

((سبحانك لا علم لنا إلا ما علمتنا إنك أنت العليم الحكيم))

أسهل الطرق للبدء مع C++

إعداد وتأليف : المبرمج العرمانى

بكالوريوس علوم حاسب (Computer Science)

المملكة العربية السعودية – المنطقة الغربية

alarmani_programmer@hotmail.com

تم الإنتهاء من إعداد وتأليف هذا الكتاب في تاريخ :

٢٠٠٧/٦/٤ م – ١٤٢٨/٥/١٨ هـ

ملاحظات حول هذا العمل :

- ١- إنطلاقاً من مبدأ لا خير في كاتم العلم شرعت في تأليف هذا الكتاب حول لغة السي ++ سائلاً الله الأجر والمثوية فهو عمل خالص لوجه الله .
- ٢- طريقة الشرح في هذا الكتاب هي طريقة تطبيقية بحتة بحيث نبدأ بالكود ثم شرح كل كبيرة وصغيرة موجودة فيه وهذه الطريقة وجدتها أسرع في الفهم وأسهل في التعلم وتضيف فهم عميق للغة مقارنة بطريقة الشرح التقليدية .
- ٣- يمكن لكل قارئ لهذا الكتاب أن يقوم بنسخه أو توزيعه سواء على صورته هذه أو غيرها ولكن أتمنى أن يتم ذكر الكاتب أو المؤلف عرفاناً لما قام به من جهد حتى وصوله للجميع وإنطلاقاً من مبدأ نسبة الشيء لصاحبه .
- ٤- قمت بإعداد وتأليف هذا الكتاب اعتماداً على مصادر مطبوعة علمية باللغتين العربية والانجليزية وليس نقلاً ولا نسخاً بحيث أعتدت على طريقة شرحة الخاصة من خلال فهمي للغة إذن فهو مجهود شخصي مني بحت .
- ٥- هذا العمل موجه إلى فئة المبتدئين مع السي ++ .
- ٦- العمل إن شاء الله سوف يكون على الإصدار السادس وهو Microsoft Visual C++ 6.0 .
- ٧- هذا العمل يعتبر أرضية خصبة لإحتراف السي ++ .
- ٨- أخيراً : أتمنى أن لاتنسوني من خالص دعائكم بالصحة والعافية والتوفيق لي ولوالدي ولجميع المسلمين .

١- مقدمة عامة عن لغة السي ++ :

تعتبر لغة السي++ من اللغات عالية المستوى high level languages وهذه اللغات تتصف بالمرونة وقابلية نقلها على أي آلة والسهولة .

والسي ++ هي لغة برمجة عامة الأغراض لذا فهي تستخدم في الكثير من التطبيقات كما أن طريقة الكتابة بها تتسم بالسهولة .

فعند كتابة البرنامج بلغة السي++ نقوم بترجمته عن طريق `compile` وهذا برنامج يقوم بتحويل الكود المصدري إلى كود بلغة الآلة . ومن ثم نقوم بتنفيذه عن طريق `execute` وهو الذي يحوله إلى كود تنفيذي .

٢- بدايتك مع Microsoft Visual C++ 6.0 :

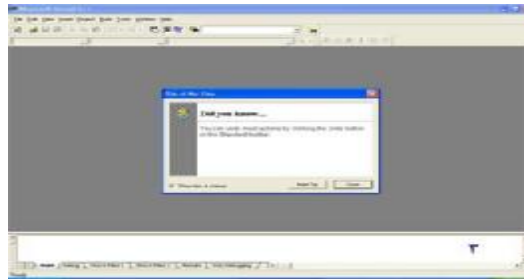
عند تحميل Microsoft visual studio 6.0 يكون من ضمن محتوياتها Microsoft Visual C++ 6.0 و Microsoft Visual basic 6.0 وغيرها . ما يهمنا هنا حقاً هي Microsoft Visual C++ 6.0 فعند تشغيله لأول مرة من خلال رمزه الموضح بالشكل ١ :



عند الضغط عليه سوف تظهر لنا هذه النافذة التي ماتلبث ثواني وإلا وهي قد أختفت كما في الشكل ٢:



بعد ذلك ستظهر لنا نافذة البرنامج الرئيسية وستظهر نافذة فرعية صغيرة نقوم بالضغط على `close` لإغلاقها كما في الشكل ٣ :

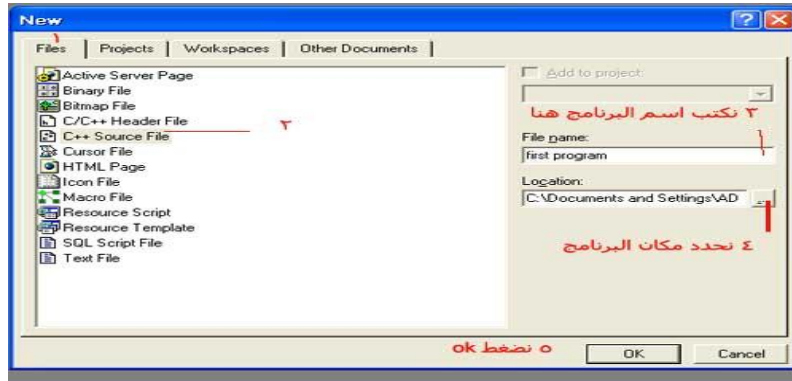


حتى نبدأ طريقنا للدخول لعالم كتابة الكود البرمجي لابد من إتباع الخطوات التالي :

- ١- من قائمة `file` نختار `new`
- ٢- من قائمة `new` التي ستظهر نختار `files` ثم نحدد `c++ source file`
- ٣- نقوم بكتابة أسم الملف ونحدد مكانه ثم نضغط موافق .

