

الدرس الثالث كتابة كود صغير وتنفيذه بعد تنصيب البرنامج

كما أشرنا ، البرنامج المستخدم في هذه الدورة هو الـ Microsoft visual ++ C الإصدار السادس النسخة القياسية.

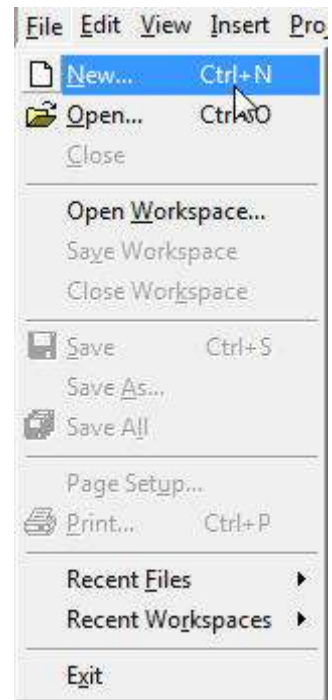
يمكن تحميله من [هنا](#).

بعد تحميل البرنامج وتنصيبه ، سوف تكون ايقونته على الشكل التالي



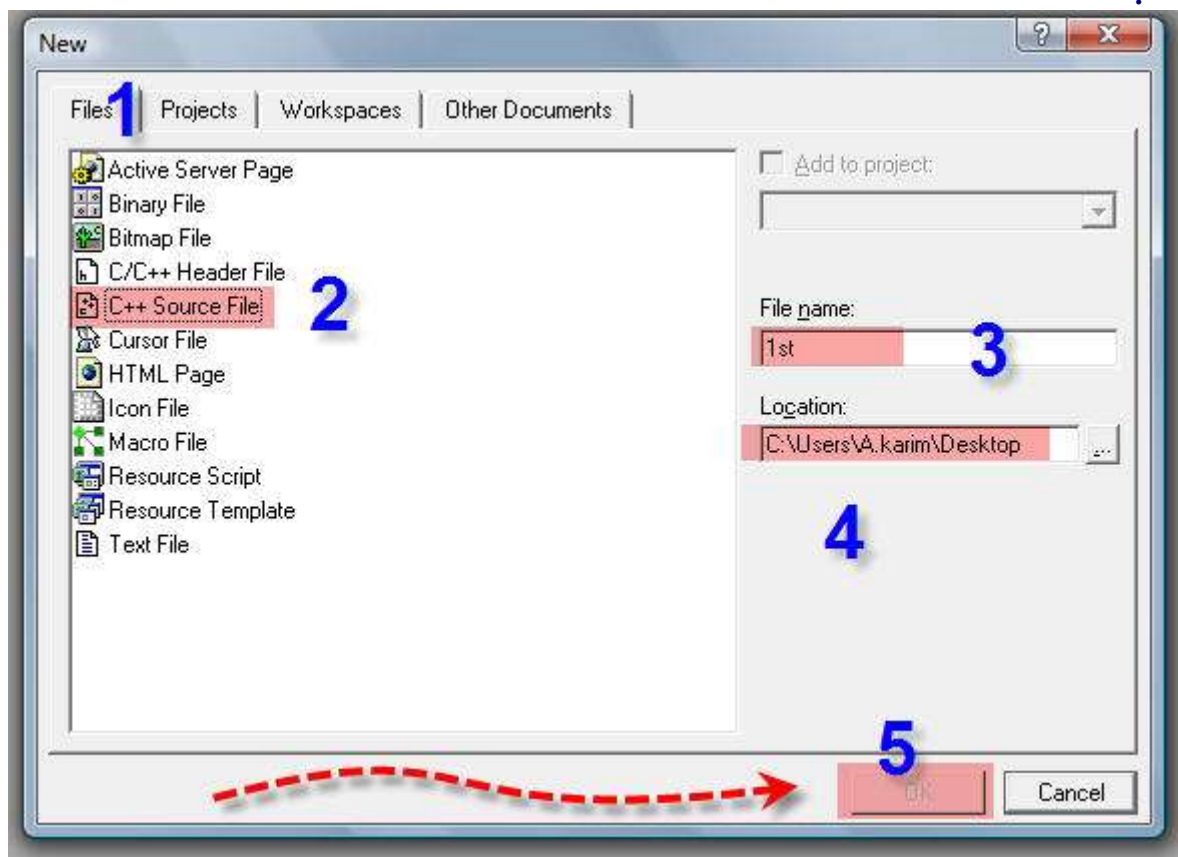
ولفتح ملف جديد ، نذهب للقائمة ملف ومن ثم اختيار جديد كما هو موضح بالصورة التالية

