

مدخل الى الخوارزميات و البرمجة

عنوان المحاضرة: مدخل الى البرمجة بلغة ال C++
المحاضر: د. حيدر خليل
الكلية: الهندسة
رقم المحاضرة 3+4.

أهداف المحاضرة:

- أساسيات لغة ال C++.
- المعارف و الكلمات المحجوزة
- أنواع البيانات
- العمليات الحسابية
- العمليات المنطقية
- الادخال و الإخراج
- أمثلة متنوعة.

1- مقدمة :

تتضمن هذه المحاضرة بعض تعليمات وبنى معطيات لغة ++C المصنفة تحت اللغات البرمجية عالية المستوى خلاف اللغات السابقة لغة الآلة ولغة التجميع.

- لغة الآلة **Machine Language** : وهي تتألف من مجموعة من الأصفار والواحدات التي تعطي الأوامر إلى الحاسب من أجل تنفيذ تعليماته كل تعليمة على حده وبالتالي تصبح صعبة على المبرمجين مما جعلهم يفكرون بتجميع عدة تعليمات بتعليمة واحدة مم أنتج لغة التجميع.

- لغة التجميع **Assembly Language** : حيث تتضمن التعليمة الواحدة أكثر من تعليمة بلغة الآلة ولكي يفهما الحاسب كان لابد من المترجم أو المفسر.

- اللغات عالية المستوى **High-Level Language** : تتضمن التعليمة الواحدة أكثر من تعليمة بلغة الآلة وكذلك تكون اللغات عالية المستوى قابلة للتطبيق على معظم أجهزة الكمبيوتر كما تمتاز هذه اللغات بالبساطة من حيث الفهم من قبل المبرمج وهي تحتاج بالطبع إلى مترجمات أو مفسرات.

يوجد طريقتان للترجمة من لغة التجميع واللغات عالية المستوى إلى لغة الآلة.

- المترجم **Compiler** : يقوم بترجمة البرامج الموجودة بلغة عالية المستوى أو لغة التجميع إلى لغة الآلة وبعدها يمكن تنفيذها، حيث يدعى البرنامج بلغة عالية المستوى ببرنامج المصدر source program والبرنامج بلغة الآلة ببرنامج الهدف object program وتحتاج المترجمات إلى زمن طويل في الترجمة وزمن قصير في التنفيذ.

- المفسر **Interpreter** : يفسر كل سطر بعد الانتقال إلى السطر الذي يليه ويحتاج إلى زمن قصير في التفسير وطويل في التنفيذ لذا تستخدم المفسرات عند تطوير البرامج وبعدها يتم ترجمتها.

تعتبر اللغتان C و ++C لغتان برمجتان للأغراض العامة حيث تتألفان من كلمات رئيسية key words مثل while , do, for, else, if وهي تشبه لغة Pascal أو Fortran....الخ، كما ويمكن أن تستخدم كلغة منخفضة المستوى نظراً لاحتوائها على سمات إضافية تمكن ذلك حيث يمكن أن تستخدم في برامج نظم التشغيل.

يعود تاريخ هذه اللغة إلى عام 1967 من قبل martin Richards لتصبح لغة خاصة بكتابة نظم التشغيل والمترجمات بعد ما طورت عام 1970 والمنتجة كانت شركة bell Telephone laboratories. ونشر أول كتاب 1978 والمؤلف من قبل Kerighan & Ritchie من دار النشر Hall ، طورت لغة C لتصبح ++C اعتباراً من الثمانينات والتي امتلكت القدرة البرمجية غرضه التوجه.

2- المعرفات والكلمات المحجوزة (Identifiers and Key words) :

المعرفات ماهي إلا أسماء عناصر البرنامج المختلفة، مثل المتغيرات والتوابع والمصفوفات. وتشكل المعرفات بواسطة أبجدية اللغة من محارف وأرقام إلا أن ذلك يشترط أن يكون الرمز الأول في المعرف محرف ويجوز أن يكون تسطيرة سفلية، وبالتالي من الخطأ البدء برقم أو أحد الرموز الخاصة كذلك يجب أن لا يحتوي المعرف على فراغ، ويجب التنويه أن لغة C تميز بعدد 8 محارف فقط من الأسماء الطويلة في حين تسمح لغة C++ بطول معرفات حسب الرغبة. مثال عن المعرفات الصحيحة:

`table names _temperature sum_1 y12 X`

مثال عن المعرفات الخاطئة:

4th معرف خاطئ لأنه يبدأ بعدد وليس بحرف.