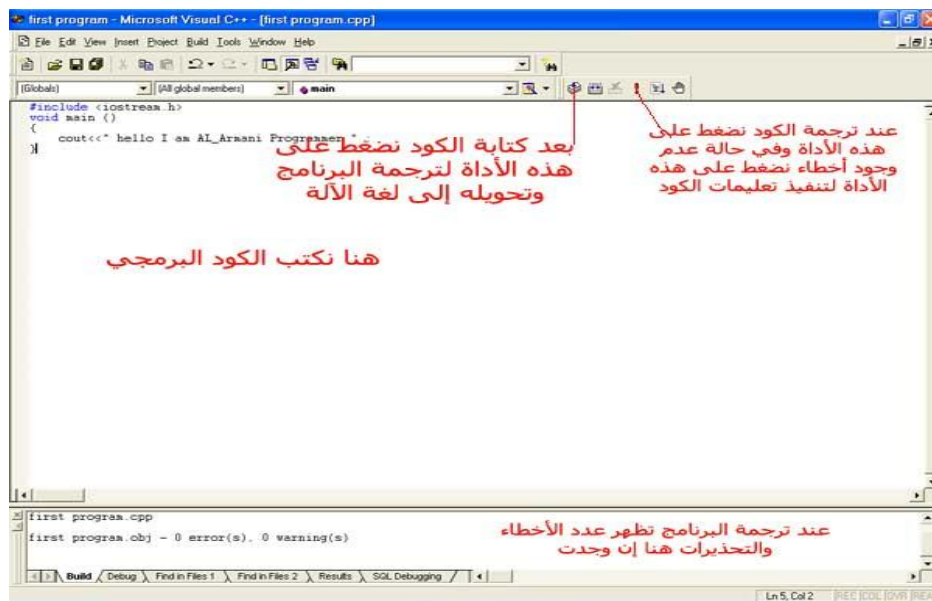
نلاحظ كما في الشكل :

أسهل الطرق للبدء مع C++

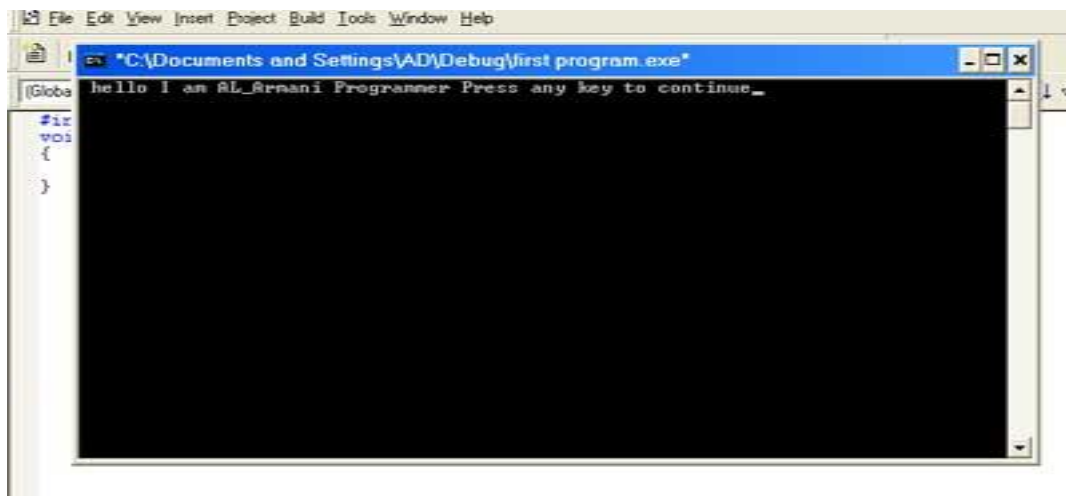


كيفية كتابة الكود :

هنا للتوضيح قمنا بكتابة كود عبارة عن طباعة رسالة ترحيب .. والشكل يوضح كيفية كتابة البرنامج والترجمة والتنفيذ ..



بعد تنفيذ البرنامج سوف تظهر لنا شاشة التنفيذ وهي الشاشة السوداء كما في الشكل التالي :



ملاحظة :

عند الضغط على رمز الترجمة compile ورمز التنفيذ execute وظهرت لنا نوافذ صغيرة بها نعم أم لا نختار نعم .

٣- كتابة البرنامج بلغة السي ++ :

قبل كتابة البرنامج يجب ان ندرك بأن لغة السي++ حساسة جداً لحالة الحروف الصغير والكبيرة ، فكلمة MOH تختلف جدا عن كلمة mOh و MOh . ويجب أن ندرك أيضا بأن كل الكلمات المحجوزة للغة جميع أحرفها صغيرة مثل void ، main ، include وغيرها . وعند ظهور أخطاء نستطيع معرفتها في النافذة الصغيرة السفلية والتي دائما تكون ظاهرة ، والاطء دائما تكون إما نسيان كتابة الفاصلة المنقوطة أو كتابة إحدى الكلمات المحجوزة بحرف كبير .

😊 البرنامج الأول : برنامج بسيط :

```
//Hello in my first program in c++ language
#include<iostream.h>
void main()
{
cout<<" I am AL_Armani Programmer ";
}
```

شرح البرنامج :

```
//Hello in my first program in c++ language
```

أي سطر في لغة السي++ يبدأ ب // فهذا يعني بأنه عبارة عن تعليق أو ملاحظة يستخدم للتوضيح فالسي++ عندما يرى هذه العلامة يعرف بأنها ملاحظة فيقوم بتجاهلها .

أيضا إذا كانت الملاحظة مكونة من عدة سطور فيصعب أن نبدأ كل سطر ب // لذا هذا نوع آخر من الملاحظات يسمى long comment وهي تبدأ ب /* وتنتهي ب */ ، مثلا :

```
/* Here we write every thins
```

```
As
```

```
Comments
```

```
*/
```



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

include هو عبارة عن موجه للمترجم لكي يقوم بإدراج ملف رأسي ضمن الملف الأساسي . ويجب أن يوضع الملف الرأسي بين هاتين العلامتين <>

iostream.h (حرف الـ i يدل على الإدخال input وحرف الـ o يدل على الإخراج output)

أسهل الطرق للبدء مع C++

هو عبارة عن ملف يحتوي على دوال وتصريحات نحتاج لإستخدامها في البرنامج وهذا الملف الرأسي ضروري كتابته إذا أردنا إستخدام `cin` و `cout` وهي أهداف للإخراج والإدخال .

```
void main()
```

هذا السطر هو عبارة عن الدالة الأساسية في أي برنامج مكتوب بلغة السي++ وأول ماينفذ من البرنامج هو هذه الدالة فهي ضرورية وتحتوي على كل شيء من نداءات لدوال أخرى وتصريحات وأهداف وغيره .

وكلمة `void` تعني ان هذه الدالة لاترجع أي قيمة . والأقواس الدائرية مهمة جدا فهي تعني أن هذا السطر هو عبارة عن دالة وليست متغير .

```
{  
}
```

هذا القوسان هما جسم الدالة الرئيسية `body of function` ونكتب بداخله الكود المراد ولايصح كتابة أي كود خارج هذين القوسين وإلا سيظهر خطأ `error` .

المبرمج العرمانى

```
cout<<" I am AL_Armani Programmer ";
```

`cout` هو عبارة عن هدف يقوم بعرض رسالة معينة أو إظهار قيمة متغير معين . أما الإشارة `<<` فهي تقوم بتوجيه ما يكتب على يمينها إلى الهدف الذي على يسارها .

