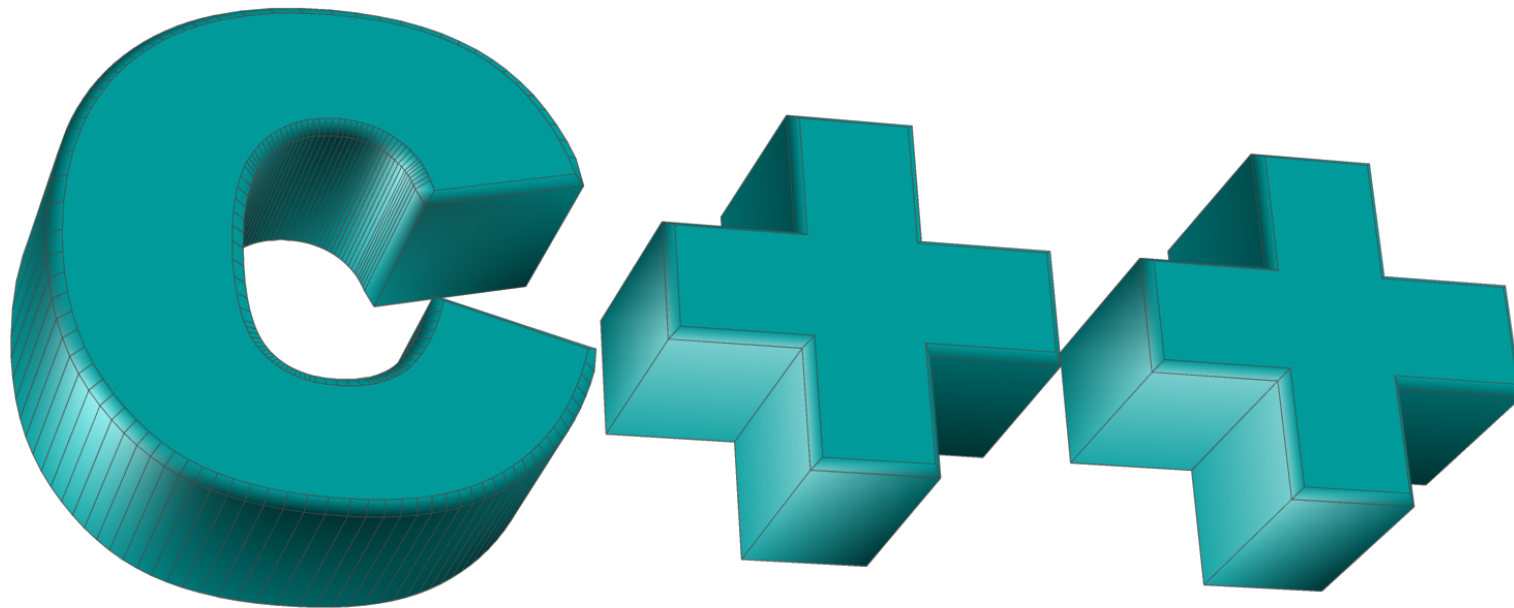


Język C++



Typ danych

- **char** - pojedynczy znak, np. 107 oznacza literę 'k'
- **string** - ciąg znaków
- **int** - liczba całkowita
- **float** - liczba rzeczywista
- **bool** - logiczny (prawda lub fałsz)

Stałe i zmienne

Tworzenie stałej:

```
const float PI=3.14
```

Tworzenie zmiennej:

```
int x=7
```

Załączanie bibliotek

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

(pozwala na stosowanie uproszczonych dyrektyw;

każda biblioteka ma swoją przestrzeń nazw)

Operatory

Matematyczne		Logiczne	
+	dodawanie	==	jest tożsame
-	odejmowanie	!=	jest różne (negacja)
*	mnożenie	<	jest mniejsze
/	dzielenie	<=	jest mniejsze lub równe
%	reszta z dzielenia	>	jest większe
++	zwiększanie o jeden (inkrementacja)	>=	jest większe lub równe
--	pomniejszanie o jeden (dekrementacja)		lub (alternatywa)
		&&	oraz (koniunkcja)
Wynik operacji: jakaś liczba.		Wynik operacji: prawda lub fałsz.	

