

مدخل إلى الخوارزميات والبرمجة هندسة الميكاترونكس سنة أولى

2023-2024

مدرس المقرر: د. عيسى الغنام

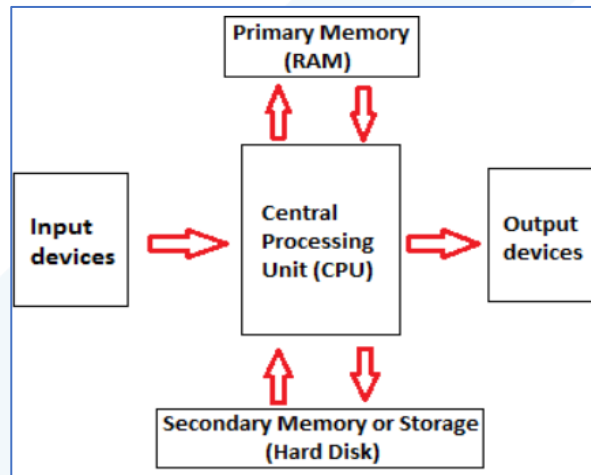
Lecture No.1

General Introduction

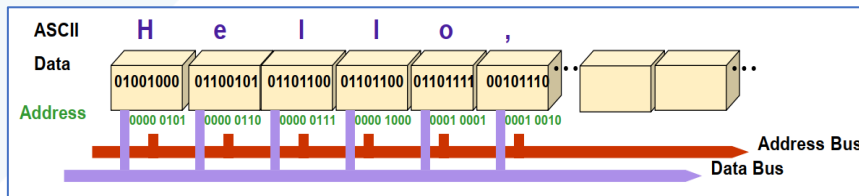
تهدف هذه المحاضرة إلى تمكين القارئ من التعرف على مفهوم البرمجة وأنواع البرمجيات ولغات البرمجة و تصنيفات الأكواد البرمجية وطرق التحويل بينها وأدواتها. وكذلك التعرف على بيئات برمجة للغة C++ مثل code blocks و visual studio وكتابة برنامج hello world بها وفهم بنية البرنامج وملفاته ضمن مجلد المشروع الجديد.

1-1 introduction:

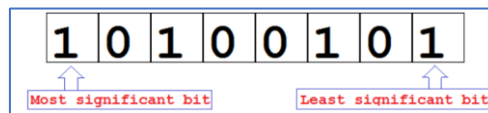
الحاسب يتكون من عتاد صلب HARDWARE يتمثل بـ CPU-Memory-Input-Output وعتاد برمجي أو برمجيات SOFTWARE



يمكن تشبيه الذاكرة ببناء فندق مكون من عدة طوابق وفي كل طابق يوجد عدة غرف كل غرفة تمثل حجرة ذاكرة



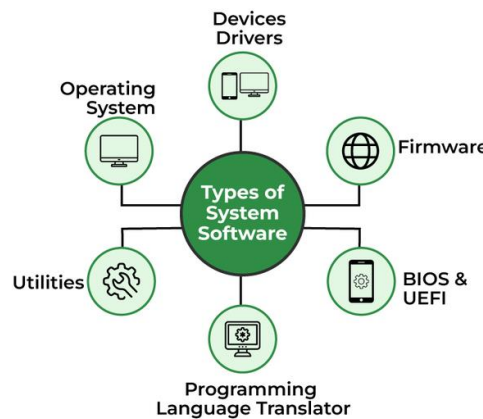
ترتيب خلية الذاكرة ذات حجم 1 بايت



وحدة المعالجة المركزية: تنجز العمليات الحسابية باستخدام وحدة الحساب/المنطق باستخدام نظام العد الثنائي وتفك شفرة الأوامر وتنفذها باستخدام وحدة التحكم.

البرمجيات Software: ولها أنواع:

- Computer Application Software (برامج تطبيقات الكمبيوتر): برامج مصممة لإنجاز مهام محددة واضحة للمستخدم. معالجات النصوص-عارض الشرائح.
- System software (برامج النظام): برامج تدعم تنفيذ وتطوير البرامج الأخرى. هناك أنواع لها:












✓ أنظمة التشغيل Operating systems مثل Windows®, UNIX®, Mac OS X تتحكم بمصادر الحاسب و تديرها نظام الملفات file system

✓ أنظمة الترجمة (أو التنسيق) Translation systems: translator هو برنامج يحول source code إلى كود الآلة machine code. بشكل عام، هناك أنواع من translators: **Compilers, Interpreters, Assemblers, linker**

1. linker هو برنامج يجمع بين ملفات الكائن والمكتبات object files and libraries لإنشاء كائن قابل للتنفيذ أو كائن قابل للمشاركة executable or a shared object. يقوم بحل الرموز (أسماء الوظائف والمتغيرات names of functions and variables) بين ملفات الكائن والمكتبات object files and libraries، ويولد الملف الثنائي النهائي الذي يمكن تنفيذه final executable binary file.

2. Assemblers يستخدم المجمع لترجمة البرنامج المكتوب بلغة Assembly إلى كود الآلة.

3. Interpreters المفسرات أو Compilers المترجمات هو برنامج خاص يقوم بترجمة الكود المصدري source code للغة البرمجة إلى كود آلة machine code أو bytecode أو لغة برمجة أخرى ذات مستوى أقل. عادةً ما تتم كتابة الكود المصدري source code بلغة عالية المستوى يمكن للبشر قراءتها high-level, human-readable language مثل Java أو C++ أو python.