`" I am AL_Armani Programmer "` هذه هي الرسالة المراد ظهورها على شاشة التنفيذ ويجب ان تكتب بين علامتي التنصيص المزدوجة . أيضا أي جملة في لغة السي++ لابد وأن تنتهي بعلامة شبه فاصلة `semicolon`

وهذه الجملة تسمى ثابتا حرفيا .

المبرمج العرمانى

😊 البرنامج الثاني : جمع عددين :

```
/* hello everyone
```

```
This program to add two integer number
```

```
*/
```

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

```
void main ()
```

```
{
```

```
int number1;
```

```
int number2;
```

```
int number3;
```

```
cout<<" please enter your first number \n";
```

المبرمج : العرمانى

أسهل الطرق للبدء مع C++

```
cin>>number1;
cout<<" please enter your second number \n";
cin>>number2;
number3=number1+number2;
cout<<" the sum of number1 and number2 which you entered is : \n" ;
cout<<number3;
}
```

شرح البرنامج :

نلاحظ بأننا أستخدمنا علامة الملاحظة الطويلة لأنها تتكون من عدة سطور .

```
int number1;
```

هذا السطر هو عبارة عن تعريف لمتغير من نوع `int` أي عدد صحيح واسمه `number1` ونلاحظ أيضا بان الجملة تنتهي بشبه فاصلة كما هو المعتاد لنهاية أي جملة في السي ++ . عرفنا هذا المتغير ليمثل الرقم الأول المراد جمعه مع الرقم الثاني .

```
int number2;
```

عرفنا هذا المتغير ليمثل الرقم الثاني المراد جمعه مع الرقم الأول .

```
int number3;
```

عرفنا هذا المتغير ليمثل حاصل جمع الرقم الاول والثاني .

إن الصيغة الرئيسية لتعريف المتغيرات هي :

```
Datatype variablename ;
```



وقفة مع المتغيرات :

المتغير هو جزء مهم جدا في الكود البرمجي وهو عبارة عن اسم يقوم بأخذ قيمة معينة سواءا عددية أو حرفية وهذه المتغيرات تخزن في الذاكرة فعندما نعرف متغير ما فإنه يحجز له مكان في الذاكرة يحمل اسمه .

أنواع المتغيرات type of variables :

- ١- Integer variable
عبارة عن متغيرات عددية صحيحة سالبة وموجبة حجمه في الذاكرة 2byte
- ٢- Long variable
عبارة عن متغيرات عددية صحيحة ولكن حجمها في الذاكرة 4byte
- ٣- Character variable
عبارة عن متغيرات حرفية حجمه في الذاكرة 1byte
- ٤- Float variable

أسهل الطرق للبدء مع ++C

عبارة عن متغيرات حقيقية عشرية ذات الفاصلة العائمة وحجمه في الذاكرة 4byte ، وهناك أيضا أنواع للمتغيرات الحقيقية العشرية والتي هي أكبر حجماً من الـ float وهي double ، longdouble

الشروط التي يجب مراعاتها عند تسمية المتغير :

- 1- يجب أن لا يبدأ أسم المتغير بفراغ أو برقم ويمكن أن يبدأ بشرطة سفلى _ .
- 2- يمكن استخدام الأحرف الكبيرة والصغيرة معاً .
- 3- لا يمكن استخدام الكلمات المحجوزة في لغى السي++ مثل void ، int وغيرها .
- 4- يفضل أن يعبر إسم المتغير عن محتواه .

المبرمج العرمانى

```
cout<<" please enter your first number \n";
```

```
cout<<" please enter your second number \n";
```

هذان السطران هما عبارة عن رسالة للمستخدم ليقوم بإدخال الرقمين الأول والثاني المراد جمعهما .
والعلامة \n تعني النزول لسطر جديد بعد هذه الجملة وهي تضيف تنظيم لشكل البرنامج حت لا يحصل تراحم .

المبرمج العرمانى

```
cin>>number1;
```

```
cin>>number2;
```

هذان السطران هما لعملية إدخال الرقم الأول والثاني وتخزين الرقم الأول في المتغير الذي عرفناه number1 والثاني في number2

المبرمج العرمانى

```
number3=number1+number2;
```

هذا هو عبارة عن متغير اسمه number3 وقيمه عبارة عن تعبير expression حاصل جمع الرقمين .

المبرمج العرمانى

```
cout<<" the sum of number1 and number2 which you entered is : \n" ;
```

هذه عبارة عن رسالة تظهر قبل عرض النتيجة .

المبرمج العرمانى

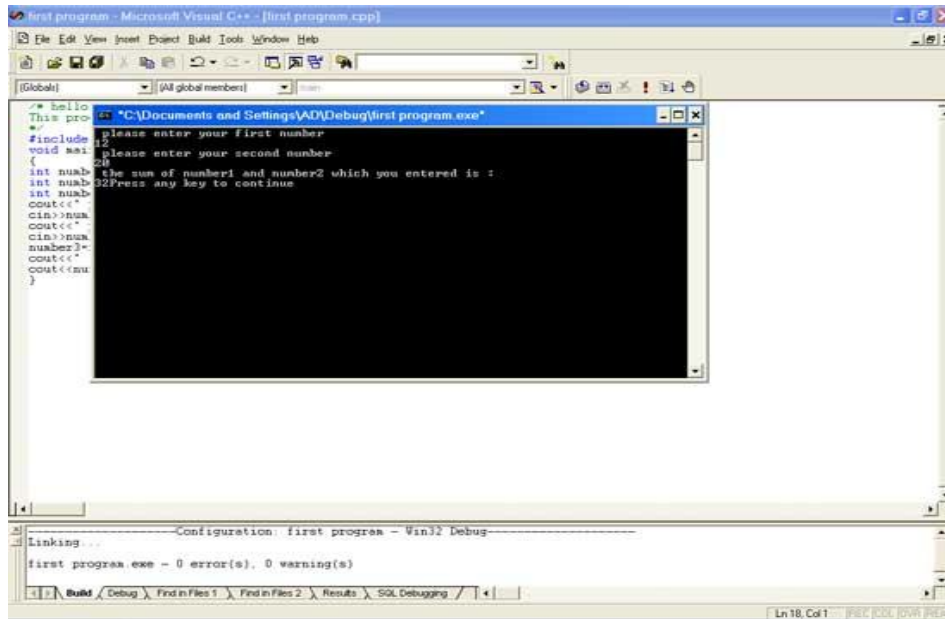
```
cout<<number3;
```

هنا يقوم بعرض قيمة number3 والذي هو حاصل جمع الرقم الأول والثاني .