Instrukcje warunkowe

if (warunek)

{

instrukcje

}

1. Jeśli warunek jest prawdziwy program wykonuje instrukcję, jeśli fałszywy nie robi nic.

2. Jeśli warunek jest prawdziwy, program wykonuje instrukcję, jeśli fałszywy wykonuje inną instrukcję

if (warunek)

{

instrukcja1

}

else

{

instrukcja2

}

Instrukcje warunkowe

switch (zmienna)

```
{  
  case zmienna1: instrukcja1; break;  
  case zmienna2: instrukcja2; break;  
  // itd...  
  default: instrukcjaN;  
}
```

Jeśli zmienna będzie mieć wartość *x*, wykonaj *instrukcję1*; jeśli zmienna będzie mieć wartość *y*, wykonaj *instrukcję2*; itd. Domyślnie (gdy nie zostanie spełniony żaden warunek) wykonaj *instrukcjęN*.

Instrukcje iteracyjne (pętle)

```
while (warunek)  
{  
  instrukcje  
}
```

Dopóki spełniony jest warunek, wykonuj instrukcje.

Instrukcje iteracyjne (pętle)

do

{

instrukcje

}

while (warunek);

Wykonuj instrukcje, dopóki spełniony jest warunek.

Instrukcje iteracyjne (pętle)

```
for (int i=0; i<10; i++)  
{  
    instrukcje  
}
```

Wykonaj pętlę określoną ilość razy (zaczynij od 0 i wykonuj aż i osiągnie wartość 9; powiększ wartość „i” po każdym przebiegu).

Funkcje

```
int nazwaFunkcji(int x, int y)
{
    instrukcje
    return x+y // W innych przypadkach może to być return EXIT_SUCCESS;
}
```

Każdą funkcję należy zainicjować i każda funkcja powinna zwrócić jakąś wartość.
Funkcje pomocnicze umieszczamy przed główną funkcją *main()*.

Instalacja Code::Blocks

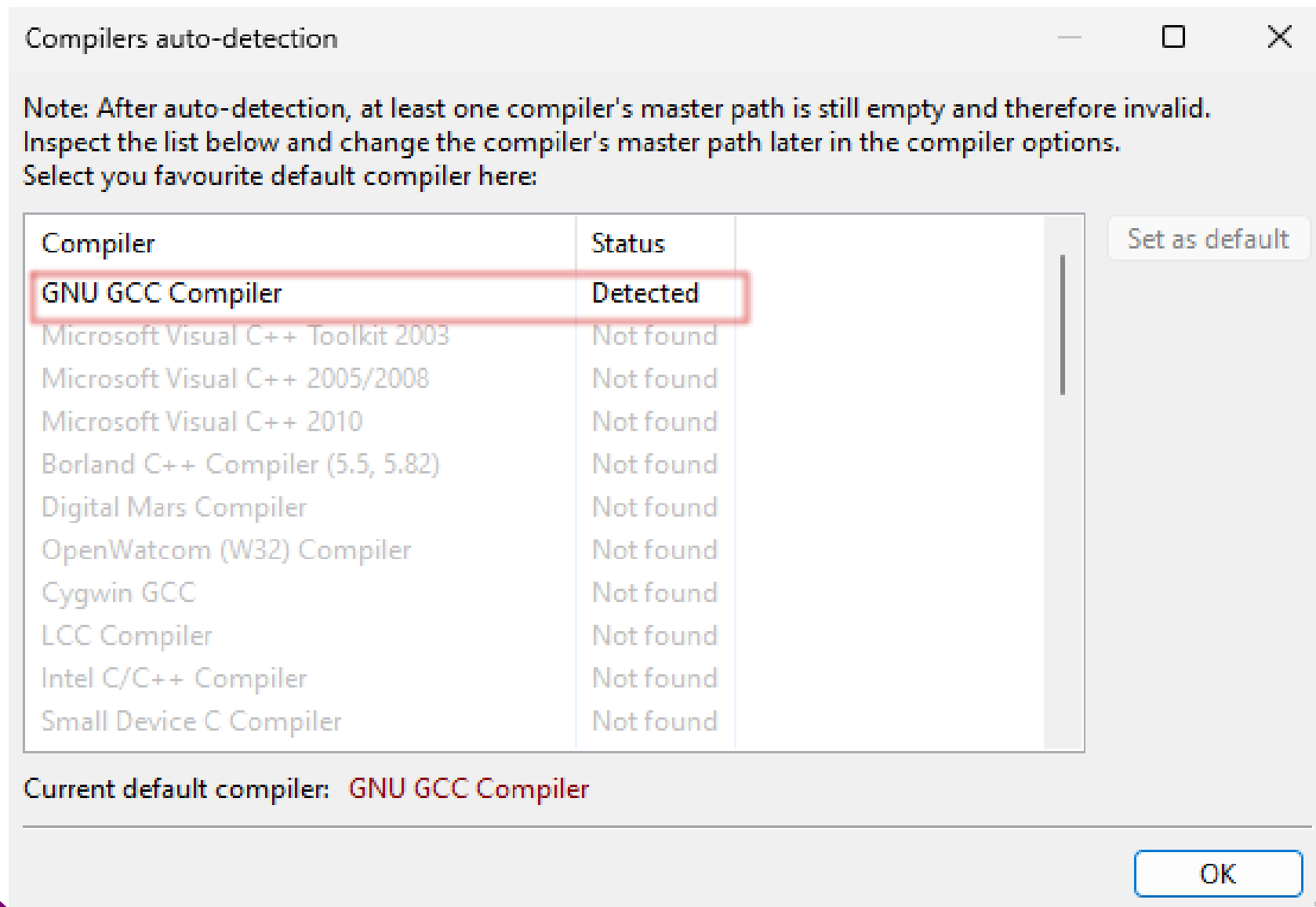
- Ściągnij program instalacyjny ze strony: codeblocks.org.

