البرمجة

=

الخوارزمية + الكود

code+ pseudo code

- الخوارزمية هي مجموعة خطوات مرتّبة ومحددة تُستخدم للوصول إلى حلّ لمشكلة ما خلال عدد محدود من الخطوات وزمن محدد.

تشبه الخوارزمية "وصفة الطبخ:"

-مدخلات (المقادير)

-خطوات معالجة

-مخرجات (الوجبة النهائية)

المكونات الأساسية للخوارزمية

كل خوارزمية تحتوي ثلاثة عناصر:

1. **المدخلات** :(Input) البيانات المطلوبة لمعالجة المشكلة.

2. **متن الخوارزمية** :(Processing) خطوات المعالجة المحددة.

3. **المخرجات** :(Output) النتيجة النهائية

الخوارزمية ليست مجرد خطوات مكتوبة، بل هي **طريقة تفكير علمية** تمكّن الطالب من:

- تحليل المشكلة
- تحديد البيانات
- صياغة خطوات الحل
- تحويل الحل لاحقًا إلى برنامج بلغة برمجة

هيكلة الخوارزميات

Structure of an Algorithm?

Problem:
A teacher wants a program that:

- Calculates the **average mark of students in class**

Problem Solving

Algorithmics

Title (Name of the algorithm)

- Represents the problem you're solving.

Example: Algorithm Calculate_Avg

Declaration (Variables / Constants)

- What information (data) the algorithm needs to start.

Example: Marks, Nb_Students

Body (Instructions / steps)

- The sequence of instructions that transform input into output.
- This is the heart of the algorithm.

Example : Avg ← Marks_sum / Nb_Students

Outputs(Instructions)

- The result or final data produced.
- May also correspond to a variable later in code.
- Example : Avg, Marks_sum

End (Marks the end of the algorithm)

```
Algorithm: Student_Average
Variables: A, B, C, Marks_sum, Avg : real
Constant: Nb_students = 3
Begin
    Read(A, B, C)                  //get students marks
    Marks_sum ← A + B + C          //calculate sum
    Avg ← Marks_sum / Nb_students
    Write("The average mark is:", Average)
End.
```

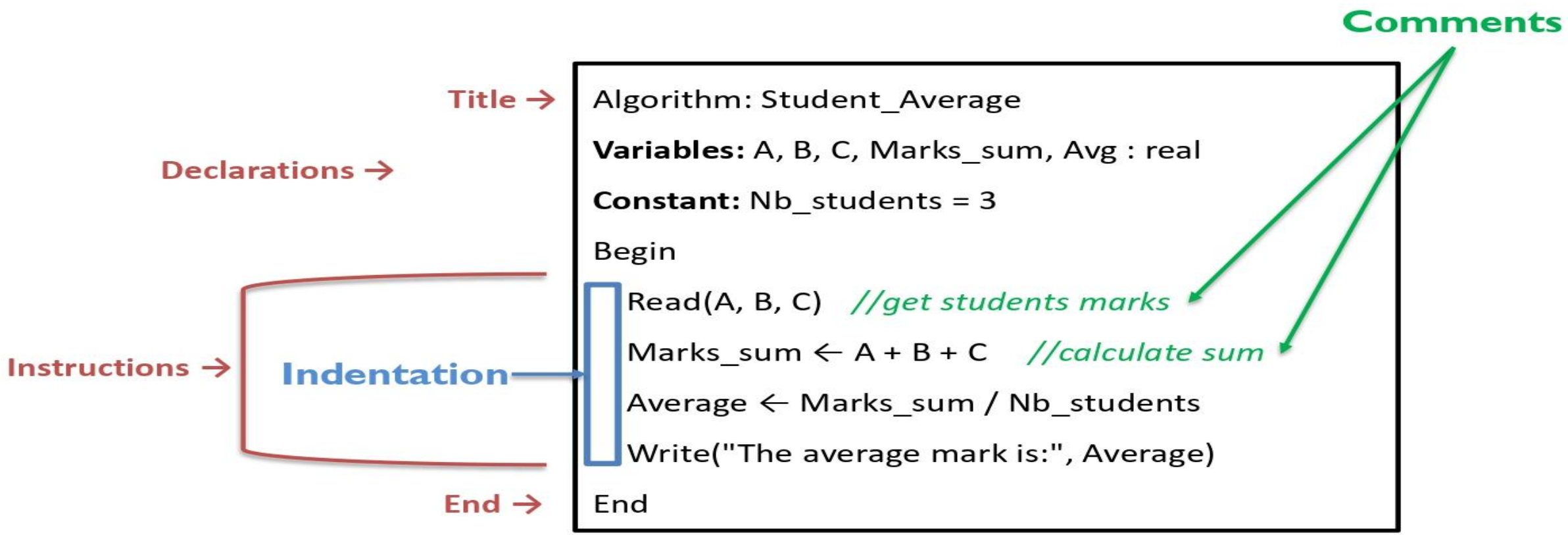
Structure of an Algorithm?