وطريقة تنفيذ البرنامج أي عند ظهور الشاشة السوداء سوف تظهر رسالة تطلب من المستخدم إدخال الرقم الأول وبعد إدخاله للرقم يضغط على مفتاح enter لوحة المفاتيح ، بعد ذلك ستظهر رسالة تطلب منه إدخال الرقم الثاني وبعد إدخاله يضغط على مفتاح enter لتظهر له نتيجة جمع الرقمين .

أسهل الطرق للبدء مع C++

وهذه هي شاشة التنفيذ :



```
first program - Microsoft Visual C++ - [first program.cpp]
File Edit View Insert Format Build Tools Window Help
[Global] [All global members] main
/* Hello
 * This program
 * #include <iostream>
 * using namespace std;
 * void main()
 * {
 *     int number1;
 *     int number2;
 *     int number3;
 *     cout<<"
 *     please enter your first number
 *     ";
 *     cin>>number1;
 *     cout<<"
 *     please enter your second number
 *     ";
 *     cin>>number2;
 *     number3=number1+number2;
 *     cout<<"
 *     the sum of number1 and number2 which you entered is :
 *     ";
 *     cout<<number3;
 *     cout<<"
 *     Press any key to continue
 *     ";
 *     getch();
 * }
 */
Configuration: first program - Win32 Debug
Linking...
first program.exe - 0 error(s), 0 warning(s)
Ln 18, Col 1: [PREV] [GO] [NEXT] [F1] [F2] [F3] [F4] [F5] [F6] [F7] [F8] [F9] [F10] [F11] [F12] [F13] [F14] [F15] [F16] [F17] [F18] [F19] [F20] [F21] [F22] [F23] [F24] [F25] [F26] [F27] [F28] [F29] [F30] [F31] [F32] [F33] [F34] [F35] [F36] [F37] [F38] [F39] [F40] [F41] [F42] [F43] [F44] [F45] [F46] [F47] [F48] [F49] [F50] [F51] [F52] [F53] [F54] [F55] [F56] [F57] [F58] [F59] [F60] [F61] [F62] [F63] [F64] [F65] [F66] [F67] [F68] [F69] [F70] [F71] [F72] [F73] [F74] [F75] [F76] [F77] [F78] [F79] [F80] [F81] [F82] [F83] [F84] [F85] [F86] [F87] [F88] [F89] [F90] [F91] [F92] [F93] [F94] [F95] [F96] [F97] [F98] [F99] [F100] [F101] [F102] [F103] [F104] [F105] [F106] [F107] [F108] [F109] [F110] [F111] [F112] [F113] [F114] [F115] [F116] [F117] [F118] [F119] [F120] [F121] [F122] [F123] [F124] [F125] [F126] [F127] [F128] [F129] [F130] [F131] [F132] [F133] [F134] [F135] [F136] [F137] [F138] [F139] [F140] [F141] [F142] [F143] [F144] [F145] [F146] [F147] [F148] [F149] [F150] [F151] [F152] [F153] [F154] [F155] [F156] [F157] [F158] [F159] [F160] [F161] [F162] [F163] [F164] [F165] [F166] [F167] [F168] [F169] [F170] [F171] [F172] [F173] [F174] [F175] [F176] [F177] [F178] [F179] [F180] [F181] [F182] [F183] [F184] [F185] [F186] [F187] [F188] [F189] [F190] [F191] [F192] [F193] [F194] [F195] [F196] [F197] [F198] [F199] [F200] [F201] [F202] [F203] [F204] [F205] [F206] [F207] [F208] [F209] [F210] [F211] [F212] [F213] [F214] [F215] [F216] [F217] [F218] [F219] [F220] [F221] [F222] [F223] [F224] [F225] [F226] [F227] [F228] [F229] [F230] [F231] [F232] [F233] [F234] [F235] [F236] [F237] [F238] [F239] [F240] [F241] [F242] [F243] [F244] [F245] [F246] [F247] [F248] [F249] [F250] [F251] [F252] [F253] [F254] [F255] [F256] [F257] [F258] [F259] [F260] [F261] [F262] [F263] [F264] [F265] [F266] [F267] [F268] [F269] [F270] [F271] [F272] [F273] [F274] [F275] [F276] [F277] [F278] [F279] [F280] [F281] [F282] [F283] [F284] [F285] [F286] [F287] [F288] [F289] [F290] [F291] [F292] [F293] [F294] [F295] [F296] [F297] [F298] [F299] [F300] [F301] [F302] [F303] [F304] [F305] [F306] [F307] [F308] [F309] [F310] [F311] [F312] [F313] [F314] [F315] [F316] [F317] [F318] [F319] [F320] [F321] [F322] [F323] [F324] [F325] [F326] [F327] [F328] [F329] [F330] [F331] [F332] [F333] [F334] [F335] [F336] [F337] [F338] [F339] [F340] [F341] [F342] [F343] [F344] [F345] [F346] [F347] [F348] [F349] [F350] [F351] [F352] [F353] [F354] [F355] [F356] [F357] [F358] [F359] [F360] [F361] [F362] [F363] [F364] [F365] [F366] [F367] [F368] [F369] [F370] [F371] [F372] [F373] [F374] [F375] [F376] [F377] [F378] [F379] [F380] [F381] [F382] [F383] [F384] [F385] [F386] [F387] [F388] [F389] [F390] [F391] [F392] [F393] [F394] [F395] [F396] [F397] [F398] [F399] [F400] [F401] [F402] [F403] [F404] [F405] [F406] [F407] [F408] [F409] [F410] [F411] [F412] [F413] [F414] [F415] [F416] [F417] [F418] [F419] [F420] [F421] [F422] [F423] [F424] [F425] [F426] [F427] [F428] [F429] [F430] [F431] [F432] [F433] [F434] [F435] [F436] [F437] [F438] [F439] [F440] [F441] [F442] [F443] [F444] [F445] [F446] [F447] [F448] [F449] [F450] [F451] [F452] [F453] [F454] [F455] [F456] [F457] [F458] [F459] [F460] [F461] [F462] [F463] [F464] [F465] [F466] [F467] [F468] [F469] [F470] [F471] [F472] [F473] [F474] [F475] [F476] [F477] [F478] [F479] [F480] [F481] [F482] [F483] [F484] [F485] [F486] [F487] [F488] [F489] [F490] [F491] [F492] [F493] [F494] [F495] [F496] [F497] [F498] [F499] [F500] [F501] [F502] [F503] [F504] [F505] [F506] [F507] [F508] [F509] [F510] [F511] [F512] [F513] [F514] [F515] [F516] [F517] [F518] [F519] [F520] [F521] [F522] [F523] [F524] [F525] [F526] [F527] [F528] [F529] [F530] [F531] [F532] [F533] [F534] [F535] [F536] [F537] [F538] [F539] [F540] [F541] [F542] [F543] [F544] [F545] [F546] [F547] [F548] [F549] [F550] [F551] [F552] [F553] [F554] [F555] [F556] [F557] [F558] [F559] [F560] [F561] [F562] [F563] [F564] [F565] [F566] [F567] [F568] [F569] [F570] [F571] [F572] [F573] [F574] [F575] [F576] [F577] [F578] [F579] [F580] [F581] [F582] [F583] [F584] [F585] [F586] [F587] [F588] [F589] [F590] [F591] [F592] [F593] [F594] [F595] [F596] [F597] [F598] [F599] [F600] [F601] [F602] [F603] [F604] [F605] [F606] [F607] [F608] [F609] [F610] [F611] [F612] [F613] [F614] [F615] [F616] [F617] [F618] [F619] [F620] [F621] [F622] [F623] [F624] [F625] [F626] [F627] [F628] [F629] [F630] [F631] [F632] [F633] [F634] [F635] [F636] [F637] [F638] [F639] [F640] [F641] [F642] [F643] [F644] [F645] [F646] [F647] [F648] [F649] [F650] [F651] [F652] [F653] [F654] [F655] [F656] [F657] [F658] [F659] [F660] [F661] [F662] [F663] [F664] [F665] [F666] [F667] [F668] [F669] [F670] [F671] [F672] [F673] [F674] [F675] [F676] [F677] [F678] [F679] [F680] [F681] [F682] [F683] [F684] [F685] [F686] [F687] [F688] [F689] [F690] [F691] [F692] [F693] [F694] [F695] [F696] [F697] [F698] [F699] [F700] [F701] [F702] [F703] [F704] [F705] [F706] [F707] [F708] [F709] [F710] [F711] [F712] [F713] [F714] [F715] [F716] [F717] [F718] [F719] [F720] [F721] [F722] [F723] [F724] [F725] [F726] [F727] [F728] [F729] [F730] [F731] [F732] [F733] [F734] [F735] [F736] [F737] [F738] [F739] [F740] [F741] [F742] [F743] [F744] [F745] [F746] [F747] [F748] [F749] [F750] [F751] [F752] [F753] [F754] [F755] [F756] [F757] [F758] [F759] [F760] [F761] [F762] [F763] [F764] [F765] [F766] [F767] [F768] [F769] [F770] [F771] [F772] [F773] [F774] [F775] [F776] [F777] [F778] [F779] [F780] [F781] [F782] [F783] [F784] [F785] [F786] [F787] [F788] [F789] [F790] [F791] [F792] [F793] [F794] [F795] [F796] [F797] [F798] [F799] [F800] [F801] [F802] [F803] [F804] [F805] [F806] [F807] [F808] [F809] [F810] [F811] [F812] [F813] [F814] [F815] [F816] [F817] [F818] [F819] [F820] [F821] [F822] [F823] [F824] [F825] [F826] [F827] [F828] [F829] [F830] [F831] [F832] [F833] [F834] [F835] [F836] [F837] [F838] [F839] [F840] [F841] [F842] [F843] [F844] [F845] [F846] [F847] [F848] [F849] [F850] [F851] [F852] [F853] [F854] [F855] [F856] [F857] [F858] [F859] [F860] [F861] [F862] [F863] [F864] [F865] [F866] [F867] [F868] [F869] [F870] [F871] [F872] [F873] [F874] [F875] [F876] [F877] [F878] [F879] [F880] [F881] [F882] [F883] [F884] [F885] [F886] [F887] [F888] [F889] [F890] [F891] [F892] [F893] [F894] [F895] [F896] [F897] [F898] [F899] [F900] [F901] [F902] [F903] [F904] [F905] [F906] [F907] [F908] [F909] [F910] [F911] [F912] [F913] [F914] [F915] [F916] [F917] [F918] [F919] [F920] [F921] [F922] [F923] [F924] [F925] [F926] [F927] [F928] [F929] [F930] [F931] [F932] [F933] [F934] [F935] [F936] [F937] [F938] [F939] [F940] [F941] [F942] [F943] [F944] [F945] [F946] [F947] [F948] [F949] [F950] [F951] [F952] [F953] [F954] [F955] [F956] [F957] [F958] [F959] [F960] [F961] [F962] [F963] [F964] [F965] [F966] [F967] [F968] [F969] [F970] [F971] [F972] [F973] [F974] [F975] [F976] [F977] [F978] [F979] [F980] [F981] [F982] [F983] [F984] [F985] [F986] [F987] [F988] [F989] [F990] [F991] [F992] [F993] [F994] [F995] [F996] [F997] [F998] [F999] [F1000]
```

