

بسم الله الرحمن الرحيم

تعلم البرمجة بلغة ++C

أن البرمجة بلغة ++C هي برمجة حديثة ولها فوائد كثيرة تختلف عما موجود في لغات البرمجة الأخرى، حيث تمتاز تلك اللغات بكونها إما أنها لغات صورية (تكوينية) أو انها لغات بيانية Data مثلاً:

لغة بيسك و فورتران و كوبل هي لغات بيانية.

لغة فبول بيسك (Visual Basic) هي لغة صورية.

اما لغة ++C فهي لغة تجمع بين الخاصيتين.

• رموز لغة ++C:

الرموز المستخدمة في هذه اللغة هي:

1- الحروف الانكليزية الكبيرة A, B, C, ...

2- الحروف الانكليزية الصغيرة a, b, c, ...

3- الأرقام 1, 2, 3, ...

4- رموز خاصة مثل:

[]	”	!	<	-	+
*	‘		>	()	=
>>	<>	<=	>=	\	/
!=	&	%	\$	#	<<

وتعد هذه الرموز المادة الخام التي تتكون منها لغة ++C.

• الكلمات في لغة ++C

الكلمات نوعين:

1- أسماء تعريفية (Identifiers).

وهي الأسماء التي نسميها نحن "المبرمجون" تُعرف الحاسوب بما تريد.

وتطلق الأسماء التعريفية على:

A- المتغيرات

B- الأختزانات (الدوال)

C- المؤشرات

• قواعد تسمية الأسماء التعريفية في لغة ++C

- 1- ان يكون الاسم مؤلفاً من سلسلة متصلة من الحروف والأرقام بشرط ان يبدأ بحرف أو بخط تحتي
"_"
- 2- أن لا يكون الأسم أحدى الكلمات المحجوزة.
- 3- أن لا يحتوي الأسم على رموز خاصة عدا الخط التحتي.

بعض الأمثلة الصحيحة على الأسماء التعريفية:

B6 -a

X-ray -b

Door 12 -c

والأمثلة التالية غير مقبولة للأسباب المبينة أزاء كل منها:

7-up لأنه بدأ برقم وليس حرف.

B6.1 لأستعمال الرمز الخاص (.).

NO#1 لأستعمال الرمز الخاص (#).

ملاحظة // لغة ++C تفرق بين الحروف الصغيرة والكبيرة فمثلاً الأسماء: SYSTEM, System, system تعامل كأسماء مختلفة عن بعضها البعض بسبب اختلاف معاملة المترجم للحروف الصغيرة والكبيرة.