تم تحميل الدرس من شبكة المنهل التعليمية
<http://111000.net>



ومن ثم نتابع

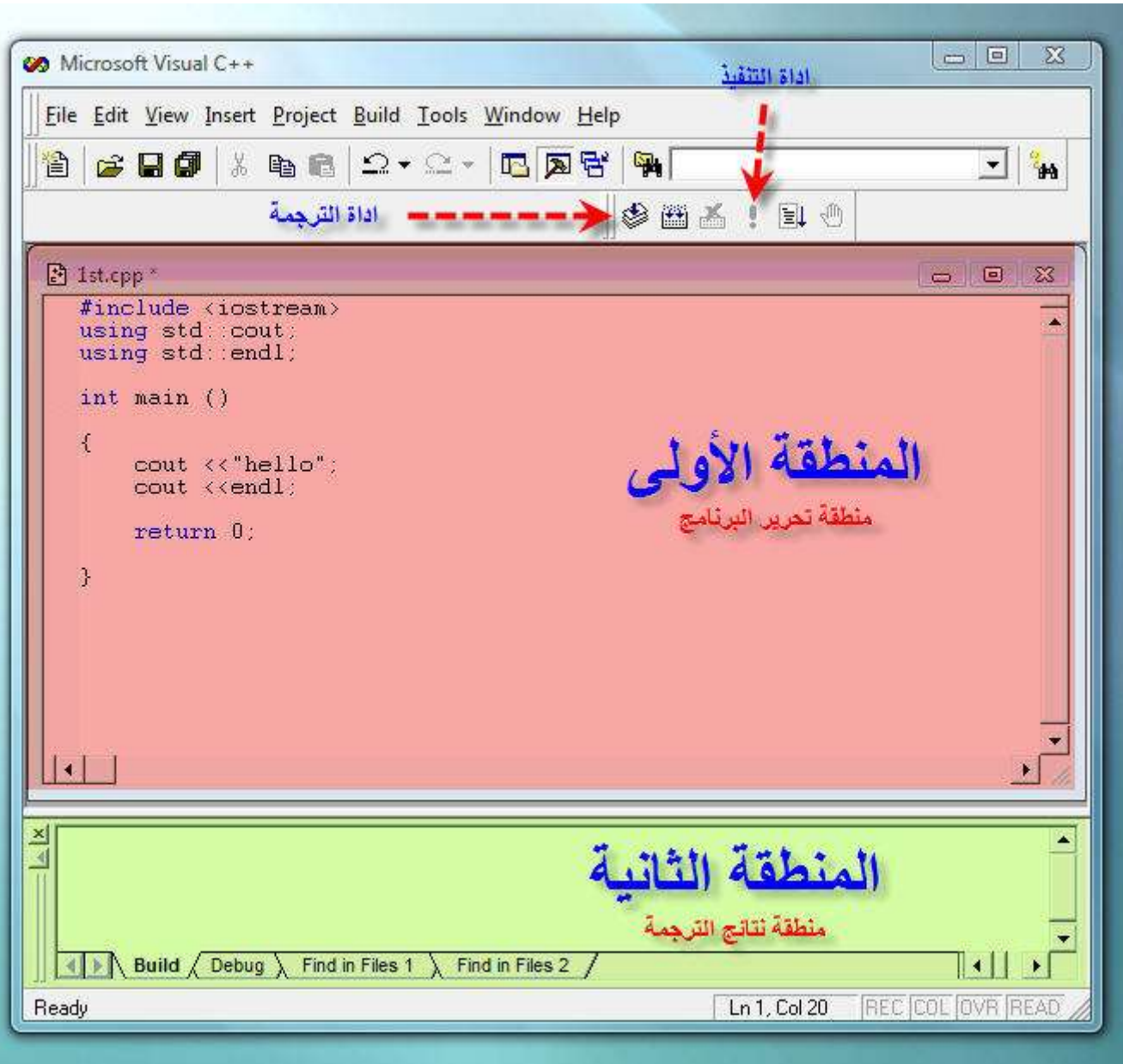
خطأ!