Compiler Software / c++		
From sources across the web		
	Microsoft Visual C++ Freeware	▼
	GNU Compiler Collection GNU General Public License	▼
	Intel C++ Compiler Proprietary software	▼
	Clang Apache License	▼
	Portable C Compiler BSD licenses	▼
	MinGW GNU General Public License	▼
	LLVM BSD licenses	▼
	Mingw-w64 GNU General Public License	▼
	Intel Fortran Compiler Freeware	▼

بعض أنواع compilers للغة c++

يقوم compiler بترجمة كود المصدر بالكامل إلى كود الآلة قبل التنفيذ، ويتطلب الكثير من الوقت لتحليل كود المصدر. يعرض compiler كافة الأخطاء بعد الترجمة compilation. وفي حال عدم وجود أخطاء يبدأ التنفيذ وذلك يؤدي إلى تنفيذ أسرع نظرًا لعدم الحاجة إلى ترجمة أثناء وقت التشغيل.

من ناحية أخرى، يقوم interpreter بترجمة التعليمات البرمجية سطرًا تلو الآخر أثناء التنفيذ، مما يسهل اكتشاف الأخطاء بوقت قليل ولكن من المحتمل أن يؤدي إلى إبطاء تنفيذ البرنامج. عادة لغة python تستخدم هذا النوع من المفسرات.

أنواع CODES:

1. Source code يمكن للبشر قراءتها... بلغة مثل C أو شيء من هذا القبيل. تتم كتابة كود المصدر في بعض لغات البرمجة، حتى يتمكن الإنسان من التعبير عنه، وقراءته، وفهمه، وتعلمه، ونقده، وما إلى ذلك. وهي تعبير بشري عما يريده المبرمج من الحاسب أن ينفذ.
2. Machine code and binary متماثلان - نظام أرقام ذو أساس 2 - إما 1 أو 0. ولكن يمكن أيضًا التعبير عن رمز الآلة بتنسيق سداسي عشري (سداسي عشري) - نظام أرقام ذو أساس 16. Object code كود الكائن هو ما تحصل عليه عندما تقوم compile ترجمة a single source code module إلى لغة آلة.
3. Executable code هو نتيجة ربط جميع وحدات كود الكائن المختلفة معًا - ويمكن بعد ذلك تحميل هذا الملف في الذاكرة بواسطة نظام التشغيل عند "تنفيذ" البرنامج. تحويل محدد للكود المصدر إلى كود كائن ثنائي قابل للتنفيذ. العملية العكسية تكاد تكون غير ممكنة على الإطلاق بطريقة مفيدة.

2-1 Programming languages:

لغة البرمجة هي نظام تدوين لكتابة برامج الكمبيوتر. البرنامج هو تسلسل من الأوامر يقوم المعالج بتنفيذها. معظم لغات البرمجة هي لغات تستند إلى النص text، ولكنها قد تكون أيضًا رسومية graphical. تصنف إلى:

1. Procedural programming languages	• High-level vs. low-level languages
2. Functional programming languages	• Front-end vs. back-end languages
3. Object-oriented programming languages	• Interpreted vs. compiled languages

4. Scripting languages	
5. Logic programming languages	

1. C++ و Python هي من أهم وأقوى اللغات المستخدمة حالياً.
2. C++ هي لغة برمجة شائعة. تُستخدم لغة C++ لإنشاء برامج تطبيقات كمبيوتر Computer Application Software وهي واحدة من أكثر اللغات استخداماً في تطوير الألعاب game development. لغة C++ ممتعة وسهلة التعلم!
3. لغة C++ يمكن تكييفها مع منصات متعددة cross-platform language . يمكن العثور على C++ في أنظمة التشغيل الحالية وواجهات المستخدم الرسومية والأنظمة المضمنة.
4. تم تطوير C++ بواسطة Bjarne Stroustrup، كامتداد للغة C. تم تطوير C++ كامتداد لـ C، ولكلا اللغتين نفس البنية تقريباً. الفرق الرئيسي بين C و C++ هو أن C++ تدعم classes الطبقات و objects الكائنات، بينما لا تفعل ذلك.
5. تعد اللغات عالية المستوى أقل كفاءة في الذاكرة less memory efficient ولكنها أكثر ملاءمة للإنسان much more human friendly هذا يجعلهم أسهل في الكتابة والفهم والصيانة والتصحيح. تعتبر لغات البرمجة الأكثر شيوعاً المستخدمة اليوم لغات عالية المستوى.
6. C++ هي لغة برمجة عالية المستوى للأغراض العامة high-level, general-purpose programming language. تسمح للمبرمج بكتابة برامج فعالة و بنيوية و غرضية التوجه. C++ هي لغة برمجة موجهة للكائنات والتي تعطي بنية واضحة للبرامج وتسمح بإعادة استخدام الكود، مما يقلل من تكاليف التطوير.
7. البرمجة الكائنية (Object-oriented programming OOP) هي نموذج برمجة كمبيوتر computer programming model ينظم تصميم البرامج software حول البيانات data أو الكائنات objects، بدلاً من التتابع functions والمنطق logic. يمكن تعريف الكائن object على أنه حقل بيانات data field له سمات وسلوك فريد unique attributes and behavior.
8. يمنح C++ المبرمجين مستوى عالياً من التحكم في موارد النظام والذاكرة.
9. تم تحديث اللغة 4 مرات رئيسية في 2011 و 2014 و 2017 و 2020 إلى C++ 11 و C++ 14 و C++ 17 و C++ 20.
10. نظراً لأن C++ قريبة من C و C# و Java، فإنه يسهل على المبرمجين التبديل إلى C++ أو العكس.
11. لبدء استخدام C++، تحتاج إلى شيتين: محرر نصوص، مثل Notepad، لكتابة كود C++ و مترجم compiler، مثل GCC، لترجمة كود C++ إلى لغة يفهمها الكمبيوتر. هناك العديد من برامج تحرير النصوص والمترجمين للاختيار من بينها. في هذا البرنامج التعليمي، سوف نستخدم IDE.

