

بسم الله الرحمن الرحيم

المحاضرة الثالثة :

بيئة الفيجوال ستوديو دوت نيت - قسم ال Console :

- قبل التحدث والدخول إلى هذه البيئة يجب أن تمتلك نسخته منها وعنوانها :
Microsoft Visual Studio .NET 2003 وهي تتكون من 6 سيديات على الأغلب وتتكون من :

2 سي دي للدوت نيت 2003

3 سي دي للمكتبة MSDN

1 سي دي Component وهو عبارة عن مكونات ما قبل البداية

1 سي دي SQL 2000 Server وهو غالباً لا يكون موجود معها ((إضافي)) للتعامل مع قواعد البيانات

لنبدأ الآن :

إذا نظرنا إلى الإصدارات القديمة من مكتبة ال Microsoft Visual Studio مثل الإصدار السادسة أو الخامسة وأخذنا على سبيل المقارنة برنامج ال Visual C++ و برنامج ال Visual Basic وفارناهم كالتالي :

من الناحية النظرية :

إذا قمنا بالعمل والتصميم على برنامج السي ++ ثم فجأة إنتقلنا إلى واجهة الفيجوال بيسك سنرى هناك إختلافاً كبيراً بينهما . وهذه نقطة ضعف لدى مايكروسوفت . بينما لو نظرنا إلى مجموعة الأوفيس 2003 بالعمل على وورد ثم الإنتقال إلى بوربوينت مثلاً سنجد هناك تشابهاً بنسبة 95% إذا فشل الإصدار السادس من الناحية النظرية .

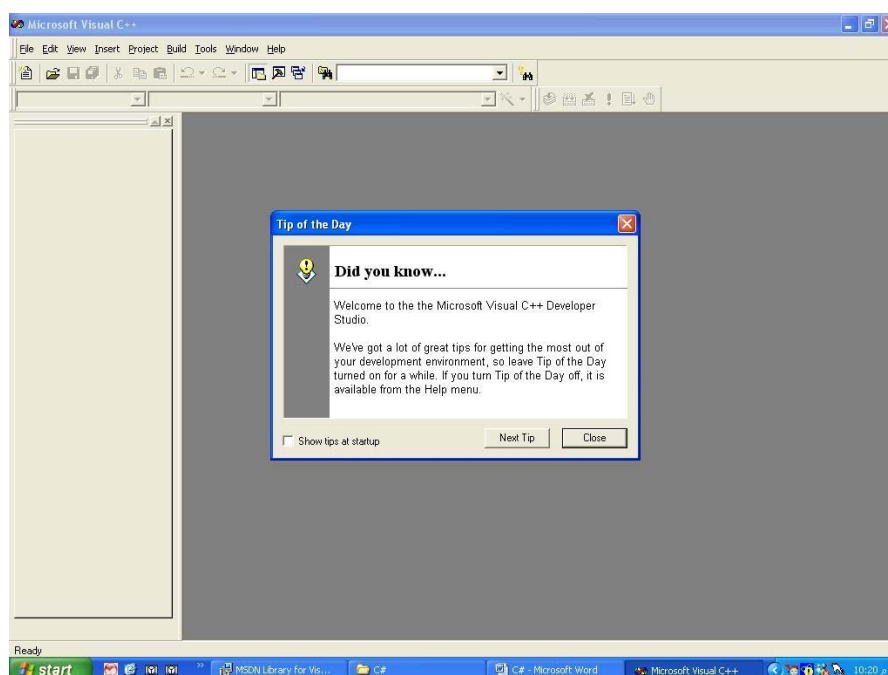
أما من الناحية العملية :

إليك التالي :

- قم بالضغط على

Start → Program File → Microsoft Visual Studio 6.0 → Microsoft Visual C++ 6.0

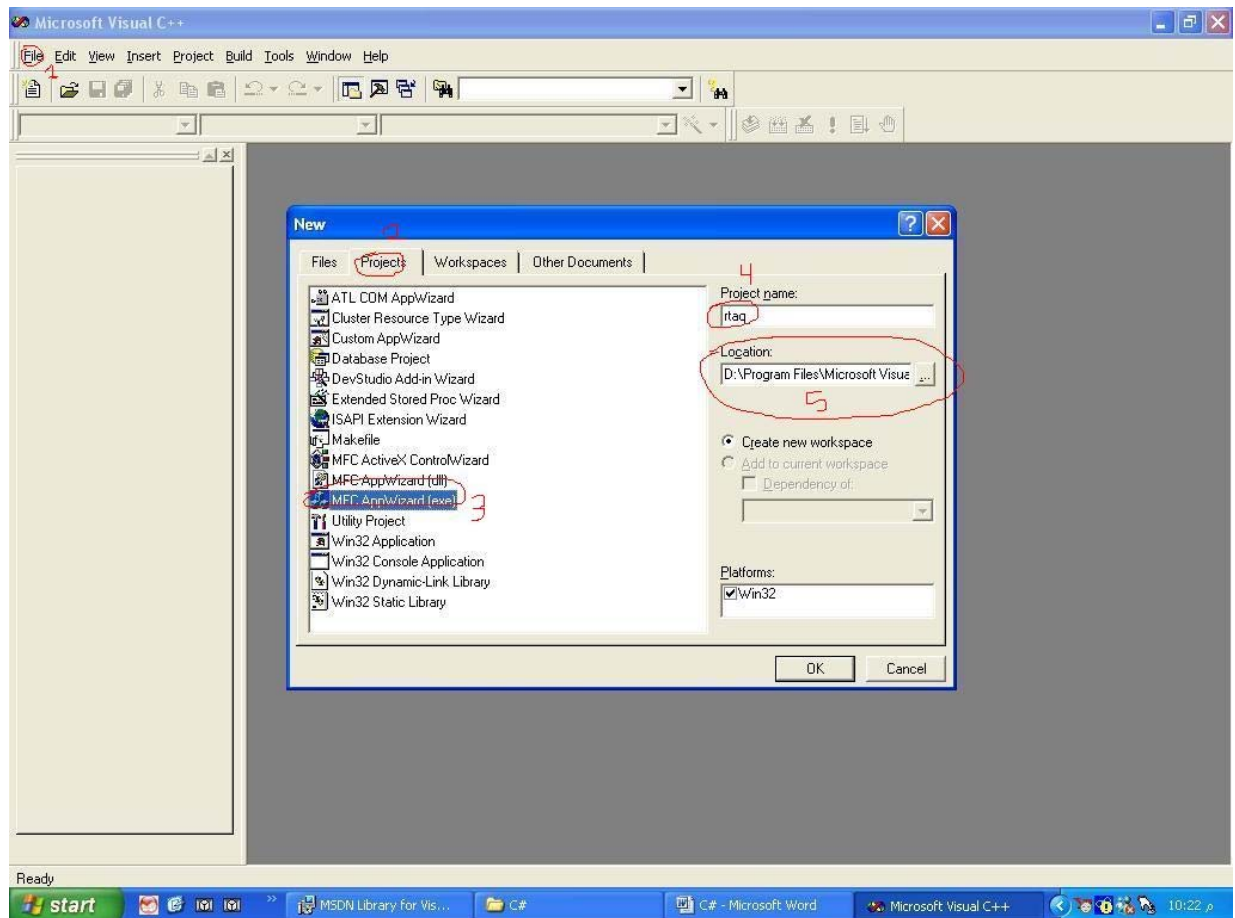
سيظهر لك الشكل التالي :



