مقدمة في البرمجة بلغة C++

* أنواع اللغات :

يمكن تقسيم اللغات المستخدمة في البرمجة إلى ثلاثة أنواع:

1- لغة الآلة ، 2. لغة المجمع 3. اللغات العالية المستوى

1. لغة الآلة:

هي اللغة التي يستطيع الحاسب أن يفهمها مباشرة وهي معرفة من قبل البنية الصلبة للحاسب ، تتألف بشكل عام من سلاسل من الأعداد (مجموعات من الأصفار والواحدات) التي تعطي الأوامر للحاسب من أجل تنفيذ تعليماته الأولية كل تعليمة على حده.

ترتبط هذه اللغة ارتباطاً وثيقاً بالآلة machine-dependent وهذا يعني أن لغة آلة ما لا تستخدم إلا لنفس النوع من الآلات فقط .

2. لغة المجتمع :

هي لغة تستخدم مصطلحات قريبة من اللغة الإنكليزية للتعبير عن العمليات الأولية للحاسب، وقد تم تطوير مترجمات للبرامج تسمى بالمجمعات assemblers تحويل البرامج من لغة المجمع إلى لغة الآلة.

3. اللغات العالية المستوى :

هي اللغات التي ظهرت لتسريع عملية البرمجة وذلك باستخدام تعليمات تقوم بالعديد من المهام الجوهرية ، وتهد اللغات C++,C من أكثر اللغات العالية المستوى قوة وانتشاراً .

- تدعى البرامج التي تقوم بتحويل النصوص من البرامج مكتوبة بلغات عالية المستوى إلى لغة الآلية بالمترجمات.

ملاحظات:

1- ÷ يوجد بعض المفسرات interpreter programs التي تستطيع تنفيذ البرامج المكتوبة بلغات عالية المستوى مباشرة دون الحاجة إلى ترجمة هذه البرامج إلى لغة الآلة.

2- البرامج المترجمة هي أسرع تنفيذاً من البرامج المفسرة عموماً .

* البرمجة بلغة C++:

تسهل لغة C++ الأسلوب المهيكل والمنهجي لعملية تصميم البرامج ، حيث تتألف برامج هذه اللغة من مكونات تسمى الصفوف classes والتوابع Functions وبالتالي يمكن تقسيم عملية تعلم لغة الـC++ إلى قسمين : يعتمد الأول منها على تعلم لغة الـC++ نفسها في حين يسمح الثاني بتعليم كيفية استخدام الصفوف الملحقة بهذه اللغة واستخدام التوابع الموجودة ضمن المكتبة المعيارية ANSI C.

*** مراحل تنفيذ برامج C++ :**

يتم التنفيذ خلال ست مراحل هي بالشكل التالي:

- **مرحلة الكتابة ضمن Edit** : وهي كتابة نص البرامج في أي محرر نصوص يستخدم لكتابة البرامج بلغة C++ .
- مرحلة ما قبل الترجمة ؛ Preprocess : هي تصحيح البرنامج من الأخطاء ومن ثم تخزينه على وحدة تخزين ثانوية مثل الأقراص بتوسع CPP, CXX وذلك حسب بيئة العمل.
- **مرحلة الترجمة Compile** : هي ترجمة البرنامج إلى لغة الآلة.
- **مرحلة الوصل Linking**: تتضمن برامج الـ C++ استدعاءات لتوابع تم تعريفها في مكان آخر مثل المكتبات المعيارية ، وبالتالي مهمة هذه المرحلة هي استخدام الواصل Linker لوصل الملف مع نصوص التوابع الناقصة من أجل الوصول إلى صورة قابلة للتنفيذ .
- **مرحلة الشحن Loading** : قبل تنفيذ البرنامج يجب وضعه في الذاكرة وذلك باستخدام الشاحن Loader الذي يقوم بأخذ الملف التنفيذي ونقله إلى الذاكرة.
- **مرحلة التنفيذ Execute**: هي مرحلة التنفيذ التي تتم تحت إشراف وسيطرة وحدة التحكم والمعالجة CPU .

* أمثلة بسيطة : لتعلم مبادئ أساسية في لغة ++ C –

1– طباعة نص مؤلف من سطر :

كل الكتابات التي تلي هذه الإشارة (//) تسمى تعليق لا يتم تنفيذه

```
// First Program
# include<iostream.h>
```

// (.h) توجيه ما قبل الترجمة حيث يتم ضمن محتوى

الملف الرأسي ذو الامتداد الحاوي على العمليات الخاصة بالدخل والخرج لنص البرنامج

```
main ( )
{
cout <<  " welcome to c++ " ;
return 0 ;
}
```

// التابع الرئيسي الذي يبدأ من عند التنفيذ

// بداية البرنامج

// تعليمة الطباعة

// إحدى طرق الخروج من التابع

// نهاية البرنامج

```
(Inactive C:\TCWIN45\BIN\NONAME00.EXE)
welcome to c++
```

2 . برنامج جمع عددين صحيحين :

```
# include < iostream.h>
main ( )
{
  int x1 , x2, x3 ;
  cout <<" enter first numbe ";
  cin >> x1 ;
```