12. بيئة التطوير المتكاملة (An integrated development environment IDE) هي تطبيق برمجي software

application يوفر تسهيلات شاملة لمبرمجي الكمبيوتر computer programmers لتطوير البرامج software

development. حيث يحوي:

1. Editor محرر
2. Compiler مترجم
3. Linker موصل
4. Debugger مصحح

13. تشمل IDE المشهورة Code :: Blocks و Eclipse و Visual Studio. هذه كلها مجانية، ويمكن استخدامها لتحرير

edit وتصحيح debug كود C ++.

14. Code :: Blocks هو IDE مجاني ومفتوح المصدر ومتعدد المنصات يدعم العديد من compilers بما في ذلك GCC

و Clang و C ++ Visual .

Downloads: There are different ways to download and install Code::Blocks on your computer:

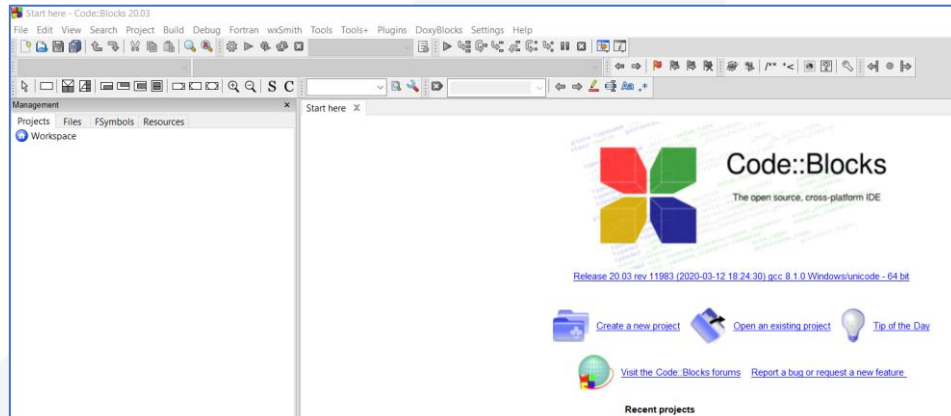
1. Download the binary release: This is the easy way for installing Code::Blocks. Download the setup file, run it on your computer and Code::Blocks will be installed, ready for you to work with it. Can't get any easier than that!
2. Download the source code: If you feel comfortable building applications from source, then this is the recommend way to download Code::Blocks. Downloading the source code and building it yourself puts you in great control and also makes it easier for you to update to newer versions or, even better, create patches for bugs you may find and contributing them back to the community so everyone benefits.

<p>Code::Blocks / Downloads / Source code</p> <h2>Source code</h2> <p>If you want to download the source code of Code::Blocks 20.03, here are the links</p> <table border="1"> <thead> <tr> <th>File</th> <th>Download from</th> </tr> </thead> <tbody> <tr> <td>codeblocks-20.03.tar.xz</td> <td>FossHUB or Sourceforge.net</td> </tr> <tr> <td>codeblocks-20.03.tar.bz2</td> <td>FossHUB or Sourceforge.net</td> </tr> </tbody> </table>	File	Download from	codeblocks-20.03.tar.xz	FossHUB or Sourceforge.net	codeblocks-20.03.tar.bz2	FossHUB or Sourceforge.net	<h2>Binary releases</h2> <p>Please select a setup package depending on your platform:</p> <ul style="list-style-type: none"> Windows XP / Vista / 7 / 8.x / 10 Linux 32 and 64-bit Mac OS X <div style="text-align: center;">  <h3>Microsoft Windows</h3> </div> <table border="1"> <thead> <tr> <th>File</th> <th>Download from</th> </tr> </thead> <tbody> <tr> <td>codeblocks-20.03-setup.exe</td> <td>FossHUB or Sourceforge.net</td> </tr> </tbody> </table>	File	Download from	codeblocks-20.03-setup.exe	FossHUB or Sourceforge.net
File	Download from										
codeblocks-20.03.tar.xz	FossHUB or Sourceforge.net										
codeblocks-20.03.tar.bz2	FossHUB or Sourceforge.net										
File	Download from										
codeblocks-20.03-setup.exe	FossHUB or Sourceforge.net										