Wersja z kompilatorem:

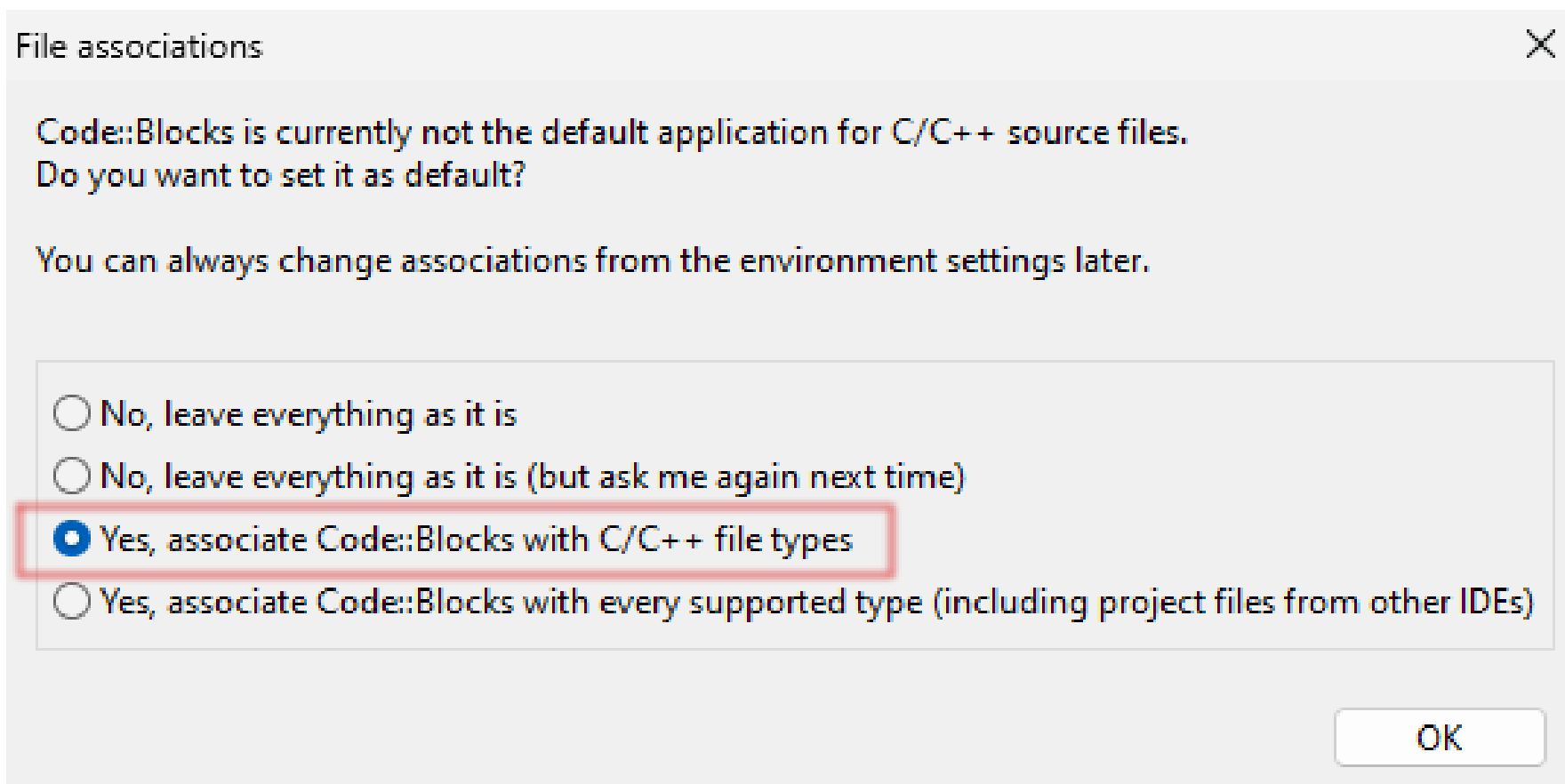
[codeblocks-*.mingw-setup.exe](#).

