

NPRG041 – C++

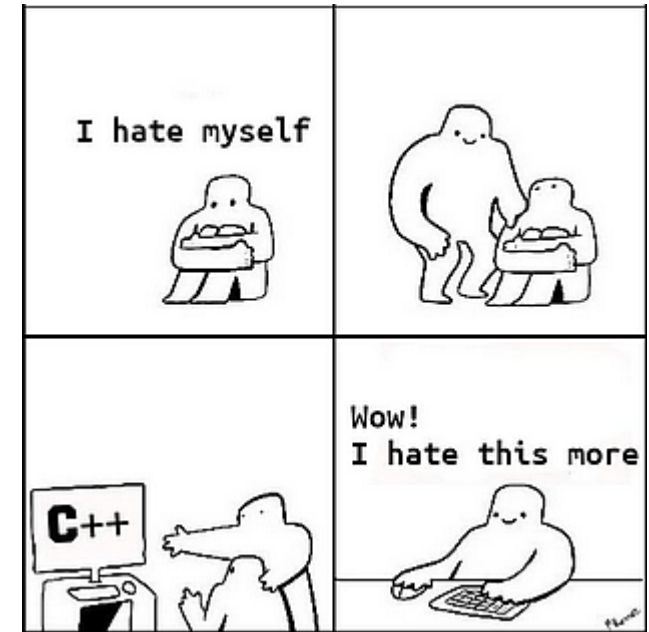
cvičení – Jiří Klepl

mattermost: ulita/2425ZS: nprg041-cpp-klepl (inv na SIS nástěnce)

Klepl@d3s.mff.cuni.cz

Agenda

- Organizační záležitosti
- Nástroje na práci v C++
 - Editor, kompilátor, debugger
 - Mattermost, Gitlab, ReCodex
- C++ programy
 - Typické typy, proměnné
 - Textový vstup a výstup
 - CMD argumenty
 - Hello World
 - Příklad: násobilka



Organizace cvičení

- **Příklady na cvičeních** – povinné, <1h 
 - Odevzdávání přes GitLab
 - Zadávány, diskutovány na cvičeních
 - Jejich **vypracování = docházka** (včetně absencí)
- **ReCodexové úkoly** – dobrovolné, ale silně doporučené, <4h 
 - Zadávány online, 2-3 za celý rok; dobrá příprava na zápočtový test
- **Závěrečný ReCodexový úkol** – povinný, ~4h 
 - Zadán v zimě, deadline před posledním cvičením, bonusový feedback
- **Zápočtový test** – na posledním cvičení (**10. 1. 2025 - POVINNÉ**)
 - Na školním počítači přes ReCodex, jen jeden opravný termín
 - Rozšíření závěrečného úkolu

Komunikace

- **Cvičení** jsou pro diskuze, kdykoliv se můžete na něco zeptat
- Mimo cvičení je nejlepší psát přes **Mattermost** (tam jsem online nejvíc)
 - Pokud je problém s invitem nebo něco podobného, pak je nejlepší **Email**
 - Mattermost workspace: **2425ZS**
 - Channel: **nprg041-cpp-klepl**
 - Na zápočtové programy, bugy v úkolech, atd.: direct message na **@jiriklepl**
- Pokud jste dlouho zaseklí na úkolu, pište

Zápočtový program (GitLab)

1. Návrh zadání do 20. 11. 2024

- Konkrétní nápad programu/nástroje, čím to je zajímavé
- Nechceme něco, co jde nakódit za den nebo najít na internetu

2. Schválení zadání do 30. 11. 2024

- „Specifikace“: co to bude dělat, jak s tím bude uživatel pracovat, ext. libky

3. Technologické demo do 12. 1. 2025

- Jde stáhnout, zbuildit, spustit (bez dalších kroků), včetně ext. libek

4. Odevzdání kompletního řešení do 15. 5. 2025

- Program by měl fungovat alespoň na Windows a Linuxu

5. Poslední opravy do 15. 6. 2025

Zaměření cvičení

- **Očekávané znalosti** z počítačových systémů a programování 2
 - Syntaxe C/C++, dekompozice programu, třídy
 - Algoritmizace
 - Visual Studio (popř. jiný editor, compiler a debugger)
- **Cíle**
 - **Důkladná znalost jazyka:** pokročilé konstrukce, efektivita
 - Knihovny
 - **Best practices:** kvalita návrhu a kvalita kódu, čemu se vyvarovat, ladění
 - **Multiplatformnost:** Visual Studio na Windows, obecně CMake
 - Moderní použití C++ (**C++20/C++23**)

Zkouška

- Praktickou formou na **školních počítačích** v laboratoři
- Je zadána úloha prověřující znalosti používání jazyka **a kvality kódu**
 - Odevzdávání do Recodexu (testy budou prověřovat základní funkčnost)
 - Lze ztratit body za prohřešky nerozpoznávány testem
 - Skryté buggy: např. memory leaky, corner-casy v algoritmu řešení, ...
 - Kvalita: nečitelnost kódu, výrazná neefektivita řešení, chybné použití prvků jazyka, ...
- Celá známka vychází právě ze zkoušky
- Lze podstoupit až **po zápočtovém testu**
- **Žádná ústní část prověřující teorii**