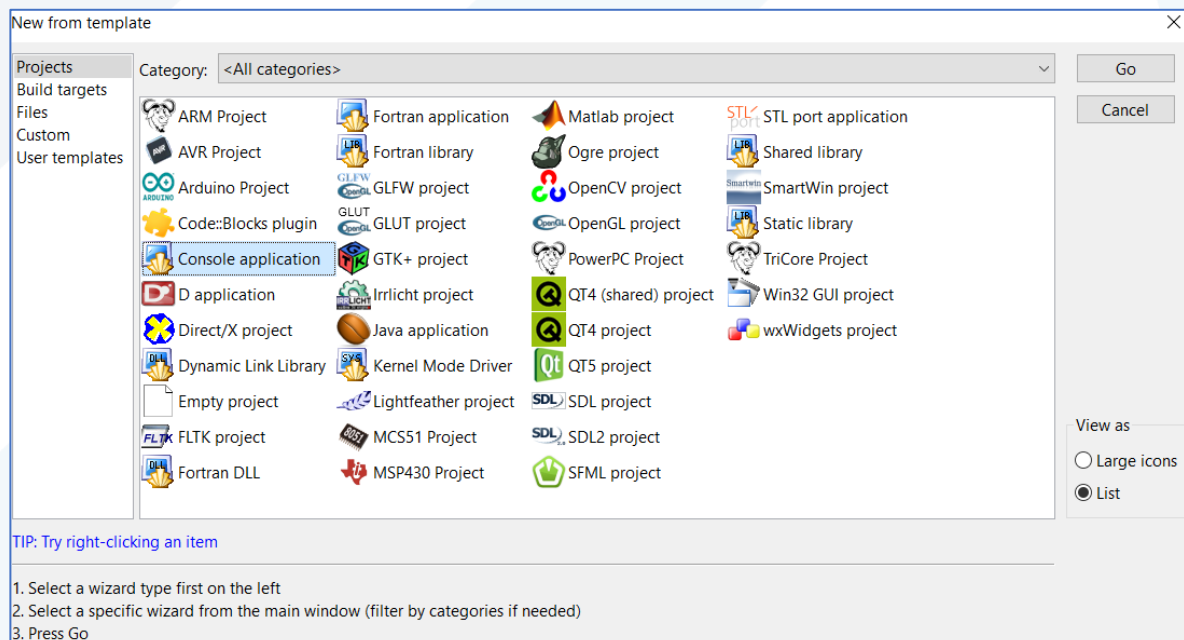
- Source code: codeblocks-20.03.tar.xz
- codeblocks-20.03mingw-setup.exe recommended

3-1 Creating a new project - The project wizard:

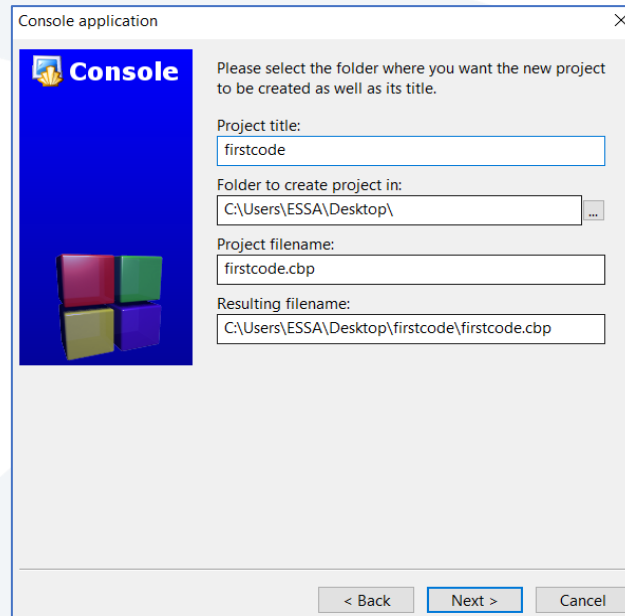
1. Launch the Project Wizard through File->New->Project... to start a new project. Here there are many pre-configured templates for various types of projects, including the option to create custom templates.



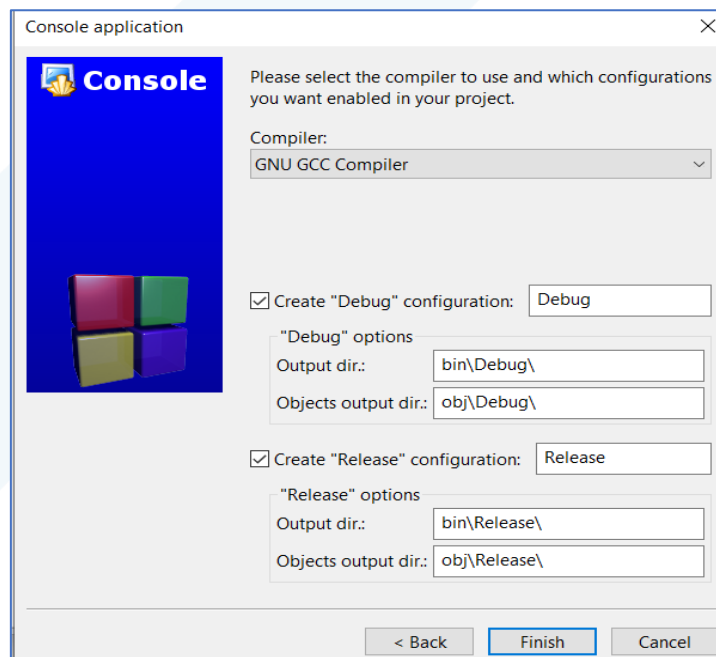
2. Select Console application, as this is the most common for general purposes, and click Go. The console application wizard will appear next. Continue through the menus, selecting C++ when prompted for a language.



3. In the next screen, give the project a name and type or select a destination folder. As seen below, Code::Blocks will generate the remaining entries from these two.



4. Finally, the wizard will ask if this project should use the default compiler (normally GCC) and the two default builds: Debug and Release. All of these settings are fine. Press finish and the project will be generated. The main window will turn gray, but that is not a problem, the source file needs only to be opened. In the Projects tab of the Management pane on the left expand the folders and double click on the source file main.cpp to open it in the editor.



GNU Compiler Collection (GCC): عبارة عن مترجم محسن optimizing compiler تم إنتاجه بواسطة مشروع GNU يدعم لغات البرمجة المختلفة various programming languages وبنى الأجهزة hardware architectures وأنظمة التشغيل operating systems.

GCC عبارة عن toolchain تقوم بترجمة compiles الكود البرمجي، وربطها links بأي library dependencies، وتحويل هذا الكود البرمجي إلى لغة تجميع assembly، ثم تحضير الملفات القابلة للتنفيذ executable files.

