الجزء النظري (المحاضرة رقم ٠٤)

ماذا يجري فعليا في الحاسوب
(كيف يقوم الحاسوب بتمثيل وتنفيذ الجملة البرمجية داخل الحاسوب)

يتكون البرنامج في لغة البرمجة C++ من المكونات التالية:

١) الملفات الرأسية
٢) المتغيرات والاعلانات
٣) جسم البرنامج

لغة سي++ هي من اللغات عالية المستوى وتستخدم كلمات انجليزية تقريباً بمعناها الأصلي ولهذا تتطلب برنامجاً يسمى مترجم "Compiler" ليقوم بتحويل الأوامر في C++ إلى لغة الآلة فالكمبيوتر لايفهم اللغات عالية المستوى بشكل مباشر , وهناك أداة اخرى نحتاج إليها وهي المربط أو Linker وهو أداة تقوم بتجميع ملفات .obj الناتج من عملية الترجمة إلى ملف تنفيذي ويكون عادة بالامتداد .exe أو مكتبات الربط الديناميكية .dll , وأداة أخرى وهي محرر النصوص والذي سنكتب فيه السطور البرمجية مثل برنامج المفكرة الموجود على نظام الويندوز، في الماضي كانت هذه البرامج متوفرة كل واحد على حدة ويتم ربط البرنامج وترجمته عن طريق أسطر أوامر console ، أما اليوم فهي موجودة في برنامج واحد يحتوي على محرر النصوص والمترجم والمربط في آن واحد ولهذا يسمونه بيئة التطوير المتكاملة أو Integrated Development Environment (اختصاراً IDE) , ومن أشهرها بيئة الفيجوال ستوديو Microsoft Visual Studio المقدمة من شركة ميكروسوفت , كذلك يمكنك استخدام برنامج Code block .

ادناه مثال لبرنامج مكتوب بلغة البرمجة (C++) :

```
#include <iostream>
using namespace std;
int main()
{
   cout << "Hello world!" << endl;
   return 0;
}
```

```
Hello world!
```

بعد إعطاء امر التنفيذ يكون الناتج كما في أعلاه جملة الترحيب وسنوضح مكونات وعمل اسطر البرنامج :

#include <iostream>

في هذا السطر استخدمنا الكلمة #include ومعناها "تضمين" (والرمز # يلفظ هاش أو باوند ويعني توجيه) ثم الكلمة iostream وهو ملف رأسي (أي أن مايكتب بين العلامتين <> في العبارة #include <> يسمى ملف رأسي ويسمى أيضاً مكتبة دوال) , ويحتوي الملف الرأسي على دوال جاهزة تساعد على كتابة البرامج , والملف الرأسي iostream يحتوي على دوال الإدخال والإخراج وهو اختصار لـ Input Output Stream وتضمينه اجباري في البرنامج اذا اردنا القيام بالإدخال والاخراج .

using namespace std;

وتعني "قم باستخدام فضاء التسمية المسمى std" وتسمى تقنية الحقل المصرح حيث نستخدم فضاءات الاسماء (Namespaces) في برامجنا باستخدام الكلمة المحجوزة using لتخبر المترجم أن الدوال والمتغيرات مصرحة في مكان محدد من (أغلب الدوال التي سنستخدمها موجودة في فضاء التسمية "namespace" المسمى std) واذا حذفت هذه العبارة ستضطر الى أن تضع std:: قبل كل أمر في البرنامج وهذا متعب جداً , و std اختصار لكلمة Standard .

int main()

وهي الدالة الرئيسية في البرنامج وهي التي يبدأ من عندها تنفيذ البرنامج وتكون موجودة في أي برنامج مكتوب بلغة (C++) مهما كان حجمه أو وظيفته ونلاحظ أنها تبدأ بالقوس المعقوف { وتنتهي بقوس اغلاق } , اما int فتعني ان الدالة main سترجع لنا قيمة