// تعريف المتحولات

// تعليمة الطباعة

// تعليمة قراءة متحول

```
  cout << " enter second number ";
  cin >> x2
  x3 = x1 + x2 ;                    //x3 إجراء عملية الجمع والإسناد إلى المتحول الجديد
  cout << "sum is " <<x3 ;          // تعليمة الطباعة المتعددة
return 0 ;
}
```

```
(Inactive C:\TCWIN45\BIN\NONAME01.EXE)
enter first numbe 10
enter second number 55
sum is 65
```

ملاحظة:

1- مفهوم يتعلق بالذاكرة ألا وهو طريقة حجز المتحولات:

كل اسم من أسماء المتحولات مثل .... , x3 , x2 ,x1 يتم وضعه في الذاكرة ويعرف بإسم name ونمط type وحجم size وقيمة value وبالتالي فإن المتحول x1 يملك الاسم x1 يملك الاسم x1 والنمط int والحجم 2 بايت والقيمة هي حسب القيمة المقروءة .

x1	5
x2	10
x3	15

مواضع المتحولات في الذاكرة مع ذكر الاسم والقيمة

2. أنواع المتحولات:

- المتحول التعدادي enum
- المتحول المحرفي char
- المتحولات الصحيحة short int, int , long int , unsigned sort int, unsigned int , unsigned long int.
- المتحولات الحقيقة float , double , long double

وبين الجدول التالي أنواع المتحولات ومجالاتها :

نوع المتحول	المجال
char	−128 to 127
int	−32768 to 32767
unsigned int	0 to 65535
short int	−32768 to 32767
Unsigned short int	0 to 65535
Long int	−2147483648 to 214748364
float	−3.4E−38 to 3.45E+38
double	−1.7E−308 to 1.7E+308
long double	−3.4E−4932 to 1.1E+4932

* **العمليات الحسابية** :

اسم العملية	الرمز الحسابي	طريقة التعبير حسب لغة C++
الجمع	+	x1 + x2
الطرح	−	x2 − x1
الضرب	*	x1 * x2
القسمة	/	x1 / x2
باقي القسمة الصحيحة	%	x1 % x2

تقوم C++ بتطبيق العمليات في العبارات الحسابية حسب ترتيب معين محدد تبعاً لقواعد الأولوية بين العمليات التي تماثل قواعد الأولوية في الجبر وذلك كما في الجدول التالي:

العملية	اسم العملية	ترتيب عملية التقسيم (الأولوية)
( )	الأقواس	**تقييم أولاً ، إذا وجد في العبارات الحسابية أقواس متداخلة ضمن بعضها البعض فالحساب يبدأ انطلاقاً من أول مجموعة في الداخل أما إذا كان لدينا مجموعة من الأقواس جانب بعضها البعض وعلى نفس المستوى عندها يبدأ الحساب من اليسار إلى اليمين.**
* ، / ، %	الضرب، القسمة ، باقي القسمة	**تقييم ثانياً ، إذا وجدت على نفس المستوى فإنها تقييم من اليسار إلى اليمين .**
+ ، −	**الجمع ، الطرح**	**تقييم في النهاية ، إذا وجدت على نفس المستوى فإنها تقييم من اليسار إلى اليمين .**

أما بالنسبة لعمليتي الإسناد والمقارنة فتتم بالشكل التالي: جميع العمليات الحسابية يتم تجميعها من اليسار إلى اليمين إلا عملية الإسناد تتم من اليمين إلى اليسار .

الشكل الجبري	الشكل الموافق حسب c++	مثال	معنى الكتابة
=	= =	x = = y	x تساوي y
≠	! =	x != y	x لا تساوي y
<	<	x <y	x أصغر من y
>	>	x >y	x أكبر من y
≤	<=	x < = y	أصغر أو يساوي y
≥	>=	x > = y	x أكبر أو يساوي y

***العملية المنطقية Logical operators :**

وهي ثلاثة :

And يرمز لها &&

Or يرمز لها ||

Not يرمز لها !

*** سلاسل الهروب :**

```
# include <iostream.h>
main ( )
{
  Cout <<"welcome to c++\ n " ;
```

حرف الهروب

```
return 0;
}
```

يدعى \ بحرف الهروب وهو يلحق بحرف يدل على معنى معين كما هو موضح في الجدول:

سلسلة الهروب	المعنى
\ n	سطر جديد أي وضع المؤشر في بداية السطر التالي
\ t	تحريك المؤشر مسافة جدولية أفقية
\"	تستخدم لطباعة علامة الاقتباس

*** بعض الأمثلة:**

1- أكتب برنامجاً يأخذ كدخل ثلاث أعداد صحيحة من لوحة المفاتيح ثم يطبع مجموعها ومتوسطها وناتج جداؤها.

```
# include < iostream.h>
main ( )
{
int a , b, c ;
cout << " enter a =" ; cin >> a ;
cout << " enter b = " ; cin >> b ;
cout << " enter c = " ; cin >> c ;
cout << " sun is " << a+b+c << " \ n" ;
cout << average is " << ( a+b+c)/3 <<" \n";
cout << product is " << a * b* c;
return ;
}
```

```
(Inactive C:\TCWIN45\BIN\NONAME02.EXE)
enter  a = 10
```

enter b = 20

enter c = 33

sun is 63

average is 21

product is 6600

2- أكتب برنامج يقرأ نصف قطر دائرة ثم يطبع قيمة قطر الدائرة ، محيطها ، مساحتها .