- Zainstaluj z domyślnymi ustawieniami.
- Utwórz pierwszy projekt zgodnie z poniższymi wskazówkami.

Pierwsze uruchomienie

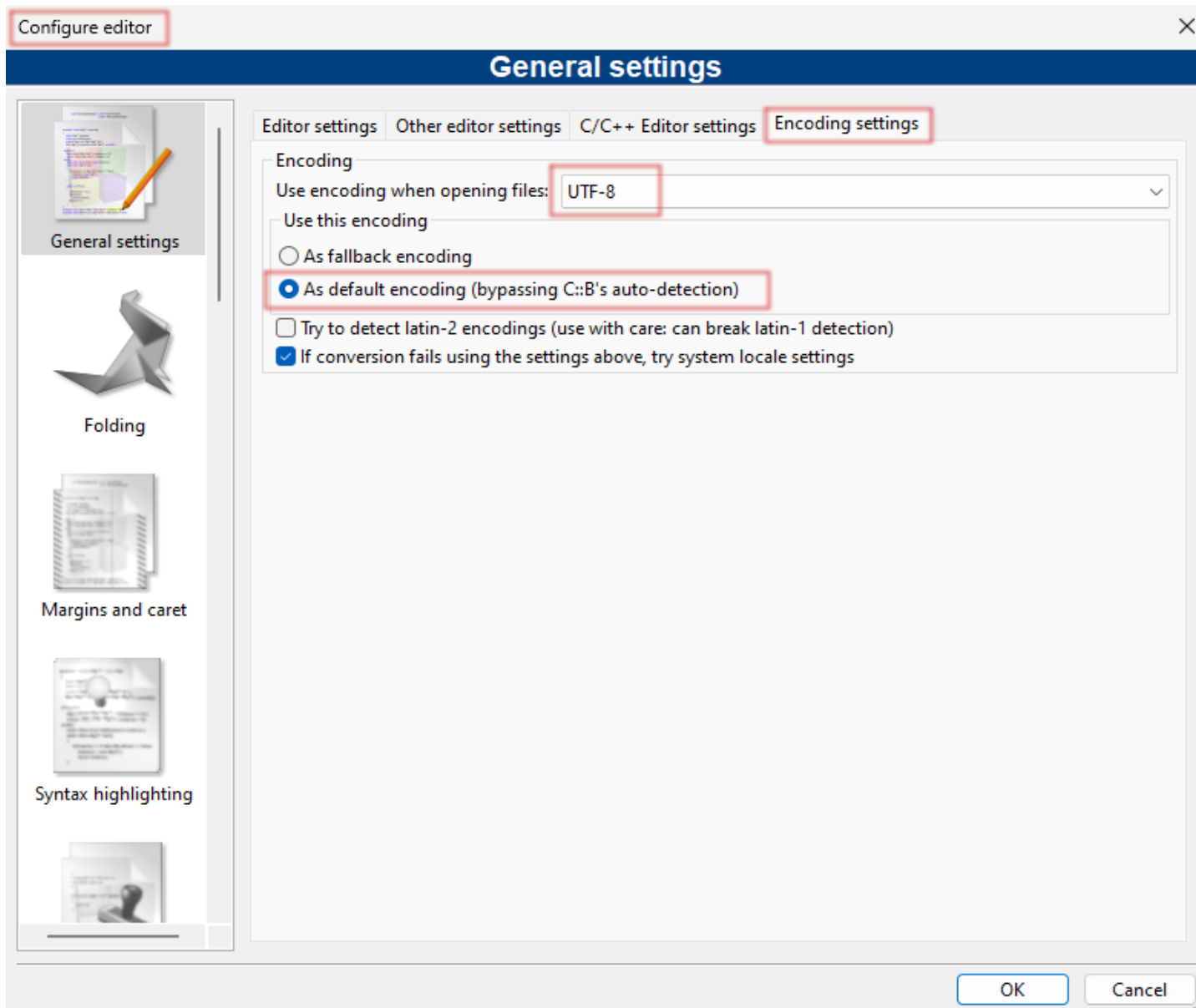


Pierwsze uruchomienie



Ustawienia poinstalacyjne

Settings / Editor...