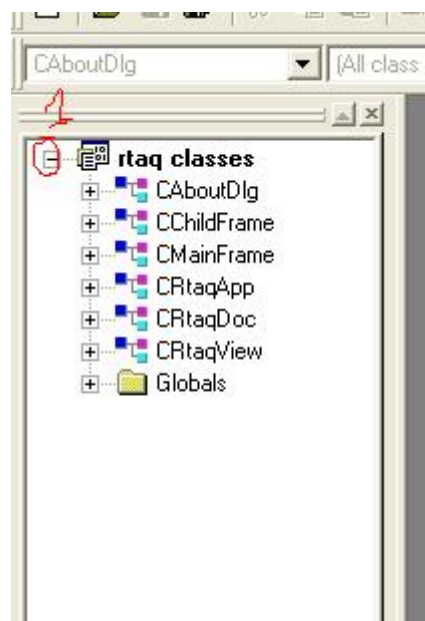
قم بالضغط على Close ثم قم باختيار

File → New → Project → MFC AppWizard (exe)

كالصورة التالية :



وقم باختيار الإسم لها كما في الرقم 4 و 5 ثم اضغط على موافق .
وسيطهر لك شاشة أخرى قم باختيار Finish منها ثم OK قم بفتح الشجرة من قسم نافذة المشروع
كالتالي :

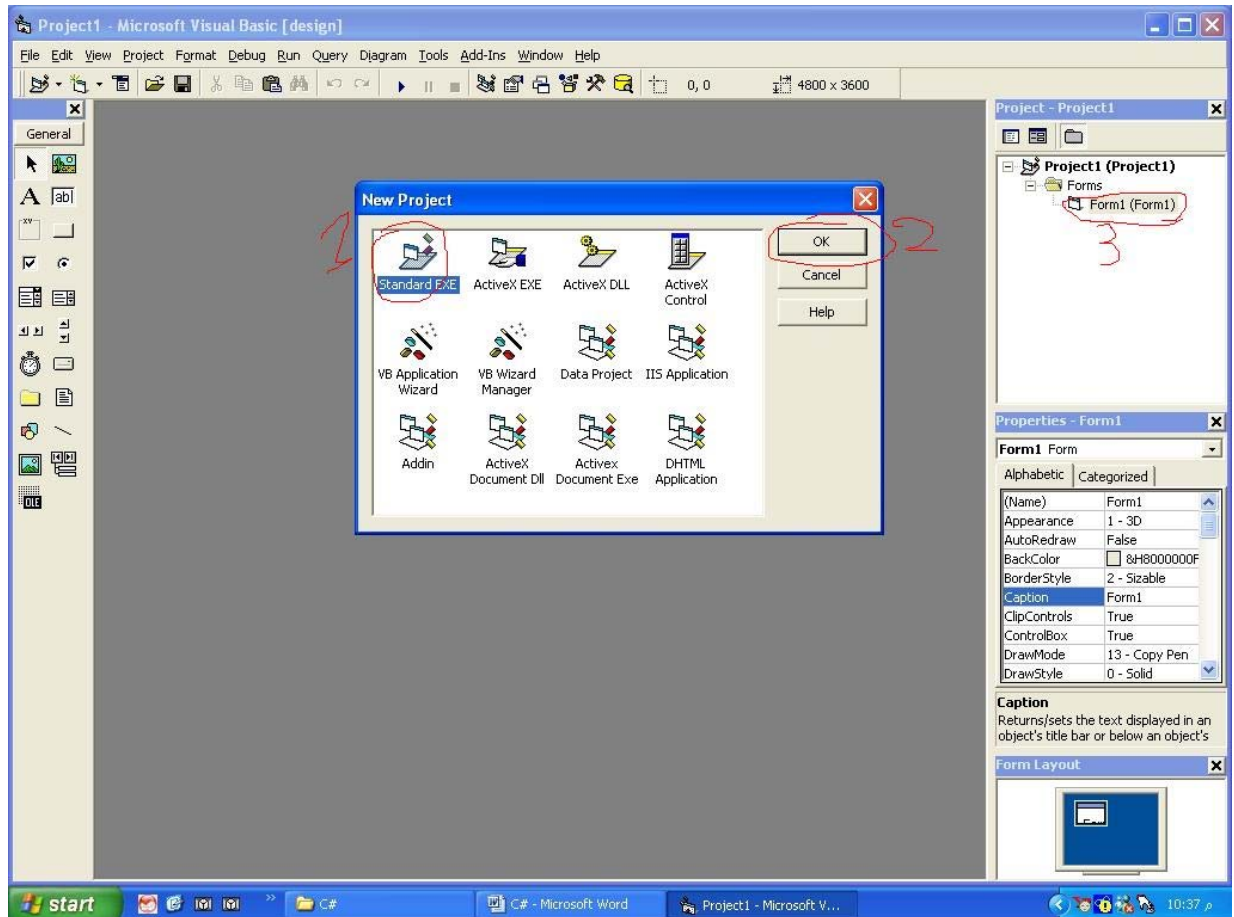


أنظر إلى عدد الـ Classes الموجودة هنا ستجدها 6 حسناً خزن هذا الرقم في مخيلتك

الآن إنتقل إلى فيجوال بيسك حسب التسلسل التالي :

