

CodeBlocks setup & use

- Από την σελίδα : <https://www.codeblocks.org/downloads/> βλέπετε την κατάλληλη έκδοση για το λειτουργικό σας. Επιλέγοντας [Download the binary release](#) , υπάρχουν οι παρακάτω επιλογές για Microsoft Windows.

Microsoft Windows

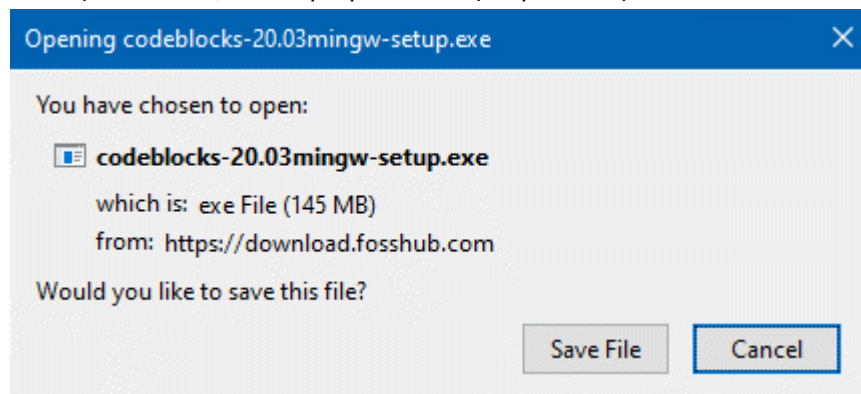
File	Download from
codeblocks-20.03-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-nosetup.zip	FossHUB or Sourceforge.net

NOTE: The codeblocks-20.03-setup.exe file includes Code::Blocks with all plugins. The codeblocks-20.03-setup-nonadmin.exe file is provided for convenience to users that do not have administrator rights on their machine(s).

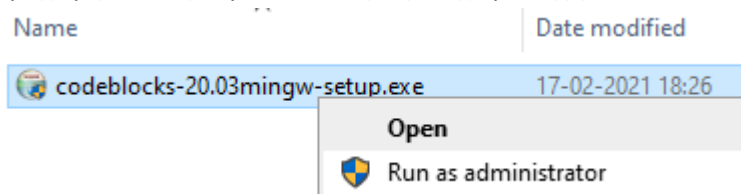
NOTE: The codeblocks-20.03mingw-setup.exe file includes additionally the GCC/G++/GFortran compiler and GDB debugger from [MinGW-W64 project](#) (version 8.1.0, 32/64 bit, SEH).

