الجامعة العربية الخاصة للعلوم والتكنولوجيا **AUST**

كلية الهندسة المعلوماتيةEngineering Faculty

قسم تقانة المعلوماتIT

- **المقرر: مدخل إلى البرمجة (١)**
- **المحاضرة : الثانية**

Dr. Mohammed AL-Mohammed

Chapter 2

لغات البرمجة

مثال: ارسم مخطط سير العمليات لحساب المقدار

$$s = \sum_{i=1}^{5} a_i$$

$$s=a_1+a_2+a_3+a_4+a_5$$

ولكن كلما زاد عدد الحدود فأن جمعها بواسطة الحاسب يصبح غير عملي ولحل هذه المشكلة لابد من ايجاد خوارزمية للحل تصلح لحل جميع المسائل المشابهة

اشتقاق خوارزمية الحل

١ . نفرض ان المجموع الابتدائي s_0 يساوي صفراً أي انه عند $i=0$ فان $s_0=0$

٢ . عند $i=1$ فان $s_1=s_0+a_1$

٣ . عند $i=2$ فان $s_2=s_0+a_1+a_2=s1+a2$

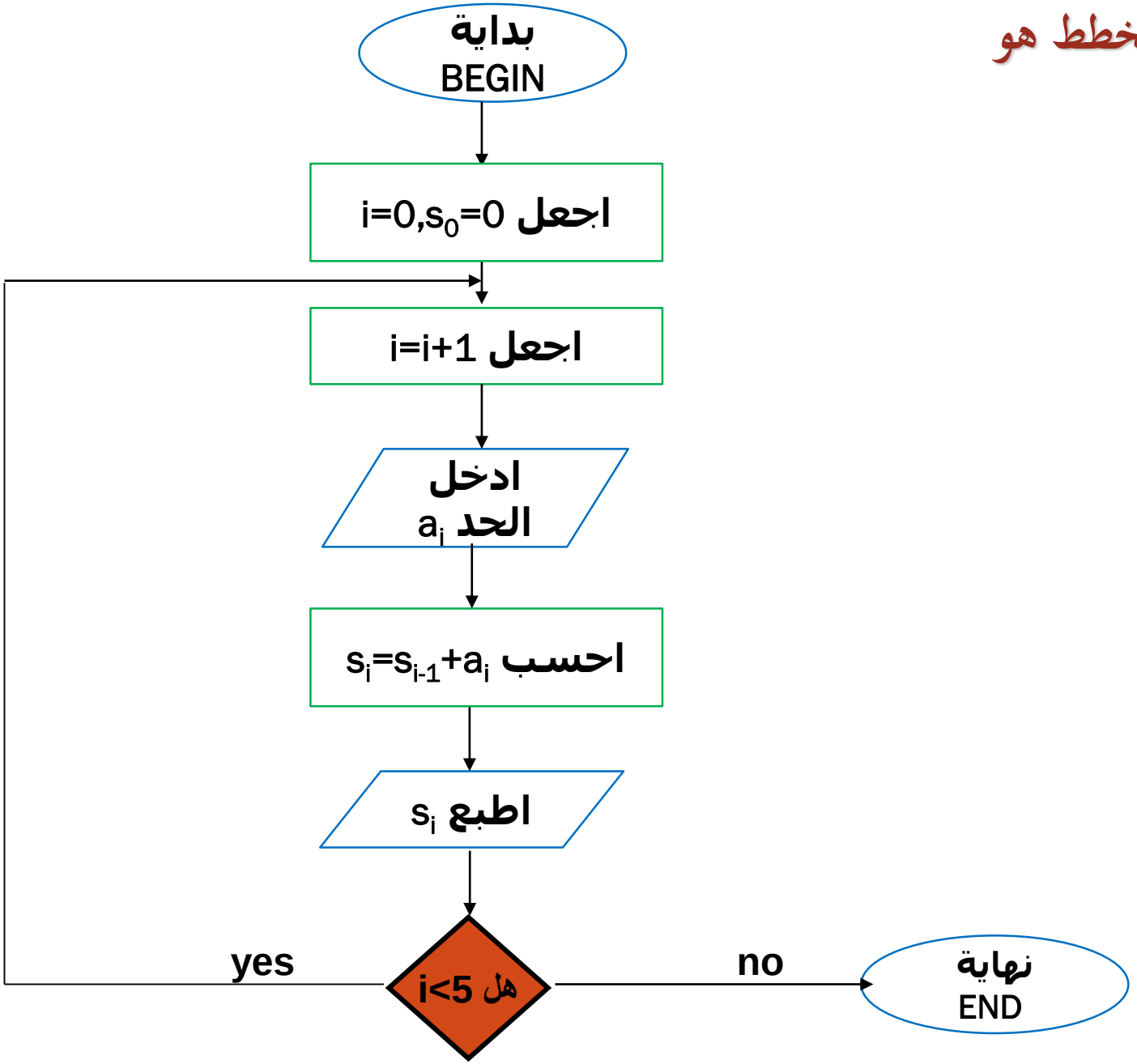
٤ . عند $i=3$ فان $s_3=s_1+a_2+a_3=s2+a3$