Start → Program File → Microsoft Visual Studio 6.0 → Microsoft Visual Basic 6.0

سيظهر لك الشكل التالي :



لاحظ أننا اخترنا المشروع التنفيذي كما في السي ++ ولكن هنا إنظر إلى الرقم 3 ماذا تلاحظ؟؟ وجود ملف واحد فقط ((طيب في السي ++ موجود 6)) إذا فشل الإصدار السادس كذلك في الاختبار من الناحية العملية لأنه يجب على كل لغة في الإصدار أن تحتوي على نفس عدد الملفات في كل مشروع مشابه

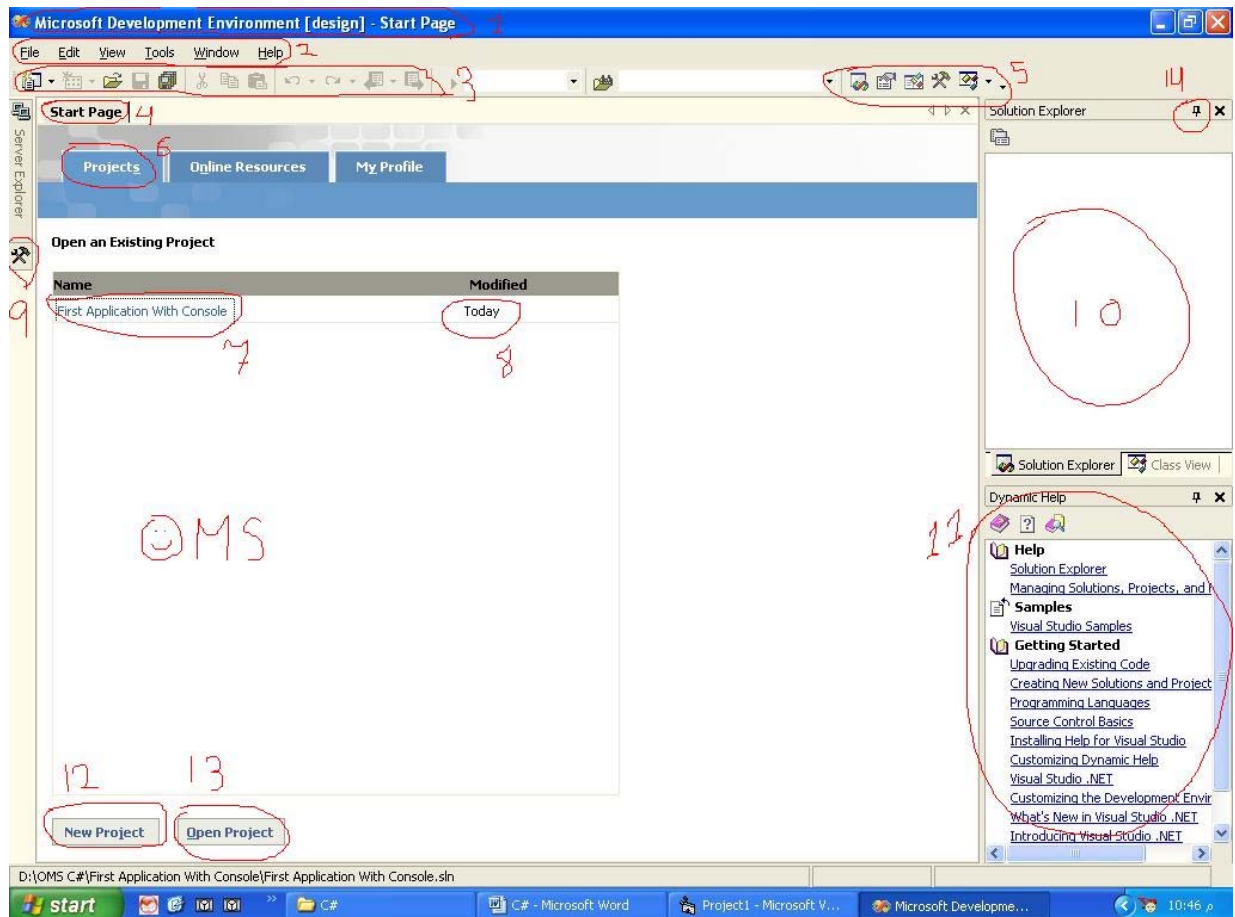
الآن وبعدما لاحظنا المقارنة في الإصدار السادس ننتقل إلى مجموعة الدوت نيت ((2003)) ونلاحظ الفرق :

قم بتتبع التسلسل التالي :

Start → Program File → Microsoft Visual Studio .NET 2003
→ Microsoft Visual Studio .NET 2003

ماذا تلاحظ من المسار؟؟ أنه لا يوجد تقسيمات للبرامج كما في الإصدار السادس يعني أنه لا يوجد سوى بيئة واحدة فقط لكل اللغات المضمنة في تلك اللغة . إذن من الناحية النظرية تمام .