Ustawienia poinstalacyjne

Settings / Compiler...

Global compiler settings

Selected compiler: GNU GCC Compiler

Buttons: Set as default, Kopiuuj, Rename, Usuń, Reset defaults

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

Batch builds

Policy:

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

Debugging

- Optimize debugging executable (compile speed, execution speed and better debugging) [-Og]
- Produce debugging symbols [-g]

Ogólne

- Have g++ follow the 1998 GNU C++ language standard (ISO C++ plus GNU extensions) [-std=gnu++98]
- Have g++ follow the 1998 ISO C++ language standard [-std=c++98]
- Have gcc follow the 1990 ISO C language standard (certain GNU extensions that conflict with ISO C90 are disabled) [-std=c90]
- Have gcc follow the 1999 ISO C language standard [-std=c99]
- Have gcc follow the 2011 ISO C language standard [-std=c11]
- Have gcc follow the 2017 ISO C language standard [-std=c17]
- In C mode, this is equivalent to -std=c90, in C++ mode, it is equivalent to -std=c++98 [-ansi]
- Position Independent Code [-fPIC]
- Static libgcc [-static-libgcc]**
- Static libstdc++ [-static-libstdc++]**
- Static linking [-static]**
- Target x86 (32bit) [-m32]
- Target x86_64 (64bit) [-m64]**

Profiling

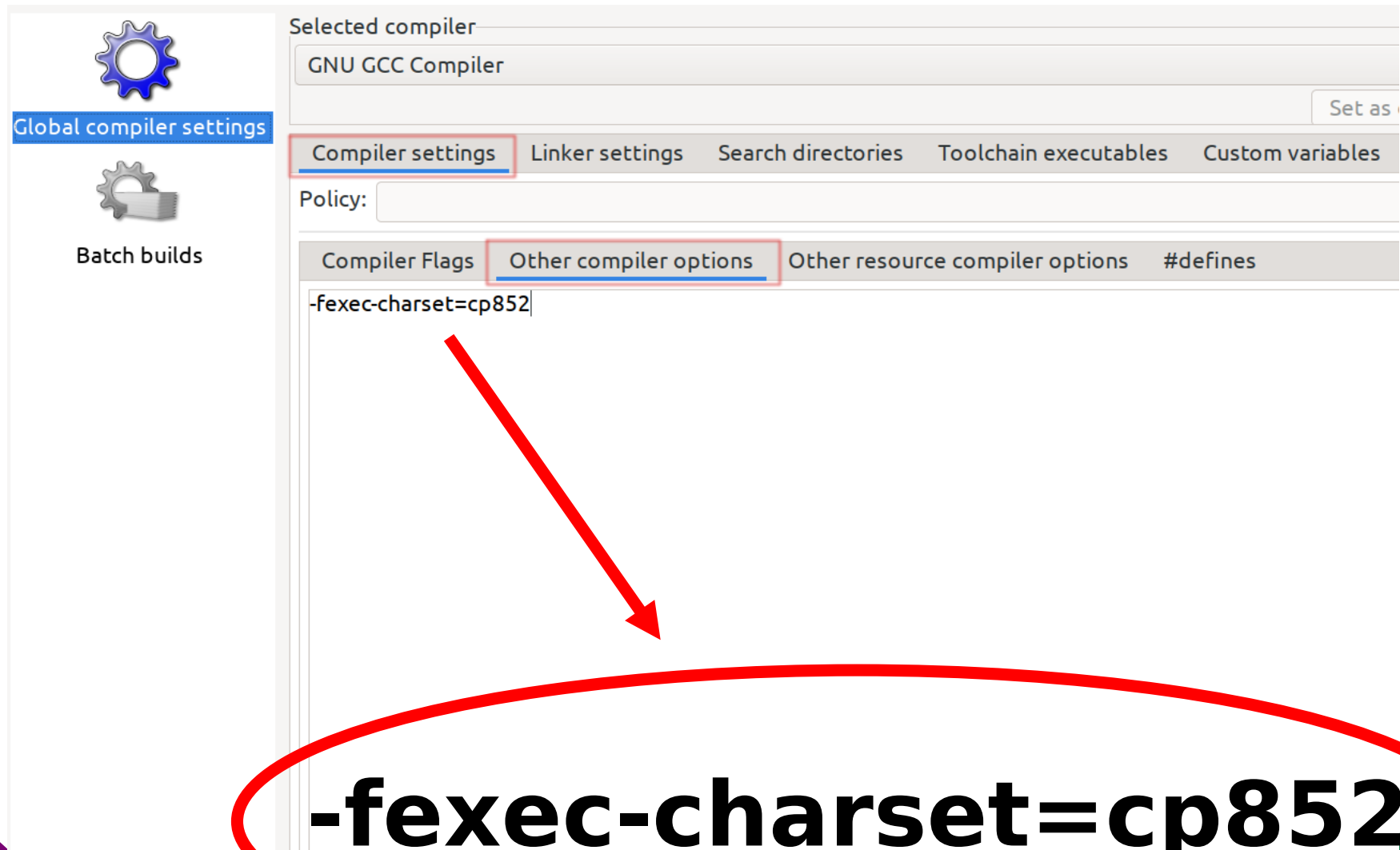
- Profile code when executed [-pg]