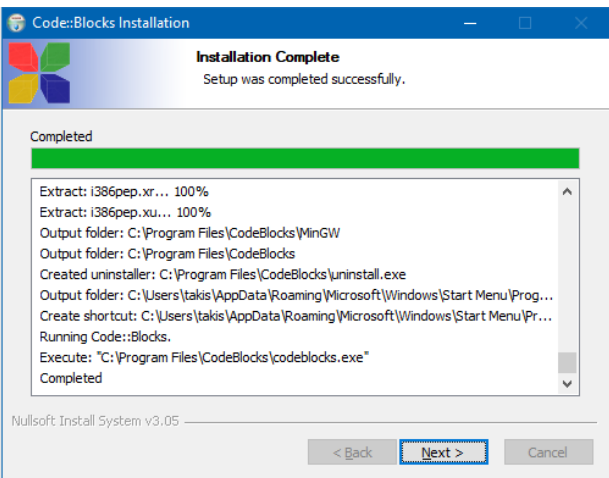
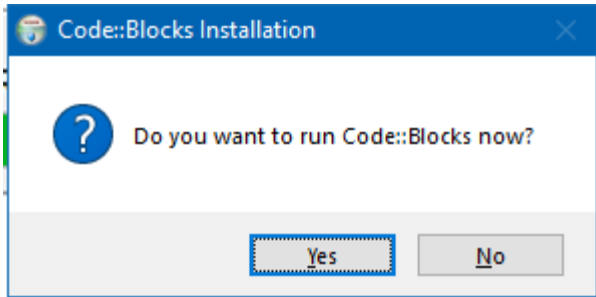
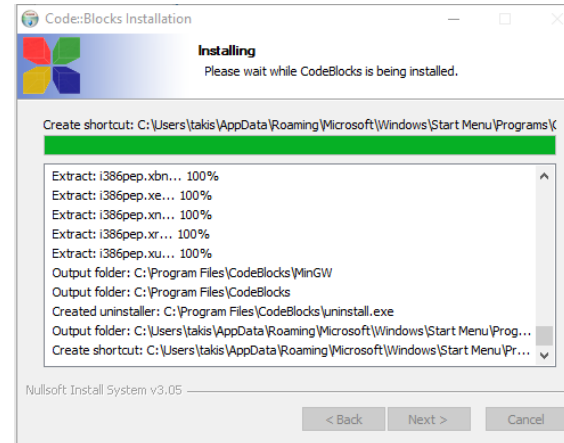
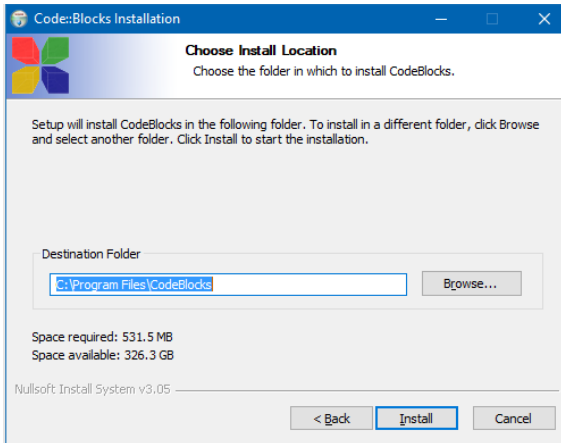
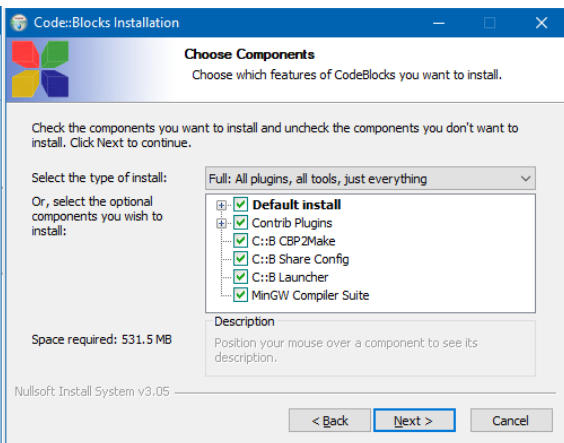
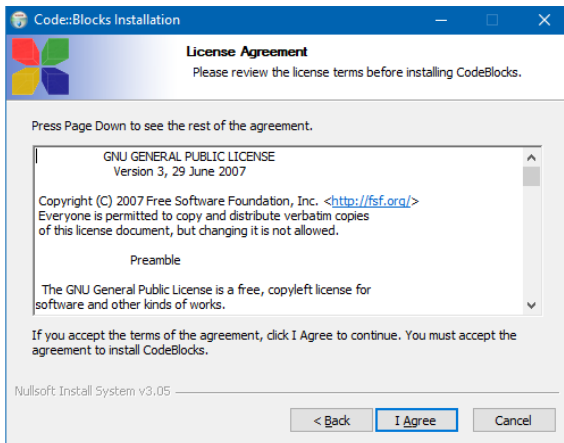
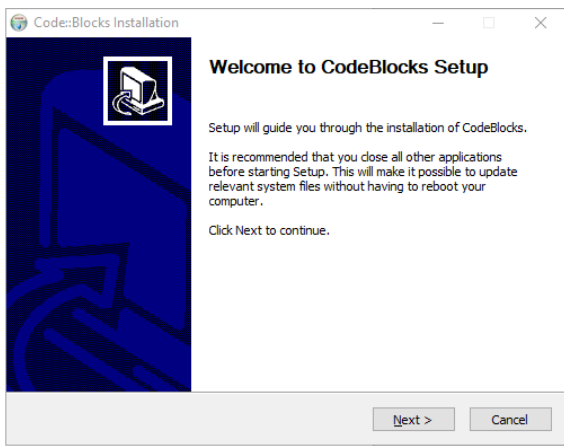
Επιλέξτε το αρχείο με όνομα : **codeblocks-20.03mingw-setup.exe** (γιατί περιλαμβάνει gcc compiler & gdb debugger).