المبرمج العرمانى

أخيراً يمكننا أن نضع قيم ثابتة للرقمين الأول والثاني بدون إدخال من المستخدم عن طريق كتابة :

```
int number1=23;
```

```
int number2=10;
```

المبرمج العرمانى

البرنامج الثالث : حساب مساحة الدائرة : 😊

```
//program which calculate the area of circuit
```

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

```
Void main()
```

```
{
```

```
Float rad;
```

```
Const float PI=3.14;
```

```
Cout<<"enter radius of circuit :";
```

```
Cin>>rad;
```

```
Float area =PI*rad*rad;
```

```
Cout<<" The area of circuit is :"<<area;
```

المبرمج : العرمانى

}

شرح البرنامج :

Float rad;

هنا قمنا بتعريف متغير من نوع float وقمنا بتسميته rad ليمثل نصف القطر .

المبرمج العرمانى

Const float PI=3.14;

كلمة const تعني ثابت فهنا قمنا بتعريف متغير ثابت من النوع float وإعطائه قيمة ثابتة وهي قيمة ثابتة رياضياً .

المبرمج العرمانى

cout<<"enter radius of circuit :";

هنا لعرض رسالة توضيحية للمستخدم ليقوم بإدخال نصف القطر .

المبرمج العرمانى

Cin>>rad;

هنا يقوم بإدخال نصف القطر .

المبرمج العرمانى

Float area =PI*rad*rad;

هنا قمنا بتعريف متغير من النوع float واسمه area وهو يمثل المساحة وقيمة هذا المتغير هو عبارة عن ناتج المعادلة

PI*rad*rad

المبرمج العرمانى

Cout<<" The area of circuit is :"<<area;

هنا يعرض رسالة توضيحية قبل إظهار المساحة بعد حسابها .