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

Anuluj OK

Ustawienia poinstalacyjne

Settings / Compiler...



Selected compiler
GNU GCC Compiler

Global compiler settings

Batch builds

Compiler settings | Linker settings | Search directories | Toolchain executables | Custom variables

Policy:

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

`-fexec-charset=cp852`

-fexec-charset=cp852

Nowy projekt

The screenshot displays the Code::Blocks IDE interface. At the top, the menu bar includes 'Plik', 'Maszyna', 'Widok', 'Wejście', 'Urządzenia', and 'Pomoc'. Below it, the main menu contains 'File', 'Edit', 'View', 'Search', 'Project', 'Build', 'Debug', 'Fortran', 'wxSmith', 'Tools', 'Tools+', 'Plugins', 'DoxyBlocks', 'Settings', and 'Help'. The toolbar contains various icons for file operations, editing, and building. On the left, the 'Management' pane shows 'Projects' and 'Workspace'. The main workspace area displays the 'Start here' screen with the Code::Blocks logo and the text 'Code::Blocks The open source, cross-platform IDE'. Below the logo, there is a release information link: 'Release 20.03 rev 11983 (2020-03-12 18:24:30) gcc 8.1.0 Windows/unicode - 64 bit'. A red arrow points to the 'Create a new project' button, which is highlighted with a red box. Other buttons include 'Open an existing project', 'Tip of the Day', 'Visit the Code::Blocks forums', and 'Report a bug or request a new feature'. At the bottom, the 'Logs & others' pane shows several open log files and their contents.

Management
Projects Files FSymbols
Workspace

Start here x

Code::Blocks
The open source, cross-platform IDE

Release 20.03 rev 11983 (2020-03-12 18:24:30) gcc 8.1.0 Windows/unicode - 64 bit

Create a new project Open an existing project Tip of the Day

Visit the Code::Blocks forums Report a bug or request a new feature

Recent projects
No recent projects

Recent files
No recent files

© 2004 - 2018, The Code::Blocks Team.

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 DoxyBlocks

Initial scaling factor is 1.000 (actual: 1.000)
Running startup script
Script plugin registered: Find Broken Files plugin
Script/function 'edit_startup_script.script' registered under menu '&Settings/-Edit startup script'
Preserving older key bindings failed.
Will create key bindings from current menu structure.
SpellChecker: Thesaurus files 'C:\Program Files\CodeBlocks\share\codeblocks\SpellChecker\th_en_US.idx' not found!