بعد أن تتبع المسار ستظهر لك النافذة الرئيسية كالتالي :



لاحظ معي وبالترتيب لكي تتعرف على واجهة البرنامج :

- 1 - وهو شريط العنوان والكل يعرفه .
- 2 - شريط القوائم و به أوامر للمساعدة في بناء المشروع في بيئة الدوت نيت
- 3 - شريط الاختصارات و به أوامر مختصرة ((كثيرة الإستعمال)) من شريط القوائم ((2))
- 4 - صفحة البداية وهي الصفحة الرئيسية لهذه البيئة وتتكون من :
 - المشاريع Projects ((6)) وتتكون من التالي :
 - نافذة المشاريع وتحتوي على آخر أربع مشاريع ((رقم 7)) قمت بالعمل بهم مع التاريخ حيث يكتب التاريخ إما اليوم أو الأمس أو تاريخ العمل به ((التعديل عليه)) .
 - أزرار إنشاء أو فتح مشاريع سابقة ((12)) ((13)) على التوالي .
 - المصادر من الإنترنت Online Resource :
 - والمهمة لهذه الصفحة الحصول على الأمثلة أو طرح المشاكل التي تواجهك في منتديات البرمجة لموقع مايكروسوفت للحلول المباشرة وهذه الخدمة متوفرة شريطة الربط مع الإنترنت .
 - ملفك الشخصي My Profile :
 - هنا يمكنك التعديل على الواجهة بشكل عام أو إختصارات لوحة المفاتيح كالتالي :
 - لو فتحت هذه النافذة لوحدت الشاشة كالتالي :



Verify that the following settings are personalized for you:

Profile:

Visual Studio Developer

Keyboard Scheme: [Default Settings]

Window Layout: Visual Studio Default

Help Filter: (no filter)

Show Help: ☒ Internal Help ☐ External Help

At Startup: Show Start Page

- ((3)) يمكنك اختيار أي اختصارات للغة قد تعودت عليها من قبل كالسي ++ 6
 ((4)) يمكنك تغيير الشكل العام أو الإطار العام لبيئة الدوت نيت
 ((5)) يمكنك أن تختار طريقة عرض المساعدة للغة قد تعودت عليها من قبل
 * كل الأرقام السابقة يمكن أن تعطيك ملف خاص كما في ((2))
 و الاختيار السادس هو إجراء حدث عند بداية تنفيذ بيئة الدوت نيت .

5 - شريط لعرض النوافذ الموجودة أمامك فإذا إختفت واحدة ما عليك سوى الضغط على إسمها وستظهر لك حالاً .

9 - شريط الأدوات ولكن في السي شارب Console لن نستعمله فهو خاص بتطبيقات النوافذ .

10 - نافذة ملفاتك في مشروعك الذي تعمل عليه الآن .

11 - نافذة المساعدة .

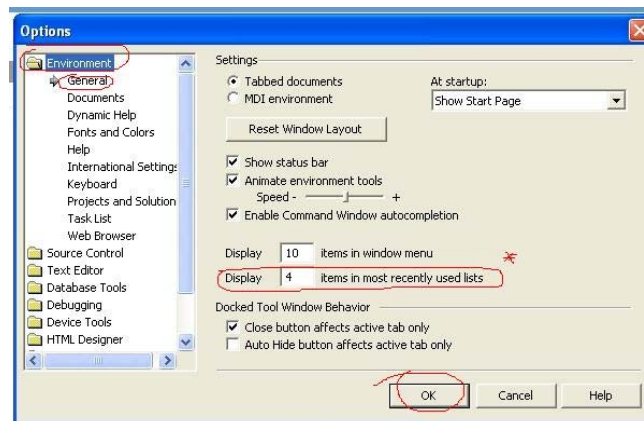
14 - في كل نافذة ستجد إشارة الدبوس هذه ومعناها أخفي تلقائياً بمعنى أنه إذا ذهب الماوس من فوق تلك النافذة قم بإخفائها تلقائياً . وإذا إقتربت منها قم بإظهارها تلقائياً .

ملاحظة (1) :

في القسم رقم ((6)) من الصورة السابقة يمكنك زيادة عدد المشاريع عن أربعة وذلك بالمسار التالي :

Tools → Option → Environment → General → Display recently

كالصورة التالية :



فقم بزيادة العدد عند إشارة النجمة ((*)) ثم قم بالنقر على موافق

ملاحظة (2) :

إذا بدأت البيئة ثم لم تظهر لك نافذة البداية ((صفحة البداية)) إذهب إلى :

Help → Show Start Page

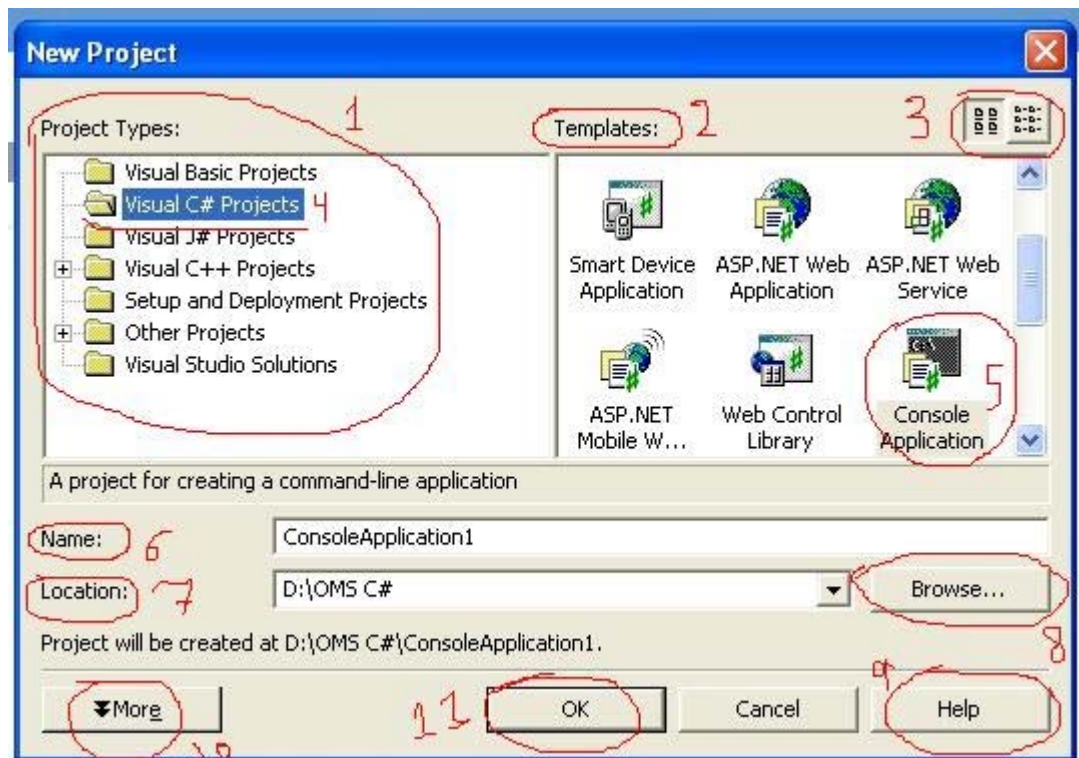
الآن وبعد ما تعرفت على أقسام النافذة الرئيسية لنبدأ بمثالنا الأول بال Console Application :

