

Temat:

Od algorytmu do programu
cz.2

Zapis algorytmu w języku programowania nazywamy **kodem źródłowym**.

Kod źródłowy tworzą dwie konstrukcje programistyczne: **instrukcje i funkcje**.

Rodzaje translatorów:

Kompitatory (*ang. compilers*)

tłumaczą od razu cały kod →

przykład **język C++**

Rodzaje translatorów:

Interpretery (ang. interpreters)
tłumaczą instrukcja po instrukcji
→ przykład **język Python**

Instrukcja to polecenie, które kompilator lub interpreter potrafi zrozumieć i wykonać

Przykłady instrukcji:

Wczytaj

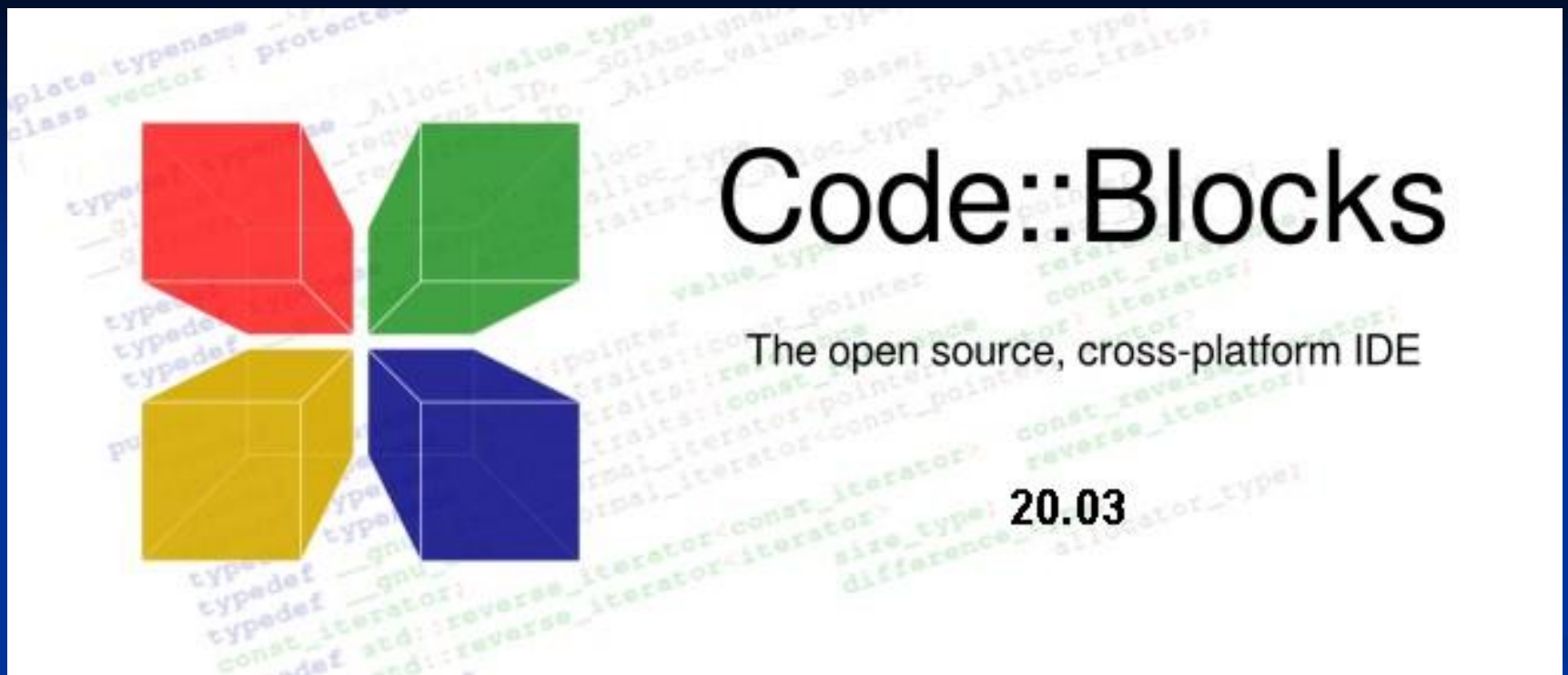
Wyświetl

Zatrzymaj się

inne..

Pojedyncze instrukcje składa się w większą całość i nazywamy **funkcją**.

Słowa kluczowe → lista
zastrzeżonych słów, które są użyte
w konkretnym celu (określone
przez twórcę języka)



Code::Blocks

Środowisko programistyczne (IDE)
umożliwiające tworzenie aplikacji w języku
C i C++ na licencji GNU.



Code::Blocks

The open source, cross-platform IDE

20.03

Code::Blocks

Wybierz właściwą wersję do pobrania

← → ↺ fosshub.com/Code-Blocks.html?dwl=codeblocks-20.03-setup.exe

Aplikacje Google Wirtualna Polska -... Zespół Szkół Techni... Dziennik UONET+ eBOK PGNiG - Załat... PGE eBOK - elektro... Facebook - zaloguj... Pekao24 - Logowanie

BAZA DANYCH

NARZĘDZIA

DEWELOPERSKIE

ANALIZATORY DYSKÓW

APLIKACJE DO E-BOOKÓW

KLIENCI POCZTY E-MAIL

SZYFROWANIE

KLIENCI FTP

POKAŻ WIĘCEJ

ŚCIGAŃ	PLIK	ROZMIAR	WERSJA	ANTYVIRUS
Code Blocks Windows 64-bitowy (w tym kompilator)	Podpis	145,4 MB	20.03	0 / 0
Code Blocks Windows 64-bitowy	Podpis	35,7 MB	20.03	0 / 0
Bloki kodu Windows 64-bitowy (bez administratora)	Podpis	35,7 MB	20.03	0 / 0
Bloki kodu Windows 64-bitowy przenośny ZIP (w tym kompilator)	Podpis	164,9 MB	20.03	0 / 0
Code Blocks Windows 64-bitowy przenośny ZIP	Podpis	35,5 MB	20.03	0 / 0
Code Blocks Windows 32-bitowy (w tym 32-bitowy kompilator)	Podpis	160 MB	20.03	0 / 0
Code Blocks Windows 32-bitowy (w tym kompilator 64-bitowy)	Podpis	164,5 MB	20.03	0 / 0
Code Blocks Windows 32-bitowy	Podpis	54,4 MB	20.03	0 / 0
Code Blocks Windows 32-bitowy (inny niż administrator)	Podpis	54,4 MB	20.03	0 / 0
Bloki kodu Windows 32-bitowy przenośny ZIP (w tym kompilator 32-bitowy)	Podpis	178,3 MB	20.03	0 / 0
Bloki kodu Windows 32-bitowy przenośny plik ZIP (w tym kompilator 64-bitowy)	Podpis	183,3 MB	20.03	0 / 0
Bloki kodu Windows 32-bitowe przenośne pliki ZIP	Podpis	54 MB	20.03	0 / 0