عددية صحيحة (في بعض المترجمات مثل code block يتوجب عليك أن تجعل الدالة ترجع قيمة عددية صحيحة أما في المترجمات الأخرى مثل الفيجوال (C++) فيمكنك أن تجعل الدالة main لاترجع أي قيمة وذلك بأن تستبدل int بـ void وتمحو الجملة return 0) .

```
cout << "Hello world!" << endl;
```

قمنا بطباعة "Hello world!" على الشاشة وهذا باستخدام الكائن cout وتنطق (سي آوت) ونضع الشيء الذي نريد طباعته داخل علامتي الاقتباس " "، والرموز المستخدمة مثل << تسمى المعاملات أو Operators وسنتعرف عليها لاحقاً، والكائن cout موجود في المكتبة iostream .

```
return 0;
```

معنى return أي ارجاع وهذا السطر يحدد القيمة التي سترجعها الدالة main بعد انتهاء تنفيذها ويتوجب ارجاع القيمة ٠ اليها لينتهي تنفيذ البرنامج وستظهر رسالة خطأ إذا لم تكتب هذه العبارة.

ملاحظة :

ويجب عليك الانتباه ان جميع اوامر لغة (C++) تنتهي بفاصلة منقوطة (;) .

بعض الاخطاء الشائعه في كتابه البرنامج:

a) أن يكّون الخطاء في الملفات الرأسية وذلك بكتابه إسم المكتبه او الملفات بشكل خاطئ.
b) أن ينْسى المبرمج قوس البدايةّ ({) بعد دالة (main) وكذلك قوس النهاية (})
c) ان ينْسى المبرمج علامه (;) في نهاية كل عبارة .
d) تخزنّ مدخلات في مواقع غير معرفه .او تخزينها في غير الموقع المعرف به.

الأساسيات
مكونات C++ وادواتها
Basic Elements of C++

رموز لغة C++ : الرموز المستخدمة في لغة (C++)

١) الحروف الإنجليزية الكبيرة (A.B.C)
٢) الحروف الإنجليزية الصغيرة (a.b.c)
٣) الأرقام العربية الأصل (1.2.3)
٤) الرموز خاصة مثل :

+	-	<	!	"	[]
_	( )	>	||	'	*
/	\	>=	<=	<>	>>
<<	#	$	%	&	!=

كلمات لغة (C++) : الكلمات نوعين

١) أسماء تعريفية (identifiers) : وهي الأسماء التي نسميها نحن المبرمجون والتي تقوم بتعريف الحاسوب بما نريد . وتطلق الأسماء التعريفية على:
A) المتغيرات
B) الدوال (Functions)
C) المؤشرات (pointers)

القواعد الواجب اتباعها لتسمية الأسماء التعريفية في لغة (C++)

١. ان يكون الاسم مكتوبا من سلسلة متصلة من الحروف او الأرقام بشرط ان يبدأ بحرف او بخط تحتي " _ ".
٢. ان لايحتوي الاسم على رموز خاصة عدا الخط التحتي .
٣. ان لايكون الاسم احد الكلمات المحجوزة .

ومن الأمثلة على الأسماء :

a. B6
b. X_ray
c. Matrix
d. Ok_
e. A
f. Soft_fine
g. Door12
h. _new

اما في ادناه أسماء تعريفية غير مقبولة وكل حسب السبب الموضح

7-up ← لانه بدا برقم وليس بحرف.
b6.1 ← لاستعماله الرمز الخاص (.)
salim! ← لاستعماله الرمز الخاص (!)
Ъ2 ← لا يجوز استعمال حروف غير إنجليزية.
No#1 ← لاستعماله الرمز الخاص (#)

ويجب ان نبين ان لغة البرمجة (C++) تفرق بين الحروف الابجدية الصغيرة والكبيرة فمثلا الأسماء System و system و SYSTEM تعامل كأسماء مختلفة عن بعضها البعض بسبب اختلاف معاملة المترجم للحروف الصغيرة والكبيرة .