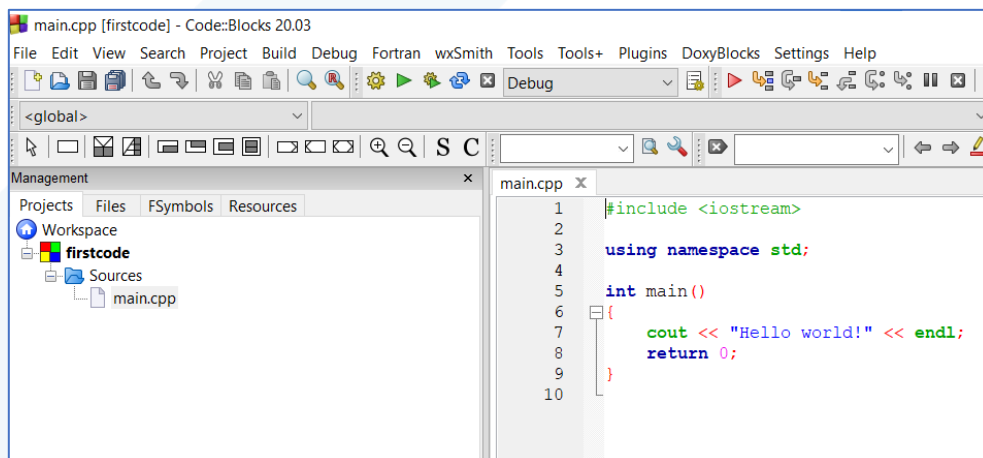
GNU عبارة عن مجموعة كبيرة من البرامج المجانية، والتي يمكن استخدامها كنظام تشغيل.

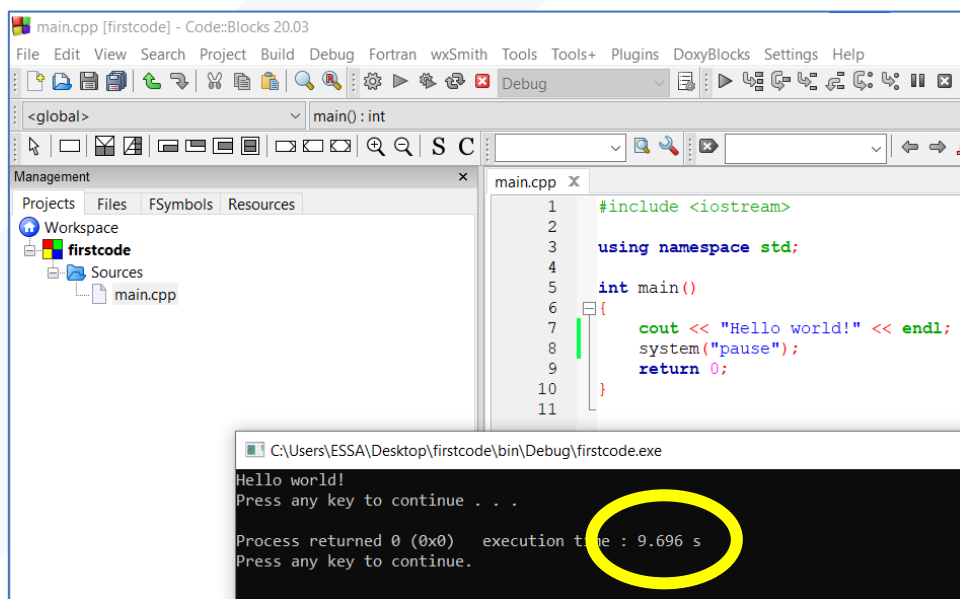
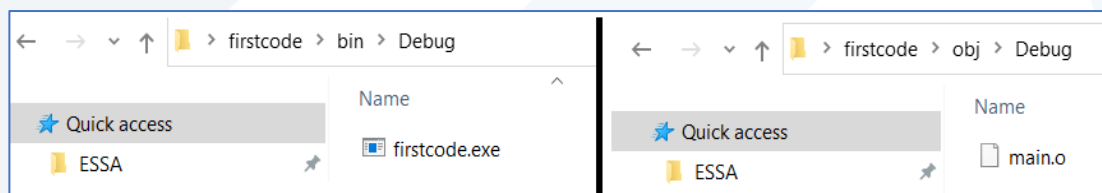
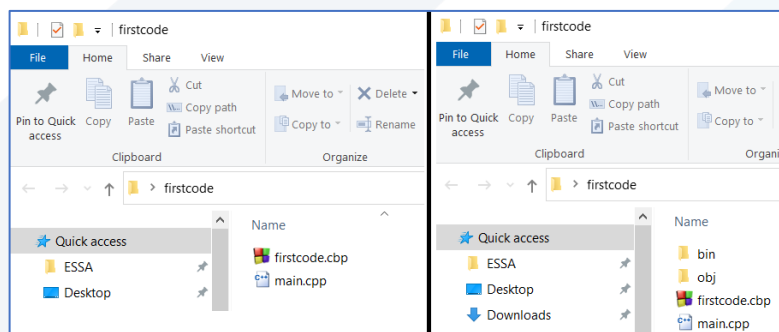
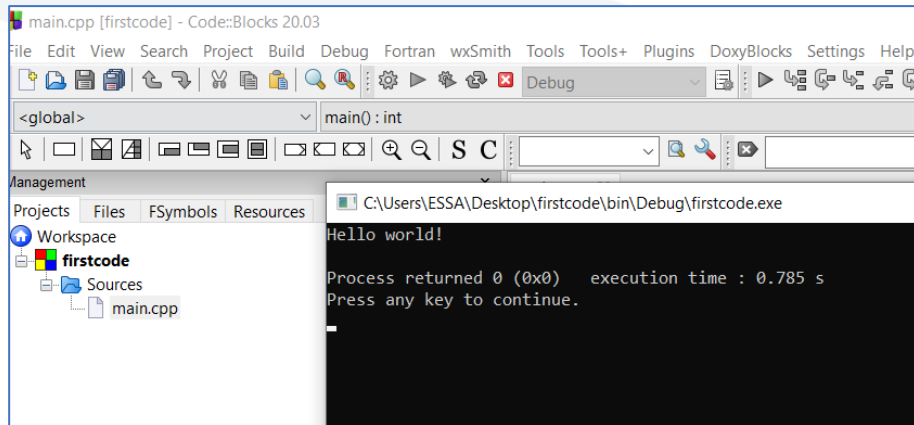
القسم السفلي من النافذة ستحدد مكان وضع نسخ Debug و Release المصرفة الخاصة ببرنامجك. اترك هذه الاعدادات كما هي واضغط Finish.