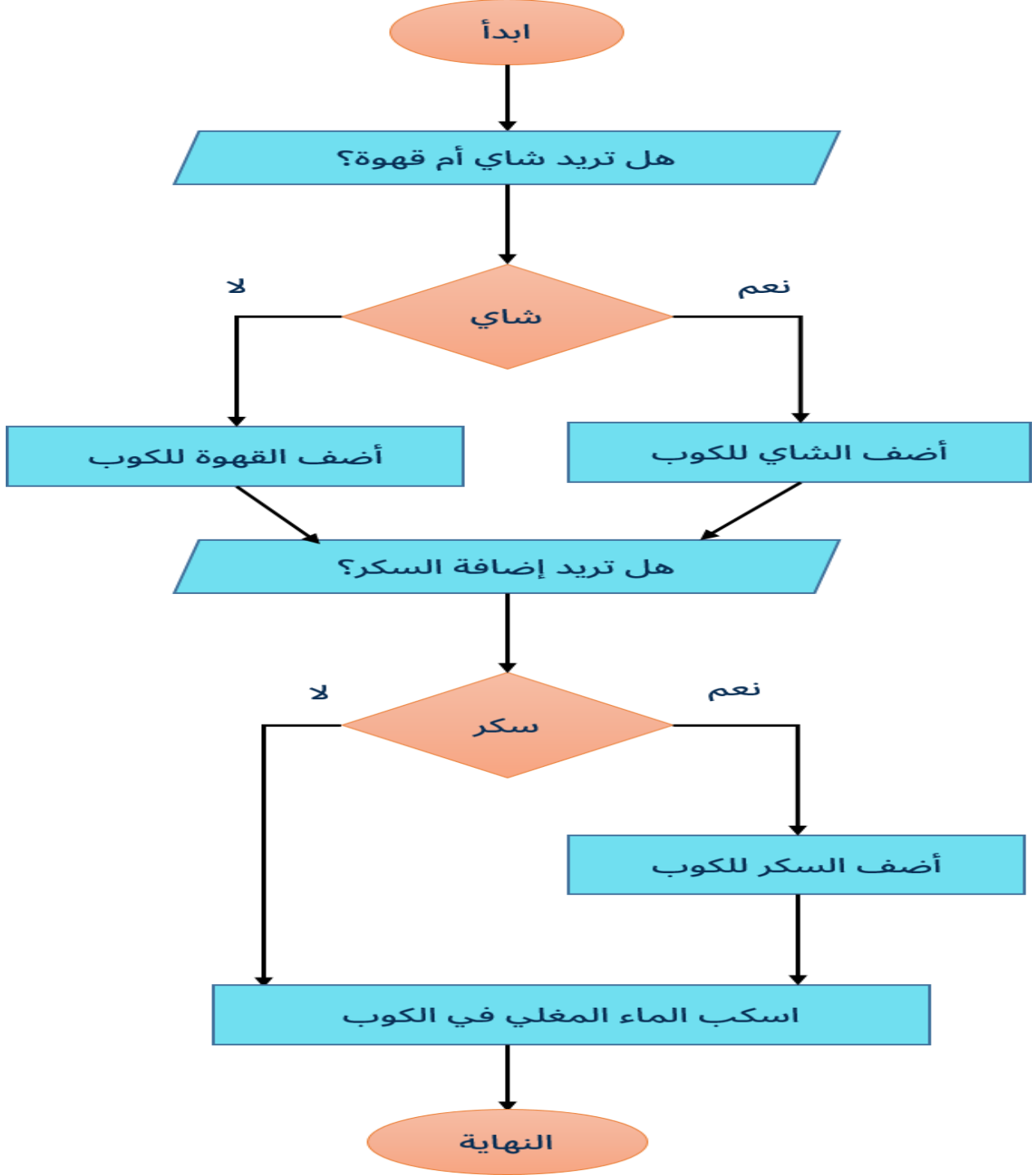
Use of VARIABLES AND CONSTANTS

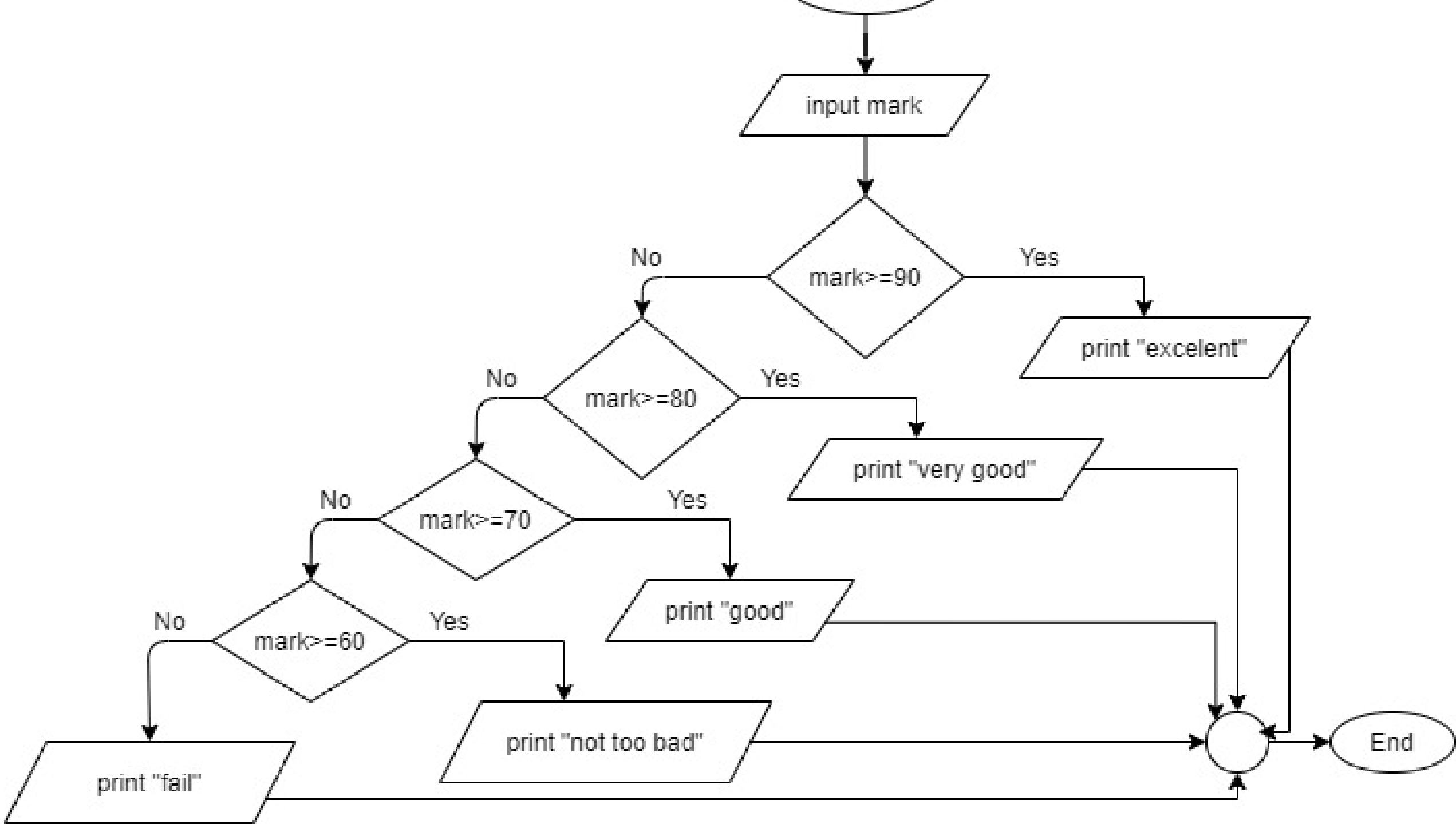
- **Variables:** store data that can change (e.g., marks, sum).
- **Constants:** hold fixed values (e.g., Nb_students = 3).

Consistent Keywords and OPERATIONS These words guide the reader, to understand the applied operations on the set of variables and constants

Commonly used keywords: Algorithm, Variables, Constant, Begin, Read, Write, End, etc.