٢) **الكلمات المحجوزة:**

وهي كلمات قياسية معروفة مسبقا للمترجم الخاص بلغة (C++) وتكتب عادة بحروف صغيرة ، ولها معان خاصة بها تؤديها في برنامج (C++) وهذه الكلمات المحجوزة مبينه حسب الترتيب الابجدي في ادناه :

near	Static	asm	Double	long	Sizeof
do	int	While	new	auto	else
For	This	Void	Delete	Goto	if
const	Entry	char	Class	Public	Case
Continue	Extern	struct	inline	float	Private
Virtual	Volatile	Frinde	enum	near	Static
cdecl	Default	inline	Overload	Unsigned	Typedef
Signed	Pascal	Operator	Switch	Template	Union
Register	Protected	far	Catch	char	Const
				break	Return

ولايجوز استخدام هذه الكلمات لغير ما خصصت لأجله. وبهذا يكون عدد الكلمات المحجوزة في لغة البرمجة (C++) هو ٥٢ كلمة فقط .

الثوابت العددية (Numeric Constants)

يمكن تمثيل الثوابت العددية بثلاث صور هي :

a) الثابت العددي الصحيح (integer) : ويوضع وفق الشروط التالية

- هو عدد مكون من الأرقام (9 – 0) .
- لايحتوي على فاصلة عشرية .
- يمكن ان يحتوي على الإشارة (+ الموجبة) او الإشارة (- السالبة) .

ومن الأمثلة للثابت العددي مثل : (61- , 321 ,1000 , 15 , 0) .
ومن الأمثلة حول الاعداد غير الصحيحة وكل حسب السبب المبين امامه :

3.31 : لانه يحتوى على فاصلة عشرية.
1,000 : لانه يحتوى على فارزة.
J72 : لانه يحتوى على حرف أبجدي.
2 4 : لوجود فراغ بين العديدين.
1992 1999 : لوجود فراغ وأيضا لان العدد كبير.

كما يمكن تصنيف الاعداد الصحيحة حسب طولها والسعة التخزينية لها في الذاكرة وكما في ادناه :

الثوابت الصحيحة 19897 , 40000 تسمى ثوابت صحيحة طويلة long int.
الثوابت 16- , 80 , 45 تسمى ثوابت صحيحة قصيرة short int.
الثوابت 20000 , 967 تسمى ثوابت صحيحة بدون إشارة unsigned int.

ان الفرق بين الثوابت الطويلة والقصيرة هو في عدد الوحدات التخزينية المطلوبة لكل نوع في الذاكرة، فالطويلة تأخذ حيز اكبر والقصيرة توفر عدد الوحدات التخزينية المستعملة اما الثوابت الصحيحة بدون إشارة (unsigned int) فان استعمالها يوفر وحدة تخزينية واحدة تستعمل للإشارة عندما تذكر كلمة (unsigned) قبل (int) وذلك بازاحة القيمة الى قيمة موجبة بدون إشارة ولكل نوع من هذه الأنواع استخدامه الخاص .

b) الثابت العددي الحقيقي (Floating-Point Constants):

- **هو عدد مكون من الأرقام (9 – 0) .**
- **يجب ان يحتوي على فاصلة عشرية .**
- **يمكن ان يحتوي على الإشارة (+) او (-) .**
- **لا يجوز ان يحتوي على فارزة (،) .**

ومن الأمثلة : الصحيحة

421.5
10.6
0.0
0
01
-68.0

الأمثلة الغير صحيحة

1000 : لانه لا يحتوى علي فاصلة عشرية.
4,000.21 : لانه يحتوى على فارزة.
2 83.4 : لان يحتوى على فراغ .