للمراسلة
w@111000.net

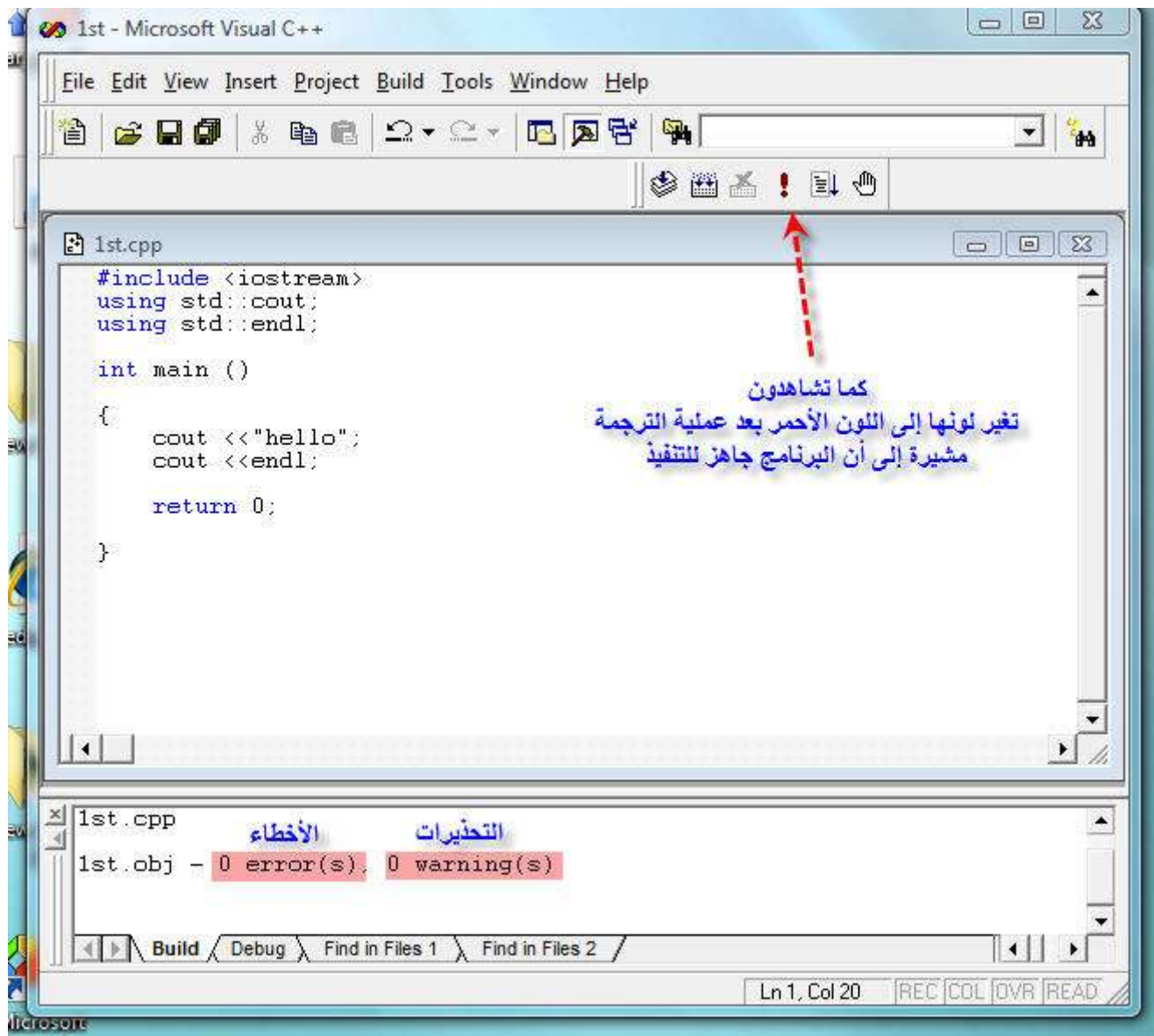
- ١- نختار التبويبة ملف.
 - ٢- لكتابة برنامج نختار من التبويبة ملف ملف ككود مصدر للـC++
 - ٣- نختار اسم للبرنامج (ضع اسم كيف تشاء وبما تقضيه غايتك)
 - ٤- اختيار موقع الملف.
 - ٥- الإعتماد النهائي لإنشاء الملف.
- بعد ذلك ، تفتح صفحة بيضاء ، يتم فيها كتابة البرنامج ، وتلاحظون الصفحة مقسمة لمنطقتين
- منطقة تحرير البرنامج (الحمراء) ومنطقة نتائج الترجمة (الخضراء)