ملاحظة : قيمة π =3.14

```
# include <iostream.h>
main ( )
{
float  r ;                    //  تعريف متحول حقيقي
float p = 3 , 14 ;            // تعريف متحول حقيقي وإسناد قيمة له
cout << " enter r =" ; cin >> r ;
cout << r * 2=" << r * 2<<"\n";
cout <<"2*p*r = " << 2*p*r<<"\n" ;
cout << "p*r*r =" << p*r*r;
return 0 ;
}
```

(Inactive C:\TCWIN45\BIN\NONAME02.EXE)

enter r = 4.5

r * 2 = 9

2 * p * r = 28.26

p*r*r = 63.585

3- أكتب برنامجاً يقوم بطباعة مستطيل

```
# include < iostream.h>
main ( )
{
  cout << " ***********\n" ;
  cout << " *\t " <<" *\n";
  cout << " *\t " <<" *\n";
  cout << " *\t " <<" *\n";
  cout << "***********\n";
return 0;
}
```

```
  (Inactive C:\TCWIN45\BIN\NONAME04.EXE)
**********
*          *
*          *
*          *
**********
```

statement 2;

مثال 1:

إذا كان علامة الطالب أكبر أو يساوي القيمة 60 درجة فيطبع كلمة "passed" وإلا فهي تطبع الكلمة "failed" عندها فإن تعليمة الـ if / else تكون بالشكل

```
if ( grad > = 60 )
    cout << " passed " ;
else
    cout << "failed" ;
```

مثال 2:

أكتب برنامجاً يقرأ عدداً صحيحاً ثم يحدد و يطبع فيما إذا كان هذا العدد زوجياً أم فردياً .

```
# include < iostream.h>
main ()
{
   int a ;
   cout <<"enter a ="; cin>>a;
   if ( a % 2 = = 0)
      cout << " not odd" ;
   else
      cout << " odd" ;
   return 0 ;
}
```

```
      Inactive C:\TCWIN45\BIN \ NONAME06.EXE)
      enter a = 13
      odd
```

ويمكن استخدام البني if / else المتداخلة من أجل القيام بفحص عدة حالات من خلال وضع البني if / else داخل بعضها البعض . على سبيل المثال إذا كانت علامة الفحص أكبر أو يساوي 90 فيتم طباعة الحرف a وإذا كانت بين 89 و 80 فتطبع الحرف b وإلا فيتم طباعة الحرف c . وبالتالي تكون العملية C++ المكافئة بالشكل:

```
if ( grad > = 90 )
        cout << "a" ;
else if ( grad >= 80)
        cout << "b" ;
else
        cout << "c" ;
```

ملاحظة :

عادة تضع تعليمة واحدة في جسم البنية الاختيارية if ولكن إذا أردنا وضع عدة تعليمات يجب أن نقوم بوضعها داخل قوسين كبيرين ({ }) . نسمى مجموعة التعليمات المحتواه ضمن زوج من الأقواس الكبيرة بالتعليمية المركبة compound statement .

مثال 1:

```
if (grad > = 60 )
        cout << " passed" ;
else
        {
        cout << " failed " ;
              cout << " you must take this course again" ;
        }
```

في هذه الحالة إذا كانت قيمة gradأصغر من 60 عندها يقوم البرنامج بتفيذ التعليمتين الموجودتين في الجزء else ويطبع ما يلي:

```
failed
```

you must take this course again

بعض الأمثلة:

1- أكتب برنامج يأخذ كدخل عددين صحيحين من لوحة المفاتيح ويفحص فيما إذا كان الثاني قاسم للأول.

```
# include<iostream.h>
main ( )
{
        int  a , b ;
        cout<<"enter a=";cin>>a;
        cout<<"enter b=;cin>>b;
        if ( b! = 0 && a % b = = 0 )
                cout << a << ' is divisible by " <<b ;
        else
                cout <<a<<is not divisible by " << b ;
        return 0 ;
}
```

```
        Inactive C\TCWIN45\BIN\NONAME00.EXE)
        enter a = 25
        enter b = 5
        25 is divisible by 5
```

2- أكتب برنامج يأخذ كدخل ثلاث أعداد صحيحة ثم يطبع أصغر هذه الأعداد.

```
# include < iostream.h>
main ()
```

```
{
  int a , b, c ;
  cin >> a >> b >> c ;
  if ( a > b )
        if (a < c ) cout << " min is" << a ;
        else cout << " min is " << c ;
  else
        if ( b < c ) cout << "min is " << b ;
  else
        cout << " min is " << c ;
  return 0;
}
```

```
        Inactive C:\TCWIN45\BIN\NONAME01.EXE)
        enter  a = 10
        enter b = 8
        enter c = 77
        min is 8
```