، "x" معرف خاطئ لأنه يبدأ برمز خاطئ.

on-order معرف خاطئ لأن الإشارة - غير مسموحة.

وسنورد في مايلي بعض الكلمات المحجوزة بلغة C وطبعاً هي الكلمات التي لا يسمح باستخدامها كمعرفات:

| | | | | | |
|-----------------------|---------------------|----------------------|---------------------|---------------------|-----------------------|
| <code>auto</code> | <code>break</code> | <code>case</code> | <code>char</code> | <code>const</code> | <code>continue</code> |
| <code>default</code> | <code>do</code> | <code>doubl</code> | <code>else</code> | <code>enum</code> | <code>long</code> |
| <code>extern</code> | <code>float</code> | <code>for</code> | <code>goto</code> | <code>if</code> | <code>int</code> |
| <code>register</code> | <code>return</code> | <code>short</code> | <code>signed</code> | <code>sizeof</code> | <code>static</code> |
| <code>struct</code> | <code>switch</code> | <code>typedef</code> | <code>union</code> | <code>while</code> | <code>volatile</code> |
| <code>unsigned</code> | <code>voil</code> | | | | |

وفي لغة C++ نجد أيضاً:

| | | | | | |
|------------------------|-----------------------|----------------------|----------------------|-----------------------|---------------------|
| <code>asm</code> | <code>catch</code> | <code>class</code> | <code>delete</code> | <code>friend</code> | <code>inline</code> |
| <code>new</code> | <code>operator</code> | <code>private</code> | <code>public</code> | <code>template</code> | <code>this</code> |
| <code>protected</code> | <code>throw</code> | <code>try</code> | <code>virtual</code> | | |

3- أنواع البيانات Data types

يوجد في لغتي C و C++ أنواع عدة من البيانات وتختلف عن بعضها البعض بعدد البايتات التي تخصص لها في ذاكرة الحاسب وفي النوع الذي تمثله، ويختلف هذا العدد من مترجم لأخر والجدول التالي يورد أنواع البيانات وعدد البايتات ومجال العمل المخصص لكل منها.

| Type | size (byte) | min (valus) | max(valus) |
|--------------------|-------------|-------------------------|-------------------------|
| int | 4 | -2147483648 | 2147483647 |
| short int | 2 | -32768 | 32767 |
| long int | 4 | -2147483648 | 2147483647 |
| unsigned int | 4 | 0 | 4294967295 |
| unsigned short int | 2 | 0 | 65535 |
| unsigned long int | 4 | 0 | 4294967295 |
| char | 1 | -128 | 127 |
| signed char | 1 | -128 | 127 |
| un signed char | 1 | 0 | 255 |
| float | 4 | 1.175494351e-38F | 3.402823466e+38F |
| double | 8 | 2.2250738585072014e-308 | 1.7976931348623158e+308 |

أنواع البيانات بلغتي C و C++ ومجال عمل كل منها

إن نوع المعطيات int ومشتقاته long int و short int يسمح بتخزين معطيات من النوع الصحيح، أما float و double فهي تسمح بتخزين معطيات من النوع الحقيقي، حيث يسمى المتغير من النوع double بالمتغير الحقيقي ذي الدقة المضاعفة أما char فيسمح بتخزين معطيات محرفيه.

4- المتغيرات variables:

المتغير يمثل قيمة معطيات بسيطة ضمن البرنامج، عددية أو حرفية وتملك هذه المتغيرات قيماً وأنواع معطيات مختلفة، ولكن المتغير الواحد يملك نوع واحد من المعطيات وقيم مختلفة منه وتعرف وفق البرنامج كما يلي:

```
int a,b,e ;    char d;
```

وهنا يجب نسب القيم الصحيحة العددية للمتحويلات الثلاث الأولى ونسب قيمة حرفية إلى المتغير d ضمن البرنامج ويمكن تغيير القيم العددية والحرفية ضمن البرنامج مع المحافظة على النوع.