٥ . عند $i=4$ فان $s_4=s_3+a_4$

٦ . عند $i=5$ فان $s_5=s_4+a_5$ (المجموع النهائي)

وبشكل عام $s_i=s_{i-1}+a_i$

اذن المخطط هو



مثال: ارسم مخطط سير العمليات لإيجاد مجموع عناصر كل سطر(صف) من صفوف المصفوفة مجموع عناصر المصفوفة كلها

الحل: اذا كانت لدينا مصفوفة b مكونة من 3 صفوف و٤ اعمدة

$$b=\begin{bmatrix} b_{11} & b_{12} & b13 & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix}$$

$$s_1=b_{11}+b_{12}+b_{13}+b_{14}$$

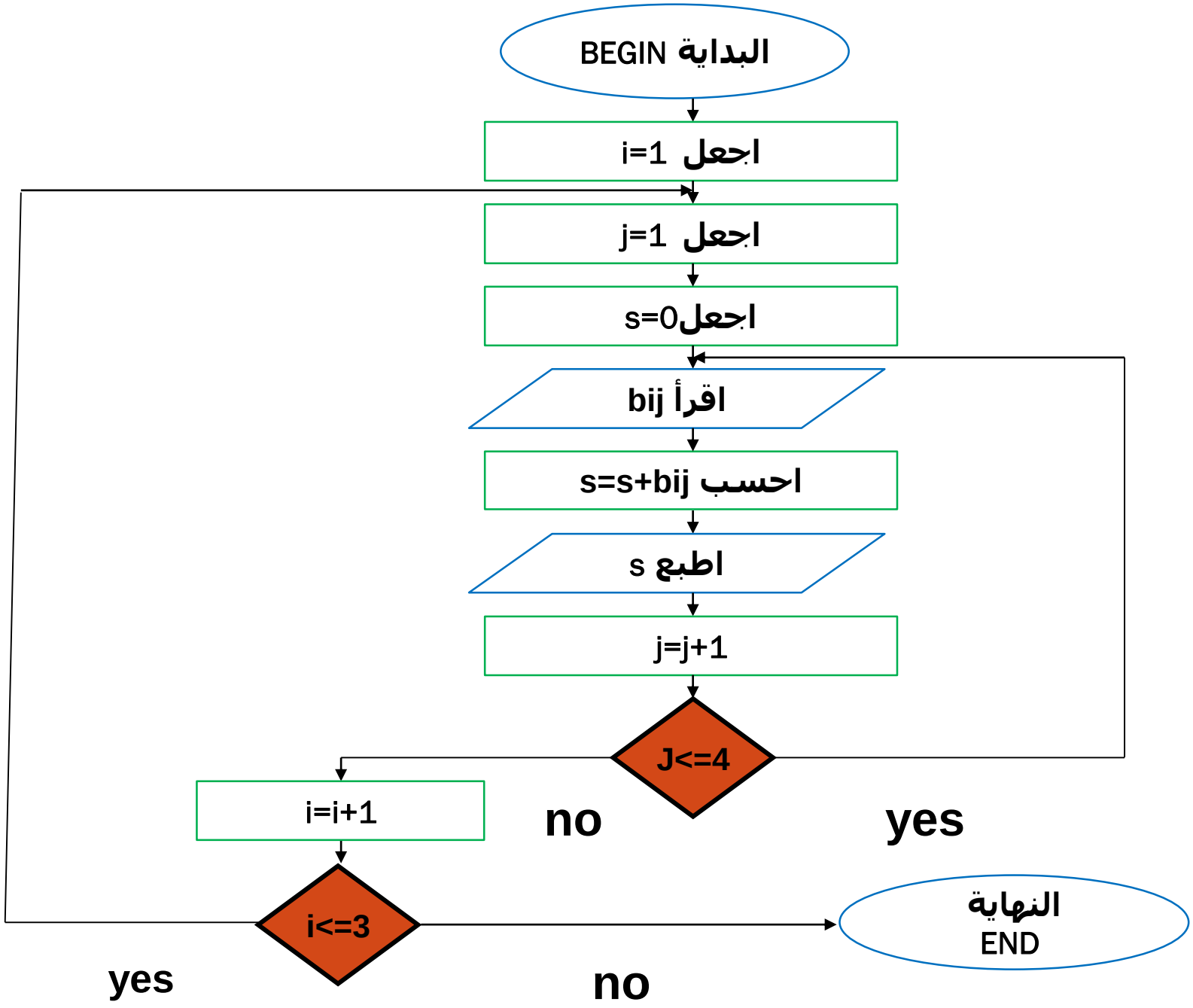
$$s_2=b_{21}+b_{22}+b_{23}+b_{24}$$

$$s_3=b_{31}+b_{32}+b_{33}+b_{34}$$

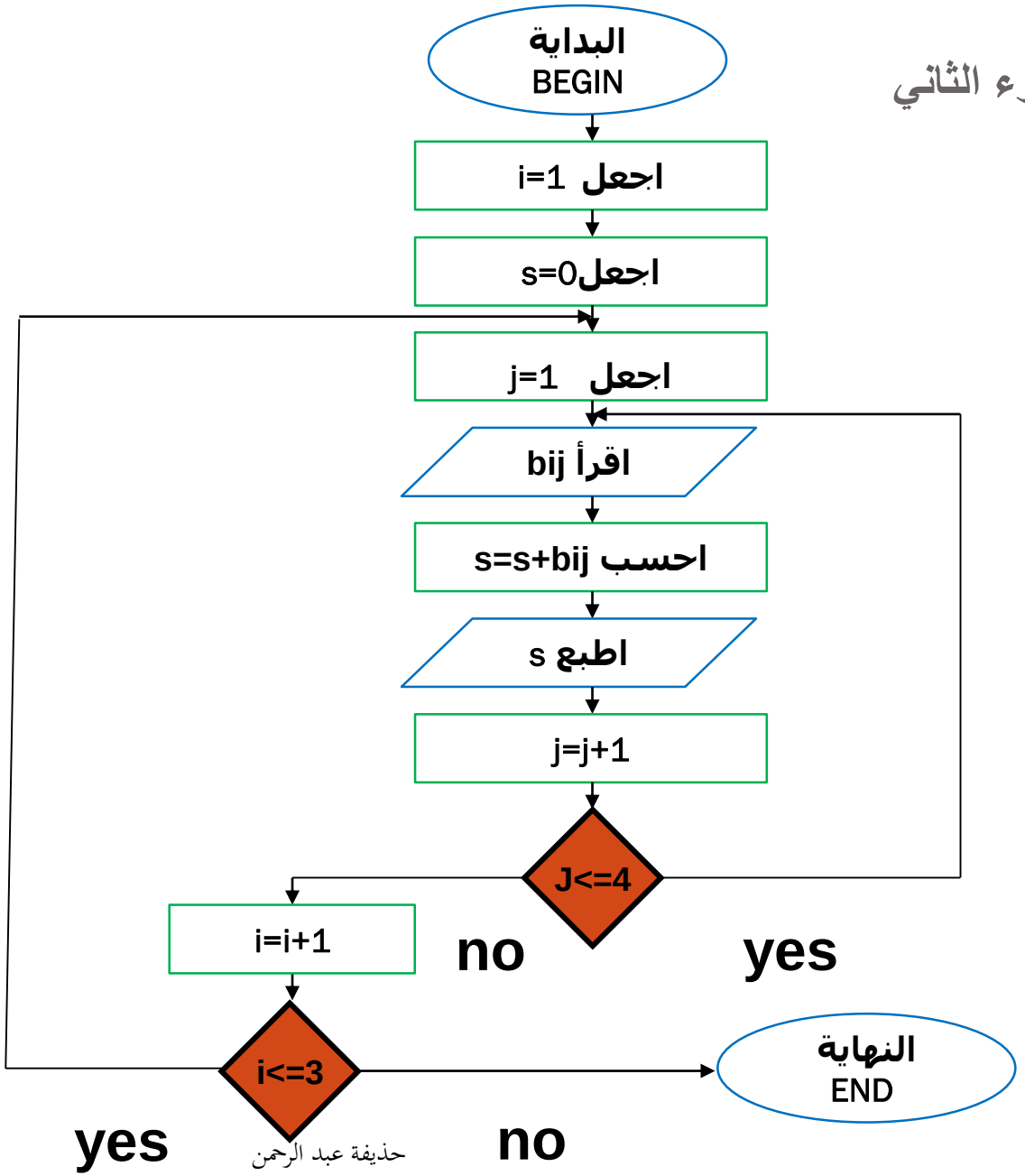