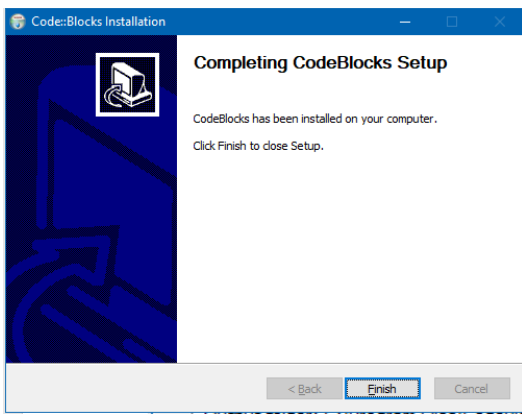
Επιλέξτε save file, στο Pop-up window (παρακάτω).



- Προχωρήστε στην εγκατάσταση ως διαχειριστής (Run as administrator)

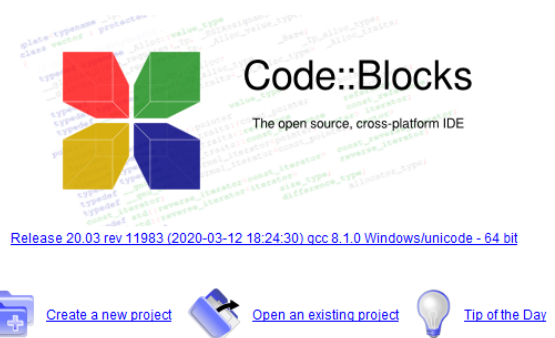
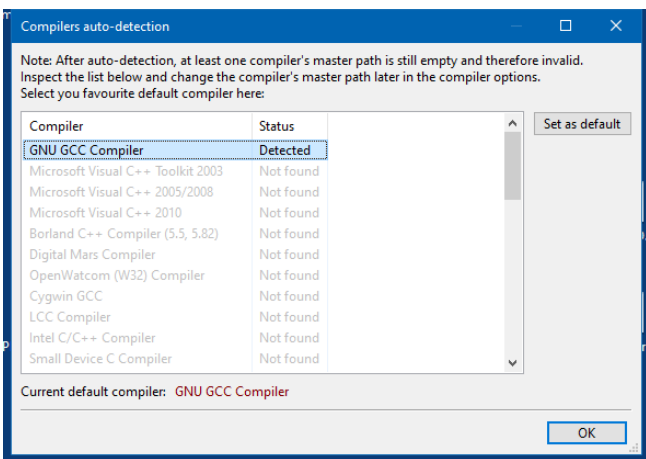






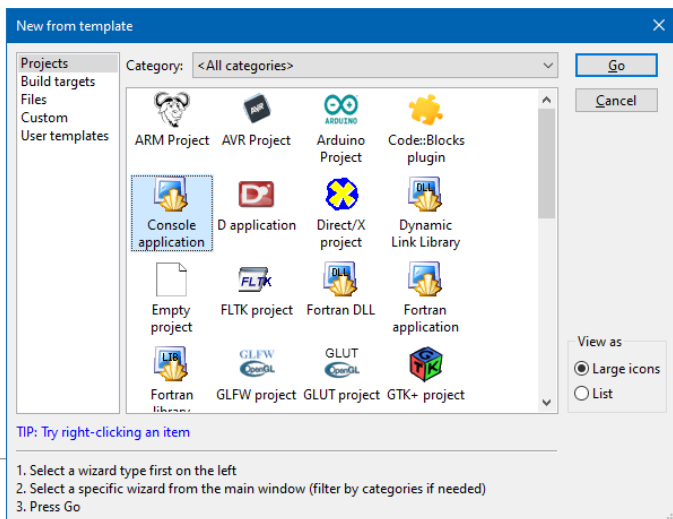
Δημιουργείται το αντίστοιχο εικονίδιο στην επιφάνεια εργασίας

Στην 1^η εκτέλεση της εφαρμογής επιλέγετε **GNU GCC Compiler (Set as default)**

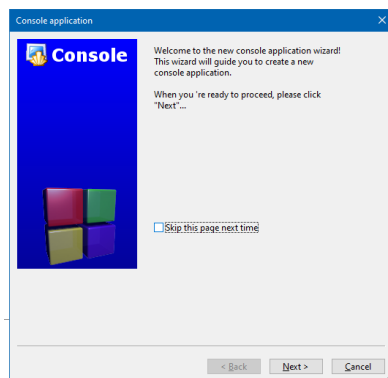


Για δημιουργία νέου project, έχετε 2 επιλογές :

1. Από το Μενού : **File -> New -> Project** ή
2. Από το εικονίδιο **Create a new project** και εμφανίζεται νέο παράθυρο.