codeblocks-20.03....exe
70,2/145 MB, Pozostały 2 min

Pokaż wszystkie

LINK DO POBRANIA ZE STRONY PRODUCENTA - TUTAJ

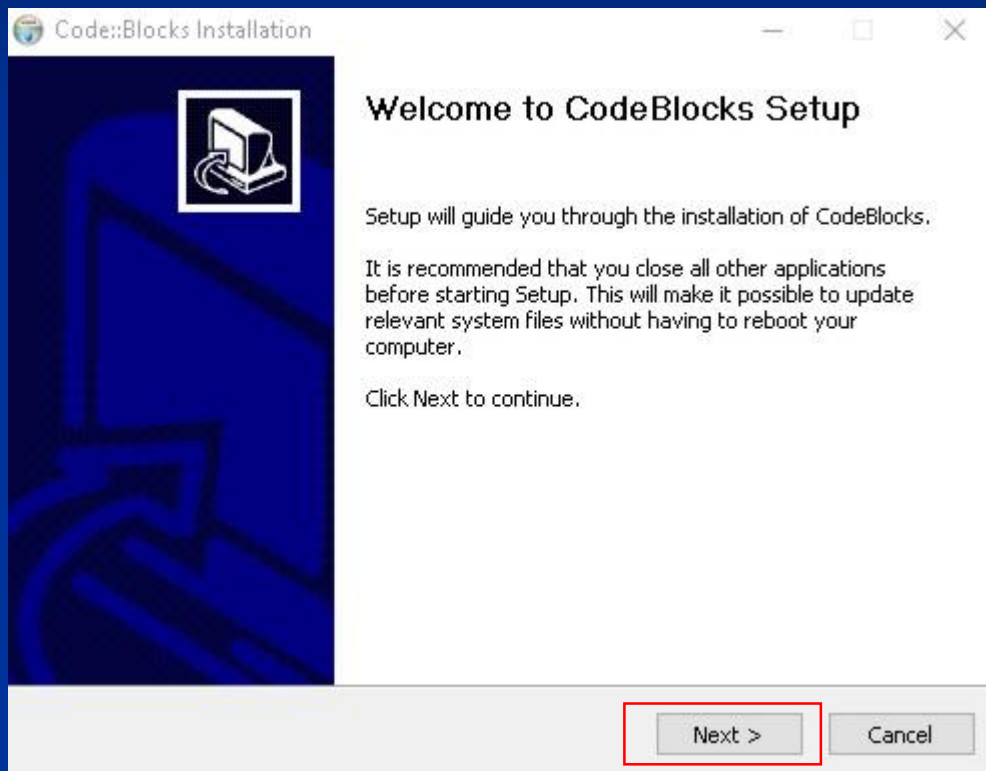


Code::Blocks

The open source, cross-platform IDE

20.03

Code::Blocks - INSTALACJA





Code::Blocks

The open source, cross-platform IDE

20.03

Code::Blocks - INSTALACJA



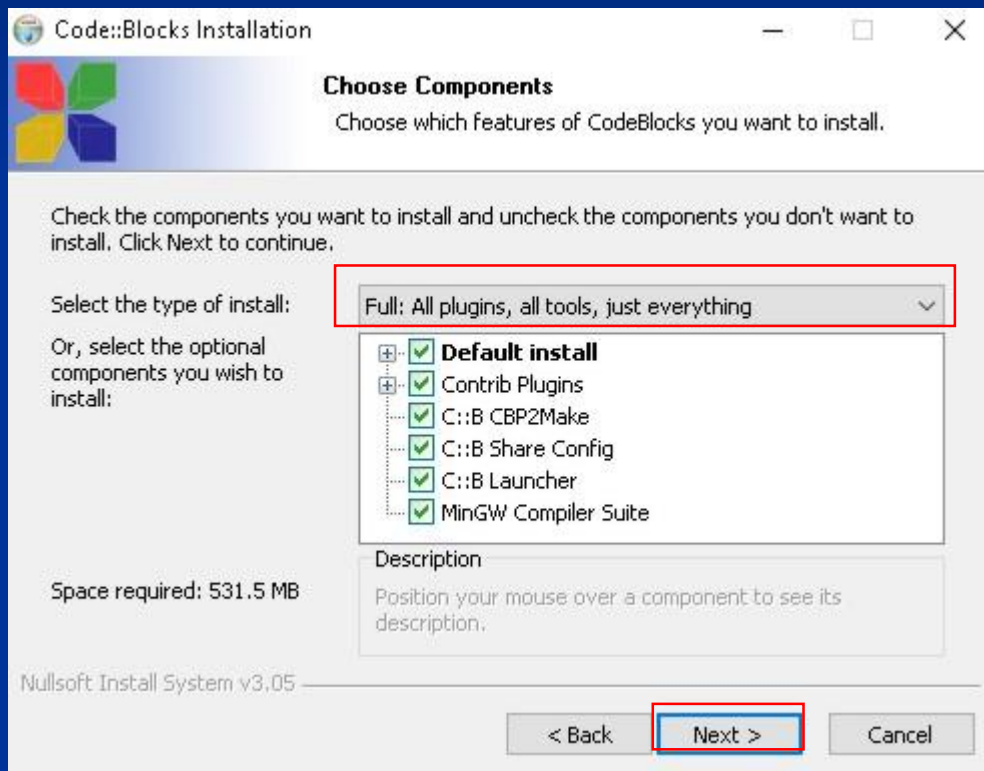


Code::Blocks

The open source, cross-platform IDE

20.03

Code::Blocks - INSTALACJA



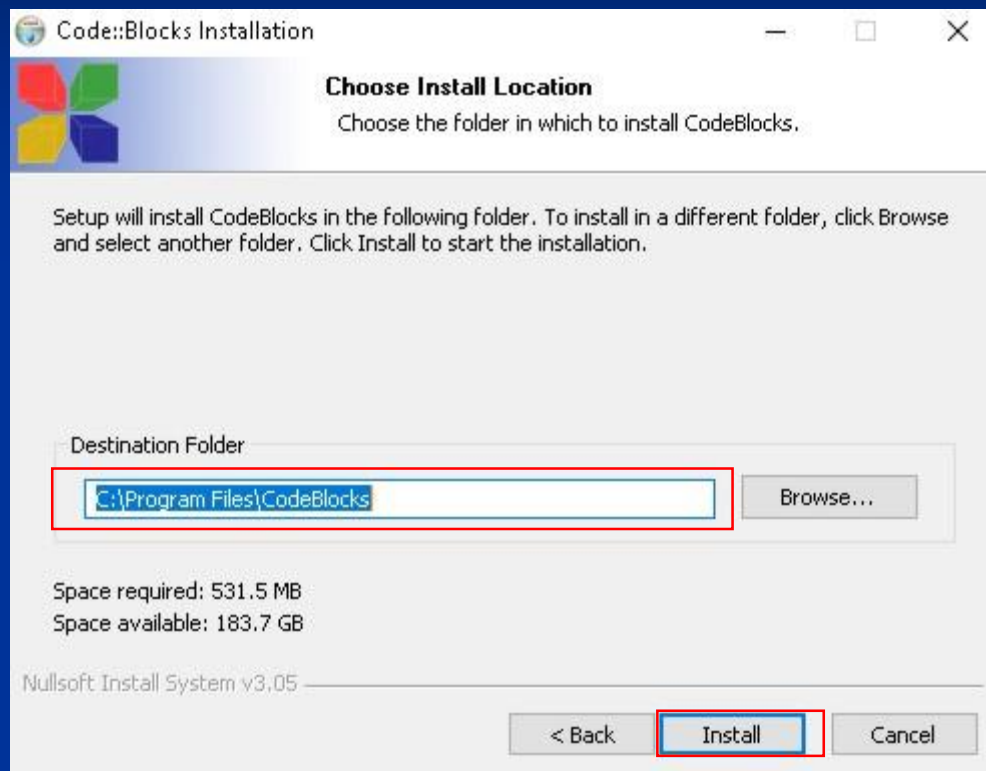


Code::Blocks

The open source, cross-platform IDE

20.03

Code::Blocks - INSTALACJA



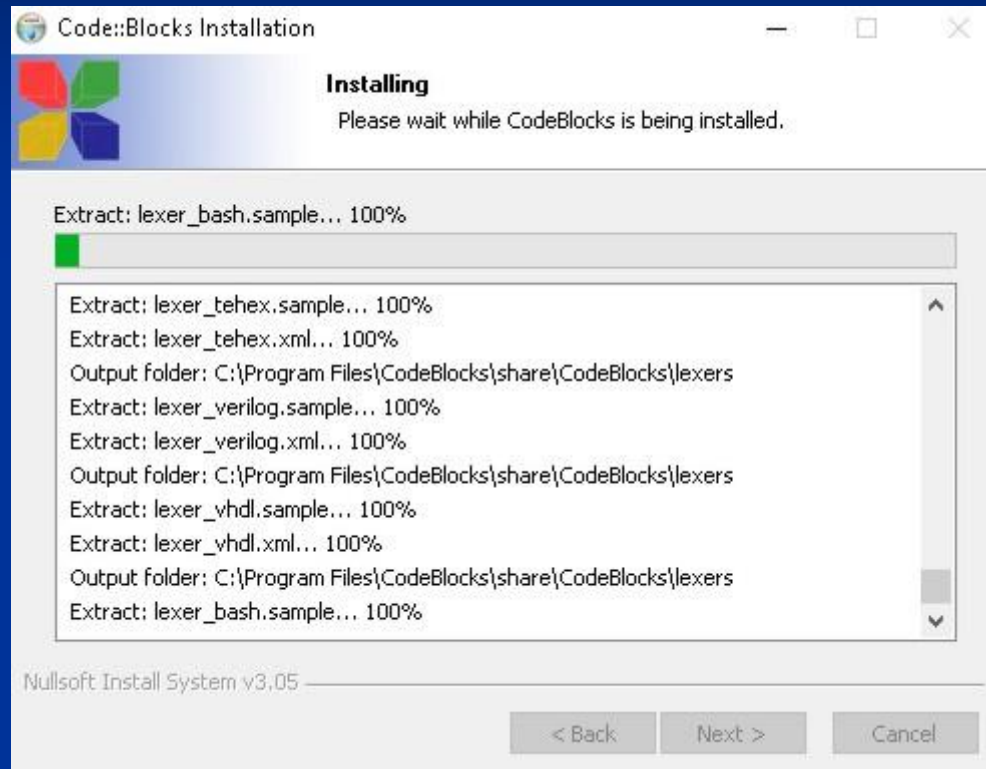


Code::Blocks

The open source, cross-platform IDE

20.03