والمطلوب هو ايجاد s_1, s_2, s_3

اذن $s=s+b_{ij}$

وبالتالي فان مخطط سير العمليات للجزء الاول هو

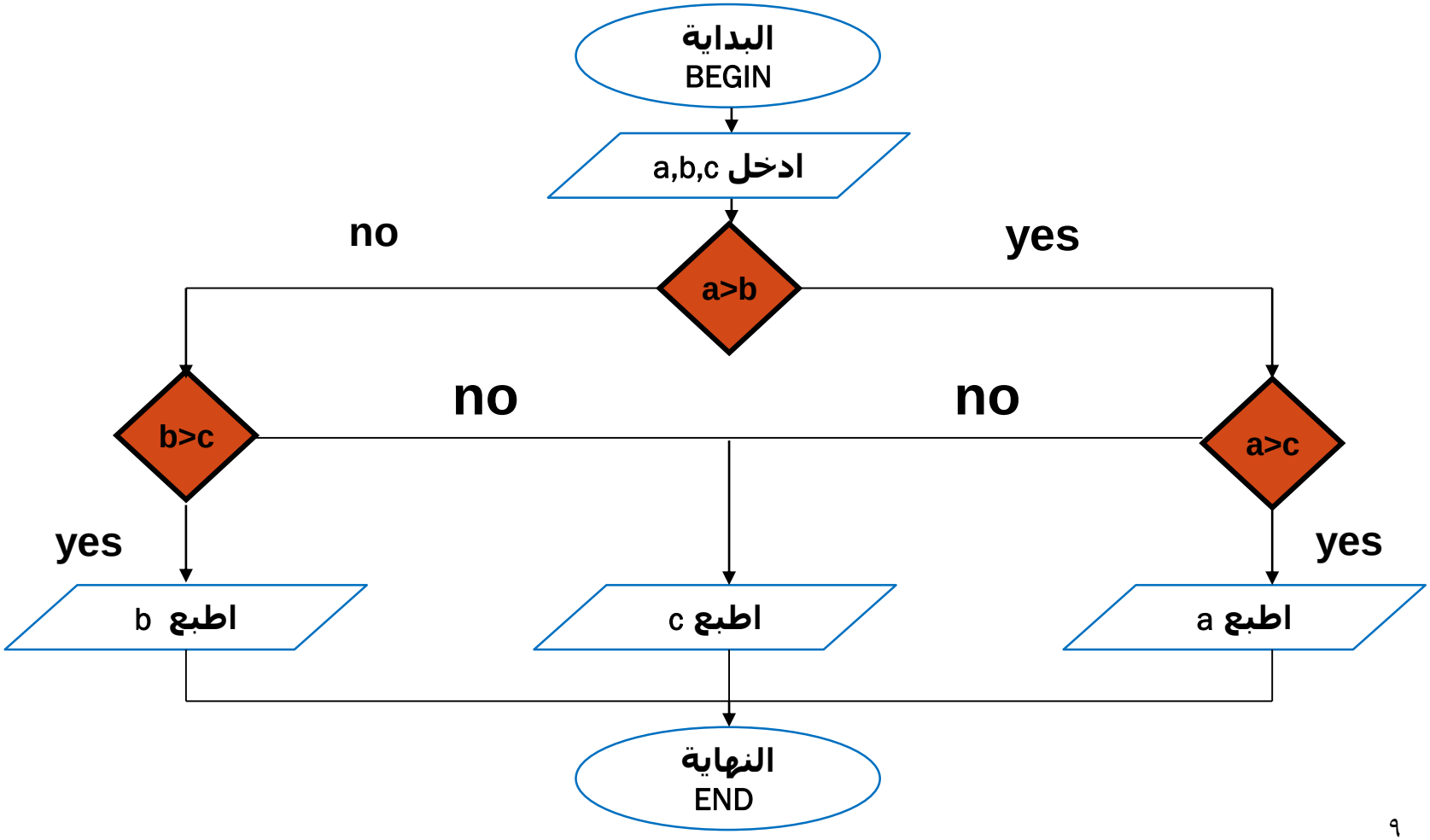

البداية BEGIN
اجعل i=1
اجعل j=1
اجعلs=0
اقرأ bij
احسب s=s+bij
اطبع s
j=j+1
J<=4
no
yes
i=i+1
i<=3
yes
no
النهاية
END

مخطط سير العمليات للجزء الثاني



مثال:ارسم مخطط سير العمليات لبرنامج يقارن بين ثلاثة اعداد موجبة غير متساوية ويطبع اكبر عدد بينها

الحل: نفرض ان الاعداد هي a,b,c



Levels of Languages

- Lower-level languages – more like the 0s and 1s the computer itself uses
- Higher-level languages – more like the languages people use