Nowy projekt

New from template

Projects
Build targets
Files
Custom
User templates

Category: <All categories>

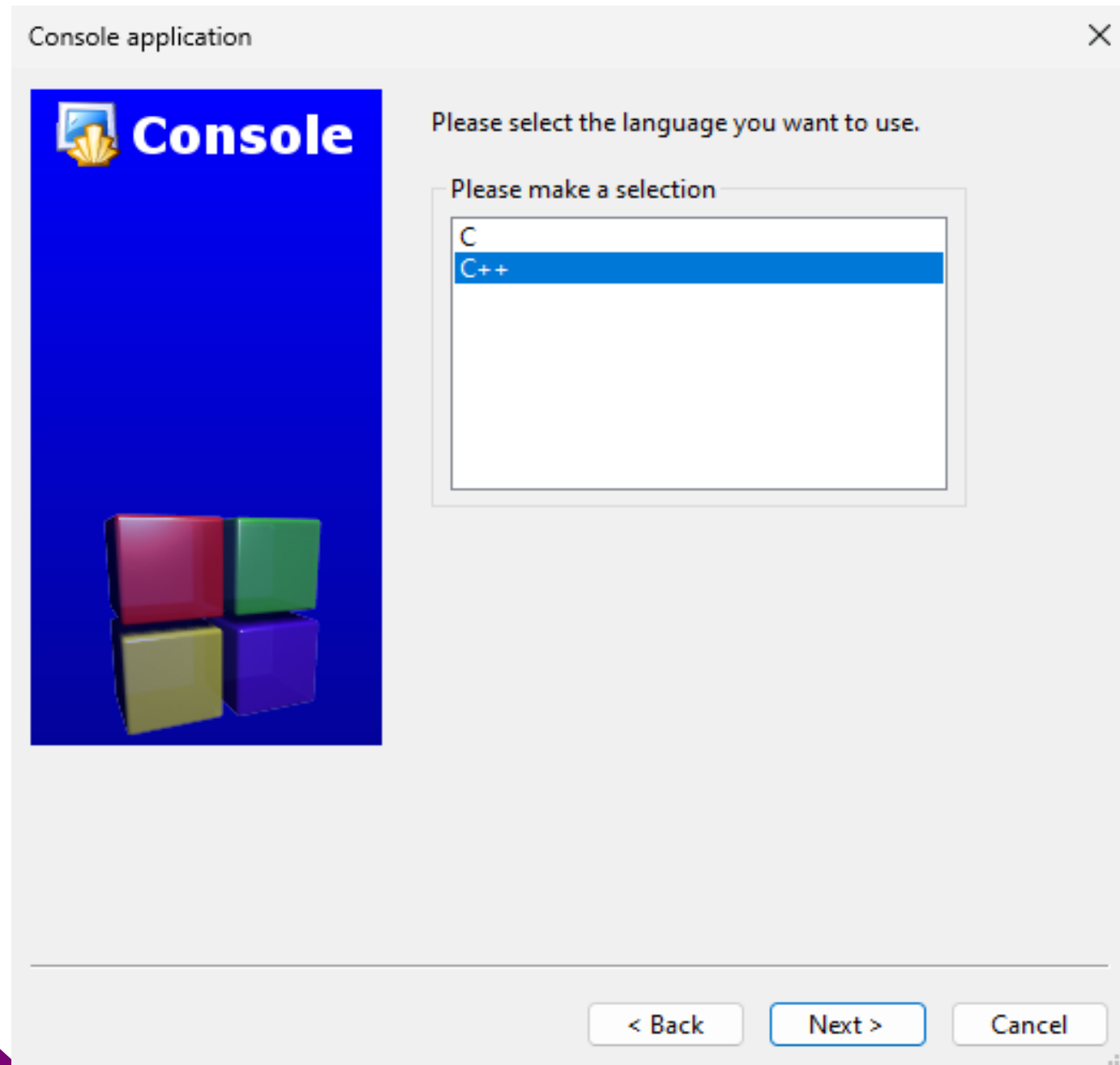
Go
Cancel

View as
 Large icons
 List

TIP: Try right-clicking an item


1. Select a wizard type first on the left
2. Select a specific wizard from the main window (filter by categories if needed)
3. Press Go

Nowy projekt



Nowy projekt

Console application ✕



Please select the folder where you want the new project to be created as well as its title.

Project title:

Folder to create project in:
 ...


Project filename:

Resulting filename:

< Back Next > Cancel

Nowy projekt

Console application

 **Console**

Please select the compiler to use and which configurations you want enabled in your project.

Compiler:

Create "Debug" configuration:

"Debug" options

Output dir.:

Objects output dir.:

Create "Release" configuration:

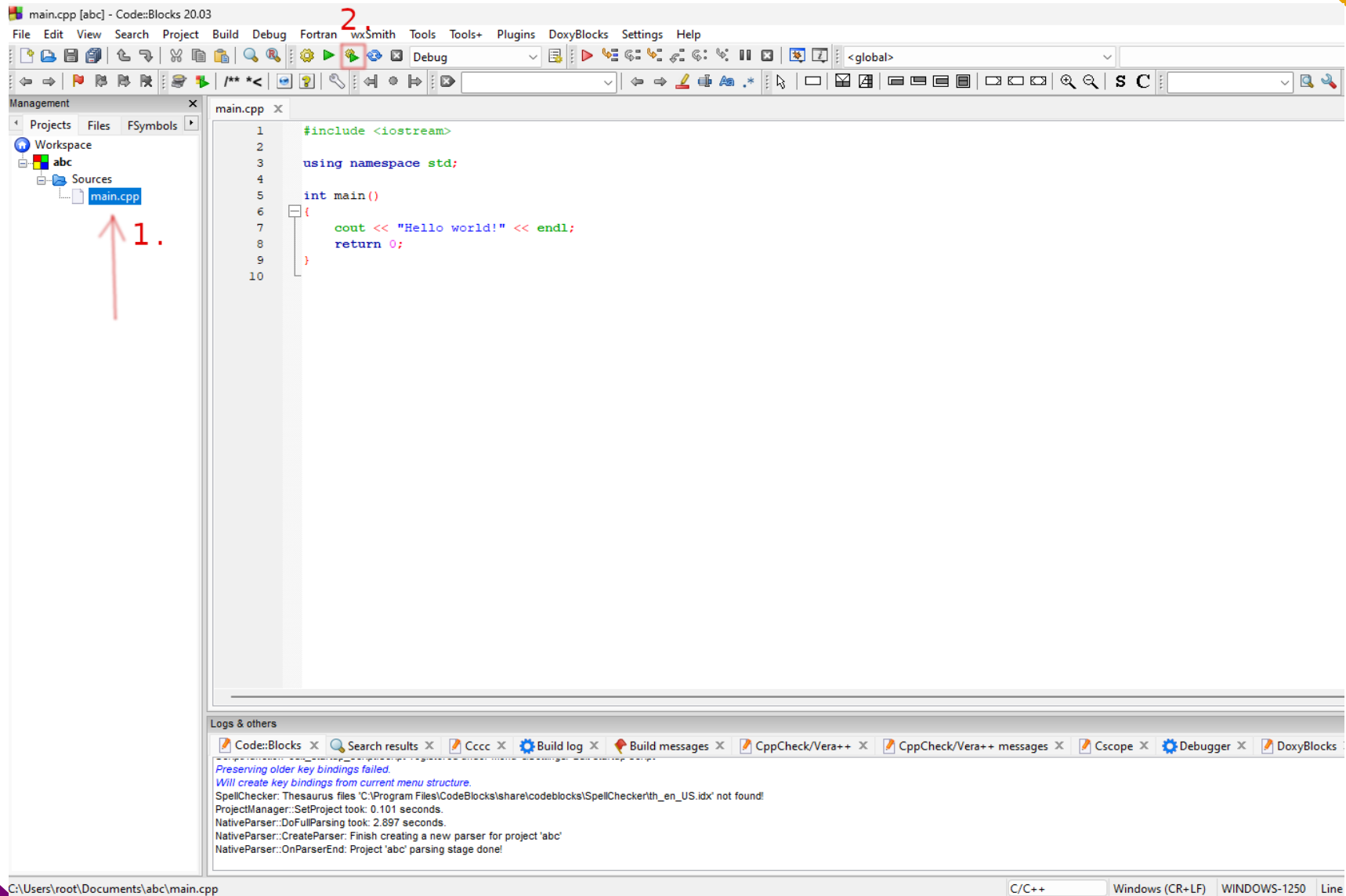
"Release" options

Output dir.:

Objects output dir.:

< Back Finish Cancel

Nowy projekt



Ćwiczenie 1

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main()
{
    cout << "Udało się! Mój pierwszy program w C++!";
    cout << "Autor: Imię Nazwisko." << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```


Gdzie jest mój *.exe?

- Zbudowany program znajduje się w:
 - *folderProjektu / bin / Debug / plik.exe.*
- Plik z kodem ma postać: *plik.cpp.*
- Plik projektu to: *plik.cbp.*
- Dokładne informacje:
 - *File / Properties...*
 - *Project / Properties...*

Ćwiczenie 2

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    float a, b;
```

```
    // Poniżej muszą być zwykłe cudzysłowy, nie apostrofy!
```

```
    cout << "Podaj liczbę A: ";
```

```
    cin >> a;
```

```
    cout << "Podaj liczbę B: ";
```

```
    cin >> b;
```

```
    if ( a < b ) {
```

```
        cout << "Liczba A jest mniejsza od B." << endl;
```

```
        cout << "Różnica pomiędzy zmiennymi wynosi " << b - a << endl;
```

```
    }
```





Ćwiczenie 2

```
if ( a >= b ) {  
    cout << "Liczba A jest większa lub równa B." << endl;  
    cout << "Różnica pomiędzy zmiennymi wynosi " << a - b << endl;  
}  
  
cout << endl << endl;  
system("PAUSE");           // Tylko system Windows  
return EXIT_SUCCESS;  
}
```

Inne pomysły

- quiz;
- „milionerzy”;
- horoskop;
- rozliczenia roczne PIT;
- „szczęśliwy numer”.