Code::Blocks - INSTALACJA



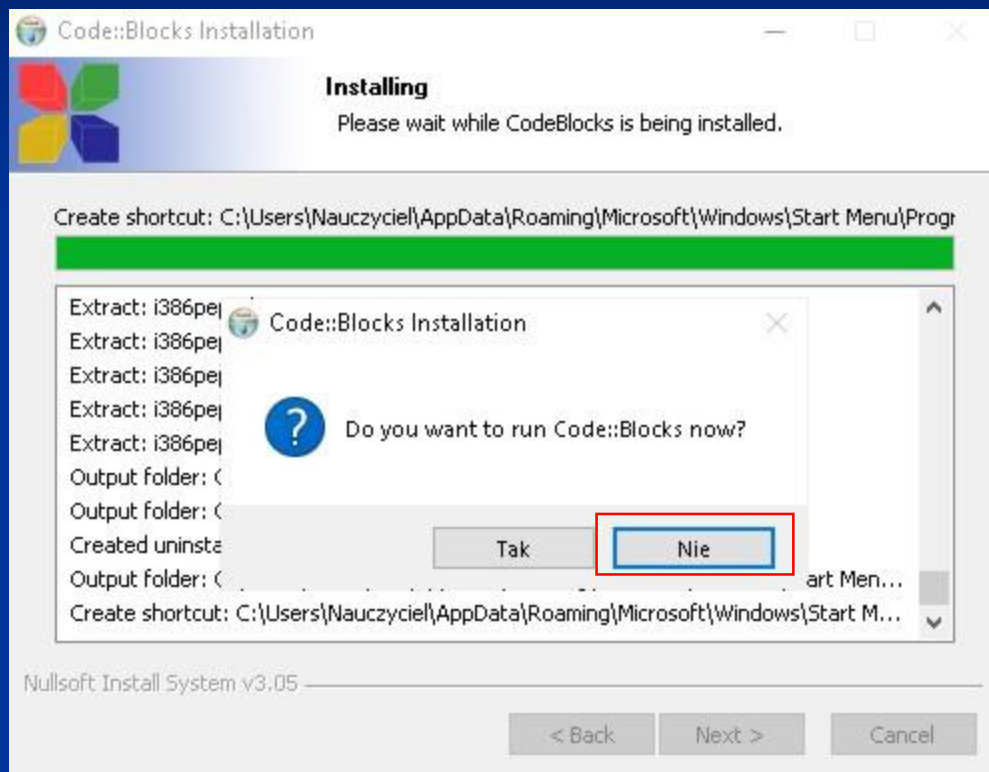


Code::Blocks

The open source, cross-platform IDE

20.03

Code::Blocks - INSTALACJA



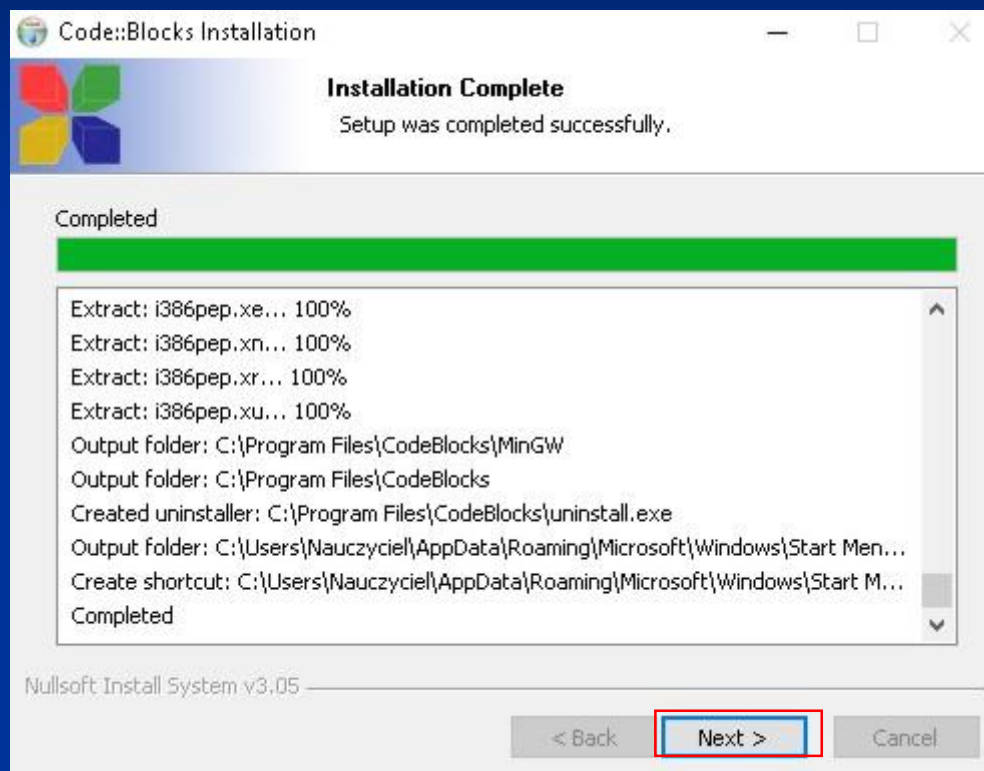


Code::Blocks

The open source, cross-platform IDE

20.03

Code::Blocks - INSTALACJA



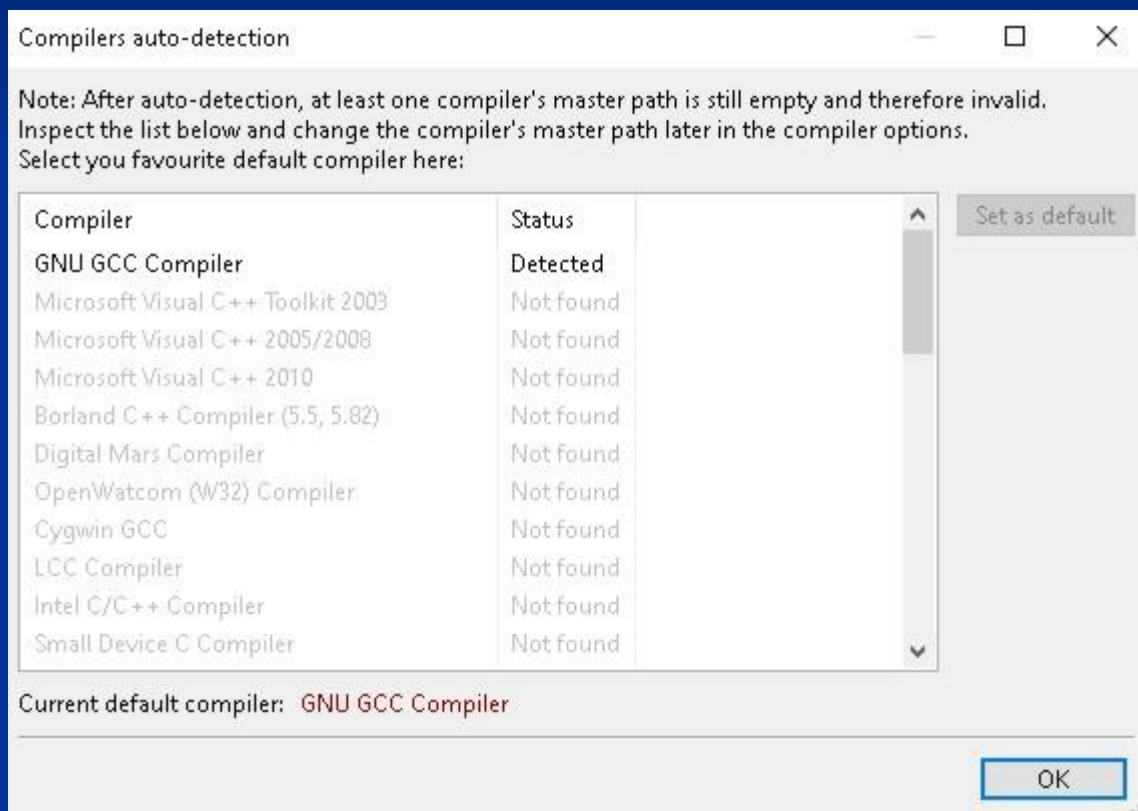


Code::Blocks

The open source, cross-platform IDE

20.03

Code::Blocks - INSTALACJA



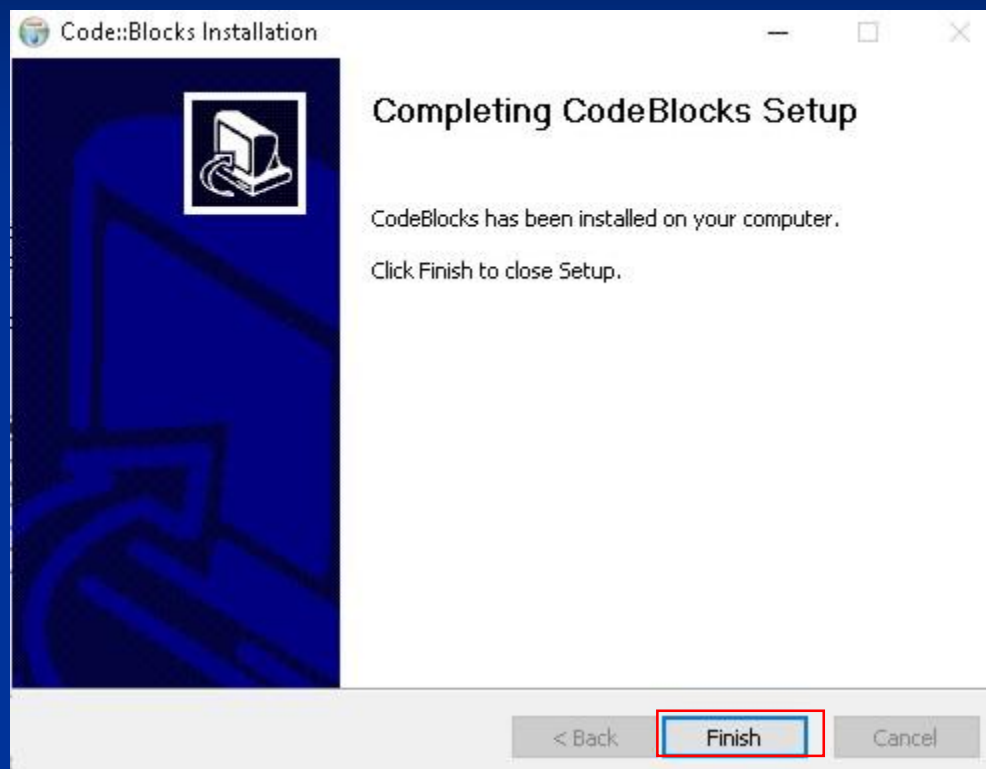


Code::Blocks

The open source, cross-platform IDE

20.03

Code::Blocks - INSTALACJA



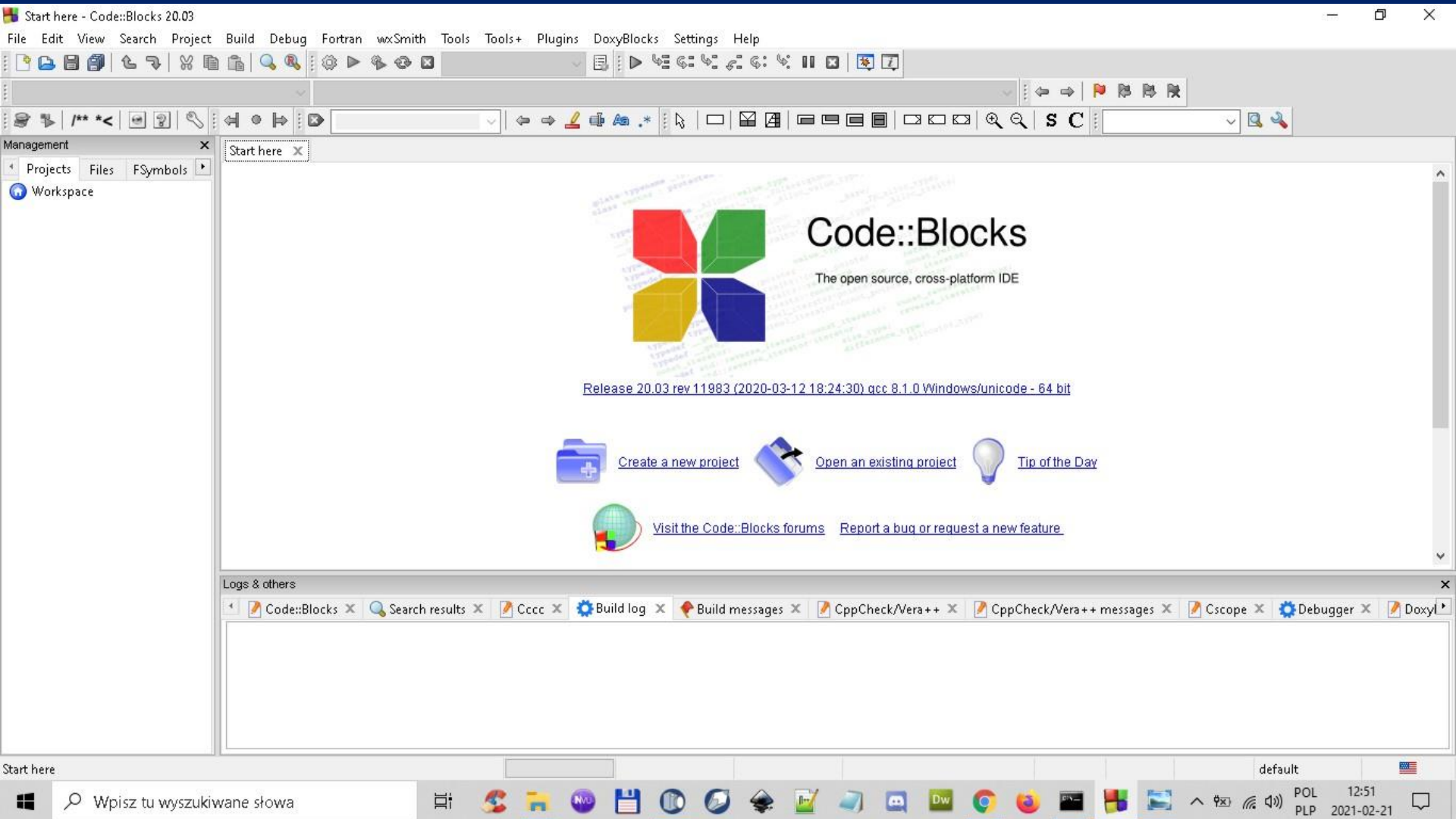


Code::Blocks

The open source, cross-platform IDE

20.03

Code::Blocks po uruchomieniu



Przykładowy kod źródłowy programu w języku C++