وهذه هي شاشة التنفيذ :

```
first program - Microsoft Visual C++ - first program.cpp
File Edit View Project Window Debug Tools Help
Debug
--(All global members)
--
--program which calculate the area of circuit
#include<iostream.h>
using namespace std;
float rad;
const float PI=3.14;
cout<<"enter radius of circuit :";
cin>>rad;
float area =PI*rad*rad;
cout<<" The area of circuit is :"<<area;
}

"C:\Documents and Settings\AD\Debug\first program.exe"
enter radius of circuit : 10
The area of circuit is : 314.159
Press any key to continue

Configuration: first program - Win32 Debug
Linking...
first program.exe - 0 error(s), 0 warning(s)
Build | Debug | FirstFile1 | FirstFile2 | Results | SQL Debugger | [x] | Ln:6, Col:2 | Page 1008 | 10:05
```

😊 البرنامج الرابع : استخدام الحلقة لحساب حاصل جمع الأعداد من ٠ إلى ٤ :

الهدف من استخدام الحلقة loop هي تكرار تنفيذ جزء من البرنامج عدد من المرات فهنا لا بد أن يكون لدينا شرط معين إذا تحقق يتم تنفيذ التكرار وإلا سيتوقف .

```
#include<iostream.h>

void main ()
{
int sum=0;
for(int i=0; i<4; i++)
{
sum+=i;
}
cout<<" the sum of number from 0 to 4 is : \n " << sum ;
}
```

شرح البرنامج :

```
int sum=0;
```

قمنا بتعريف متغير من النوع int وأعطيناها القيمة صفر لأن متغير لحفظ مجموع الأعداد من ٠ إلى ٤ ويجب علينا تصفيره في البداية .

المبرمج العرمانى

```
for(int i=1; i<=4; i++)
```

هذه هي حلقة التكرار وبعد الأمر for من أوامر الحلقات والأكثر استخداماً ونستخدمها عندما نعرف عدد تكرار تنفيذ الجزء ، والضيعة العامة لها كما يلي :

(زيادة العداد ; الشرط ; القيمة الابتدائية للعداد)

نستطيع تعريف المتغير العداد وهو i داخل الأقواس الدائرية للحلقة أو ماقبلها أي بجانب تعريف الـ sum

هنا قمنا بتعريف i كعداد وأعطيناها قيمة ابتدائية واحد وبعد ذلك وضعنا شبه الفاصلة ثم الشرط للتكرار وهنا نريد أن يكرر أربعة أعداد من الواحد إلى الأربعة وبعد ذلك يقوم بالزيادة .

المبرمج العرمانى

```
{
```

المبرمج : العرمانى

أسهل الطرق للبدء مع C++

```
sum+=i;
}
```

هنا الأقواس تمثل جسم الحلقة أي كلما كان الشرط صحيح نفذ ما بداخل هذا الجسم .

فهنا كلما كان الشرط صحيح كلما جمع قيمة العداد مع قيمة sum .

المبرمج العرمانى

```
cout<<" the sum of number from 0 to 4 is : \n " << sum ;
```

هنا قمنا بإظهار النتيجة النهائية لحاصل جمع الأرقام ، وعمدنا إلى كتابة هذا السطر خارج جسم الحلقة لكي يقوم بعرض النتيجة النهائية بشكل صحيح أي بعد إنتهاء الحلقة .

المبرمج العرمانى

طريقة سير الحلقة :

عند الدخول إلى الحلقة للمرة الأولى تكون قيمة العداد ١ ويرى هل ١ أقل من أو يساوي ٤ ، نعم يقوم بالدخول إلى جسم الحلقة ويقوم بجمع ١ وهي قيمة العداد على الصفر وهي قيمة الـ sum ويصبح الـ sum بواحد ، بعد ذلك يرجع إلى جملة الـ for ويقوم بزيادة العداد بواحد ليصبح قيمته ٢ ثم يرى هل الشرط تحقق أم لا ، نعم يدخل إلى جسم الحلقة ويقوم بجمع ٢ وهي قيمة العداد على ١ وهي قيمة الـ sum ليصبح الـ sum قيمته ٣

وهكذا إلى أن تصبح قيمة العداد أكبر من ٤ أي ٥ فيخرج من جسم الحلقة ليجد جملة طباعة الـ sum .

المبرمج العرمانى

البرنامج الخامس : استخدام حلقة التكرار while :

يستخدم الأمر while إذا كنا لا نعلم عدد مرات التكرار .

هذا البرنامج يقوم بإدخال أي رقم ماعدا الصفر بحيث إذا قام المستخدم بإدخال الصفر يتم الخروج من البرنامج .

```
#include<iostream.h>
void main ()
{
int r ;
Cout<<" you can enter any value to exit enter 0 ";
while ( r!=0)
{
cout<<"\n";
Cin>>r;
```

المبرمج : العرمانى

أسهل الطرق للبدء مع C++

```
}  
}
```

شرح البرنامج :

```
int r ;
```

قمنا بتعريف متغير اسمه r من النوع الصحيح وقيمة هذا المتغير يقوم بإدخالها المستخدم .

المبرمج العرمانى

```
Cout<<" you can enter any value to exit enter 0 ";
```

هذه الرسالة توضيحية للمستخدم ليقوم بإدخال أي عدد ماعد الصفر.

المبرمج العرمانى

```
while ( r!=0)
```

هنا طريقة كتابة أمر التكرار while وبداخل الأقواس الدائرية كتبنا الشرط بأن القيمة المدخلة لاتكون صفراً .

المبرمج العرمانى

```
{
```

```
cout<<"\n";
```

```
Cin>>r;
```

```
}
```

هذا يعتبر جسم الحلقة وبداخلها جملة ;cout <<"\n" وهي للنزول إلى سطر جديد بين الأرقام المدخلة .

المبرمج العرمانى

```
Cin>>r;
```

هذه الجملة لإدخال قيمة المتغير وهي أي عدد ماعد الصفر حسب شرط الحلقة لكي تقوم بالتكرار .

المبرمج العرمانى

😊 البرنامج السادس : استخدام حلقة التكرار do-while :

هذه الحلقة تختلف عن حلقة التكرار while في أنها تنفذ الجملة ثم تنظر إلى الشرط .