- The main file contains the following standard code.

main.cpp:

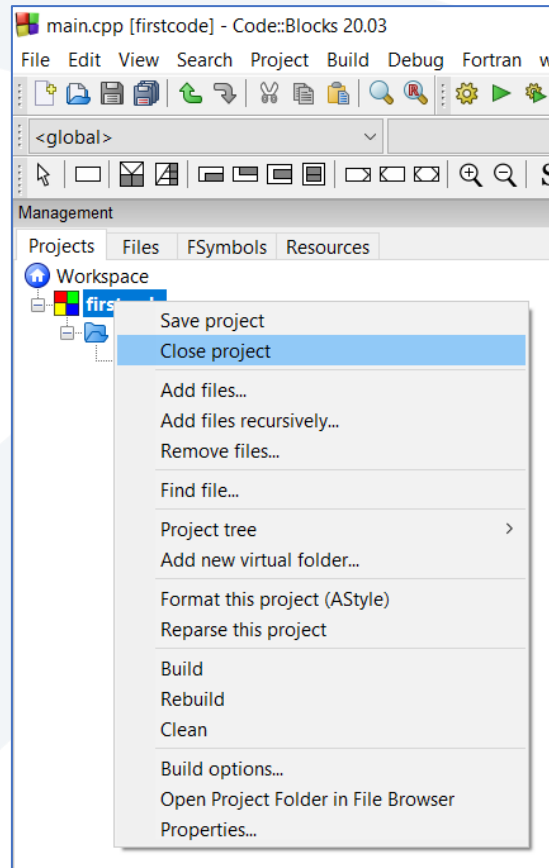
```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```





غير نمط debug build إلى نمط release build وقارن مجلدات المشروع الجديد.

Note: Debug builds can be very slow



4-1 Explanation of C++ Hello World Program Code:

1. `#include <iostream>`
2. `using namespace std;`
3. `int main()`
4. `{`
5. `cout << "Hello world!" << endl;`
6. `return 0;`
7. `}`

ملاحظة: يجب انهاء كل تعليمة بفاصلة منقوطة semicolon.

- # وهو عبارة عن توجيه يرشد المترجم (المعالج المسبق) إلى تضمين ملفات C++ رأسية القياسية هنا مكتبة الإدخال/الإخراج القياسية iostream.h في البرنامج. بدون تضمين هذا الرأس، لن يتم ترجمة التعبير الموجود في السطر 5.

iostream هي مكتبة ملفات رأسية تتيح لنا العمل مع كائنات الإدخال والإخراج، مثل cout المستخدم في السطر 5. تضيف ملفات الرأس وظائف إلى برامج C++. iostream: هو ملف رأسي لمكتبة قياسية يحتوي على تعريفات الإدخال والإخراج القياسية. يتم تضمين هذه التعريفات في namespace std. يوفر ذلك طرقاً للبرامج للحصول على الإدخال والإخراج من نظام خارجي - عادةً ما يكون الجهاز الطرفي.

• <> يعني أن الملف الموجود بداخله هو ملف رأس قياسي لـ C++.

• سطر الكود 2: using namespace std يعني أنه يمكننا استخدام أسماء للكائنات والمتغيرات من المكتبة القياسية.

• سطر الكود 3: يحدد تابع جديد يسمى main. وهو التابع الرئيسي للبرنامج. يتم الإشارة إلى التابع بواسطة الأقواس (). قبل التابع يوجد "int". هذا يعني أن التابع الرئيسي سيعيد عددًا صحيحًا إلى التابع أو العملية التي استدعتها.

• يحتوي القوسان المتعرجان curly braces { } على تعليمات داخل التابع. ينتهي البرنامج بنهاية الـ main المشار إليها بـ {

