The default GCC toolchain that comes with CodeBlocks for windows doesn't support OpenMP, so we have to change the toolchain with one that supports OpenMP.

The following steps will describe how to configure CodeBlocks with the OpenMP enabled MinGW toolchain.

- Download the <u>TDM-GCC</u> zip and extract it.
- keep the location of extracted files because we will need that shortly.
- Start CodeBlocks, and go to "compiler..." options under settings



Settings -> Compiler...

A window like the following will appear

	GNU GCC Compiler	
203	Set as default Copy Rename Delete Reset de	fault
lobal compiler settings	Compiler settings Linker settings Search directories Toolchain executables Custom variables Build options	1
Name of Street	Policy:	
	Compiler Flags Other compiler options Other resource compiler options #defines	
	General	^
Desflorentingen	Have g++ follow the 1998 ISO C++ language standard [-std=c++98]	
Prohier setungs	Have g++ follow the C++11 ISO C++ language standard [-std=c++11] ▼	
	Have g++ follow the C++14 ISO C++ language standard [-std=c++14]	
0.0	Have g++ follow the coming C++0x (aka c++11) ISO C++ language stan	
5000	Have g++ follow the coming C++1y (aka C++14) ISO C++ language star	
	Have g++ follow the coming C++1z (aka C++17) ISO C++ language star	
	Have gcc follow the 1990 ISO C language standard (certain GNU extensio	
Batch builds	Have gcc follow the 1999 ISO C language standard [-std=c99]	
butter builds	Have gcc follow the 2011 ISO C language standard [-std=c11]	
	In C mode, this is equivalent to -std=c90, in C++ mode, it is equivalent to	
	Position Independent Code [-fPIC]	
	Static libgcc [-static-libgcc]	
	Static libetic ++ [-static-libetic ++]	~

• Under "Compiler settings", select "Other Compiler Options". On the textbox bellow add "-fopenmp" flag

Compiler settings	— 🗆 X										
Global compiler settings											
	Selected compiler GNU GCC Compiler Set as default Copy Rename Delete Reset defaults										
Global compiler settings	Compiler settings Vinker settings Search directories Toolchain executables Custom variables Build options										
Profiler settings	Policy:										
Batch builds											

• Select "Linker settings" tab, add "-lgomp" flag on the "other linker options" box

Compiler settings		— 🗆 X											
Global compiler settings													
	Selected compiler												
	GNU GCC Compiler	~											
	Set as default Copy Rename De	Reset defaults											
Global compiler settings	Compiler setting Linker settings Sarch directories Toolchain executables Custom	variables Build options											
1	Policy:	~											
Profiler settings	Link libraries: Other linker options:												

- Finally select "toolchain executable" tab. Change the red circled fields as described:
  - Compiler's installation directory: "<path to TDM-GCC-64>"
  - C compiler: gcc.exe
  - C++ compiler: g++.exe

Linker for dynamic libs: g++.exe

Selected compiler				
GNU GCC Compiler				
	Set as default	Copy Rename	Delete	Reset defaul
Compiler settings Linker	settings Search directories	Toolchain executables	Custom variables	Build options
Compiler's installation d	irectory			
D:\IPDC\TDM-GCC-64				Auto-detect
NOTE: Air programs mu	ust exist either in the "bin" sub	-directory of this path, o	r in any of the "Ad	ditional paths"
Program Files Addition	nal 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 : Defau	ult		~
Resource compiler:	windres.exe			
Make program:	mingw32-make.exe			
	GNU GCC Compiler  Compiler settings Linker  Compiler's installation d  D:\IPDC\TDM-GCC-64  NOTE: All programs m  Program Files Addition C compiler: C++ compiler: Linker for dynamic libs: Debugger: Resource compiler: Make program:	GNU GCC Compiler         Set as default         Compiler's installation directory         D:\IPDC\TDM-GCC-64       NOTE: Air programs must exist either in the "bin" sub         Program Files       Additional Paths         C compiler:       gcc.exe         C++ compiler:       Ig++.exe         Linker for dynamic libs:       gr.exe         Debugger:       GDB/CDB debugger : Defau         Resource compiler:       windres.exe         Make program:       mingw32-make.exe	GNU GCC Compiler       Set as default       Copy       Rename         Compiler settings       Linker settings       Search directories       Toolchain executables         Compiler's installation directory       D:\IPDC\TDM-GCC-64       Toolchain executables         NOTE: Air programs must exist either in the "bin" sub-directory of this path, or       Program Files       Additional Paths         C compiler:       gcc.exe       C       C++ compiler:       Ig++.exe         Linker for dynamic libs:       gr.exe       Debugger:       GDB/CDB debugger : Default         Resource compiler:       windres.exe       Make program:       mingw32-make.exe	GNU GCC Compiler       Set as default       Copy       Rename       Delete         Compiler settings       Linker settings       Search directories       Toolchain executables       Custom variables         Compiler's installation directory       D:\IPDC\TDM-GCC-64       Toolchain executables       Custom variables         Program Files       Additional Paths       C       Compiler:       gcc.exe         C ++ compiler:       gc+.exe       Linker for dynamic libs:       gr+.exe         Linker for static libs:       ar.exe       Debugger:       GDB/CDB debugger : Default         Resource compiler:       windres.exe       Make program:       mingw32-make.exe

Press "OK" and OpenMP should work with CodeBlocks now.

Note: in case there are existing project, after configuring OpenMP with CodeBlocks, clean and rebuild the existing project.

Test:

- Download <u>openmp\_confirm.cpp</u>
- Open it in CodeBlocks
- Build and run (F9 is a shortcut for this)
- You should see:
  - hi from thread 0
  - hi from thread 1