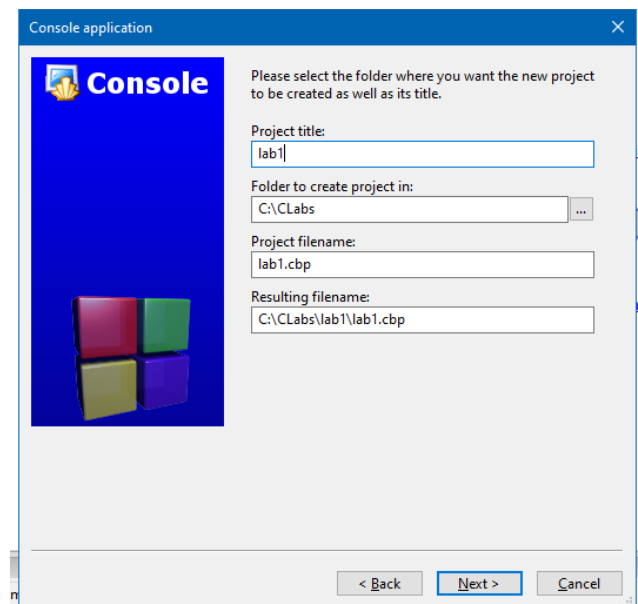
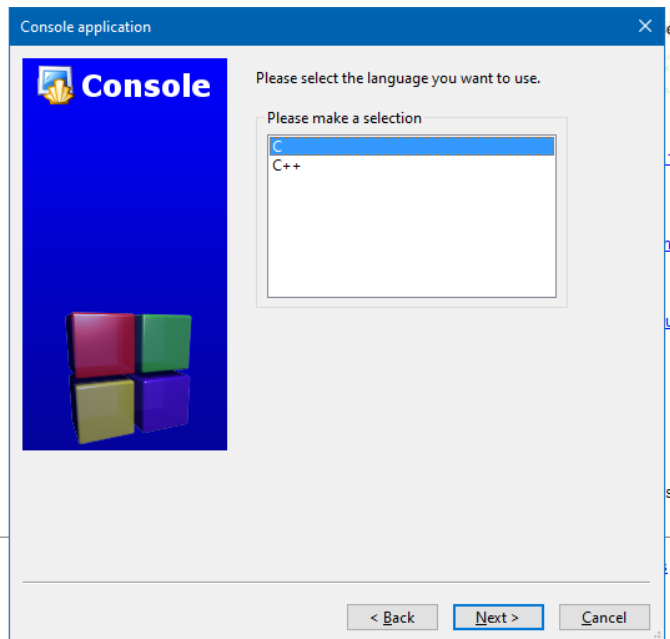
Επιλέγετε **Console application** , **Go**



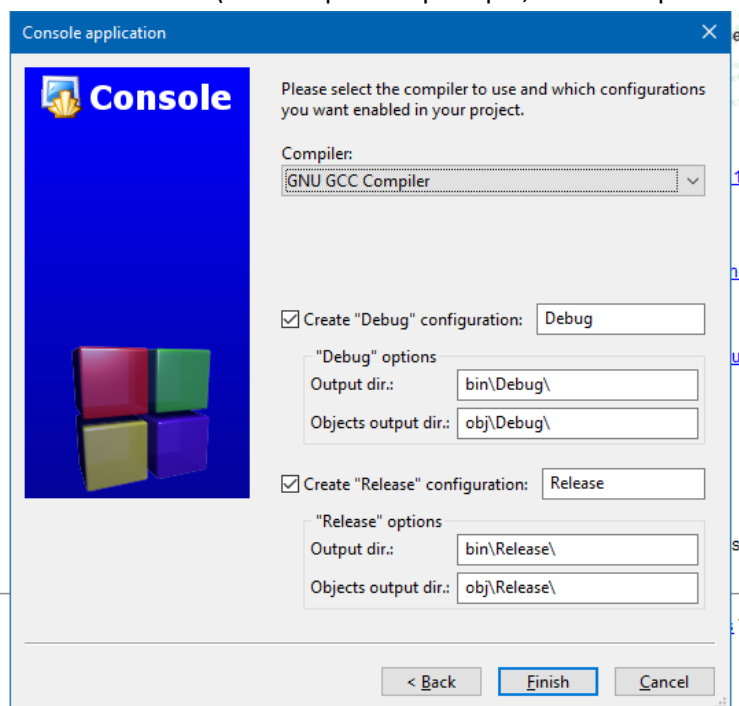
Πατάτε **Next** & επιλέγετε γλώσσα **C**.