Mattermost, GitLab, ReCodex

- Invite na Mattermost je na nástěnce předmětu v SISu
- GitLab: <https://gitlab.mff.cuni.cz/teaching/nprg041/2024-25/klepl>

The screenshot displays the ReCodEx web application interface. On the left is a dark sidebar with a 'MENU' section containing links to Dashboard, MFF, Exercises, Archive, SIS Integration (highlighted in green), and FAQ. The main content area is titled 'UK SIS Integration' and shows a user profile for 'Jiří Klepl' with a message: 'Your ReCodEx account is associated with SIS identity identifier "4784...". Below this is a section 'Create Groups Associated with UK SIS Courses' with a warning icon and text: 'Please note that when a group is created from or bound to a SIS course, only one group is allowed to join it.' A section 'Year and term: 2024-1' shows a course entry: 'Fr 10:40 SW2 Programming in C++ (24a|NPRG041x05)'. Below this, a table lists course groups, with one entry circled in red: 'Programming in C++ / 2024/25 1-Winter / Programming in C++ (Fri, 10:40, SW2)'. To the right, a detailed view of the course 'Programování v C++ - NPRG041' is shown. It includes fields for English name, responsible department, faculty, start year, semester, E-credits, content, exam type, number of seats, minimum occupancy, 4EU status, virtual mobility, subject status, language, teaching methods, contact info, and notes. On the right side of this view, a list of related courses and prerequisites is shown, with a link to 'Nástěnka' (Noticeboard) circled in red. At the bottom right, a green callout bubble contains the text 'Link na stránku s invitationem'.

ReCodEx

Jiří Klepl

Settings Logout

MENU

- Dashboard
- MFF
- Exercises
- Archive
- SIS Integration**
- FAQ

UK SIS Integration

Your ReCodEx account is associated with SIS identity identifier "4784...".

Create Groups Associated with UK SIS Courses

SIS courses you teach in particular semesters and which have mapping to ReCodEx

Please note that when a group is created from or bound to a SIS course, only one group is allowed to join it.

Year and term: 2024-1

Fr 10:40 SW2 Programming in C++ (24a|NPRG041x05)

Name
Programming in C++ / 2024/25 1-Winter / Programming in C++ (Fri, 10:40, SW2)

Programování v C++ - NPRG041

Anglický název: Programming in C++

Zajišťuje: Katedra softwarového inženýrství (204. • 32-KSI)

Fakulta: Matematicko-fyzikální fakulta

Platnost: od 2021

Semestr: zimní

E-Kredity: 5

Rozsah, examinační: zimní s.:2/2, Z+Zk [HTI]

Počet míst: neomezen

Minimální obsazenost: neomezen

4EU+: ne

Virtuální mobilita / počet míst pro virtuální mobilitu: ne

Stav předmětu: vyučován

Jazyk výuky: čeština, angličtina

Způsob výuky: prezenční

Způsob výuky: prezenční

Další informace: <https://www.ksi.mff.cuni.cz/teaching/nprg041-web/>

Poznámka: povolen pro zápis po webu

Garant: RNDr. David Bednárek, Ph.D.
RNDr. Filip Zavoral, Ph.D.

Vyučující: RNDr. David Bednárek, Ph.D.
Mgr. Jiří Klepl
Ing. Pavel Koupil, Ph.D.
RNDr. Martin Svoboda, Ph.D.
Mgr. Petr Škoda, Ph.D.
RNDr. Filip Zavoral, Ph.D.

Třída: Informatika Bc.

Klasifikace: Informatika > Programování

Prerekvizity: {NPRG031 nebo NPRM045 nebo NOFY056 nebo NMIN102}, {NPRG031 nebo NPRM045 nebo NOFY056 nebo NMIN102}

Neslučitelnost: NPRX041

Záměnnost: NPRX041

Je korekvizitou pro: NPRG051, NPRX051

Je neslučitelnost pro: NPRX041, NPRG048

Je záměnnost pro: NPRX041, NPRG029

Ve slož. korekvizitě pro: NPRG024

Ve slož. prerekvizitě: NSWI152, NSWI202, NSWX152

Výsledky anket Zapsaní studenti Rozvrh ZS Nástěnka

Link na stránku s invitationem

GitLab

- Struktura

- finalhw ← sem nahrát poslední úkol zadaný na ReCodexu
- homeworks ← sem nahrávat úkoly z ReCodexu (kromě posledního)
- labs ← sem nahrávat úkoly ze cvičení
- project ← adresář na zápočtový program
 - Budou zde zdrojáky, knihovny, data, specifikace, dokumentace