خطأ!



بعدها كتابة البرنامج ، سوف تقوم بتصحيحة عن طريقة أداة الترجمة وسوف يظهر لك البرنامج بالشكل التالي

خطأ!



وعندما تظهر لك ايقونة التنفيذ (على شكل علامة تعجب) باللون الأحمر ، هذا يعني ان البرنامج جاهز للتنفيذ ، اضغط على هذه الأيقونة لتنفيذه.

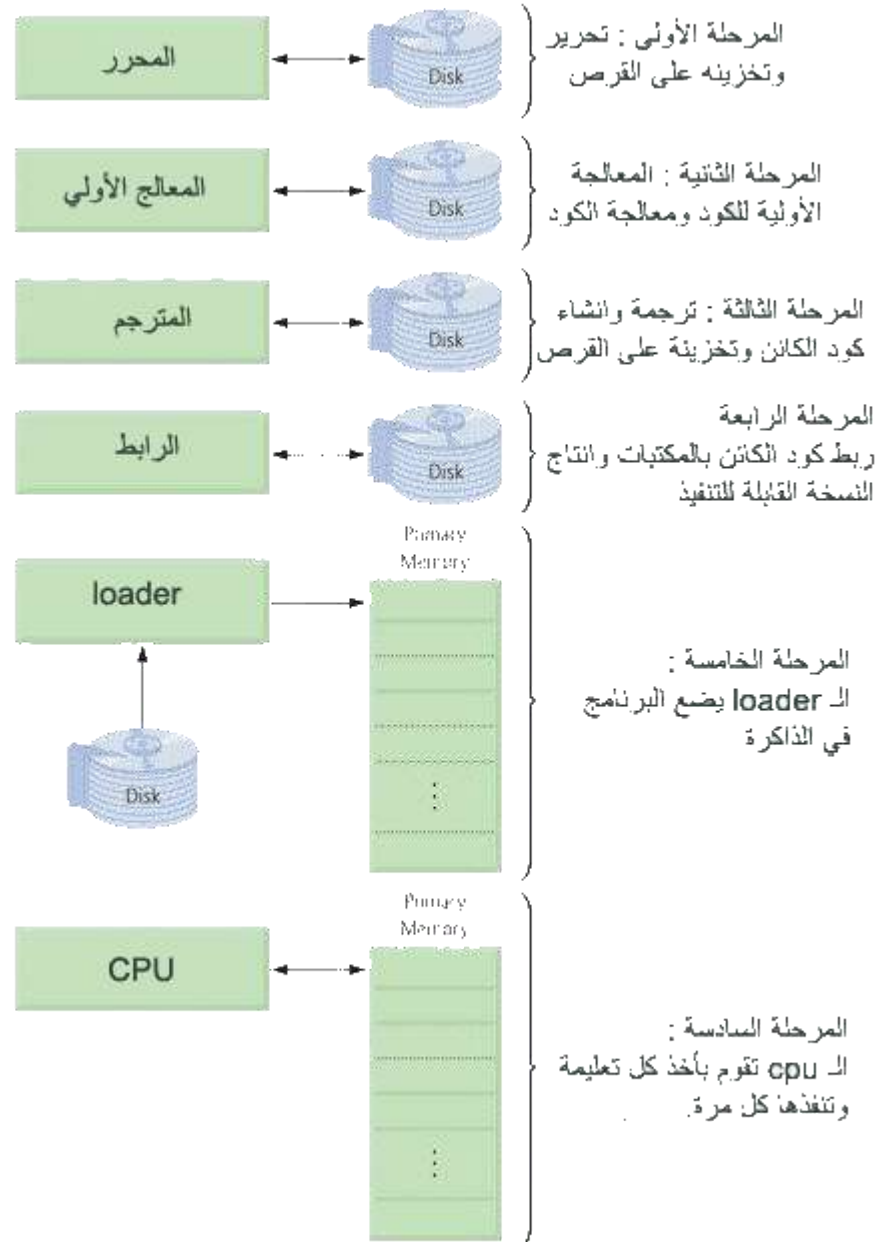
بيئة تطوير الـ C++ النمذجية

للمراسلة

w@111000.net

تم تحميل الدرس من شبكة المنهل التعليمية
<http://111000.net>

لنعتبر الخطوات في إنشاء وتنفيذ تطبيق الـ ++C باستخدام بيئة التطوير الموضحة في
الرسم التالي :



نظام ++C بشكل عام يتكون من ثلاثة اجزاء : بيئة تطوير البرلمج ... اللغة ... مكتبة الـ ++C القياسية.

برنامج الـ ++C عادةً يمر بستة مراحل : التحرير ... المعالجة الأولية .. الترجمة .. الربط .. التحميل والتنفيذ .

المرحلة الأولى ، انشاء البرنامج :

للمراسلة

w@111000.net

وتدرج خطوات هذه المرحلة كالتالي : طباعة سطور البرنامج (يشار إليه بكود المصدر source code) باستخدام أي محرر اكود ، عادة ما يسمى المحرر بال editor. ومن ثم تقوم باتباع الكود وتقوم بالتصحيحات المطبعية والمنطقية اللازمة ، وبعد ذلك تقوم بحفظ الكود بأحد الإمتدادات التالية .cpp ، .cxx ، .cc ، .c (لاحظ أن حرف السي كبير). وهذه الإمتدادات تعني أن الكود مكتوب بلغة الـ C أو الـ ++C .