- Divided into five generations

Five Generations of Languages

- Machine language
- Assembly languages
- High-level languages
- Very high-level languages
- Natural languages

Major Programming Languages

- FORTRAN
- COBOL
- BASIC
- RPG
- Visual Basic
- C
- Java

FORTRAN

- The first high-level language
- Stands for FORmula TRANslator
- Used primarily for engineering, mathematical, and scientific tasks

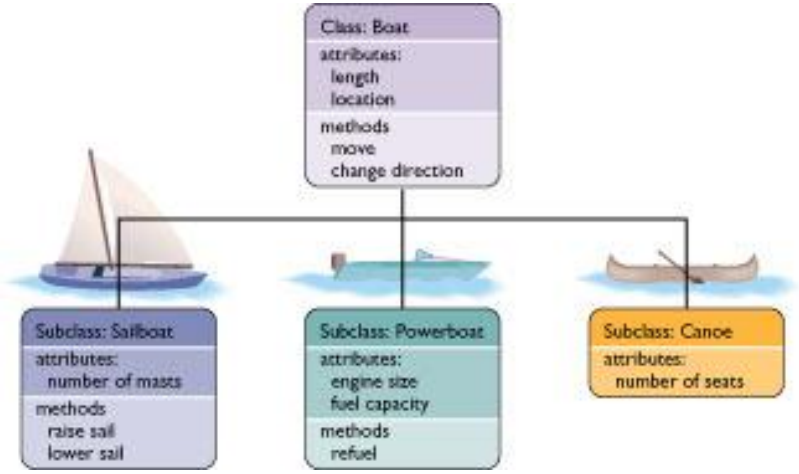
```
C     FORTRAN PROGRAM
C     AVERAGING INTEGERS ENTERED THROUGH THE KEYBOARD
      WRITE (6,10)
      SUM = 0
      COUNTER = 0
      WRITE (6,60)
      READ (5,40) NUMBER
    1 IF (NUMBER .EQ. 999) GOTO 2
      SUM = SUM + NUMBER
      COUNTER = COUNTER + 1
      WRITE (6,70)
      READ (5,40) NUMBER
      GO TO 1
    2 AVERAGE = SUM / COUNTER
      WRITE (6,80) AVERAGE
   10 FORMAT (1X, 'THIS PROGRAM WILL FIND THE AVERAGE OF',
     * 'INTEGERS YOU ENTER ',/1X, 'THROUGH THE ',
     * 'KEYBOARD. TYPE 999 TO INDICATE END OF DATA.'/)
   40 FORMAT (I3)
   60 FORMAT (1X, 'PLEASE ENTER A NUMBER ')
   70 FORMAT (1X, 'PLEASE ENTER THE NEXT NUMBER ')
   80 FORMAT (1X, 'THE AVERAGE OF THE NUMBERS IS ',F6.2)
      STOP
      END
```