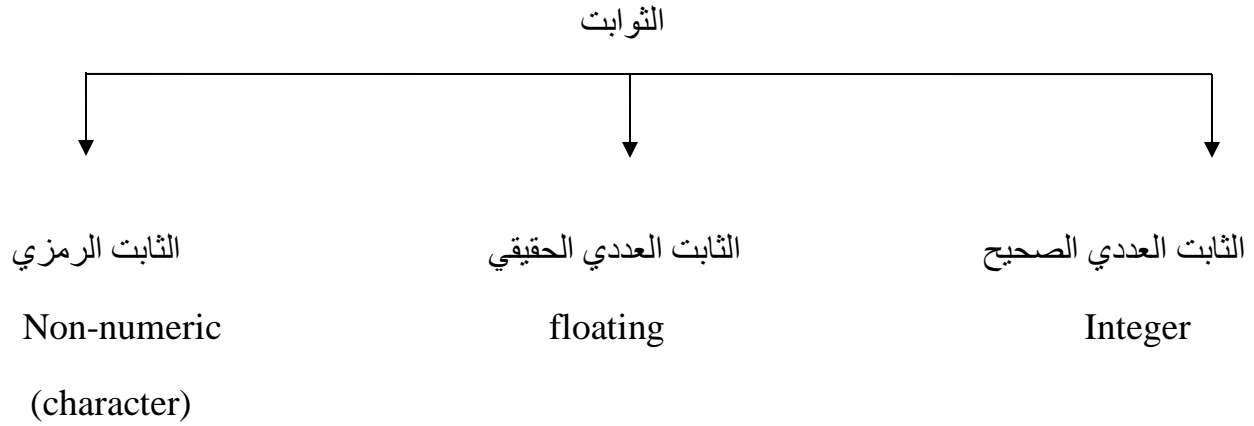
2- الكلمات المحجوزة

وهي كلمات قياسية معروفة مسبقاً لمترجم ++C، وتكتب بحروف صغيرة وينبغي التنبيه الى ان هذه الكلمات المحجوزة لايجوز إعادة تعريفها او استعمالها لغير ما خصصت له والجدول التالي يذكر بعضاً منها.

near	state	while	double	sizeof
do	int	new		else
for	void	delete	goto	if
const	char	class	public	case
continue	float	inline	private	default

• تمثيل الثوابت العددية Numeric constants

يمكن تمثيل الثوابت في لغة ++C بثلاث صور رئيسية هي:



-a الثابت العددي الصحيح integer

- هو عدد مكون من الأرقام من 0 إلى 9 .
- لا يحتوي على فاصلة عشرية.
- يمكن ان يحوي الإشارة "+" أو "-" .

أمثلة غير صحيحة

أمثلة صحيحة

3.21 لأنه يحتوي على الفاصلة العشرية.
1,000 لأنه يحتوي على فارزة.
J72 لأنه يحتوي على حرف أبجدي.
2 4 لوجود فراغ بين العددين.

0

15

1000

321

كما ويمكن تصنيف الأعداد الصحيحة في لغة ++C، حسب طولها والسعة التخزينية لها مثل:

ثوابت صحيحة طويلة long int

ثوابت صحيحة قصيرة short int

ثوابت صحيحة بدون إشارة unsigned int

-b الثابت العددي الحقيقي floating

- هو عدد مكون من الأرقام من 0 إلى 9 .
- يجب ان يحتوي على فاصلة عشرية.
- يمكن ان يحتوي على الإشارة "+" أو "-" .

الثابت	العملية	الثابت	النتيجة
int float float	+	int int float	int float float
int float float	-	int int float	int float float
int float float	*	int int float	int float float
int float float	/	int int float	int or float float float
int float float	%	int int float	int Not valid (false) Not valid (false)

-c الثوابت الرمزية (Non-numeric (character))

سلسلة من رموز اللغة (أحرف، أرقام، رموز خاصة) محصورة بين حواصر علوية مزدوجة (علامات اقتباس أو تنصيص)، مثال:

"first"

"my name is"

"30+50=80"

"my, no=123.04"

وكذلك يستخدم نوع اخر من المتغيرات هو (المتغيرات المنطقية) Boolean قيمها اما True او False ويمكن التعبير عنها بـ bool.

ملاحظة// السلاسل الحرفية ممكن ان تمتد الى اكثر من سطر واحد وذلك عن طريق وضع العلامة (\) عند نهاية كل سطر (خط) غير منتهي.

"string expressed in \

two lines"

مما يتضح ان هناك بعض الرموز الخاصة الواجب ذكرها هنا وتعتبر من خصائص كائن الأخراج cout الذي سيتم التطرق له لاحقاً، والقائمة التالية توضح بعض من تلك الرموز التي قد تكون مهمة في هذا المجال.