ابدأ
هل تريد شاي أم قهوة؟
شاي
لا
نعم
أضف القهوة للكوب
أضف الشاي للكوب
هل تريد إضافة السكر؟
سكر
لا
نعم
أضف السكر للكوب
اسكب الماء المغلي في الكوب
النهاية


input mark
mark>=90
No
Yes
print "excelent"
mark>=80
No
Yes
print "very good"
mark>=70
No
Yes
print "good"
mark>=60
No
Yes
print "not too bad"
print "fail"
End

المتغيرات

- هو مساحة في ذاكرة الكمبيوتر تحمل اسمًا رمزيًا لتخزين قيمة قابلة للتغيير أثناء تشغيل البرنامج

- المتغير (**Variable**) عبارة عن مكان يتم حجزه في الذاكرة (**RAM**) بهدف تخزين قيمة معينة فيها أثناء تشغيل البرنامج.

Variables / Constants
المتغيرات والثوابت

Variables: store data that can change (e.g., marks, sum).

المتغير تحمل قيم قابلة للتغير

- Constants: hold fixed values (e.g., Nb_students = 3).

الثوابت تحمل قيم ثابتة غير قابلة للتغير

أنواع المتغيرات

الاعداد الصحية Integer

- مثل:-5, 0, 12, 2000

الأعداد الحقيقية/الفاصلة

Floatأو Double

مثل: 1.5 , 3.14 , -7.01

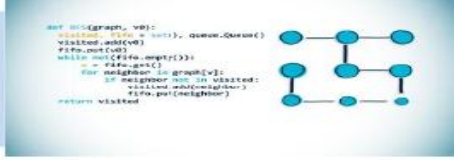
النصوص Strings مثل:

"Hello"

القيم المنطقيةBoolean

تأخذ أحد قيمتين فقط: Trueأو False

Data Declaration Rules & Types



"We said algorithms manipulate data, but how does the computer know what kind of data it's working with?

Is 12 a number? A letter? Or maybe a code like '12'?"

A **type** defines **the nature of data** and tells the computer **how to interpret and store** it.

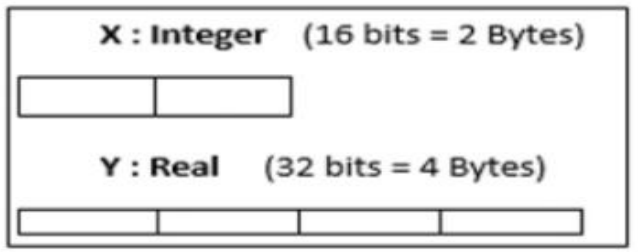
Basic data types

Type	Meaning	Examples
Integer	Whole numbers (positive, negative, or zero)	-2, -1, 0, 5, 12
Real	Numbers with decimals	3.14, -0.5, 100.0
Character	A single symbol (letter, digit, or special sign) written between ' '	'A', '9', '+'
String	A sequence of characters, written between " "	"Ahmed", "CS101", "0792/00"
Boolean	Logical value (truth-based)	True, false

Data Declaration Rules & Types

Representation of Types in Memory

- Every variable occupies a specific number of **bytes** in the **RAM**.
- The **type** determines how many bytes are needed and **how the value is encoded**.



Type	Example (C)	Size (approx.)	Representation
Integer	int x = 5;	4 bytes	Stored in **binary** form: 0000 0101
Real (float)	float y = 2.5;	4 bytes	Encoded using **IEEE 754** format
Double (real)	double z = 2.5;	8 bytes	Higher precision than float
Character	char c = 'A';	1 byte	Stored as **ASCII code**: 'A' → 65
Boolean	bool b = true;	1 byte	Stored as **1 (true)** or **0 (false)**

Data Declaration Rules & Types

Identify the Type of Each Variable

Variable Name	Description	Type
Student_Name	Full name of the student	.....
Student_ID	Registration number of the student	.....
Age	Student's age in years	.....
Average	Student's average grade	.....
Group	Student's group number	.....
Passed	Indicates whether the student passed the exam	.....
Gender	Student's gender ('M' or 'F')	.....
Math_Grade	Grade obtained in mathematics	.....
Birth_Date	Date of birth of the student	.....
Is_Redoubling	Indicates whether the student is repeating the year	.....
Comment	Teacher's remark ("Excellent", "Needs improvement")	.....

Data Declaration Rules & Types

Identify the Type of Each Variable

Variable Name	Description	Type
Student_Name	Full name of the student	String
Student_ID	Registration number of the student	String (or integer)
Age	Student's age in years	Integer
Average	Student's average grade	Real
Group	Student's group number	Integer
Passed	Indicates whether the student passed the exam	Boolean
Gender	Student's gender ('M' or 'F')	Character
Math_Grade	Grade obtained in mathematics	Real
Birth_Date	Date of birth of the student	String
Is_Redoubling	Indicates whether the student is repeating the year	Boolean
Comment	Teacher's remark ("Excellent", "Needs improvement")	String