Object-Oriented Programming

- Object – a self-contained unit that contains both data and its related functions
- Key terms in object-oriented programming
  - Encapsulation – an object isolates both its data and its related instructions
  - Attributes – facts that describe the object
    - Also called properties
  - Methods – instructions that tell the object to do something
  - Messages – an outside stimulus that results in the change of the state of an object

Using Objects

- Programmers define classes of objects
  - The class contains all attributes that are unique to objects of that class
  - An object is an instance (occurrence) of a class
- Objects are arranged hierarchically ir classes and subclasses
  - Subclasses are derived from classes
  - Inheritance – a subclass possesses all attributes of the class from which it is derived
  - Additional attributes can be coded in the subclasses



Activating the Object

- A message is sent to the object, telling it to do something
  - The object's methods tell it how to do it
- Polymorphism – each object has its own way to process the message
  - For example, the class may have a Move method, but each subclass implements that method differently

Object-Oriented Languages

- C++
- Java
- C#
- Visual Basic

مقدمة عن لغات البرمجة

نعلم أن دراسة الحاسب تنقسم إلى قسمين هما:

- العتاد Hardware.
- البرمجيات Software.

ولتسهيل الدراسة يتم تجزئة كل قسم على حده. فتم تقسيم العتاد إلى وحدات الإدخال، وحدات الإخراج، و وحدة النظام وتم تقسيم البرمجيات إلى **نظم التشغيل، لغات البرمجة، والبرامج التطبيقية**.

لغات البرمجة

تنقسم لغات البرمجة بصفة عامة إلى مستويين أساسين هما:

- لغات المستوى المنخفض Low-Level Languages.
- لغات المستوى العالي High-Level Languages.

و بالطبع هناك فارق كبير بين هذين المستويين في الإمكانيات، وسهولة التعامل مع الحاسب، بالإضافة إلى سهولة تعلم اللغة وفهمها. وبما أن لغات المستوى العالي تستخدم كلمات إنجليزية معينة ورموز رياضية مألوفة، فهي أسهل في تعلمها وفهمها.

لغات المستوى المنخفض Low-Level Languages

- تنقسم لغات هذا المستوى إلى قسمين هما:
- – لغة الآلة Machine Language.
- – لغة التجميع Assembly Language.

لغة الآلة Machine Language

هي أول اللغات ظهور، وهي اللغة الوحيدة التي يفهمها الحاسب مباشرة دون وسيط. وتتكون من رمزين هما: الصفر و الواحد.هذان الرمزان يعبران عن الأوامر المختلفة والبيانات التي يتكون منها البرنامج. إلا أن هذه اللغة صعبة التعلم وخاصة أن لكل حاسب لغة آلة خاصة به، وتتطلب معرفة واسعة في تصميم الحاسب، بالإضافة إلى صعوبة في اكتشاف الأخطاء. مما أدى إلى تطوير هذه اللغة إلى لغة التجميع.

لغة التجميع Assembly Language

تعتمد هذه اللغة على الرموز المختزلة، أي اختصارات لكلمات ذات مدلول لغوي محدد مثل:

ADD تدل على الجمع، و MOV تدل على النقل، وهكذا...، مما جعل تعلم هذه اللغة أسهل نسبياً من لغة الآلة. ولكن البرنامج في هذه اللغة يتم تجميعه وتحويله إلى لغة الآلة عن طريق ما يسمى بالمجمع Assembler. مما يؤكد أن الحاسب لا يتعامل مباشرة إلا مع لغة الآلة. ومع ذلك تبقى هذه اللغة صعبة التعلم، ولها عيوب من أبرزها ارتباطها بالآلة ، فكل آلة لها لغة تجميع خاصة بها، ويقصد بالآلة هنا تحديداً المعالج Processor .

بناءً على ما سبق نقول إن كتابة البرامج بلغات المستوى المنخفض تعتمد على معرفة واسعة بالتصميم الداخلي للحاسب (المعالجات، المقاطعات، مسارات البيانات، عناوين الذاكرة،...). مما جعل العلماء يفكرون بلغات تعزل المبرمج نسبياً عن التصميم الداخلي للحاسب.