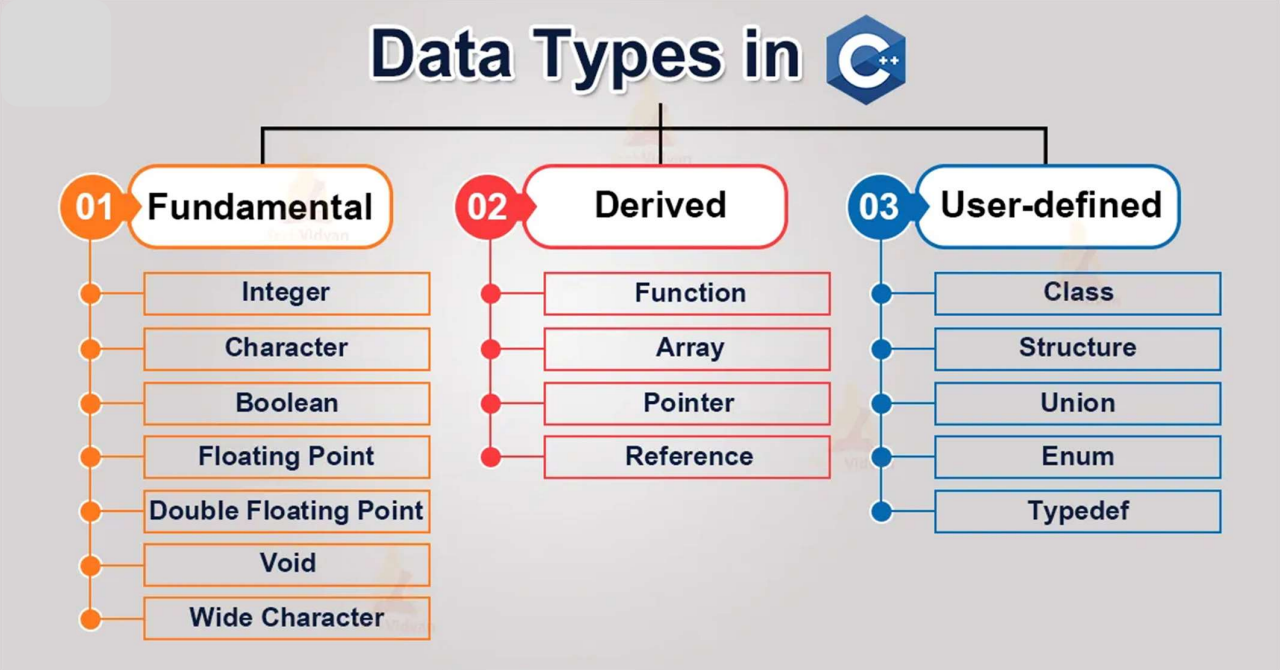
تمثيل الثوابت الرمزية (Non-numeric)

وهي سلسلة من رموز اللغة (احرف – ارقام – رموز خاصة) محصورة بين حواصر علوية مزدوجة (علامات تنصيص او اقتباس) (" ") وكما في الأمثلة ادناه :

```
"first"
"my name is"
"30+50=80"
"my,no=123.04"
"Islam"
```

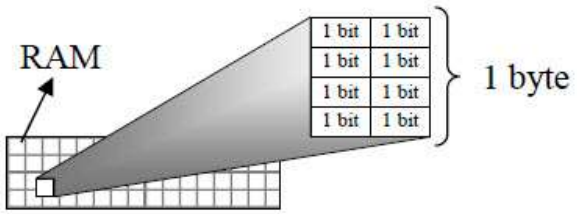
كل الثوابت الرمزية وان استخدمت أرقاما حسابية داخلها ولكن لاتحمل أي قيمة حسابية وليس لها معنى حسابي وتستخدم في العادة كمعلومات توضيحية مع نتائج البرنامج.

أنواع البيانات:



INTEGER	Bytes	REAL	Bytes	STRING	Bytes	LOGIC	Bytes
Short	2	Float	4	Char	1	Bool	1
Int	4	Double	8	String	8		
Long	4						

هذه المسميات موجودة في الذاكرة العشوائية (RAM) ولكل نوع من هذه الأنواع تقسيم (حجم) معين:



النوع	الحجم
Bit	2 (0/1)
Byte	8 bit
Kilobyte	1,000 byte
Megabyte	1,000,000 byte
Gigabyte	1,000,000,000 byte

- **(int) :** يستخدم للتعريف عن عدد صحيح ليس به كسر او فاصلة عشرية مثل (4-,10,2) ويستخدم كما في ادناه :

  Int a,b,c

  ويعني ان هناك موقع باسم (a) في الذاكرة يمكن نحفظ بداخله عدد صحيح .