Για μια φορά και πριν τη δημιουργία του πρώτου project, να δημιουργήσετε φάκελο π.χ. **C:\Clabs** που θα τον επιλέγετε πάντα κατά τη δημιουργία των project, ώστε αυτά να μπαίνουν στο συγκεκριμένο φάκελο.

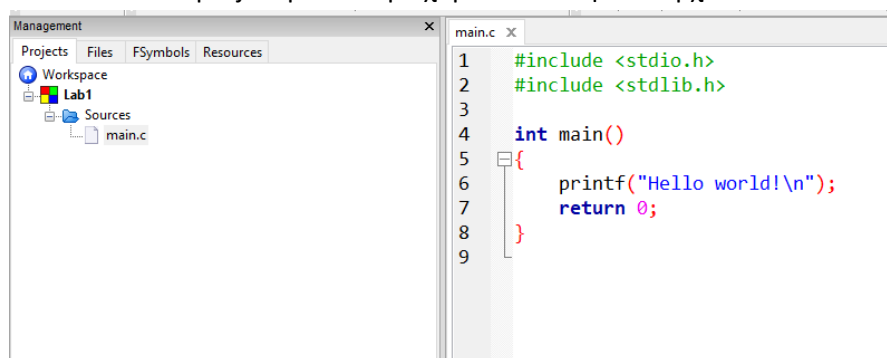
Πατάτε Next και δίνετε όνομα για το project, καθώς και το φάκελο (directory) που θα δημιουργηθεί / αποθηκευτεί το project.



και πατάτε **Finish** (στο επόμενο παράθυρο , που επιλέγει compiler & debugger).



Βλέπετε το νέο project με τα περιεχόμενά του δηλ. το αρχείο **main.c**





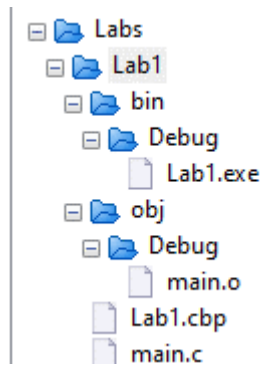
Με τα 3 εικονίδια έχετε τις επιλογές : **Build (Ctrl+F9)** , **Run (Ctrl+F10)**, **Build and run (F9)**. Έτσι βλέπετε το αποτέλεσμα της εκτέλεσης του προγράμματός σας.

```
main.c X
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      printf("Hello world!\n");
7      return 0;
8  }
9
```

D:\Labs\Lab1\bin\Debug\Lab1.exe

```
Hello world!
Process returned 0 (0x0)   execution time : 0.020 s
Press any key to continue.
```

Μετά από σωστό compilation (χωρίς errors) δημιουργείται το εκτελέσιμο αρχείο (.exe)



Ρυθμίσεις Compiler (από το μενού : Settings -> Compiler)

The screenshot shows the 'Global compiler settings' dialog box for the 'GNU GCC Compiler'. The 'Compiler Flags' tab is active, displaying a list of compiler options with checkboxes. The 'Enable warnings demanded by strict ISO C and ISO C++ [-pedantic]' option is highlighted in blue.

Global compiler settings

Selected compiler: GNU GCC Compiler

Buttons: Set as default, Copy, Rename, Delete, Reset defaults

Policy: [Dropdown]

Compiler Flags | Other compiler options | Other resource compiler options | #defines