- Co tam patří

- Zdrojáky (.cpp, .hpp)
- Projektové soubory (Visual Studio: .prj, .sln; makefile; CMake*)

- Co tam nepatří

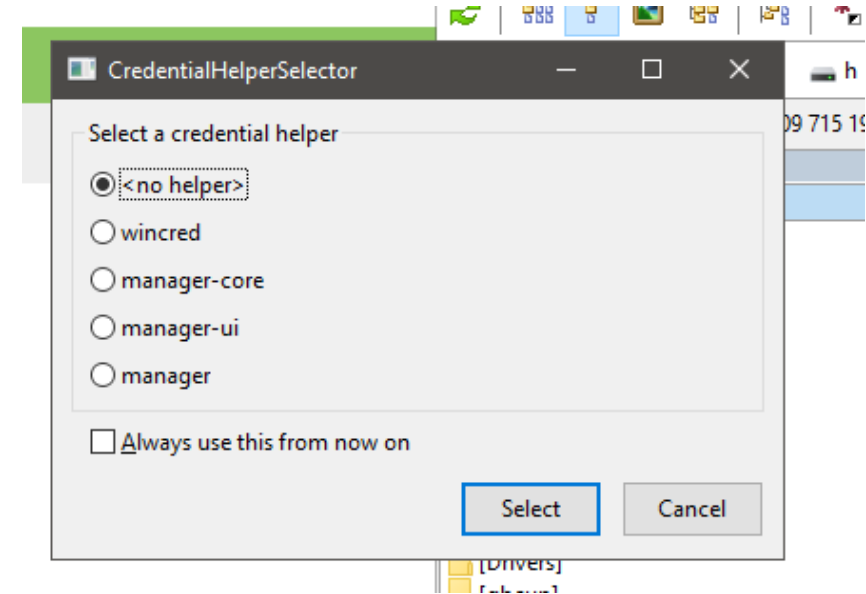
- Vše vygenerované (.o/.obj soubory, .tmp, bin složka, vygenerovaný program, ...)

GIT CLI cheatsheet

- Basic commands
 - Clone(copy) remote repository: `git clone https://url.com/repository.git`
 - Update changes (by coworkers) to the local repository: `git pull`
 - Creating a new GIT commit:
 - Register (stage) a changed file: `git add path/file.cpp`
 - Wrap-up the commit: `git commit -m "What the commit achieves (e.g., Add file.cpp)"`
 - Upload changes to the origin (the cloned one) remote repository : `git push`
 - Upload to a specific remote repository: `git push REPOSITORY [branch]`
 - Queue-up current state of the local repository: `git status`
 - Create new branch: `git branch new-branch`
 - switch branch: `git checkout branch-name`
- More useful commands: `git commit --amend`, `git checkout -b new-branch`