المحرر المعتمد في هذه الدورة هو المحرر نفسه الذي يأتي مع الـ Microsoft visual ++C ٦ ، وهو ملحق ببيئة البرنامج ، وأنت تستطيع أيضا استخدام محرر نصوص بسيط مثل المفكرة notepad الذي يأتي مع الويندوز لتكتب كود المصدر .

وبافتراض أنك تعلم طريقة تحرير البرنامج (طباعة الحروف من لوحة المفاتيح) نكمل باقي المراحل.

الخطوات ، الثانية والثالثة ، المعالجة الأولية والترجمة : preprocessing and compiling

في المرحلة الثانية ، المبرمج يقوم بإعطاء الأمر للترجمة، برنامج المعالج الأولي ينفذ الأوامر وهي ما تسمى بموجهات المعالج الأولي وهي ما تشير إلى القيام بعمليات معينة قبل عملية الترجمة. هذه العمليات عادةً ما تكون الحاق ملفات نصية للبرنامج لي تتم ترجمته وأيضا إجراء بعض التبادلات النصية المتنوعة. أكثر موجهات المعالج الأولي شيوعا تمت الإشارة لها في الفصول الأولى بعد هذا الفصل. والشرح المفصل عن مكونات المعالج الأولي تكون مشروحة بشكل مفصل في الملحق F .

في المرحلة الثالثة ، يقوم المترجم compiler بترجمة البرنامج إلى لغة الآلة وينتج لدينا كود الكائن (object code).

عملية الترجمة ، هي عملية اكتشاف الأخطاء النصية والمنطقية في البرنامج ، قبل هذه العملية ، يقوم المعالج الأولي بعمليات أولية قبل الترجمة ، وهي بعض الأوامر تسمى (موجهات المعالج الأولي directives preprocessor). هذه العمليات عادةً ما تكون الحاق ملفات نصية بالبرنامج وإجراء بعض التبادلات النصية المتنوعة لكي تتم ترجمة البرنامج.