البرنامج هذا يقوم بضرب عددين وشرط التوقف هو إدخال حرف a .

```
#include<iostream.h>

void main()

{

long d1 , d2;

char let;

do{

cout<<"enter the first number\n";

cin>>d1;

cout<<"enter the second number\n";

cin>>d2;

cout<<" the result of multiple two numbers is :\n";

cout<<d1*d2;

cout<<"\n if you want to do again press any letter or number unless a \n";

cin>>let;

cout<<"\n";

}while(let!='a');

}
```

شرح البرنامج :

long d1 , d2;

هنا قمنا بتعريف متغيرين ليمثلا الرقمين المراد ضربيهما وهما من النوع long لأن دائماً حاصل ضرب الأرقام يكون عدد كبير وبإمكانك تعريفهم على انهم int ولكن الأفضل في الضرب هو long ، أيضاً قمنا بتعريف المتغيرين للإختصار بسطر واحد عن طريق الفاصلة بدلاً من كتابة سطرين كالتالي :

Long d1;

Long d2;



```
char let;
```

هنا عرفنا متغير من النوع الحرفي او الرمزي ليمثل قيمة الشرط المدخلة وكما نعلم بأن النوع char يقبل الأرقام والحروف ولكن لانستخدمه للأرقام عندما نريد إجراء عملية حسابية عليهم .



```
do{
```

من هنا تبدأ حلقة التكرار do-while وصيغتها كالتالي :

```
Do{
```

هنا نكتب الجمل المراد تنفيذها أو الأوامر ..

```
while(condition);
```

هنا بعد قفل قوس الحلقة نكتب الـ while ثم قوسين دائريان وبداخلهما الشرط ولاننسى بالنهاية ان نضع فاصلة منقوطة .



```
cout<<"enter the first number\n";
```

```
cin>>d1;
```

```
cout<<"enter the second number\n";
```

```
cin>>d2;
```

```
cout<<" the result of multiple two numbers is :\n";
```

```
cout<<d1*d2;
```

الجمل الستة العليا هي عبارة عن :

- 1- رسالة توضيحية للمستخدم ليقوم بإدخال الرقم الأول
- 2- إدخال الرقم الأول
- 3- رسالة توضيحية للمستخدم ليقوم بإدخال الرقم الثاني
- 4- إدخال الرقم الثاني
- 5- رسالة توضيحية للمستخدم قبل عرض نتيجة حاصل الضرب
- 6- إظهار أو عرض حاصل ضرب الرقم الأول في الرقم الثاني



أسهل الطرق للبدء مع C++

```
cout<<"\n if you want to do again press any letter or number unless a \n";
```

```
cin>>let;
```

```
cout<<"\n";
```

هذه الجملة عبارة عن :

- 1- رسالة توضيحية للمستخدم ليقوم بإدخال أي رقم أو أي حرف لكي يعيد تنفيذ الأوامر من جديد أي إعادة عملية إدخال الرقمين وضربهما بإستثناء حرف الـ a فإذا قام بإدخاله فإنه يخرج من الحلقة .
- 2- إدخال الحرف أو الرقم وتخزينه بالمتغير الذي عرفناه مسبقاً وأعطيناه الاسم let
- 3- الجملة الأخيرة هي لتنظيم البرنامج للنزول لسطر جديد .



```
}while(let!='a');
```

هنا نلاحظ قفل قوس جسم الحلقة ومن ثم الشرط وهو أن الحرف الذي تم إدخاله لتكرار الحلقة لا يكون حرف a ونلاحظ بأن حرف تم وضعه بين علامة تنصيص مفردة ، وفي نهاية الشرط تم وضع فاصلة منقوطة .



😊 البرنامج السابع : استخدام أمر الشرط if :

في بعض البرامج قد نحتاج إلى إتخاذ قرارات معينة والتي من خلالها تحدد سير البرامج وهذا يتم من خلا أمر الشرط والذي يكون محقق أو غير محقق لإكمال مسار جزء معين من البرنامج أو البرنامج ككل .
هذا البرنامج يقوم بإظهار جميع الأرقام المدخلة من قبل المستخدم والتي لا بد أن تكون أصغر من ٥٠ .

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

```
Void main ()
```

```
{
```

```
Int k;
```

```
Cout <<" enter your number : \n ";
```

```
Cin>> k;
```

```
If ( k<50)
```

```
Cout<<" the number is less than 50 ";
```

```
}
```

شرح البرنامج :

كل الجملة قد مرت علينا أثناء شرح البرامج السابقة ماعدا :

المبرمج : العرمانى

if (k<50)

وهذه الجملة هي شرط أن تكون الأرقام أقل من ٥٠ لكي يكمل البرنامج .



😊 البرنامج الثامن : استخدام عبارة switch :

نستخدم هذه العبارة عندما تكون لدينا خيارات متعددة .

هذا البرنامج عبارة عن آلة حاسبة للجمع والضرب والطرح والقسمة .

```
#include<iostream.h>

Void main ()
{
Float num1,num2;
Char symbol;
Float result;
Cout<<"please enter your two numbers :\n";
Cin>>num1;
Cout<<"\n";
Cin>>num2;
Cout<<"please enter the symbol of operator which you want to execute:\n":
Cin>>symbol;
Switch(symbol)
{
Case '+':
Result=num1+num2;
Break;
Case '-':
Result=num1-num2;
Break;
```


أسهل الطرق للبدء مع C++

```
Case '*':  
Result=num1*num2;  
Break;  
Case '/':  
If(num2!=0.0)  
Result=num1/num2;  
Else  
{  
Cout<<"error happned , you devided by zero ";  
Result=0.0;  
}  
Break;  
Default:  
{  
Cout<<"you entered unknown symbol ";  
Result=0.0;  
}  
Cout<<"the result is : \n" <<result;  
}  
}
```

شرح البرنامج :

Float num1,num2;

Char symbol;

Float result;

١- قمنا بتعريف متغيرين من النوع float لأن عملية القسمة ربما تكون بباقي ويكون الناتج أعداد عشرية .

٢- قمنا بتعريف متغير ليمثل الإشارة المدخلة سواءا ضرب طرح قسمة جمع .

المبرمج : العرماني

أسهل الطرق للبدء مع C++

٣- أخيراً عرفنا متغير ليقوم لتخزين نتيجة العملية الحسابية .

المبرمج العرمانى

```
Cout<<"please enter your two numbers :\n";
```

```
Cin>>num1;
```

```
Cout<<"\n";
```

```
Cin>>num2;
```

```
Cout<<"please enter the symbol of operator which you want to execute:\n";
```

```
Cin>>symbol;
```

أعتقد أخي القارئ بأن هذه الجمل أصبحت معروفة لديك ، فهذه مجرد رسائل توضيحية للمستخدم ليقوم بالإدخال وأيضا إدخال الرقمين والإشارة الحسابية .

المبرمج العرمانى

```
Switch(symbol)
```

```
{
```

من هنا يبدأ إستخدامنا لعبارة switch بحيث نعطيها الإشارة المدخلة وهي تقوم بتوجيهها إلى حالتها .

المبرمج العرمانى

```
Case '+':
```

```
Result=num1+num2;
```

```
Break;
```

هذه تمثل الحالة الأولى وهي إذا كانت الإشارة المدخلة جمع فإنه يتم الدخول لهذه الحالة نلاحظ أولاً كتبنا كلمة case وهي تعني حالة ثم عنوان الحالة بعدها وهنا نقصد علامة الجمع وقمنا بوضعها بين علامة تنصيص مفردة ثم نقطتين عموديتين .