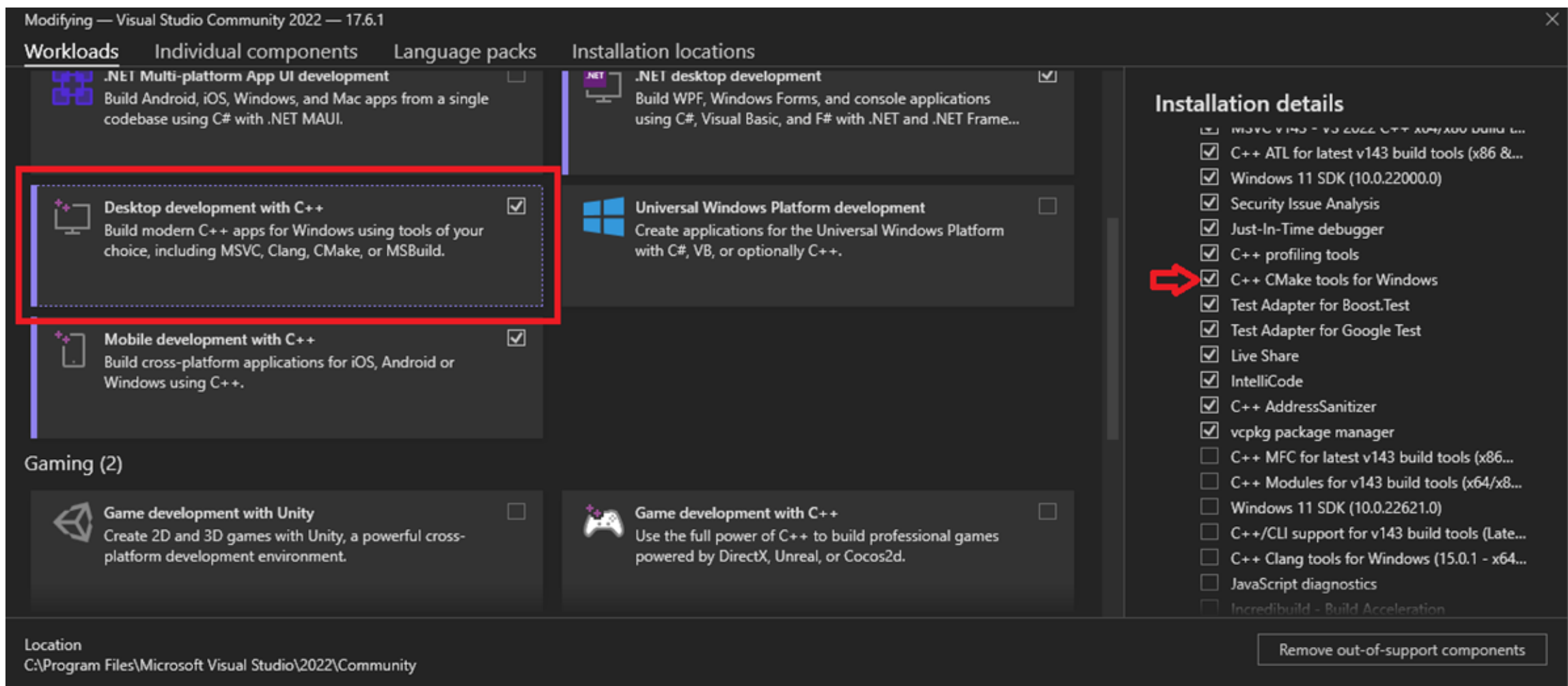
GIT na školních počítačích

- Nenahrávejte SSH klíče ze školních počítačů!
 - **Všichni studenti sdílejí stejný profil**
- Nastavte jméno a email (pro přehlednost)
 - git configure user.name "Jiří Klepl"
 - git configure user.email "klepl@d3s.mff.cuni.cz"
 - **Nepoužívejte flag --global**
- Používejte HTTPS verzi (kopírujte https link z gitlabu, ne ssh)
 - Nepovolujte **žádné credential managery**



Visual Studio IDE

- Stačí Community verze
- Nainstalovat "**Desktop Development with C++**" a "**CMake tools**,"
 - Ukázáno na dalším slidu
- Dobré alternativy (obě multiplatformní)
 - **Visual Studio Code**
 - To používám já
 - Není IDE, je potřeba nainstalovat compiler+CMake zvlášť, spousta rozšíření
 - C/C++ Extension Pack (nebo clangd), CMake, CMake Tools
 - **CLion**
 - Používá Clang compiler
 - Nejčastější IDE na MacOS



Konfigurace Projektu

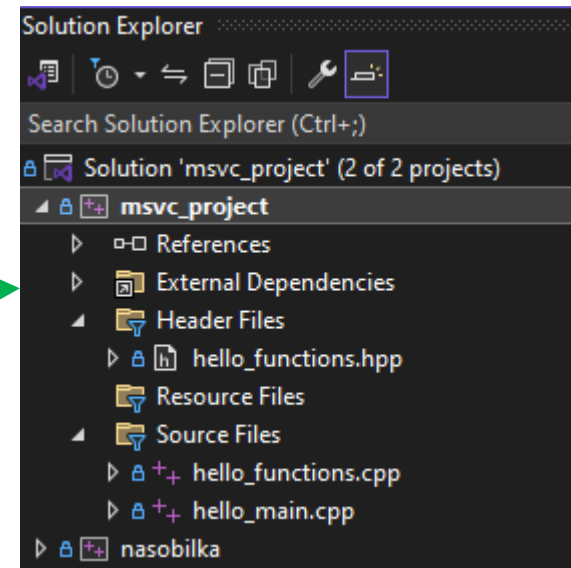


Clone a repository
Get code from an online repository like GitHub or Azure DevOps

Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started



msvc_project Property Pages

Configuration: Active(Debug) Platform: Active(x64) Configuration Manager

Configuration Properties

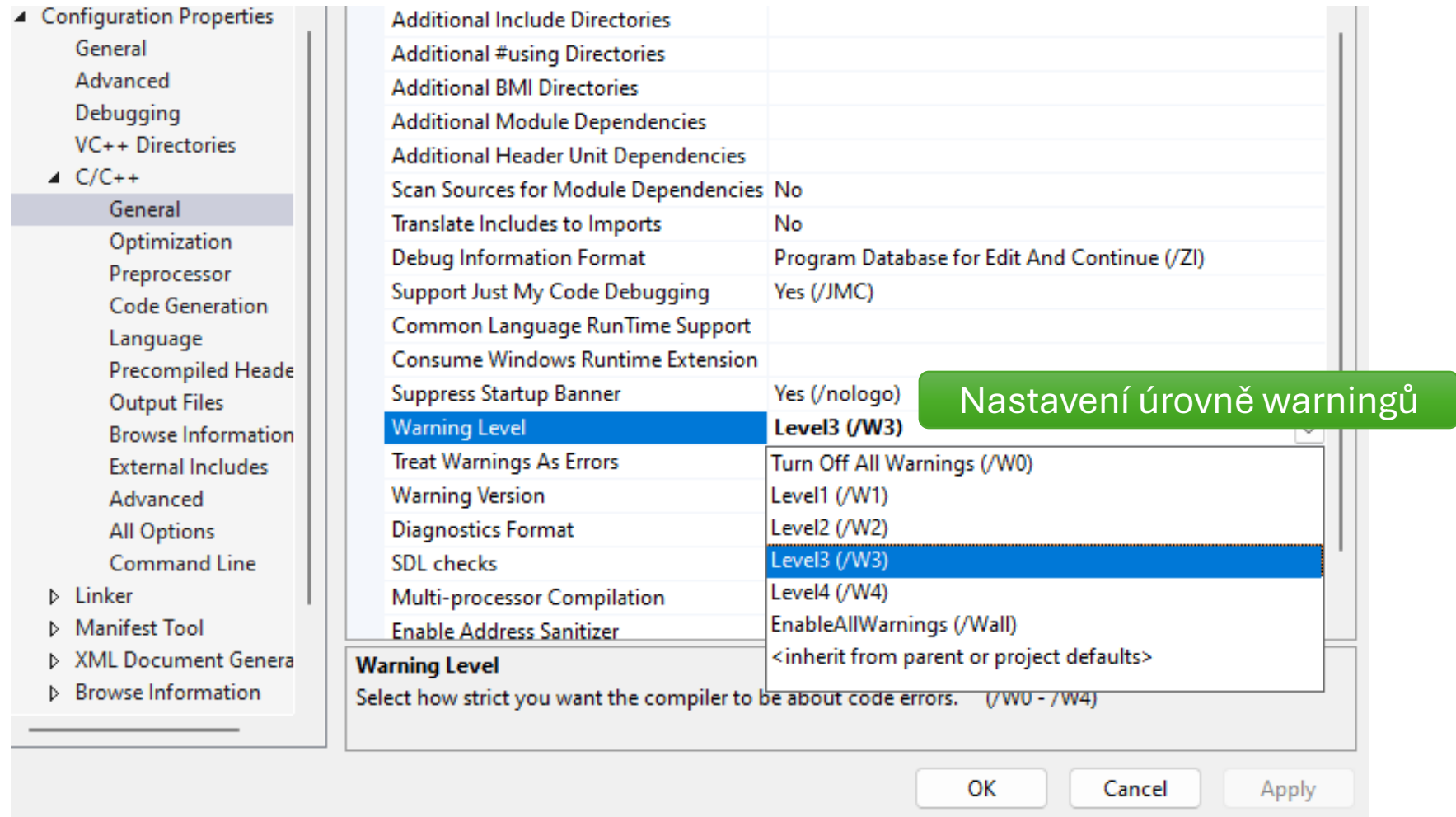
- General
- Advanced
- Debugging
- VC++ Directories
- C/C++
- Linker
 - General
 - Input
 - Manifest File
 - Debugging
- Optimization
- Embedded IDL

General Properties

Output Directory	\$(SolutionDir)\$(Platform)\\$(Configuration)\
Intermediate Directory	\$(ShortProjectName)\\$(Platform)\\$(Configuration)\
Target Name	\$(ProjectName)
Configuration Type	Application (.exe)
Windows SDK Version	10.0 (latest installed version)
Platform Toolset	Visual Studio 2022 (v143)
C++ Language Standard	ires from the Latest C++ Working Draft (/std:c++latest)
C Language Standard	Default (ISO C++14 Standard)
	ISO C++14 Standard (/std:c++14)
	ISO C++17 Standard (/std:c++17)
	ISO C++20 Standard (/std:c++20)
	Preview - Features from the Latest C++ Working Draft (/std:c++latest)
	<inherit from parent or project defaults>

Nastavení verze jazyka

Warningy pomáhají předcházet chybám



Na co si dávat pozor při sestavování

- **C++ standard** (většina compilátorů nedefaultuje na C++20 ani C++23)
- Nejlepší tři compilery (pokud jde o pokrytí C++ featur)
 - MSVC 19.28+ (ověřit v *Developer PowerShell for VS*: `cl --version`)
 - Součást Visual Studio 2022
 - Kliknout project > Properties > C/C++ > Language > C++ Language Standard => `/std:c++20`
 - Pozor na přepínač Debug/Release (Debug verze může být OPRAVDU pomalejší)
 - GCC 13+ (ověřit: `g++ --version`)
 - Nastavení standardu (to samé clang): `-std=c++20`
 - Clang
- Nastavení warning levelu: `/W4` (msvc) `-Wall -Wextra` (gcc, clang)
- Nastavení optimalizací (VS to řeší za nás; manuálně `/O2` a `-O2`)