Have gcc follow the 2017 ISO C language standard [-std=c17]	<input type="checkbox"/>
In C mode, this is equivalent to -std=c90, in C++ mode, it is equivalent to -std=c++98 [-ansi]	<input checked="" type="checkbox"/>
Position Independent Code [-fPIC]	<input type="checkbox"/>
Static libgcc [-static-libgcc]	<input type="checkbox"/>
Static libstdc++ [-static-libstdc++]	<input type="checkbox"/>
Static linking [-static]	<input type="checkbox"/>
Target x86 (32bit) [-m32]	<input type="checkbox"/>
Target x86_64 (64bit) [-m64]	<input type="checkbox"/>
Debugging	
Optimize debugging executable (compile speed, execution speed and better debugging) [-Og]	<input type="checkbox"/>
Produce debugging symbols [-g]	<input type="checkbox"/>
Profiling	
Profile code when executed [-pg]	<input type="checkbox"/>
Warnings	
Enable all common compiler warnings (overrides many other settings) [-Wall]	<input checked="" type="checkbox"/>
Enable Effective-C++ warnings (thanks Scott Meyers) [-Weffc++]	<input type="checkbox"/>
Enable extra compiler warnings [-Wextra]	<input checked="" type="checkbox"/>
Enable warnings demanded by strict ISO C and ISO C++ [-pedantic]	<input checked="" type="checkbox"/>
Inhibit all warning messages [-w]	<input type="checkbox"/>
Stop compiling after first error [-Wfatal-errors]	<input checked="" type="checkbox"/>
Treat as errors the warnings demanded by strict ISO C and ISO C++ [-pedantic-errors]	<input checked="" type="checkbox"/>
Warn about uninitialized variables which are initialized with themselves [-Winit-self]	<input type="checkbox"/>
Warn if '0' is used as a null pointer constant [-Wzero-as-null-pointer-constant]	<input type="checkbox"/>
Warn if a class has virtual functions but no virtual destructor [-Wnon-virtual-dtor]	<input type="checkbox"/>
Warn if a function can not be inlined and it was declared as inline [-Winline]	<input type="checkbox"/>

NOTE: Right-click to setup or edit compiler flags.

Buttons: OK, Cancel

The screenshot shows the 'Global compiler settings' dialog box for the 'GNU GCC Compiler'. The 'Toolchain executables' tab is active, displaying fields for the compiler's installation directory and various toolchain executables.

Global compiler settings

Selected compiler: GNU GCC Compiler

Buttons: Set as default, Copy, Rename, Delete, Reset defaults

Compiler settings | Linker settings | Search directories | Toolchain executables | Custom variables | Build options | Other settings

Compiler's installation directory: C:\MinGW

NOTE: All programs must exist either in the "bin" sub-directory of this path, or in any of the "Additional paths"...

Program Files | Additional Paths

C compiler:	gcc.exe
C++ compiler:	g++.exe
Linker for dynamic libs:	g++.exe
Linker for static libs:	ar.exe
Debugger:	GDB/CDB debugger: Default
Resource compiler:	windres.exe
Make program:	mingw32-make.exe

Buttons: OK, Cancel

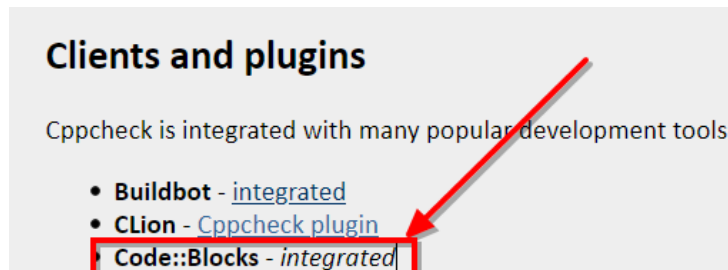
----- Build: Debug in Lab1 (compiler: GNU GCC Compiler)-----

```
gcc.exe -Wall -pedantic-errors -pedantic -Wfatal-errors -Wextra -Wall -ansi -g -pedantic-errors -pedantic -Wfatal-errors -Wextra -Wall -ansi -IC:\MinGW\bin -c D:\Labs\Lab1\main.c -o obj\Debug\main.o
```

```
[100.0%] gcc.exe -o bin\Debug\Lab1.exe obj\Debug\main.o
```

CppCheck plugin settings

Το επίσημο [site](#) για το CppCheck αναφέρει ότι είναι ενσωματωμένο στο Code::Blocks.



Επομένως έχετε έτοιμο το περιβάλλον.

Αν όμως υπάρξει πρόβλημα, από την ιστοσελίδα : <http://cppcheck.sourceforge.net/> κατεβάζετε την έκδοση για το λειτουργικό σας.

Download

Cppcheck 2.3

Platform	File
Windows 64-bit (No XP support)	Installer
Source code (.zip)	Archive
Source code (.tar.gz)	Archive

Packages

Cppcheck can also be installed from various package managers; however, you might get an outdated version then.