إذهب الآن إلى File → New → Project

أو قم بالضغط على Ctrl + Shift + N

أو قم بالنقر على زر المشروع الجديد من شريط الاختصارات .

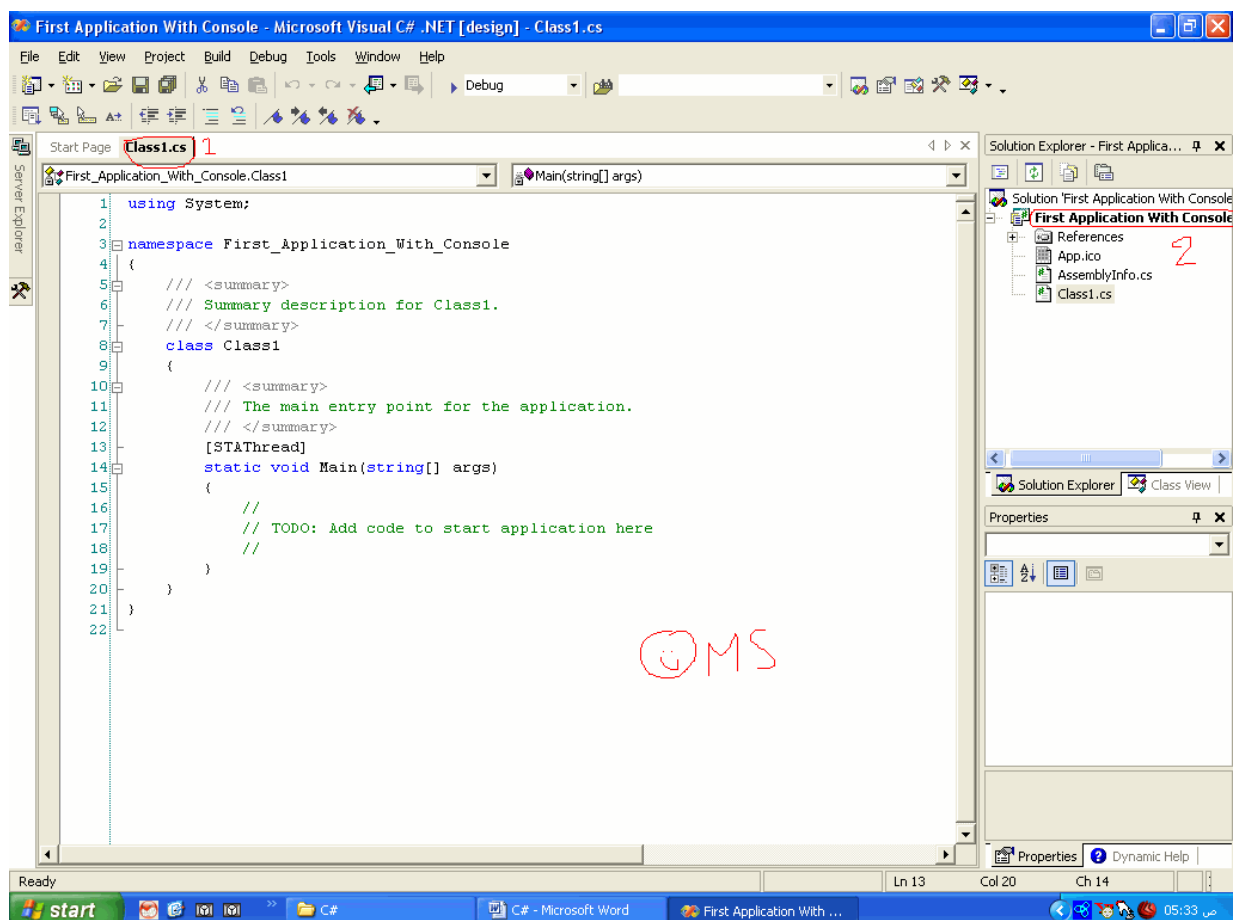
إذا نجحت في ذلك ستظهر لك النافذة التالية :



لنقف قليلاً عند هذه النافذة لكي نبين أجزائها :

- 1 - أنظر هنا في هذا القسم ((1)) ستلاحظ أسماء جميع اللغات المستعملة في بيئة الدوت نت وبهذا تكون شركة مايكروسوفت قد نجحت في تجميع جميع اللغات في إطار واحد .
- 2 - تجد جميع الأقسام ((القوالب)) التي يمكنك تصميمها مثل تطبيقات الويندوز وتطبيقات الكونسول وتطبيقات السمات ديفيس ((Pocket PC)) وذلك في القسم ((2)) .
- 3 - يمكنك أن تصغر وتكبر الأيقونات الموجودة في القسم ((2)) بواسطة الأزرار في القسم ((3)) .
- 4 - إذن الآن لنقم بفتح مشروع سي شارب فقم بالضغط على ((4)) .
- 5 - ثم بعدها قم بالضغط على ((5)) ولاحظ أن تطبيقات ال Console سبق لنا تعريفها وهي التطبيقات التي تفتح شاشة سوداء كنظام ال DOS .
- 6 - يمكنك تسمية المشروع من الرقم ((6)) وتذكر أنه سيسمي الملفات كلها بهذا الاسم ((ملفات العمل)) .
- 7 - من إسمها ((7)) تعرف أنها مكان منطقة العمل Directory . وبإمكانك تحديد مكان معين بالضغط على الزر الذي يحمل الرقم ((8)) .
- 8 - إذا ضغطت هذا المفتاح ((9)) سيفتح لك نافذة مساعدة عن هذه الصفحة فقط .
- 9 - بإمكانك زيادة الخيارات المستخدمة عندك بإضافة عملك في مجلد جديد على المسار الذي حددته في الخطوة رقم ((7)) وذلك بالضغط على الرقم ((10)) والذي يحمل كلمة More .
- 10 - وأخيراً قم بعمل OK ((موافق)) لنبدأ مشروعنا اليوم .