• يتم استدعاء main عند تنفيذ البرنامج. يجب أن تكون هناك main واحد فقط في برنامج C++، ويجب أن تُرجع دائمًا رقمًا من النوع int. هنا، int هو ما يسمى بنوع الإرجاع الخاص بالتابع. القيمة التي يتم إرجاعها بواسطة main هي رمز الخروج.

• سطر الكود 5: cout: هو كائن الإخراج القياسي، المحدد في iostream، ويطبوع إلى الإخراج القياسي cout. (تُنطق "see-out") هو كائن يُستخدم مع عامل الإدراج (<<) لإخراج/طباعة النص. في مثالنا سيتم إخراج "Hello World!".

• السلسلة الحرفية عبارة عن سلسلة من الأحرف محاطة بعلامات اقتباس مزدوجة.

• يقوم endl بإدراج حرف السطر الجديد على نفس السطر.

• يمكن حذف السطر std الخاص بـ namespace واستبداله بالكلمة الأساسية std، متبوعة بالعامل :: لبعض الكائنات: مثل:

std::cout << "Hello world!" << endl;

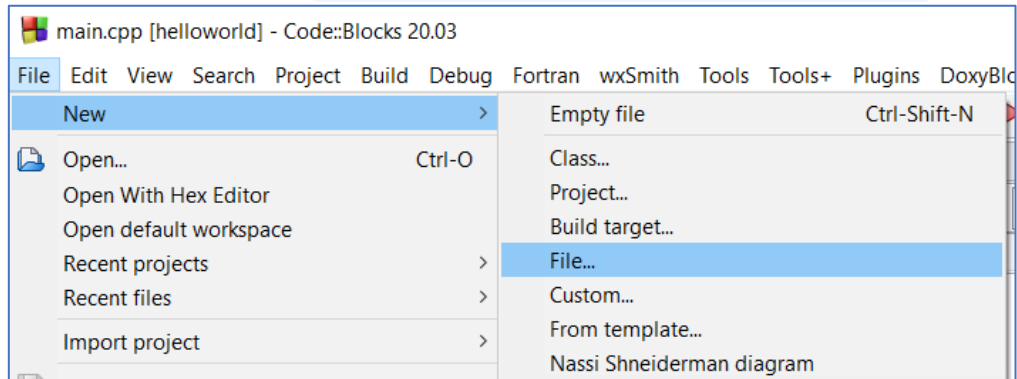
• سطر الكود 6: إرجاع 0: هذا هو الأمر الأخير في main، بيان الإرجاع. والغرض منه هو فقط إرجاع قيمة إلى الوظيفة أو العملية التي تستدعي main. يجب أن يُرجع صفرًا من main مما يعني أن البرنامج يعمل بنجاح ويخرج.

5-1 Changing file composition:

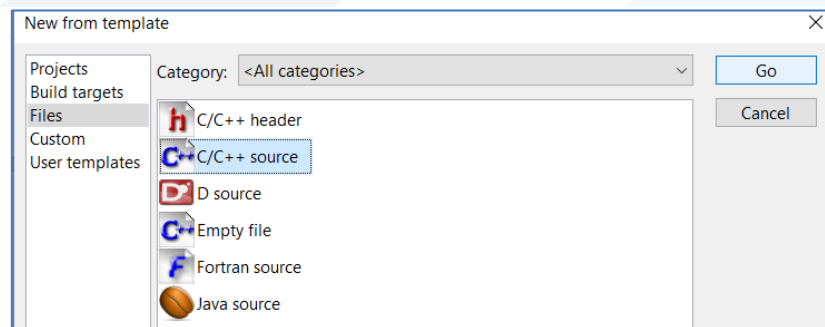
Code::Blocks has several very simple methods of adding additional files to the project.

Adding a c++ source file and c++ header file: In this example, we will be splitting the main.cpp file into two separate files c++ source file "hello.cpp" and c++ header file "hello.h".

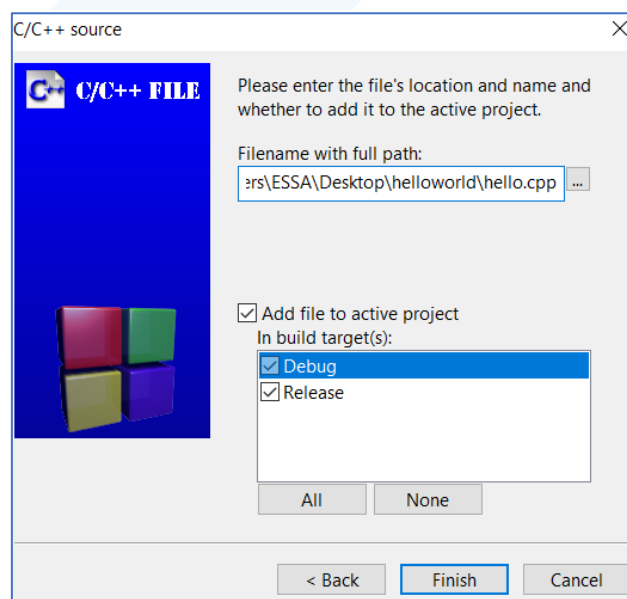
C++ source file "hello.cpp"



Then:



Select c++ then name file hello.cpp:

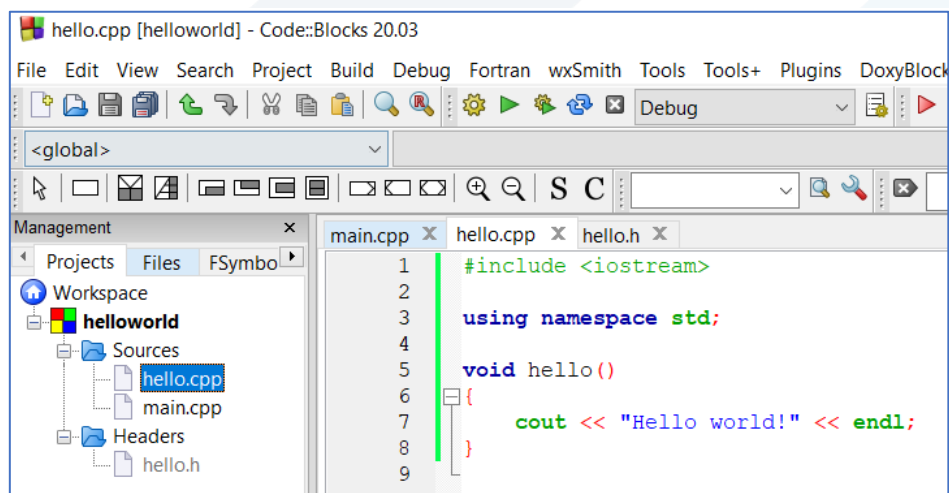


Add this code and save the file "hello.cpp":

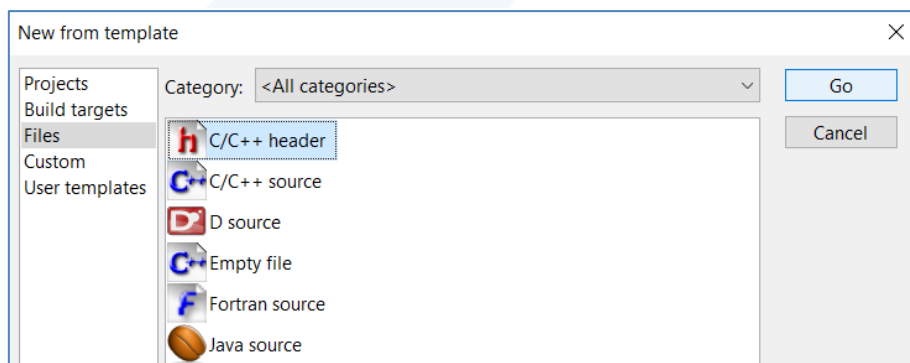
```
#include <iostream>