To polecenie dołącza bibliotekę z obsługą standardowego wejścia i wyjścia. Dzięki niemu program będzie mógł m.in. wczytywać dane z klawiatury i wypisywać wyniki na ekranie monitora.

```
1. #include <iostream>
2.
3. using namespace std;
4.
5. int main() //nagłówek funkcji głównej
6. {
7.     cout << "Hello world!" << endl;
8.     return 0;
9. }
```

Ten zapis pozwala programowi wykorzystać tzw. standardową przestrzeń nazw. Dzięki temu instrukcje w kodzie można zapisywać w wygodniejszej krótszej wersji, np. `cout` zamiast `std::cout`

Funkcja `main` to główny blok kodu każdego programu. Początek i koniec bloku oznacza się nawiasami klamrowymi

cout

Instrukcja wyjścia → jej działanie polega na wyświetleniu tekstu na ekranie

<<

operator, którego zadaniem jest przekazywanie danych

```
cout << "Hello world!" << endl;
```

end of line → znak końca wiersza

```
return 0;
```

Poprawne zakończenie programu

Zasady tworzenia kodu w języku C++

Wielkość liter w kodzie ma znaczenie – zmienne `a` i `A` mogą przechowywać różne wartości.

Nie używamy polskich znaków w nazwach tworzonych zmiennych i funkcji.

Każdą instrukcję kończymy średnikiem.

Tekst po znakach `//` jest komentarzem – można używać w nim polskich znaków.

Warto stosować w kodzie wcięcia i puste wiersze. Zwiększają one czytelność programu.

Spacje, wcięcia i inne tzw. białe znaki nie mają znaczenia w procesie kompilacji.