إذا نجحت في تطبيق السابق ستظهر معك النافذة التالية :



أنظر إلى النافذة السابقة ولاحظ معي :

1 – قام بفتح نافذة جديدة ((1)) وسماها Class1.cs .
يتكون الإسم من قسمين كالتالي :

القسم الأول يحمل إسم Class1 وهذه تعتبر الكائن الرئيسي في المشروع لأنه سبق لنا أن قلنا أننا نسعى للغة تدعم البرمجة الكائنية OOP ولأننا قلنا أنه يجب أن يكون هناك على الأقل كائن واحد وهو يحمل الدالة الرئيسية Main Function وبهذه الخطوة إرتقت مايكروسوفت إلى مستوى البرمجة بالكائنات والتي لم تكن موجودة بالإصدارات السابقة .

والقسم الثاني يحمل إسم الإمتداد cs ومعناه C Sharp أي إسم اللغة

2 – في القسم ((2)) ماذا تلاحظ ؟

المشروع يحتوي على 3 ملفات فقط . ملف للأيقونة الناتجة في المشروع وملف التحويل للغة الأسميلي والملف الثالث المحتوي على الكود الذي نقوم بكتابته .
ملاحظة سريعة هنا :

حاول و افتح مشاريع غير السي شارب ((ماذا تلاحظ)) ؟
ستقول لي كذلك تفتح 3 ملفات . وفي هذه الخطوة إستطاعت شركة مايكروسوفت توحيد أنماط جميع لغاتها ضمن باقة دوت نيت .

- لنأتي الآن إلى منطقة العمل ونشرحها بالتفصيل :
لاحظ معي النافذة كالتالي :

```

1  using System;
2
3  namespace First_Application_With_Console
4  {
5      /// <summary>
6      /// Summary description for Class1.
7      /// </summary>
8      class Class1
9      {
10         /// <summary>
11         /// The main entry point for the application.
12         /// </summary>
13         [STAThread]
14         static void Main(string[] args)
15         {
16             //
17             // TODO: Add code to start application here
18             //
19         }
20     }
21 }
22

```

((اقرأ الملاحظة بعد الموضوع مباشرة))

في السطر رقم (1) جملة `using System;` نستطيع إستنتاج التالي :

- * جميع الكلمات الموجودة في منطقة العمل والتي تحمل اللون الأزرق هي كلمات محجوزة Keyword لا نستطيع إستعمالها كمتغيرات .
- * تعتبر لغة السي شارب لغة حساسة Case Sensitive يعني أن المتغيرات (Age, AGE, aGE ,) كلها متغيرات لا يشبه بعضها بعضاً وتعاملها هذه اللغة كل واحدة على حده .
- * نهاية كل جملة تحتوي على فاصلة منقوطة ; وهي تعبر عن نهاية السطر .

ووظيفة السطر الأول هي إستدعاء مكتبة ((سننطق على تسميتها `namespace`)) للتعامل مع المشروع بشكل جيد مثل حمل الإدخال والإخراج وتعادل هذه الجملة بكلمة `#include` في لغة السي هنا إستدعى مكتبة ال `System` ولاحظ أن أول حرف كبير وهذه المكتبة مختصة بالدوال الرئيسية التي تستخدم بكثرة كحمل الإدخال والإخراج وتعادل هذه المكتبة مكتبة ال `iostream.h` المستخدمة في لغة السي .

في السطر رقم (3) جملة `namespace First_Application_With_Console` نستطيع إستنتاج التالي :

- * قام بإنشاء مكتبة تحتوي على المشروع الذي نكتبه الآن .
- * قام بوضع علامة تحت السطر (_) بدلاً من الفراغات والتي أصلاً إسم مشروعنا الحالي .
- * تلاحظ أنه يوجد مربع صغير يحتوي على إشارة ناقص (-) ووظيفته إخفاء تفاصيل الكلاس أو الدالة المشار إليها وبعد الضغط عليه يظهر لنا مستطيل يحتوي على ثلاث نقاط (...) إذا حركت الماوس عليه يعطيك محتوى الكلاس أو الدالة المشار إليها كاملاً كشكل ملاحظة Tag بمستطيل أصفر اللون فيعرض لك محتوياتها مهما بلغت من الطول .
- والهدف من هذا المربع هو إخفاء دالة أو كلاس سبق لنا أن كتبناها ولا نريد إظهارها .

ومعنى هذه الجملة أنه قام بإنشاء مكتبة خاصة والتي تحتوي على الـ Classes الموجودة في مشروعنا الحالي فمثلاً إذا أردنا إستعداد دالة من الدوال في مشروع آخر ما علينا سوى كتابة إسم المشروع الحالي ثم إتباعه بنقطة ثم إسم الكلاس أو الدالة التي نريد إستعمالها .

في السطر رقم (4) القوس المشهور ({}) والذي يدل على بداية الدالة أو الكلاس وطبعاً نغلقها بالمثل بإستخدام القوس المثل (}) كما في السطر رقم (21) .

في السطر رقم (5) <summary> /// جملة تعيق ولكنها للغة الـ XML دعها جانباً لن نغيدنا الآن في الوقت الحالي فلها وقتها . لاحظ أنها تحتوي على ثلاث أقواس .

في السطر رقم (7) لاحظ وجود إشارة (-) على العمود وهذا يعني بداية الجملة الأولى في الكلاس أو الدالة Function .