CheatSheet pro nastavení Visual Studio

- **Set the C++ language standard version.**
 - Right click project > Properties > C/C++ > Language > C++ Language Standard => /std:c++20 or latest.
 - Equivalent of ``-std=c++20`` in GCC/Clang.
- **Add command line args to your debugged program.**
 - Right click project > Properties > Debugging > Command Arguments => ``-t -v --some=cool``.
- **Set up a warning level.**
 - Right click project > Properties > C/C++ > General > Warning Level => W4.
 - (Somewhat) equivalent of ``-Wall`` in GCC/Clang.
- **Add additional include directories.**
 - Right click project > Properties > C/C++ > General > Additional Include Directories => Add the desired dirs.
 - Equivalent of ``-I <some dir>`` in GCC/Clang.
- **Add additional directories where to look for libs for linking.**
 - Right click project > Properties > Linker > General > Additional Library Directories=> Add the desired libs.
 - Equivalent of ``-L /opt/libdir/`` in GCC/Clang.
- **Add additional libraries to link with.**
 - Right click project > Properties > Linker > Input > Additional Dependencies => e.g. somelib.lib.
 - Equivalent of ``-l somelib`` in GCC/Clang.

Clone a repository (ne na školních počítačích)

Clone a repository

Enter a Git repository URL

Repository location

`https://gitlab.mff.cuni.cz/teaching/nprg041/2023-24/klepl/kleplj.git`

Path

`E:\source\kleplj`

✓ Your new project access token

.....

Connect to GitLab

GitLab

Sign in

`https://gitlab.mff.cuni.cz/`

Token Password

Username or email (optional)

.....

Sign in

Add a project access token

Token name

Token name

For example, the application using the token or the purpose

Expiration date

YYYY-MM-DD

Select a role

Developer

Grants read access (pull) to the repository.

☒ write_repository
Grants read and write access (pull and push) to the repository.

☐ ai_features
Grants access to GitLab Duo related API endpoints

Create project access token Cancel

Project

K kleplj

Pinned

Merge requests 0

Manage

Plan

Code

Build

Secure

Deploy

Operate

Monitor

Analyze

Settings

General

Integrations

Webhooks

Access tokens

Sestavení programu (build) a čtení outputu

```
#include "hello_functions.hpp"

#include <iostream>
#include <iomanip>
#include <stdexcept>

int main(int argc, char * * argv)
{
    try {
        int i = "hello";
        say_hello();
    }
    catch (const std::exception & e)
    {
        std::cout << e.what() << std::endl;
    }
    catch (...)
    {
        std::cout << "Unknown exception." << std::endl;
    }
    return 0; // return code 0 required for recodex success
}
```

Program (s chybou)

Release konfigurace (optimalizuje kód)

Release x64 Local Windows Debugger



Spuštění sestavení (buildění)

Build Debug Test Analyze Tools Extensions Window Help
Build Solution Ctrl+Shift+B