• أمثلة عن حجز المتغيرات في الذاكرة من الأنواع المختلفة من البيانات:

```
int age = 25;           // عدد صحيح عادي
short smallNumber = 100; // عدد صحيح قصير
unsigned int positiveOnly = 42; // عدد موجب فقط
```

```
float pi = 3.14f;      // عدد عشري بدقة منخفضة
double salary = 1500.50; // عدد عشري بدقة عالية
char letter = 'A';     // حرف واحد
```

| أمثلة عن المعرفات الخاطئة | أمثلة عن المعرفات الصحيحة |
|---|---|
| <code>int 2students;</code> // لا يمكن البدء برقم ❌ | <code>int age;</code> // صحيح ✅ |
| <code>float student-name;</code> // لا يمكن استخدام شرطة ❌ | <code>float _price;</code> // صحيح ✅ |
| <code>string student name;</code> // لا يمكن استخدام مسافات ❌ | <code>string student_name;</code> // صحيح ✅ |
| <code>int class;</code> // كلمة محجوزة ❌ | <code>bool isValid;</code> // صحيح ✅ |
| <code>bool value@;</code> // لا يمكن استخدام @ ❌ | <code>int x, y, z;</code> // صحيح ✅ |
| | <code>int __count;</code> // صحيح ✅ |
| | <code>int MAX_VALUE;</code> // صحيح ✅ |

5- العمليات الحسابية في لغة السي بلس:

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

6- المعاملات العلائقية

| 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 |
| ≥ | >= | $x >= y$ | أكبر أو يساوي y |

والمعاملات الستة لها عاملين وتكون نتيجة تطبيقها على عواملها إما true (1) أو false (0).

وقد تكون العوامل جمل أو تعابير مثلاً:

$(i+f) \leq 10 ; c \geq 10*(i+f) ;$

7- المعاملات المنطقية:

المعاملات المنطقية

Logical operators

كما وتملك لغتي C و C++ التابعين المنطقيين `or` (`||`) و `and` (`&&`).
يفرض أن هذين التابعين يطبقان على معاملين `a,b` عندئذ يكون جدول الحقيقة لهما بالشكل:

| a | b | a&&b |
|---|---|------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

الجدول يوضح عمل `and`

| a | b | a b |
|---|---|------|
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

الجدول يوضح عمل `or`

هنا `&&` لها أولوية أكبر من `||` وتقع بعد المساواة وعدم المساواة.
بالإضافة للمعاملين `&&`, `||` هناك معاملي النفي الأحادي `not` (!) وهو يقوم بنفي التعبير المنطقي وله أولوية على المعاملات الأحادية.

الجدول التالي يوضح أولوية المعاملات:

| | | |
|------------------|--|----------------------|
| الأحادية | <code>-, ++, --, !, sizeof (type)</code> | من اليمين إلى اليسار |
| الحسابية | <code>*, /, %</code> | من اليسار إلى اليمين |
| الحسابية | <code>+, -</code> | من اليسار إلى اليمين |
| العلاقية | <code><, <=, >, >=</code> | من اليسار إلى اليمين |
| التسوي | <code>==, !=</code> | من اليسار إلى اليمين |
| And | <code>&&</code> | |
| Or | <code> </code> | |
| الإدخال والإخراج | <code><<, >></code> | |
| النسب | <code>=, +=, -=, *=, /=, %</code> | من اليمين إلى اليسار |

أمثلة:

```
int result = 10 * 5 / 2;           // = (10 * 5) / 2 = 25
int remainder = 10 % 3;           // = 1 (باقي القسمة)
int result = 10 - 5 + 3;         // = (10 - 5) + 3 = 8
```

```
bool b = 5 < 3 <= 10;    // = (5 < 3) <= 10 = false <= 10 = true
bool b1 = 5 == 5;       // true
bool b2 = 5 != 3;       // true
```

8- إدخال و إخراج البيانات بلغة ال C++:

• التابع COUT:

يستخدم التابع cout لإخراج البيانات إلى وحدات الخرج القياسية (شاشة الحاسب).
مثال 1:

```
#include<iostream >
using namespace std;
// this program print hello world.
void main()
{
cout<<"hello world.\n";
}
```

خروج البرنامج:

hello world.

شرح البرنامج السابق:

- السطر الأول في البرنامج هو عبارة عن توجيه يخبر المترجم بضرورة ضم الملف الرأسي iostream إلى ملف البرنامج لإحتوائه علي تعليمات ضرورية لعمليتي الإدخال والإخراج.
- السطر الثاني ماهو إلا تعليق لايراه المترجم (لأنه سبق ب-//) يوضح الغاية من البرنامج.
- إذا امتد التعليق لأكثر من سطر يجب أن يسبق كل سطر بالرمز // لكن هناك طريقة أخرى بأن يوضع التعليق بين الرمزين /* في بداية التعليق و */ في نهاية التعليق .
- السطر الثالث يعلن عن بدء التابع main() يليه قوس البداية الذي يحصر تعليماته مع قوس النهاية.
- السطر الخامس يتضمن التعبير:

```
cout<<"hello world.\n";
```

- هذا التعبير يعني إرسال الرسالة hello world بواسطة cout إلى مجرى خرج قياسي عادة يكون شاشة الحاسب، والرمز \n يخبر المترجم بالانتقال لسطر جديد بعد إخراج الرسالة السابقة، والرمز << هو معامل الإخراج فهو يضع الرسالة المراد إخراجها في مجرى الخرج القياسي.