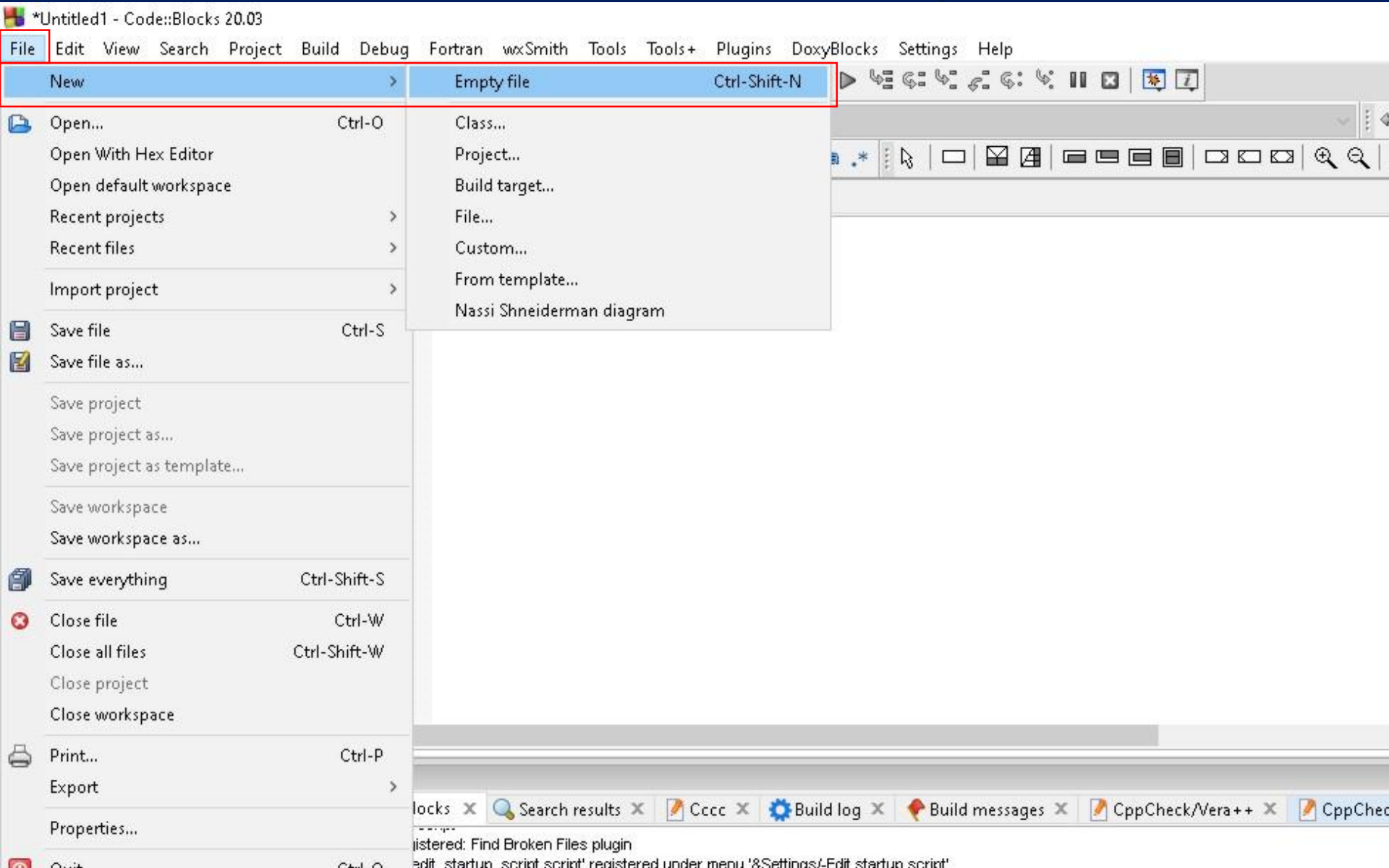
Code::Blocks

The open source, cross-platform IDE

20.03

Tworzenie nowego kodu

Podręcznik str. 101



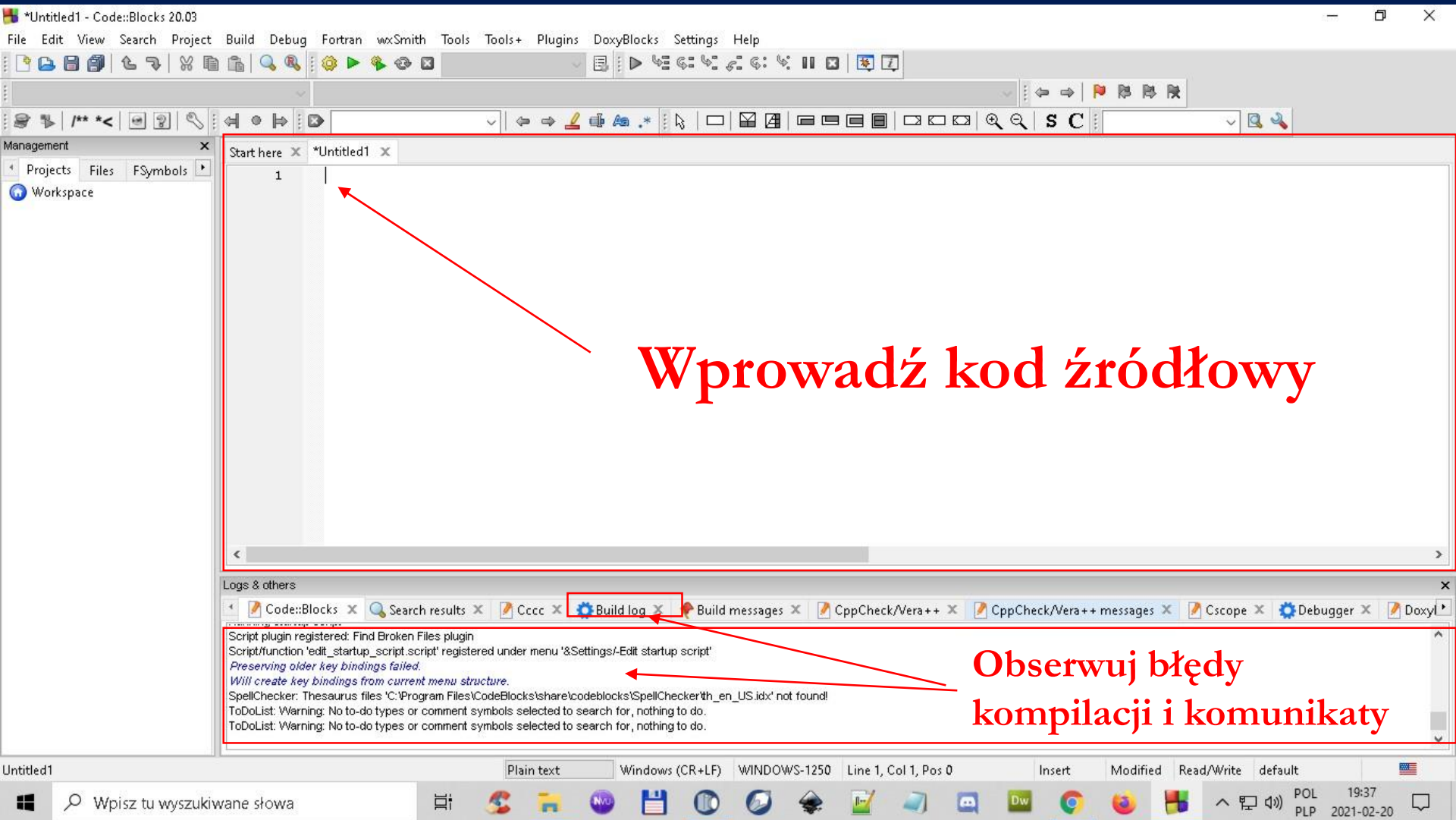


Code::Blocks

The open source, cross-platform IDE

20.03

Tworzenie nowego kodu





Code::Blocks

The open source, cross-platform IDE

20.03

Program Hello World!