لغات المستوى العالي High-Level Languages

- تعتمد هذه اللغات على كلمات إنجليزية واضحة المدلول مثل:write, read, input print، مما سهل تعلم هذه اللغات والإقبال عليها لحل المشاكل والتطبيقات العلمية والتجارية وغيرها. إلا أن تنفيذ البرنامج بهذه اللغات يحتاج إلى كشف الأخطاء وتتبع التعليمات خطوة خطوة وذلك عن طريق ما يسمى بالمفسر Interpreter ثم ترجمته وتحويله إلى لغة الآلة عن طريق ما يسمى بالمترجم Compiler.
- أما اللغات التي ظهرت في هذا المستوى فهي كثيرة جداً، من أبرزها وأشهرها: الفورتران *fortran* ، الكوبول *cobol* ، البيسك *Qbasic* ، الباسكال *pascal* ، السي *c* ، C++...الخ.

مقدمة إلى لغة C++

هي إحدى لغات المستوى العالي High level language التي تستخدم في كثير من التطبيقات وأهمها أنها تدخل في بناء نظم التشغيل وتعتبر امتداد إلى لغة c.

مزايا لغة C++ :توجد العديد من المزايا للغة c++ نذكر منها:

١. السرعة speed: تعتبر لغة C++ سريعة في إنجاز المهام.

٢. الحجم size :تحتوي على مجموعة ضخمة من الدوال جعلت منها لغة كبيرة نسبياً ولحل هذه المشكلة تم توزيع هذه الدوال على مجموعة من المكتبات المتخصصة ، كل مكتبة لها اسم معين وتحتوي على دوال معينة.

وبالتالي يحتاج البرنامج الواحد إلى مكتبات معينة يتم استخدامها في الوصول إلى الحل ولذلك أصبحت صغيرة الحجم.

مكونات برنامج C++

يتكون أي برنامج مكتوب بلغة c++ من أربعة عناصر أساسية هي:

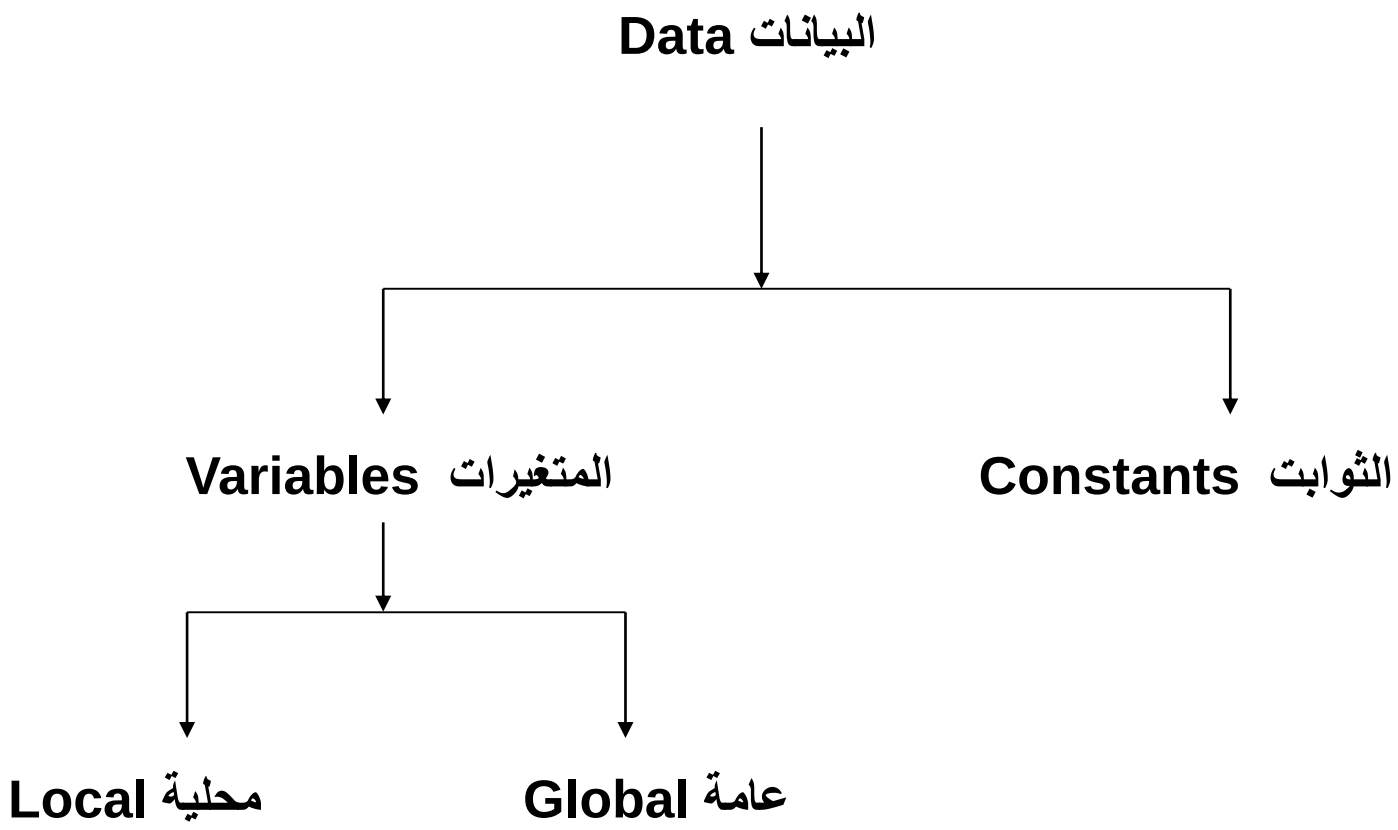
١. استدعاء المكتبات المستخدمة في البرنامج.

