## المحاضرة الثالثة

# لغات البرمجة ومدخل إلى لغة C++

## لغات البرمجة

نعلم أن دراسة الحاسب تنقسم إلى قسمين هما:

- الاجهزة Hardware.
- البرمجيات Software.

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

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.

## مدخل 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:

1. 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.

مثال:- المكتبة الرياضية math تحتوي على كافة الدوال الرياضية

مثل: sin,cos,tan…….

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

2. الإعلان الحر عن المتغيرات free variable declaration

أصبح من الممكن في لغة c++ الإعلان الحر عن المتغيرات في أي موقع من البرنامج مما يتيح ربط المتغير بالوظيفة التي من أجلها تم الإعلان عنه مما يزيد من سهولة ومتابعة وفهم البرنامج.

3. الإعلان عن الثوابت constants declaration

في لغة c يتم الإعلان عن الثوابت باستخدام الماكرو define # وفي هذه الطريقة لا يتم الإعلان عن نوع الثابت المعلن عنه أما في لغة C++ يتم استخدام الكلمة المحجوزة const للإعلان عن الثوابت وفيها يتم تحديد نوع الثابت ومن مزايا هذه الطريقة تساعد المترجم على فحص الأنواع type checking وحجز ذاكرة تتناسب ونوع الثابت.

4. الدوال الخطية Inline function

وهي ميزه تتعلق بالدوال ذات الحجم البسيط، حيث يتم إدخال سطور الدالة ضمن البرنامج الرئيسي أثناء زمن الترجمة حتى يتسنى تنفيذ البرنامج بشكل أسرع .

ملاحظة// الدوال ذات الحجم الكبير والدوال التي تحمل حلقات تكرارية لا يتم اعتمادها على أنها خطية.

5. التعليقات comments

وهي عبارة عن جمل إيضاحية وليس لها أي تأثير على تنفيذ البرنامج وتستخدم لتسهيل إعادة قراءة البرنامج أو تعديله من طرف المبرمج ويبدأ التعليق بالرمزين(*/)وينتهي بالرمزين(/*)سواءً كان التعليق لسطر واحد أو لعدة أسطر بينما الرمزين(//)يستخدمان للتعليق لسطر واحد.

**مثال:**

```
/* this is a comment statement */
// this is a comment statement
```

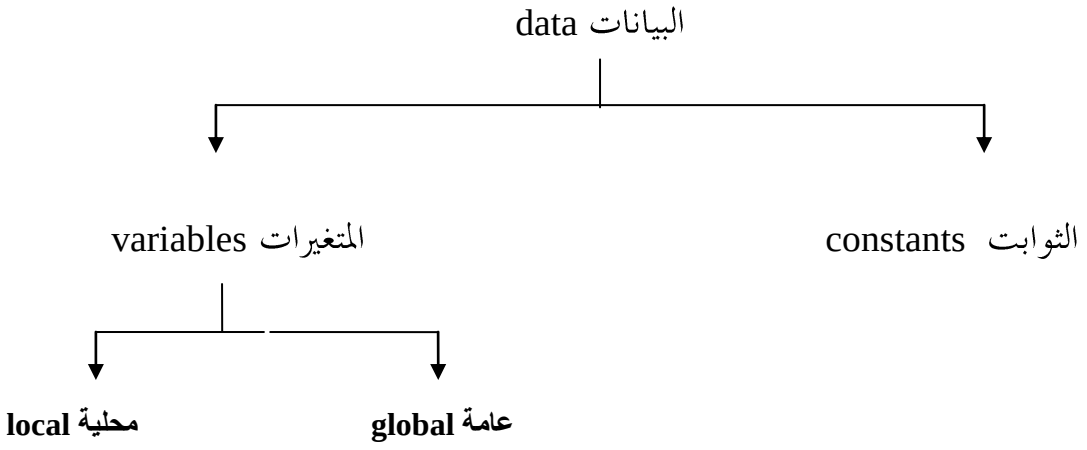
**مكونات برنامج c++:-**

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

1. استدعاء المكتبات المستخدمة في البرنامج.
2. تعريف المعطيات أو البيانات المستخدمة في البرنامج.
3. جمل(عبارات)البرنامج program statement

4. إذا احتوى البرنامج على دوال مستخدم user define functions(UDFs) تكتب هذه الـــدوال في الجـــزء 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 (علـــى مســـتوى 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

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

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

## جسم البرنامج program body :–

ويحتوي على تعريف المتغيرات المحلية وعبارات (جمل) البرنامج ويمكن أن يكتب في جزء واحد أو في مجموعـــة مـــن الأجزاء ونسمي الجزء في برنامج c++ بالدالة function

توجد دالة قياسية تمثل الجزء الرئيسي في البرنامج تسمى بالدالة main وقد يحتوي البرنامج على دوال فرعية أخـــرى تسمى بدوال المستخدم

## مكونات الدالة

اسم الدالة function_ name

بداية الدالة ويكون بالرمز {

الإعلان عن المتغيرات المحليةlocal variables

عبارات البرنامج program body

المردود return

النهاية {

## المردود:-

أي دالة لها مردود يمثل نتيجة المعالجة التي نجريها على المتغير المستقل فمثلاً $f(x)=x^2$ فعندما تكون f دالة 2 أي f(2) مردود هذه الدالة يساوي 4 نفس هذا الإجراء ينطبق تماماً على دوال برنامج c++ .

## 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

function_name اسم الدالة

{ بداية الدالة

local variables المتغيرات المحلية

statements العبارات

return المردود

} نهاية الدالة

**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:-**

1. المكتبة math تحوي كل الدوال الرياضية.
2. المكتبة conio تحوي دوال التعامل مع الأحرف وتثبيت المخرجات.
3. 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>
```