Debian:

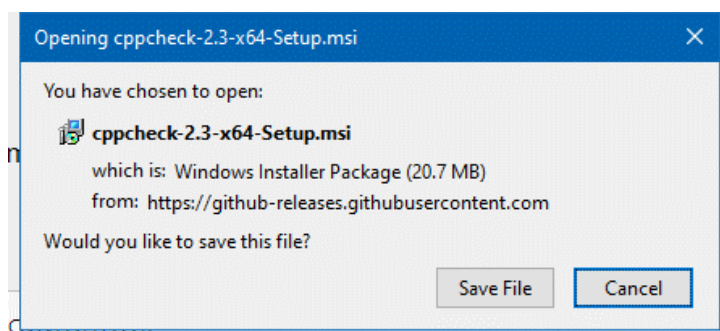
```
sudo apt-get install cppcheck
```

Fedora:

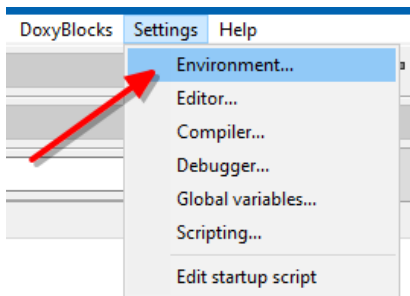
```
sudo yum install cppcheck
```

Mac:

```
brew install cppcheck
```



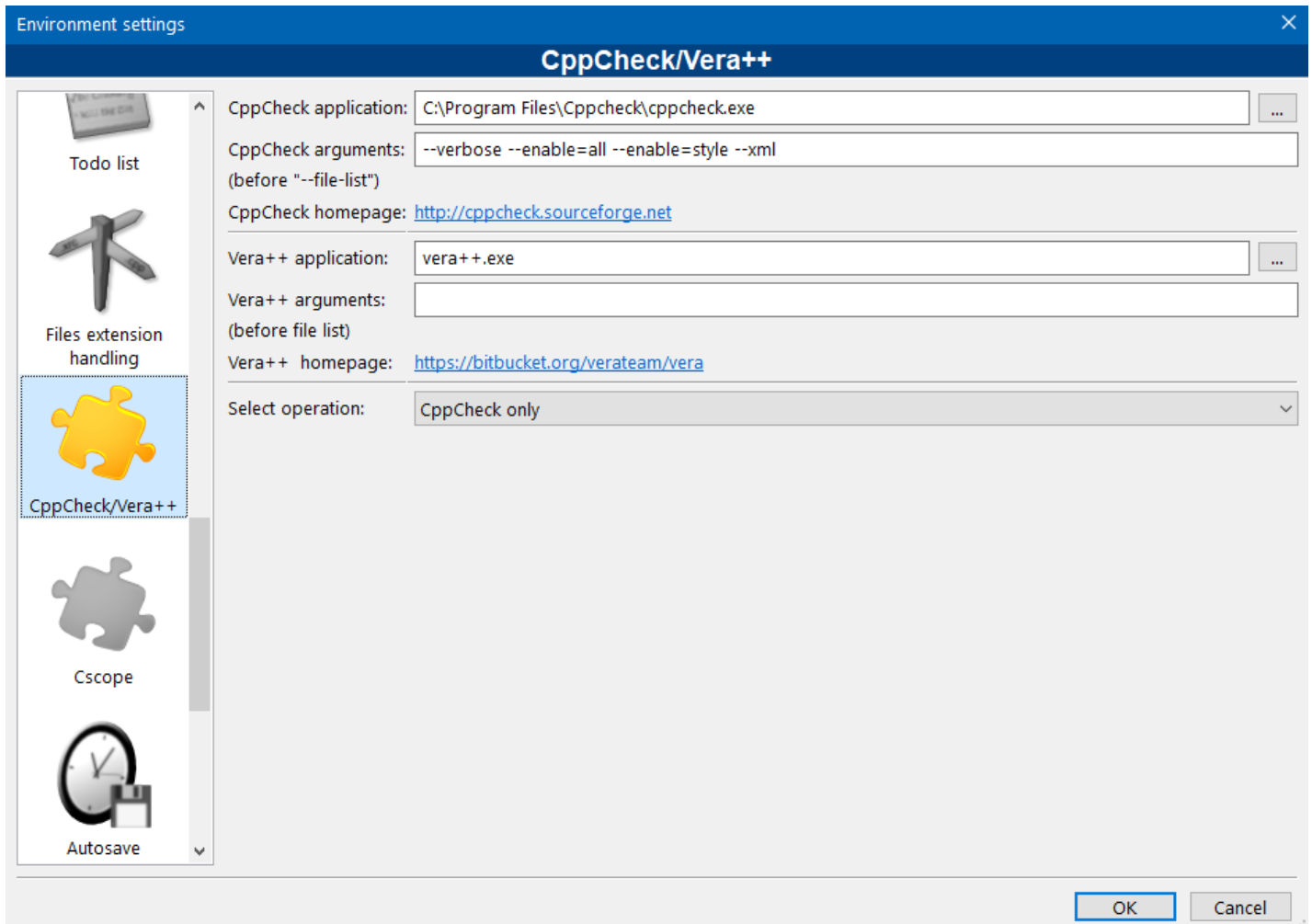
Για τα Windows είναι το αρχείο **cppcheck-2.3-x64-Setup.msi**