٢. تعريف المعطيات أو البيانات المستخدمة في البرنامج.

٣. جمل(عبارات)البرنامج

٤. إذا احتوى البرنامج على دوال مستخدم تكتب هذه الدوال في الجزء الأخير من البرنامج.

تعريف البيانات(المعطيات) data declaration



عادة عند تعريف المتغير يجب أن نحدد هل هو متغير عام بمعنى أنه يمكن استخدامه في جميع الأجزاء (على مستوى البرنامج)أم هو متغير محلي بمعنى أنه يستخدم داخل الجزء الخاص به فقط.

نميز بين المتغيرات العامة والمحلية من خلال وضع عملية التعريف داخل البرنامج

المتغيرات التي يتم تعريفها قبل بداية البرنامج (الدالة main) تمثل متغيرات عامة.

المتغيرات التي يتم تعريفها داخل الدالة main أو داخل أي برنامج فرعي تعتبر متغيرات محلية تستخدم فقط داخل الجزء المعرفة فيه.

الشكل العام لبرنامج c++:

Libraries Call	استدعاء المكتبات
Constants Declaration	الإعلان عن الثوابت
Global Variables Declaration	الإعلان عن المتغيرات العامة
main( )	الدالة الرئيسية
{	بداية البرنامج
Local Variables Declaration	الإعلان عن المتغيرات المحلية
Program Statements	عبارات البرنامج
return	مردود الدالة
}	نهاية البرنامج

استدعاء المكتبات Libraries Call

تحتوي لغة c++ على مجموعة من المكتبات،كل مكتبة تحتوي على مجموعة من الدوال وكل دالة لها وظيفة معينة تؤديها داخل البرنامج

الخطوة الأولى في البرنامج تتمثل في استدعاء المكتبات التي يتطلبها البرنامج وتسمى هذه العملية بالترويسة Header لذلك نجد المكتبات تنتهي أسماؤها بواسطة (.h)

ومن أشهر المكتبات في لغة c++ المكتبة المتضمنة دوال الإدخال والإخراج وتسمى iostream

الصيغة العامة لاستدعاء المكتبة:

#include <libarary Name.h>

اسم المكتبة

مثال: #include<iostream.h>

أمثلة لبعض المكتبات والدوال التي تحويها:

- المكتبة math تحوي كل الدوال الرياضية.
- المكتبة conio تحوي دوال التعامل مع الأحرف وتثبيت المخرجات.
- المكتبة string تحوي دوال التعامل مع السلاسل النصية.

مثال:-

إذا رغبنا في كتابة برنامج يحتوي على عمليتي الإدخال والإخراج بالإضافة إلى بعض الدوال الرياضية

فيجب استدعاء المكتبات التالية

```
#include<iostream.h>
#include<math.h>
```