Error

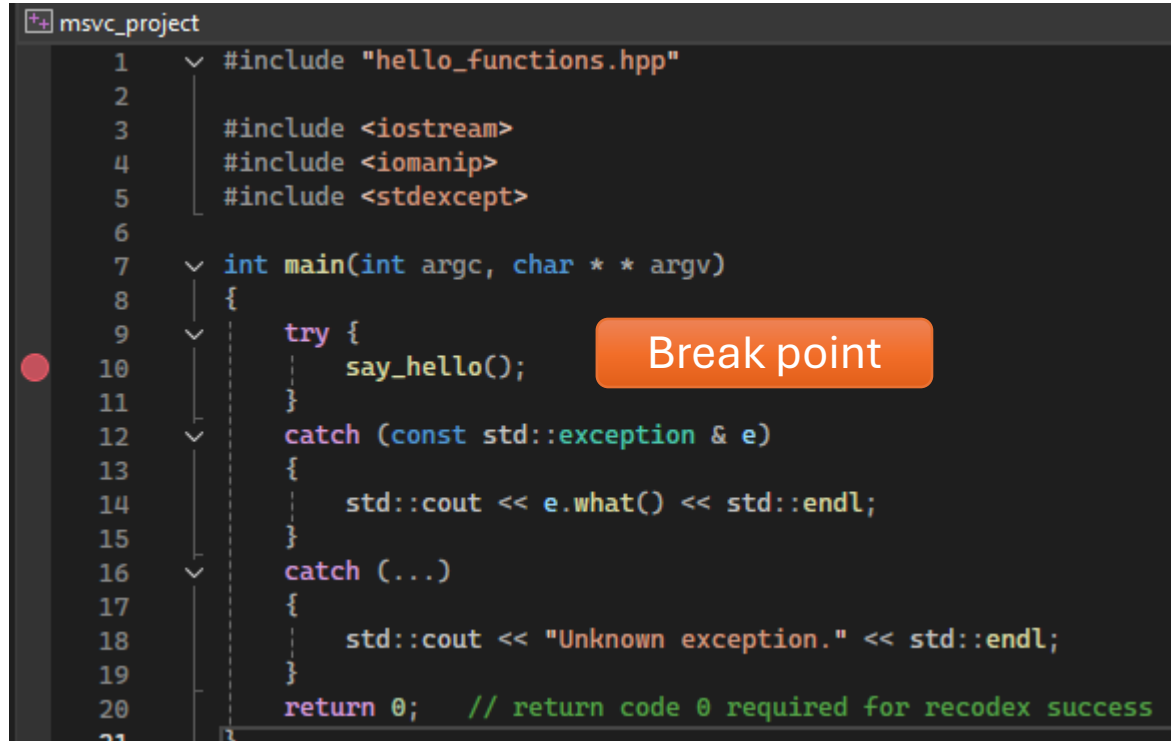
Output

Show output from: Build

```
2>hello_main.cpp
2>C:\Users\Jirka\source\kleplj-nprg041\labs\msvc_project\hello_main.cpp(10,15): error C2440: 'initializing': cannot convert from 'const char [6]' to 'int'
2> C:\Users\Jirka\source\kleplj-nprg041\labs\msvc_project\hello_main.cpp(10,15):
2> There is no context in which this conversion is possible
2>Done building project "msvc_project.vcxproj" -- FAILED.
1>Generating code
```

Error List Developer PowerShell Developer PowerShell Developer Powershell 7 Output Find Symbol Results

Debugger

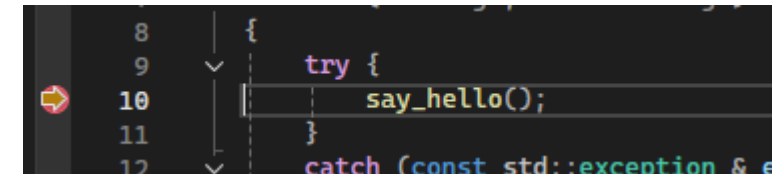


```
msvc_project
1  #include "hello_functions.hpp"
2
3  #include <iostream>
4  #include <iomanip>
5  #include <stdexcept>
6
7  int main(int argc, char * * argv)
8  {
9      try {
10         say_hello();
11     }
12     catch (const std::exception & e)
13     {
14         std::cout << e.what() << std::endl;
15     }
16     catch (...)
17     {
18         std::cout << "Unknown exception." << std::endl;
19     }
20     return 0; // return code 0 required for recodex success
21 }
```

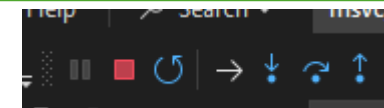
Spuštění Debug konfigurace (F5)



Zastavený program



Pauza, stop, restart | Krokování



Příklad: násobilka - příprava

The screenshot shows the Visual Studio 'Create a new project' dialog. On the left, under 'Recent project templates', 'Console App' and 'Empty Project' are listed. The main area shows a list of templates: 'Empty Project', 'Console App', 'CMake Project', 'Windows Desktop Wizard', 'Windows Desktop Application', and 'Dynamic-Link Library (DLL)'. The 'Console App' template is selected. On the right, the configuration for the 'Console App' is shown. The 'Project name' is 'nasobilka'. The 'Location' is 'C:\Users\Jirka\source\kleplj-nprg041\labs\'. The 'Solution' is 'Create new solution'. The 'Solution name' is 'nasobilka'. The checkbox 'Place solution and project in the same directory' is checked. At the bottom, it says 'Project will be created in "C:\Users\Jirka\source\kleplj-nprg041\labs\nasobilka\"'. There are two green callout boxes: one pointing to the 'Location' field with the text 'Cesta: REPOZITÁŘ/labs' and another pointing to the 'Place solution and project in the same directory' checkbox with the text 'Projekt společně se solutionem'. The path 'C:\Users\Jirka\source\kleplj-nprg041\labs\nasobilka\' is circled in red.

Create a new project

Recent project templates

- Console App
- Empty Project

Search for templates (Alt+S)

C++ Windows All project types

Empty Project
Start from scratch with C++ for Windows. Provides no starting files.
C++ Windows Console

Console App
Run code in a Windows terminal. Prints "Hello World" by default.
C++ Windows Console

CMake Project
Build modern, cross-platform C++ apps that don't depend on .sln o
C++ Windows Linux Console

Windows Desktop Wizard
Create your own Windows app using a wizard.
C++ Windows Desktop Console Library

Windows Desktop Application
A project for an application with a graphical user interface that runs
C++ Windows Desktop

Dynamic-Link Library (DLL)
Build a .dll that can be shared between multiple running Windows ap
C++ Windows Library

Console App C++ Windows Console

Project name
nasobilka

Location
C:\Users\Jirka\source\kleplj-nprg041\labs\

Solution
Create new solution

Solution name
nasobilka

☒ Place solution and project in the same directory

Project will be created in "C:\Users\Jirka\source\kleplj-nprg041\labs\nasobilka\'

Cesta: REPOZITÁŘ/labs

Projekt společně se solutionem

Hello World

```
#include <iostream>

int main(int argc, char* argv[]) {
    // argv[0] is the program name
    // argv[1] (optional) is name of the person
    if (argc == 2) { // one argument
        std::cout << "Hello, " << argv[1] << "!" << std::endl;
    } else { // no arguments
        std::cout << "Hello, World!" << std::endl;
    }
    return 0; // success
}
```