في السطر رقم (8) تلاحظ وجود إسم الكلاس المستعملة في مشروعنا الحالي .

في السطر رقم (13) [STAThread] أي كلمة موجودة بين قوسين ((مربعين)) تسمى خاصية Attribute وسنقوم بشرحها لاحقاً .

في السطر رقم (14) static void Main(string[] args) هنا توجد الدالة الرئيسية لمشروعنا .

والجملة تتكون من :

* static void تحديد نوع الدالة الرئيسية فهي من نوع Void التي تعني أن الدالة لا ترجع أي قيمة وهي من القسم Static من النوع الإستاتيكي .

والنوع الإستاتيكي يمكن شرحه كالتالي : لو أنك عرفت متغير ما بالنوع الإستاتيكي في دالة معينة في داخل كلاس معين ثم إستدعيت الدالة وأجريت تعديلات على هذا المتغير وخرجت من الدالة فإن الأصل أن يحذف المتغير من الذاكرة ولكن المتغير الإستاتيكي يقوم بتسجيل نفسه في الذاكرة ما دام البرنامج أو المشروع الذي صممته في وقت التنفيذ . فمثلاً لو عرفت في دالة معينة المتغير X من نوع Int من النوع الإستاتيكي وقمت في سطر تالي بزيادة هذا المتغير بقيمة واحد فإنه كلما قمت بإستدعاء الدالة سينفذ السطر الثاني فقط ويقفز عن السطر الأول لأنه موجود في الذاكرة.

دعه الآن له وقت سنشرحه بالتفصيل .

* Main لاحظ أن أول حرف كبير .

* (string[] args) وهي هنا تعني أننا بإمكاننا أن نستخدم الوسائط ((البارامترات))

فمثلاً لو أنشأنا مشروع لجمع عددين وقمنا بتسميته Sum طبعاً ستقول لي بعد تنفيذ المشروع نكتب الجمل اللازمة لكي يقوم بالحساب في ما بينهما وذلك بالطلب من المستخدم أن يدخل رقمين مباشراً من طريق جمل الإدخال . حسناً هنا بإمكانك قبل تنفيذ المشروع أن تدخل العددين وتقوم بالتعامل معهما فمثلاً نذهب إلى محرر الدوس ونقوم بكتابة الجملة التالي :

C:\> Sum.exe 152 965

فنستطيع مباشرة وبأول جملة في المشروع أن تعطيه الناتج .

في السطر رقم (16) // لاحظ وجود قوسين هنا وهما لجمل التعليقات ((لاحظ الفرق في السطر الخامس)) . أي أنها جمل لا معنى لها تضع التوضيح أو التعليق على الجمل في هذا السطر

ملاحظة :

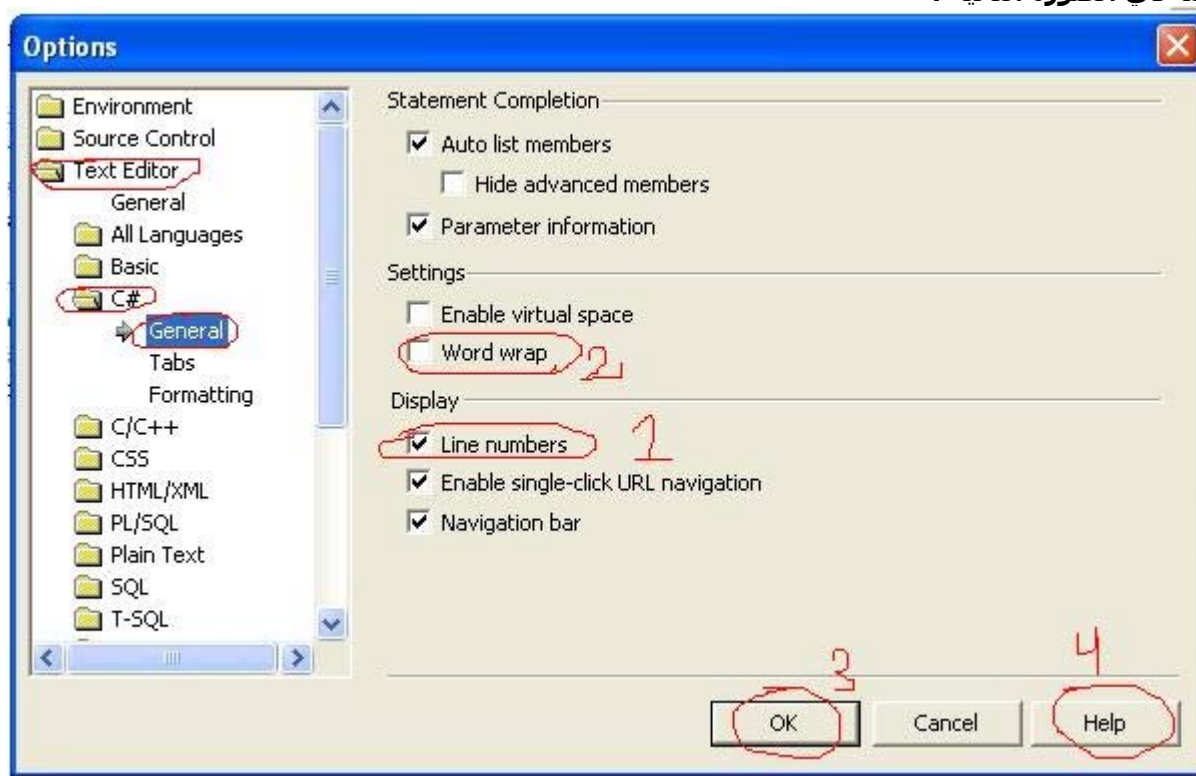
إذا واجهتك أي مشكلة في أي سطر وتريد معرفة المزيد قم بالنقر مرتين مزدوجتين على الجملة ثم قم بالضغط على F1 لظهور نافذة المساعدة بالجملة التي تريد فقط ((يجب أن تمتلك MSDN))

ملاحظة :

إذا لم تحتوي منطقة العمل على أرقام بإمكانك إضافتها بإتباع المسار التالي :

Tools → Option → Text Editor → C# → General → Line Numbers

كما في الصورة التالية :



قم بوضع ✓ على الرقم ((1))

كذلك بإمكانك أن تفعل خاصية Word Wrap ((2)) والتي تعني أنه بعد حد معين للسطر قم بإنزاله إلى السطر الجديد وبذلك أنت بغنى عن شريط ال Scroll Bar الأفقي .
أيضاً تستطيع إظهار معلومات أخرى عن هذه النافذة بالضغط على الرقم ((3)) .
الآن اختر كما في النافذة السابقة و اضغط الزر رقم ((3)) OK .