أكثر الموجهات شيوعا سوف يتم ذكرها في المحاضرات الأولى من هذه الدورة. والمزيد عنها تجدونه في الملحق F .

في المرحلة الثالثة يقوم المترجم الـ compiler بترجمة البرنامج إلى لغة الآلة لينتج لنا الكود الموجه للآلة object code.

المرحلة الرابعة ، الربط (linking) :

من الصعب في الـ ++C كتابة كل شيء أثناء البرنامج ، فمن نعم الله علينا في الـ ++C أن يكون هناك أشياء جاهزة للإستخدام نضع عناوينها فقط في البرنامج لأجل استثمار الكثيبيبيير من الوقت. كثير من الدوال والبيانات يتم تعريفها في أماكن أخرى خارج

البرنامج ، كالمكتبة القياسية مثلا ، وأيضا كمتبات لمشاريع خاصة ، تم تدوينها عن طريق مجموعة مبرمجين لغرض انجاز مشروع برمجي ما. أيضا ، هناك اشياء اخرى في كائن الكود ناتجة عن اجزاء ناقصة، يتم ربطها وسد الثغرات.

كل هذا يقوم به الرابط linker ، بعدها تنتج لدينا نسخة من البرنامج قابلة للتنفيذ ، تسمى image executable .

المرحلة الخامسة ، التحميل ، loading :

قبل تنفيذ البرنامج يجب تحميله للذاكرة ورفعها للذاكرة المؤقتة، وهذا يتم عن طريق ال loader الذي يقوم بأخذ ال executable image من القرص ووضعها على الذاكرة، الأجزاء الإضافية من المكتبات المشتركة هالذي يدعم البرنامج سوف تتحمل ايضا.

يقوم ال loader بحمل ال executable image ووضعها على الذاكرة ، ويلحق بها بعض الأجزاء الإضافية من المكتبات المشتركة التي تدعم البرنامج وذلك من أجل الجهيـز لعملية التنفيذ.

المرحلة السادسة ، التنفيذ ، executing :

آخر خطوة ، يأتي دور ال cpu ، حيث يقوم بتنفيذ الكود ، حيث يقوم تنفيذ تعليمة واحدة في كل مرة.

[color/]

[color="#ff0000"]مشاكل ربما تحدث عند عملية التنفيذ:

ربما يكون هناك اخطاء تمنع تنفيذ البرنامج سوف يتم شرحها خلال هذه الدورة. على سبيل المثال ، عند عملية القسمة على الصفر سوف يعترض عليها المترجم وذلك لانها عملية ليس لها ناتج معروف لديه، فهذه القيمة خيالية وغير معرفه، بالتالي سوف يتم عرض رسالة خطأ على الشاشة وعندها سوف نحلج إلى الرجوع للبرنامج وتصحيح الكود.

خطأ برمجي شائع :

الخطأ عن طريق القسمة على الصفر يحدث عند تنفيذ البرنامج. يصف هذا الخط من ا " اخطاء وقت التشغيل runtime errors أو " اخطاء وقت التنفيذ " execution-time errors . اخطاء وقت التشغيل الفادحة errors fatal runtime تنهي البرنامج

مباشرة ، أخطاء وقت التشغيل الغير فادحة *unfatal runtime errors* تسمح للبرنامج أن يكمل التنفيذ ولكن تظهر نتائج غير صحيحة. [ملاحظة : بعض أنظمة التشغيل تعتبر أن القسمة على الصفر ليست خطأ فادح].

برنامج صغير جاهز للتشغيل والتعود على محادثة التطبيق.

في المرفق برنامج صغير للتعرف على شاشة تنفيذ البرنامج ، وهي الشاشة السوداء (console) ، وذلك للتعود على شاشة المحادثة بين المستخدم والبرنامج.
حاول التعرف عليه بنفسك.
يمكن التحكم بتنسيق الشاشة عن طريق خصائص عرض الشاشة ، وذلك عن طريق الضغط بالزر الأيمن على شريط العنوان ومن ثم اختيار خصائص **properties**.

<http://www.arabteam2000-forum.com>