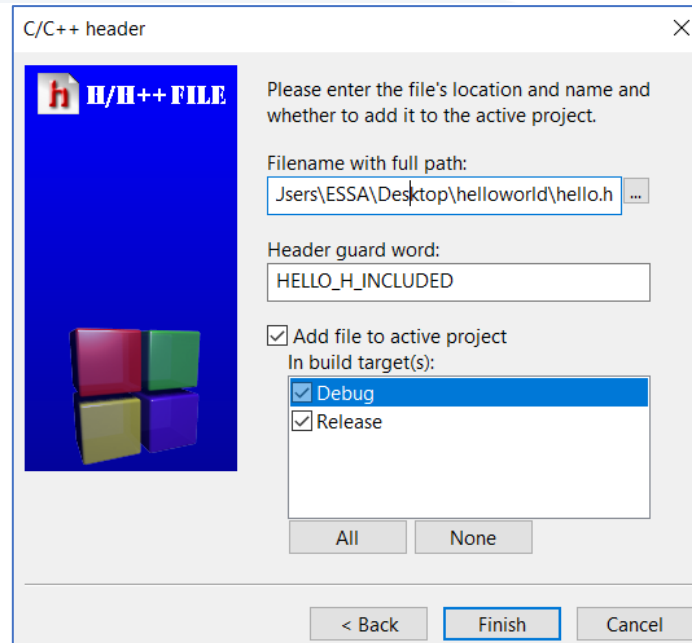
using namespace std;

void hello()
{
    cout << "Hello world!" << endl;
}
```



C++ header file "hello.h":



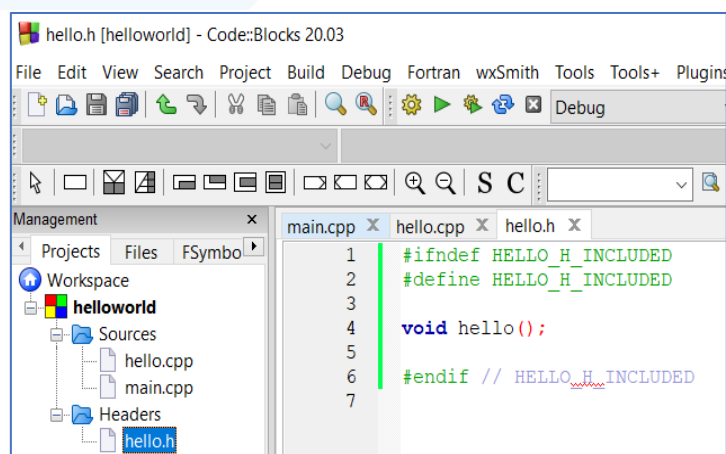


Add this code and save the file “hello.h”:

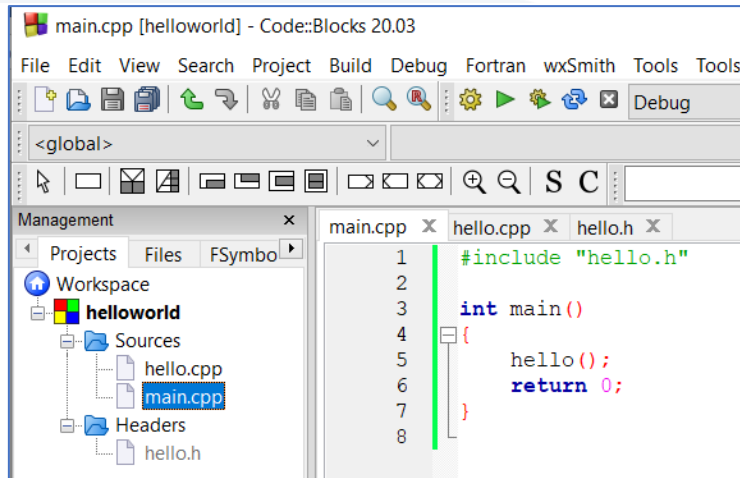
```
#ifndef HELLO_H_INCLUDED
#define HELLO_H_INCLUDED

void hello();

#endif // HELLO_H_INCLUDED
```



Modify main.cpp file:



Save project then build and run:



- Note: Single-line comments start with two forward slashes (//). Any text between // and the end of the line is ignored by the compiler (will not be executed).

```
cout << "Hello World!"; // This is a comment
```

- Note: Multi-line comments start with /* and ends with */. Any text between /* and */ will be ignored by the compiler:

```
/* The code below will print the words Hello
World! to the screen, and it is amazing */
cout << "Hello World!";
```

انتهت المحاضرة