وقمنا بكتابة معادلة الجمع وتخزين النتيجة في المتغير result .

أخيراً كتبنا كلمة break وهي تعني الخروج من الـ switch نهائياً .

المبرمج العرمانى

Case '-':

```
Result=num1-num2;
```

```
Break;
```

Case '*':

```
Result=num1*num2;
```

```
Break;
```

هذه هي حالتي الطرح والضرب وهي بنفس طريقة حالة الجمع .



Case '/':

```
If(num2!=0.0)
```

```
Result=num1/num2;
```

```
Else
```

```
{
```

```
Cout<<"error happned , you divided by zero ";
```

```
Result=0.0;
```

```
}
```

```
Break;
```

هذه هي حالة القسمة وهي نفس الحالات السابقة ولكن قمنا بإضافة شرط لها وهو ان المقسوم عليه يجب ان لا يكون صفرا `If(num2!=0.0)`

وقمنا بكتابة الصفر عشريا لاننا عرفنا الارقا في الاعلى على انها `float` .

فإذا تحقق الشرط وكان المقسوم عليه ليس صفرا فإن عملية القسمة سوف تتم .

```
Result=num1/num2;
```

وإذا لم يتحقق الشرط فإنه يقوم بالدخول إلى `else`

وتظهر رسالة بأنه تم القسمة على صفر .

ويوضع ال `result` بصفر .

ويتم الخروج من `switch` .



```
Cout<<"the result is : \n" <<result;  
}
```

هنا يتم طباعة النتيجة .

أخيراً : يمكننا أن نكرر العمليات الحسابية بإضافة حلقة تكرارية والأفضل أن نستخدم `do-while` قبل عملية إدخال الرقمين والإشارة ، كما في البرنامج الذي تم شرحه مسبقاً بأنه لو تم إدخال حرف معين فإنه يتم الخروج من الحلقة .

٤- مقدمة عامة عن البرمجة موجهة الهدف (Introduction to Object Oriented Programming)

تعتبر كتابة البرامج الطويلة والضخمة أمر معقد ، ولكن نستطيع حل هذا التعقيد بواسطة تقسيم هذا البرنامج إلى وحدات أو أجزاء **functions** وكل جزء يقوم بعملية معينة ، ولربما ضربنا مثلاً في حياتنا : وهو أننا لا نستطيع أكل بطيخة دفعة واحدة وإلا سيكون الأمر متعب للغاية للأسنان والفم والمعدة بالطبع ولكن يمكننا تقطيعها إلى قطع صغيرة مما يسهل علينا عملية الأكل (بالعافية على الجميع الظاهر فتحت شهيتكم) ، وهذا ينطبق على كتابة البرامج الكبيرة فكلما قمنا بتجزئته إلى وحدات كلما كان أصبحت عملية الكتابة سهلة جداً وأيضاً حتى عندما نواجه مشكلة تساعد عملية التجزئة إلى حل المشكلة من خلال تحديد مكانها بأي جزء هي .

عند التحدث عن لغة C++ لابد لنا جبراً أن نتحدث ولو باختصار عن مفهوم البرمجة موجهة الهدف .

ماهي البرمجة موجهة الهدف (Object Oriented Programming) :

هي أسلوب برمجي يعتمد على توحيد أو جمع البيانات **data** والدوال **functions** في وحدة برمجية تسمى الهدف **object** ، فلكي نستطيع الوصول إلى البيانات لابد لنا من استخدام الدوال والتي تسمى دوال العضوية **member functions** . والبرنامج المكتوب بلغة السي بلس بلس يمكن أن يحتوي على العديد من الأهداف ، والأهداف هي عبارة بالتالي عن أعضاء من الصنف **class** ، إذن نستطيع القول بأن الصنف هو عبارة عن مجموعة من الأهداف ، فمثلاً : الحشرات ، الزواحف ، الثدييات هم أعضاء في صنف الحيوانات .

أخيراً إن أسلوب البرمجة موجهة الهدف يسمح لنا بتنظيم البرنامج لكي يسهل من فهمه ، أيضاً يسمح لنا بتعريف أكثر من نوع للبيانات .

خصائص اللغات التي تتميز بالبرمجة موجهة الهدف :

١- الوراثة Inheritance

سوف نقوم بشرح هذه الخاصية من خلال ضربنا لهذا المثل : لدينا صنف اسمه الحيوانات وهو يمثل الخصائص العامة لجميع الحيوانات نستطيع من خلاله أن نشق صنف جزئي يرث منه صفات عامة يسمى الطيور والتي لها خصائص خاصة بها من الجناحان ، المنقار والتي لا توجد في الزواحف والثدييات . إذن فالصنف المسمى بالحيوانات يطلق عليه الصنف الرئيسي **base class** والصنف الجزئي المسمى بالطيور يطلق عليه الصنف المشتق **derived class** .

٢- إعادة الاستخدام Reusability

بعد كتابة البرنامج يستطيع المبرمجون إضافة مميزات له وإمكانيات بواسطة الإشتقاق لصنف جدي وما إلى ذلك وهذا ما يسمى بإعادة الاستخدام .

٣- تغليف البيانات Encapsulation

وهي عملية إخفاء البيانات لمنع الوصول إليها من دوال غير مسموح لها .

بقي أن نقول فيما يخص البرمجة موجهة الهدف بأن عندما نواجه مشكلة ما فإننا نسأل كيف سنقسمها إلى أهداف وليس إلى توابع .

الفرق بين لغة C و C++ :

تعتبر لغة السي لغة إجرائية أي أنها عبارة عن سلسلة من الأوامر التي ينفذها الكمبيوتر فهي ليست من اللغات التي تعتمد على البرمجة موجهة الهدف .

(السي ، باسكال ، فورتران ، بيسك هي لغات لاتعتمد على البرمجة موجهة الهدف) .

خاتمة

أخيراً أسأل الله العلي القدير أن يجعل هذا الكتاب لبنة من لبنات المكتبة العربية العلمية وأن ينفع به العرب والمسلمين ولو بشكل بسيط ... وأتمنى من الله أن يمدني بالصحة والعافية لإكمال مسيرة التأليف للجزء الثاني من هذا الكتاب وعند وجود أي ملاحظات حول الكتاب يمكنكم مراسلتي عبر الإيميل .

والله من وراء القصد .

أخوكم : المبرمج العرماني

جمادي الأول لعام ١٤٢٨ هـ

AL_ARMANI