نكون هنا قد شرحنا كود البداية لملف السي شارب بطريقة ال Console .

الآن سنقوم بكتابة أول برنامج شهير وهو طباعة جملة " Hello World " بواسطة المكتبة System :

الآن قم بكتابة السطر التالي في الدالة الرئيسية :

```
static void Main(string[] args)
{
    Console.WriteLine("Hello World")
}
```

ولاحظ أنه يوجد حروف كبيرة كما في الشكل ((تحته خط))
الآن للناقش السطر السابق :

* إستخدمنا الدالة WriteLine والتي تسمح لنا بإظهار سطر على الشاشة وهي تأخذ قيمة من نوع String وبإمكانك أن تطبع المتغيرات والأسماء والأرقام .
* لاحظ أن الدالة السابقة خرجت من كلاس اسمه Console لتطبيق مبدأ ال Full OO .

يوجد أكثر من صيغة للدالة WriteLine . إليك الأمثلة كالتالي :

- 1 - Console.WriteLine("555");
- 2 - Console.WriteLine(555);
- 3 - Console.WriteLine("Hello To RTAQ");
- 4 - Console.WriteLine("Hello To {0}", "RTAQ");
- 5 - Console.WriteLine("Sum {0} + {1} = {2}", 5, 4, 5+4);

في المثال الأول :

سيكون الناتج 555 وسيعتبرها كأنها نص .

في المثال الثاني :

سيكون الناتج 555 وسيعتبرها كأنها رقم .

في المثال الثالث :

سيكون الناتج Hello To RTAQ وسيعتبرها كأنها نص .

في المثال الرابع :

سيكون الناتج Hello To RTAQ وسيعتبرها كأنها نص ولكن إنتبه هنا فيوجد تكنيك جديد وهو أننا بإمكاننا فصل الكلام إلى مواقع Index تعتبرها البيئة بالترتيب بداية من صفر فهنا قمنا بإضافة متغير ((نص)) إلى الجملة الأصلية Hello To الذي يمتلك Index = 0 فقمنا باستدعائه بالرمز {0} والموجود بعد الجملة الأولى مباشرة بعد الفاصلة .

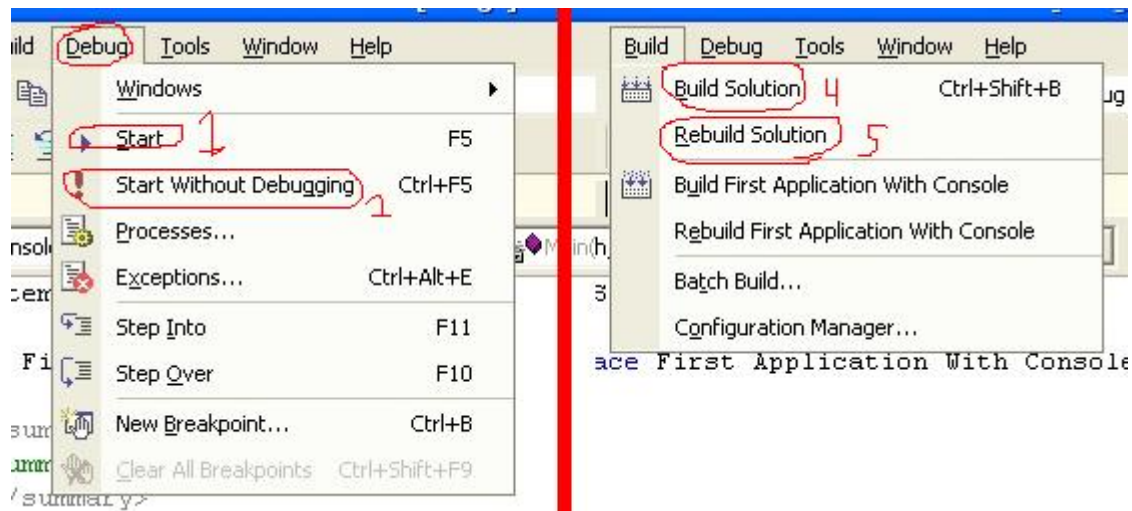
في المثال الخامس :

سيكون الناتج 5 + 4 = 9 هنا أيضاً قام بأخذ المواقع Index بالترتيب ووضعها في الجملة الأولى .

والآن إلى كيفية تنفيذ البرنامج :

فيعد كتابة الكود قم بالذهاب إلى Debug → Start Without Debugging أو قم بالضغط على الأزرار التالية : Ctrl + F5

لعلك تتساءل : لماذا هذه مع أنه يوجد أوامر كثيرة مثل ال Debug وال Start ؟؟ سأقوم بتوضيح الأمر لك . أنظر إلى الصورة التالية :



الرقم 1 : وهو مسئول مباشرة عن ظهور النافذة السوداء وتطبيق الكود .

الرقم 2 : كوظيفة ((1)) ولكن هنا يقوم بوضع سطر إضافي وهو سطر إيقاف لمشاهدة الكود ففي الكود السابق وبطريقة الرقم ((1)) تظهر الشاشة السوداء ولكن سرعان ما تنتهي .

الرقم 3 : يقوم بإنشاء الملفات اللازمة للمشروع بما فيها الملف التنفيذي EXE .

الرقم 4 : يقوم بحذف الملفات السابقة وإنشاء ملفات جديدة حسب التغيرات التي حدثت .