الرمز (سلسلة الافلات)	المعنى (التعبير)
\t	جدولة افقية تترك 3 فراغات أو اكثر (tab)
\n	الانتقال الى سطر جديد new line
\r	إعادة المؤشر الى بداية السطر carriage return
\a	يقوم باصدار صوت تنبيه alert (beep)
\b	الحذف الخلفي backspace
\v	جدولة عمودية vertical tab

• المتغيرات

هي اسماء (عناوين) لمواقع في ذاكرة الحاسوب، يخزن بها رموز او اعداد. وبما ان انواع المعلومات المراد تخزينها تكون عادة مختلفة مثل القيم الصحيحة او الحقيقية او الرمزية الخ، فانك تحتاج ان تعلم المترجم في بداية البرنامج عن انواع المتغيرات التي تريد استعمالها في البرنامج كما في المثال التالي:

float x,y;

x,y عدنان حقيقيان

int k;

k عدد صحيح

char a,b;

a,b رموز

x=2.0;

y=10.6*x;

k=100;

a="t";

b="4";

في هذا المثال تم تعيين جميع قيم المتغيرات بأستخدام عامل الاسناد = والذي يسند ما موجود على يمينه الى ما موجود على يساره.

هناك عامل اخر سوف نتطرق له لاحقاً وهو ايعاز الادخال cin.

الأدوات (المؤثرات) المستعملة في لغة ++C

يوجد ثلاثة انواع من الادوات وهي:

1- الادوات الحسابية (المؤثرات الحسابية) Arithmetic operators

2- الادوات المنطقية والعلاقة.

3- الادوات الدقيقة.

1- الأدوات (المؤثرات) الحسابية

يمكن ان نلخص الادوات الحسابية المستعملة في لغة ++C فيما يأتي:

وظيفةها	الأداة
للطرح أو كأشارة سالبة	-
للجمع	+
للضرب أو (المضاعفة)	*
للقسمة	/
لباقى القسمة الصحيحة	%

ملاحظة // عند اجراء عملية القسمة لعددتين صحيحين مثلاً لقسمة 8 على 3 فهي قسمة صحيحة (8/3) فان الناتج يكون صحيحا وهو 2 فقط، واذا ما رغبت ان تحافظ على الجزء الكسري الذي اهمل، يمكنك ان تحول القسمة الى قسمة حقيقية (8.0/3.0) حينئذ فإن الناتج سيكون (2.667) لهذا السبب ادخلت لغة ++C أداة باقى القسمة % ويسمى Modulus Operator ويستعمل على النحو التالي:

a=7%3;

قيمة a هي 1.

• التعبير وعمليات الاسناد Assignment Operator and Expressions

1- إشارة المساواة او التعيين (=) Assignment

a=5;

الجزء على يسار مؤثر المساواة يعرف بالقيمة اليسرى والتي تكون على شكل متغير ما، إما الجزء الأيمن فيمثل قيمة المتغير (ثابت). كما ويمكن لهذا المؤثر ان يسند المتغير الذي في يمينه الى الذي في يساره كما في المثال:

a=b;

في هذا السطر المتغير a يأخذ قيمة b مع بقاء قيمة b ثابتة.

كما ويمكن ان يستخدم مع المتغيرات الحرفية كما يلي:

char x= "y";

تمتاز لغة ++C بمؤثر المساواة والذي يمكن استخدامه بالطريقة التالية:

a=2+(b=5);

```
b=5;
```

```
a=2+b;
```

والذي يكافئ

والتعبيرات التالية ايضا صحيحة في لغة C++ وهي:

```
a=b=c=5;
```

```
i=i+2
```

لنفرض ان لدينا متغير عددي هو i ونريد جمعه بالعدد 2 حينها سنقوم بكتابة:

لذا فان هذه اللغة توفر لنا معامل اسناد اسرع من معامل الاسناد = وأكثر اختصارا هو (=+) كما في المثال التالي:

```
i+=2;
```

مثال بسيط //

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

```
void main ( )
```

```
{
```

```
int a,b; //a:?, b:?
```

```
a=10; //a:10,b:?
```

```
b=4; //a:10, b:4
```

```
a=b; //a:4, b:4
```

```
b=7; //a:4, b:7
```

```
cout<<"a:";
```

```
cout<<a;
```

```
cout<<"b:";
```

```
cout<<b;
```

```
}
```