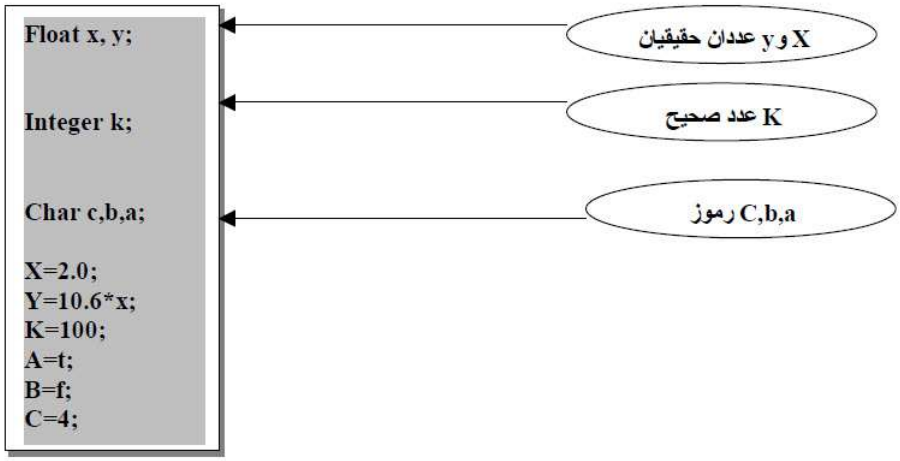
- **(char) :** يستخدم للتعريف عن حروف ويضم أيضا الرموز مثل (a , / , * , = , +) .
- **(flot) :** ويستخدم للتعريف عن عدد حقيقي او كسري مثل (5.0 , 1.4) .

القواعد التي يجب الالتزام بها في كتابة الأسماء التعريفية :

١) لايمكن ان يبدأ الاسم التعريفي برقم مثل (int 1a;) .
٢) لايمكن ان يحتوي الاسم على فراغ مثل (char a ge;) .
٣) لايمكن ان يحتوي الاسم على أي رمز خاص مثل (int %;) .
٤) لايمكن ان يكون الاسم التعريفي احد الكلمات المحجوزة
عند كتابة برنامج بلغة (C++) تكتب الأوامر والكلمات بحروف صغيرة .

المتغيرات Variable:

هي أسماء (عناوين) لمواقع في ذاكرة الحاسوب ، يخزن بها رموز او اعداد .
وبما ان أنواع المعلومات المراد تخزينها تكون عادة مختلفة مثل القيم الصحيحة او الحقيقية او الرمزية .... الخ فاننا نحتاج ان نعلم المترجم في بداية البرنامج عن أنواع المتغيرات التي نريد استعمالها في البرنامج ، وفي المثال ادناه نبين التالي :



- المتغيرين X و y حقيقيان .
- المتغير K صحيح .
- المتغيرات a و b و c رمزية

وتنقسم المتغيرات الى نوعين :

١) متغيرات عددية : وهي عبارة عن مواقع في الذاكرة تخزن بها اعداد .
٢) متغيرات رمزية : وهي مواقع في الذاكرة تخزن فيها رموز .
٣) متغيرات منطقية : وتخزن فيها قيمة منطقية اما (False=0) او (True=1) .

- **شروط تعريف المتغيرات :**

١) لايبدأ برقم او عملية حسابية او رمز ما عدا (_) (underscore) .
٢) الا يحتوي على عملية حسابية او رمز او فراغ .
٣) ان لايزيد عن ٢٥٥ حرفا .

مثـــال :

إسناد قيم للمتغيرات:

```
1. int x = 5 ;
2. x = 10 ;
3. x = 20 ;
4. x = 3 + 5 ;

5. cin >> x ;
6. char ch = 'y' ;
```

تعريف متغير x يحمل قيمة ابتدائية 5.
إسناد قيمة جديدة لـ x.
تغيير القيمة السابقة بقيمة جديدة أخرى.
إسناد ناتج القيمة الحسابية() للمتغير
(المتغير سيحمل القيمة ٨)
تغيير القيمة أثناء التشغيل من قبل المستخدم().
إسناد قيمة حرفية لمتغير (يكتب داخل تعليق مفرد ' ')