Podręcznik str. 101

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() //nagłówek funkcji głównej
6 {
7     cout << "Hello World!" << endl;
8     return 0;
9 }
```

Wprowadź kod źródłowy

Untitled1

Plain text

Windows (CR+LF)

WINDOWS-1250

Line 1, Col 20, Pos 19

Insert

Modified

Read/Write

default



Wpisz tu wyszukiwane słowa



POL 21:18
PLP 2021-02-21



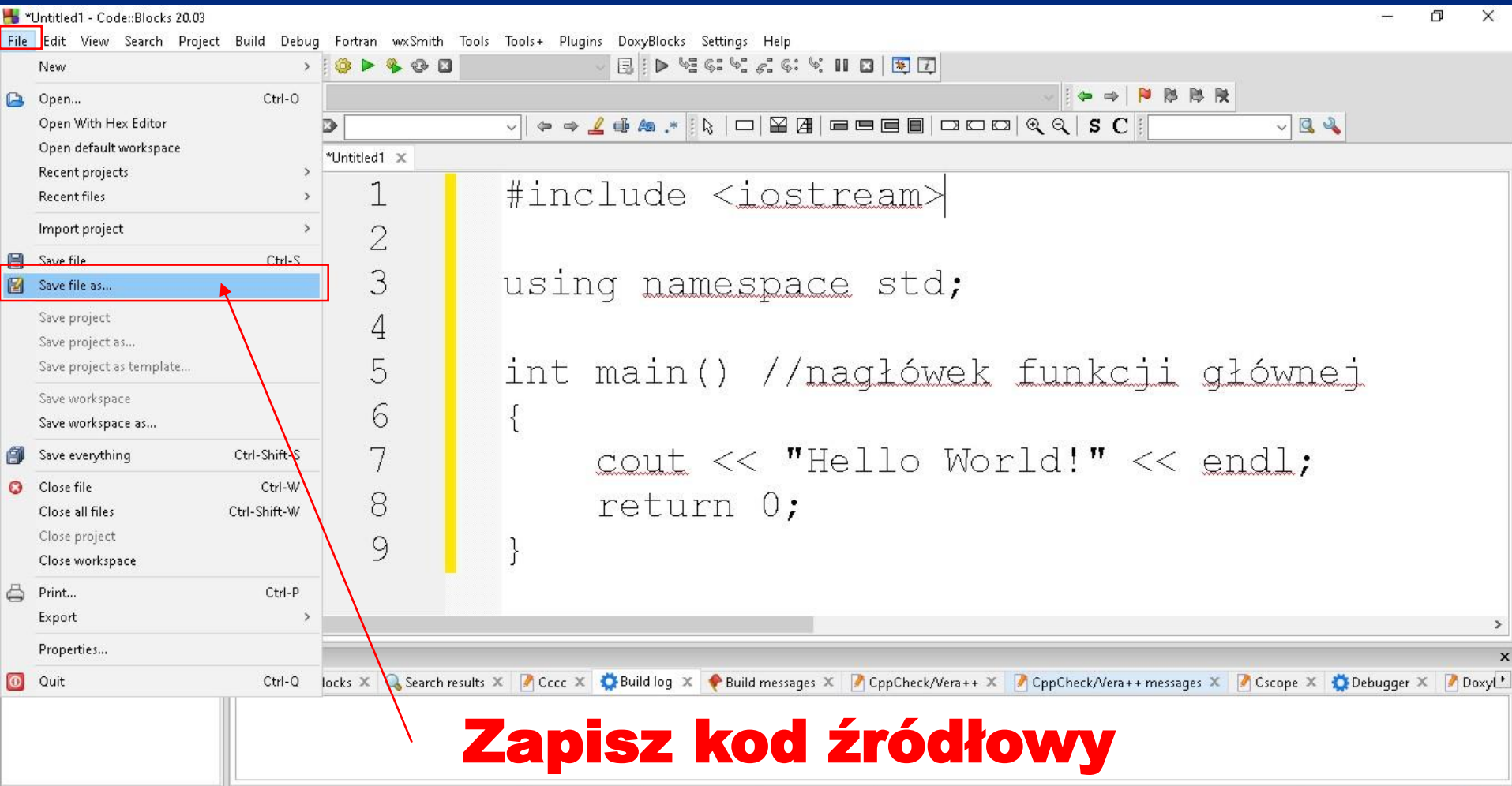
Code::Blocks

The open source, cross-platform IDE

20.03

Program Hello World!

Podręcznik str. 102



Zapisz kod źródłowy

Save the active file under a different name

Plain text

Windows (CR+LF)

WINDOWS-1250

Line 1, Col 20, Pos 19

Insert

Modified

Read/Write default



Wpisz tu wyszukiwane słowa



POL
PLP
21:20
2021-02-21



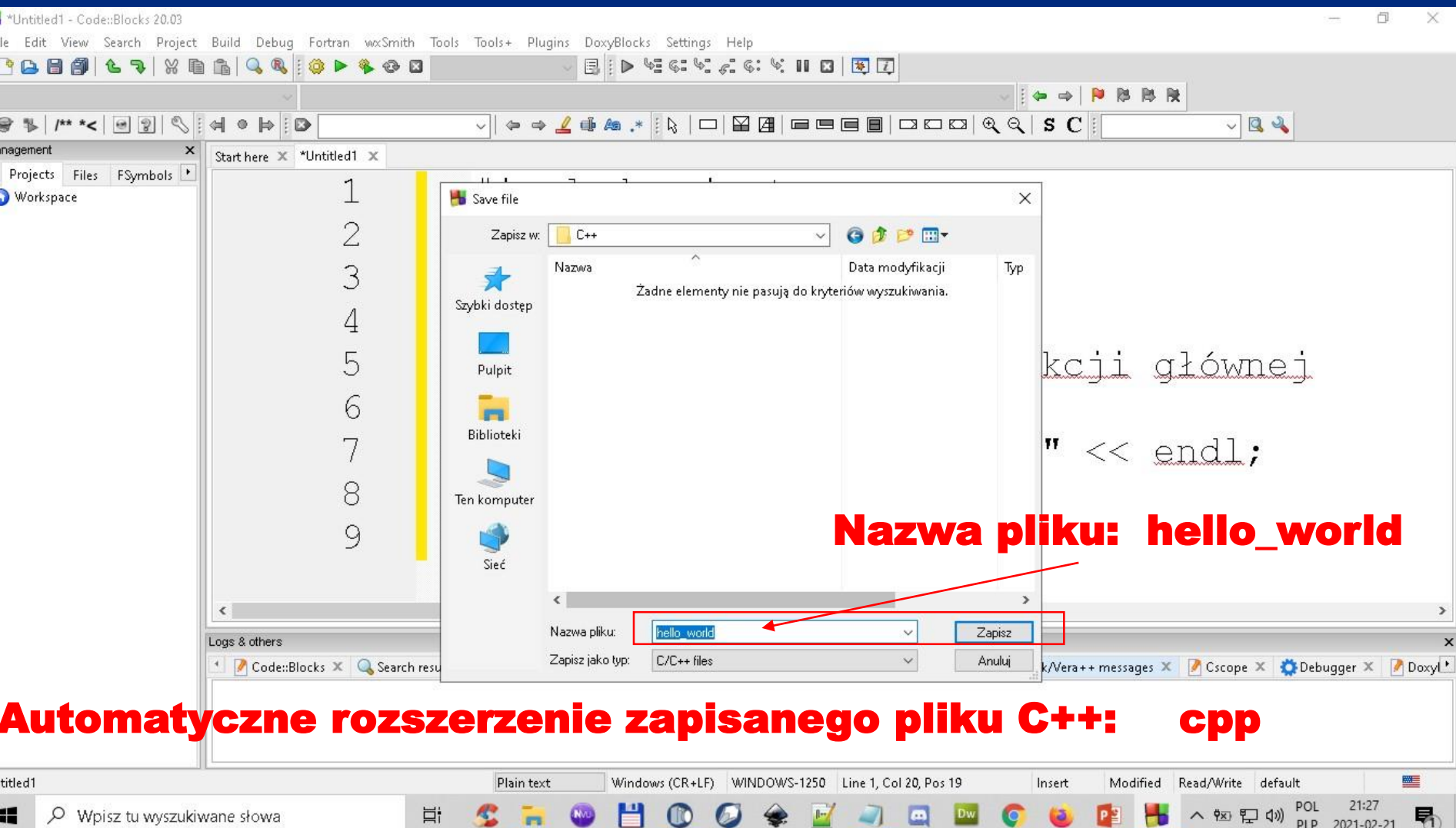
Code::Blocks

The open source, cross-platform IDE

20.03

Program Hello World!

Podręcznik str. 102





Code::Blocks

The open source, cross-platform IDE

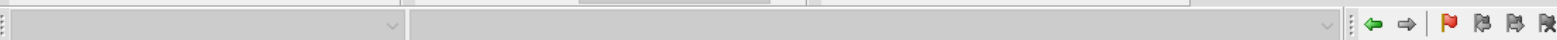
20.03

Program Hello World!

Podręcznik str. 102

hello_world.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help



Management

Projects Files FSymbols

Workspace

Start here x hello_world.c x

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() //nagłówek funkcji głównej
6 {
7     cout << "Hello World!" << endl;
8     return 0;
9 }
10
```