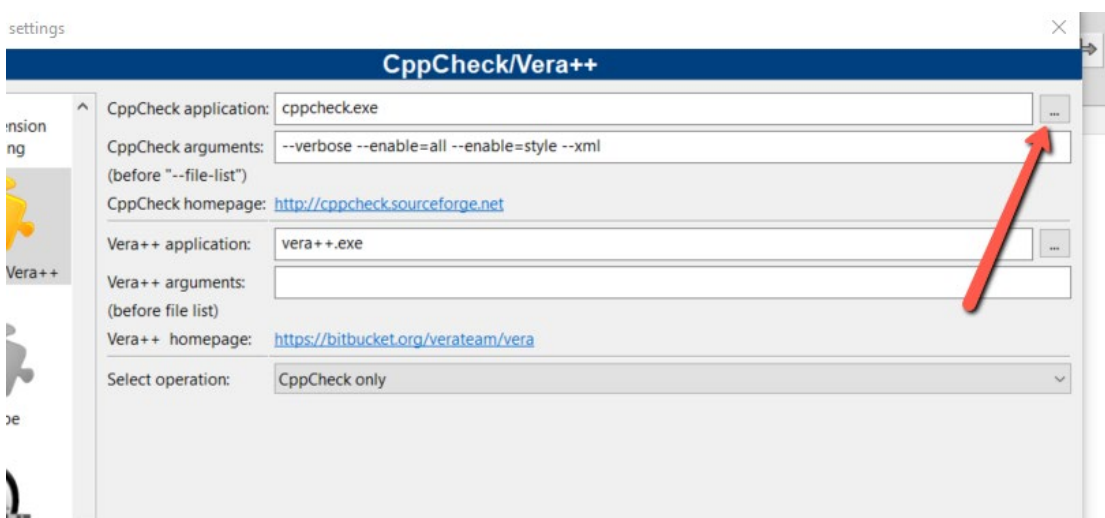
Το εγκαθιστάτε και μετά μπορείτε να κάνετε τις ρυθμίσεις στο Code::Blocks.
Επιλέγετε : **Settings -> Environment**, και κάνετε scroll down μέχρι να δείτε την επιλογή **CppChek/Vera++**

Προσέχετε να είναι σωστή η ρύθμιση του path στο 1^ο πλαίσιο δηλ. **CppCheck application**


Τα σωστά arguments είναι : **--verbose --enable=all --enable=style --xml**



Αν δεν βλέπει σωστά το path τότε κάνετε κλικ στο εικονίδιο με τις 3 τελείες και επιλέγετε το σωστό path για το cppcheck.exe.



CppCheck/Vera++

CppCheck application: 

CppCheck arguments: (before "--file-list")

CppCheck homepage: <http://cppcheck.sourceforge.net>

Vera++ application:

Vera++ arguments:

Vera++ homepage: <https://bitbucket.org/verateam/vera>

Select CppCheck application

← → ↑

Organize ▾ New folder

Name	Date modified	Type	Size
addons	24/2/2021 11:52	File folder	
cfg	24/2/2021 11:52	File folder	
lang	24/2/2021 11:52	File folder	
platforms	24/2/2021 11:52	File folder	
sqldrivers	24/2/2021 11:52	File folder	
styles	24/2/2021 11:52	File folder	
cppcheck.exe	5/12/2020 9:33	Application	136 KB
cppcheckgui.exe	5/12/2020 9:30	Application	4.206 KB

File name: Executable files (*.exe)

