enter a ***number***

sum = sum + ***number***

count = count +1

}

go to step 3 if count < 10

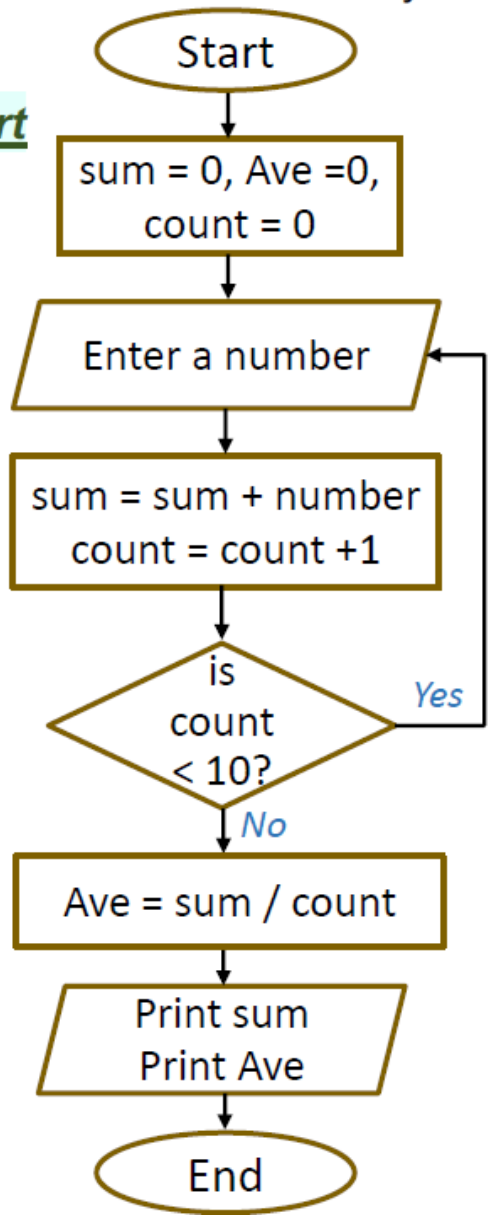
Step 4: *Ave = sum/count*

Step 6: Print sum

Print Ave

Step 7: End

Flowchart



Qualities of Good Algorithms

1. ***Precision*** – الدقة the steps of input and output should be precisely defined.
2. ***Clarity*** – الوضوح each step in the algorithm should be clear and unambiguous.
3. ***Inputs/Outputs*** – المدخلات والمخرجات has inputs that lead to the production of the output
4. ***Effectiveness*** – الفعالية Algorithms should solve problems in effective ways such that getting an
5. output from each step based on its inputs and outputs of the previous step
6. ***Generality*** – العمومية/ الشمول the algorithm should be written in such a way that it can be used in
7. different programming languages.

THANK YOU