مثال 2:

```
#include<iostream>
using namespace std;
void main()
{
int n;
n=66;
cout<<n<<endl;}
```

خرج البرنامج:

66

السطر الأول بين قوسين يعلن عن n كمتغير من النوع الصحيح int ، السطر الذي يليه يخصص القيمة 66 للمتغير n. والثابت الرمزي endl يعمل عمل \n فهو يسبب الانتقال لسطر جديد بعد طباعة القيمة 66. الإعلان int n; يخبر المترجم شيئاً : الأول اسم المتغير هو n، الثاني هو النوع الصحيح للمتغير، ومن خلال هذا النوع يتعرف المترجم إلى كيفية الحجز للمتغير وعلى قيم المتغيرات واستخدامها.

مثال 3 :

```
#include<iostream>
using namespace std;
void main()
{
int x,y1;
x=77;
y1=88;
int y2=55;
cout<<x<<","<<y1<<","<<y2<<endl;
}
```

خرج البرنامج:

77 , 88 , 55

في هذا البرنامج تم إخراج قيم المتحولات بالتتالي، ونلاحظ أنه يمكن الإعلان عن المتغير في أي مكان ضمن البرنامج المهم أن يعلن عنه قبل استخدامه، كما نلاحظ أنه بالإمكان تخصيص المتغير عند الإعلان عنه.

• التابع cin:

يستخدم التابع cin لإدخال المعطيات من لوحة المفاتيح لتتم معالجتها بعد تخزينها في ذاكرة الحاسب.
مثال 1:

```
#include<iostream >
using namespace std;
void main()
{
int x,y;
cin>>x>>y;
cout<<x<<'+ '<<y<<'='<< x+y <<endl;
}
```

5 7
5+7=13

في هذا البرنامج تم استخدام التابع cin لإدخال البيانات من لوحة المفاتيح، والمعامل >> يسمى معامل الإدخال حيث يقوم بتلقي البيانات المدخلة من لوحة المفاتيح و إرسالها الى العنوان المناسب في الذاكرة، مثلا يتم اخذ القيم التي تم إدخالها من لوحة المفاتيح و من ثم تحويلها الى امكنتها x و y و هب المتغيرات الموجودة في الذاكرة.

9- مراحل تنفيذ أي برنامج C++ :

يتم التنفيذ خلا ست مراحل هي بالشكل التالي:

- **مرحلة الكتابة ضمن Edit :** وهي كتابة البرنامج في أي محرر نصوص يستخدم لكتابة البرامج بلغة C++ .
- **مرحلة ما قبل الترجمة ؛** 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 ال 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 صور 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 .

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

```
# include <iostream >
using namespace std;
int main ( )
{
cout <<"welcome to c++ \n " ;
cout <<"Mechatronic Engineering " ; return 0;
}
```

الخرج

```
welcome to c++
Mechatronic Engineering
```

11- أمثلة متنوعة :

مثال 1 : أكتب برنامج يأخذ كدخل ثلاثة أعداد صحيحة من لوحة المفاتيح ثم يطبع مجموعها ومتوسطها وناتج جدائها.

```
# include <iostream >
using namespace std;
int main ( )
{
int a , b, c ;
cout << " enter a =" ;
cin >> a ;
cout << " enter b = " ;
cin >> b ;
cout << " enter c = " ;
cin >> c ;
cout << " sunm is " << a+b+c << " \ n" ;
cout << "average is " << ( a+b+c)/3 <<" \n";
cout << "product is " << a * b* c;
return 0 ;
}
```

الخرج في حال تم اسناد القيم ل a,b,c على التوالي 3,5,7.

```
enter a =3
enter b = 5
enter c = 7
sum is 15
average is 5
product is 105
```

مثال 2: أكتب برنامج يقرأ نصف قطر دائرة ثم يطبع قيمة قطر الدائر ، محيطها و مساحتها.

```
# include <iostream >
using namespace std;
int main ( )
{
float r ;
float p = 3.14 ;
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 ;
}
```