Logs & others

Code::Blocks x Search results x Cccc x Build log x Build messages x CppCheck/Vera++ x CppCheck/Vera++ messages x Cscope x Debugger x Doxy

Widok zapisanego pliku hello_world.cpp

C:\Users\Nauczyciel\Desktop\C++\hello_world.c

C/C++

Windows (CR+LF)

WINDOWS-1250

Line 10, Col 1, Pos 148

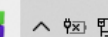
Insert

Read/Write

default



Wpisz tu wyszukiwane słowa



POL
PLP
2021-02-21





Code::Blocks

The open source, cross-platform IDE

20.03

Kompilacja programu

Podręcznik str. 102

hello_world.c - Code::Blocks 20.03

File Edit View Search Project



Management

Projects Files FSymbols

Workspace

Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Build Ctrl-F9

Compile current file Ctrl-Shift-F9

Run Ctrl-F10

Build and run F9

Rebuild Ctrl-F11

Clean

Build workspace

Rebuild workspace

Clean workspace

Abort

Errors

Select target

Export Makefile

CTRL+F9

```
6 {
7
8
9 }
10
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main() //nagłówek funkcji
```

```
cout << "Hello World!";
return 0;
```

Logs & others

Code::Blocks Search results Compiler Build log Build messages Compiler Messages



Code::Blocks

The open source, cross-platform IDE

20.03

Kompilacja programu

Podręcznik str. 102

hello_world.cpp - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main() : int

Management

Projects Files FSymbols

Workspace

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() //naglowek funkcji glownej
6 {
7     cout << "Hello World!" << endl;
8     return 0;
9 }
```

Logs & others

Code::Blocks Search results Cccc Build log CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger DoxyBlocks Fortran in

----- Build file: "no target" in "no project" (compiler: unknown)-----

```
g++.exe -c C:\Users\Nauczyciel\Desktop\C++\hello_world.cpp -o C:\Users\Nauczyciel\Desktop\C++\hello_world.o
g++.exe -o C:\Users\Nauczyciel\Desktop\C++\hello_world.exe C:\Users\Nauczyciel\Desktop\C++\hello_world.o
Process terminated with status 0 (0 minute(s), 1 second(s))
0 error(s), 0 warning(s) (0 minute(s), 1 second(s))
```

C:\Users\Nauczyciel\Desktop\C++\hello_world.cpp

C/C++ Windows (CR+LF) WINDOWS-1252 Line 5, Col 34, Pos 80 Insert Read/Write default

Wpisz tu wyszukiwane słowa

POL 22:19 2021-02-21

**Sprawdź czy
nie ma błędów
w kodzie**



Code::Blocks

The open source, cross-platform IDE

20.03

Plik wykonywalny programu

Podręcznik str. 102

hello_world.cpp - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Build Ctrl-F9

Compile current file Ctrl-Shift-F9

Run Ctrl-F10

Build and run F9

Rebuild Ctrl-F11

Clean

Build workspace

Rebuild workspace

Clean workspace

Abort

Errors

Select target

Export Makefile

Management

Projects Files FSymbols

Workspace

CTRL+F10

```
#include <iostream>
```

```
using namespace std;
```

```
int main() //naglowek fun
```

```
cout << "Hello World  
return 0;
```

Logs & others



Code::Blocks

The open source, cross-platform IDE

20.03

Plik wykonywalny programu

Podręcznik str. 102

hello_world.cpp - Code::Blocks 20.03

```
File
C:\Users\Nauczyciel\Desktop\C++\hello_world.exe
Hello World!
Process returned 0 (0x0)   execution time : 0.313 s
Press any key to continue.
```

Logs & others

Code::Blocks x Search results x Cccc x Build log x CppCheck/Vera++ x CppCheck/Vera++ messages x Cscope x Deb

0 error(s), 0 warning(s) (0 minute(s), 1 second(s))

Checking for existence: C:\Users\Nauczyciel\Desktop\C++\hello_world.exe

Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\Nauczyciel\Desktop\C++\hello_world.exe" (in 'C:\Users\Nauczyciel\Desktop\C++\')