Hello World

```
#include <iostream>

int main(int argc, char* argv[]) {
    // argv[0] is the program name
    // argv[1] (optional) is name of the person
    if (argc == 2) { // one argument
        std::cout << "Hello, " << argv[1] << "!" << std::endl;
    } else { // no arguments
        std::cout << "Hello, World!" << std::endl;
    }
    return 0; // success
}
```

Počet argumentů + 1

{prg. name, arg1, arg2, ...}

C++ indexuje od 0

Výstup do stdout

Hello World

```
#include <iostream>

int main(int argc, char* argv[]) {
    // argv[0] is the program name
    // argv[1] (optional) is name of the person
    if (argc == 2) { // one argument
        std::cout << "Hello, " << argv[1] << "!" << std::endl;
    } else { // no arguments
        std::cout << "Hello, World!" << std::endl;
    }
    return 0; // success
}
```

Počet argumentů + 1

{prg. name, arg1, arg2, ...}

C++ indexuje od 0

Výstup do stdout

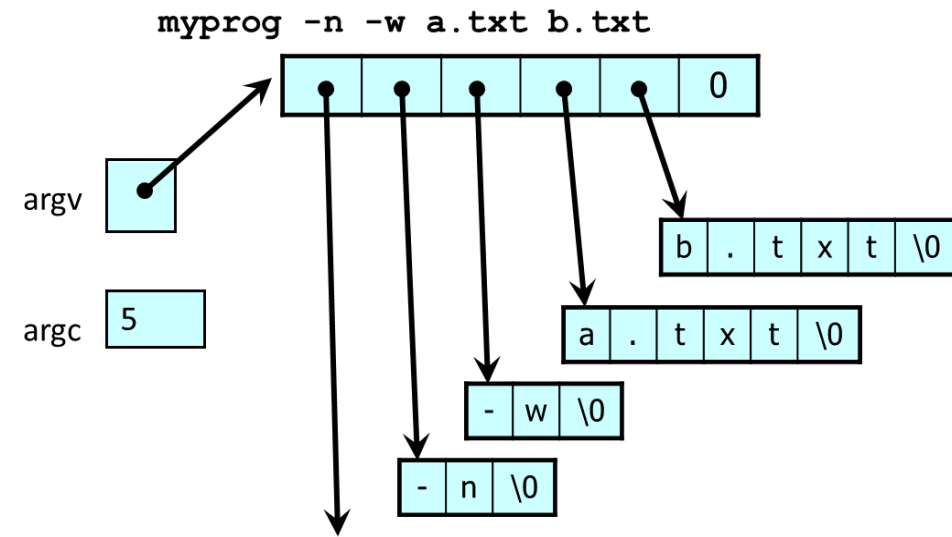
```
std::cout << std::format("Hello, {:20}!", argv[1]) << std::endl;
```

Vytvoří string s formátovaným textem

Vytisknout název ve sloupci o 20 znacích

Funkce main

- Volaná při startu programu, **nikdy jindy**
- Vrací integer
 - **0** značí úspěch
- Pokud má argumenty, tak jsou následující:
 - **int argc** - počet cmd argumentů (včetně názvu programu)
 - **char *argv[]** - null-terminated pole pointerů na jednotlivé cmd argumenty
 - **argv[0]** je název programu
 - Argumenty reprezentovány **C-stringy** (pole bajtů zakončeny 0)



Příklad: násobilka - zadání

(kostra programu je na stránce cvičení)

1. Vypište násobilku všech čísel z cmd parametrů

- Výstup podle ukázky
- Všechny čísla hezky ve sloupcích

2. Rozšíření: parametry

- Určují rozsah násobilky:
 - -f N ← from (default: N = 1)
 - -t N ← to (default: N = 10)
 - V libovolném pořadí, ale vždy před čísla

3. Udělat třídu `MulTable`

- Na dalším slidu je její náčrt

./nasobilka 7				
1	*	7	=	7
2	*	7	=	14
...				
10	*	7	=	70

```
int main(int argc, char* argv[]) {  
    // Pohodlnější zpracování argumentů:  
    // args = {argv[1], argv[2], ..., argv[argc-1]}  
    std::vector<std::string> args(argv + 1, argv + argc);  
    int from = 1;  
    int to = 10;  
    // TODO: podpora `-f FROM` a `-t TO`  
  
    for (size_t i = 0; i < args.size(); i++) {  
        int number = std::stoi(args[i]);  
        print_mul_table(number, from, to);  
    }  
}
```

MulTable

```
class MulTable {  
public:  
    MulTable() {}  
    MulTable(int f, int t) : from_(f), to_(t) {}  
  
    void print(int n);  
private:  
    int from_ = 1;  
    int to_ = 10;  
};  
  
void MulTable::print(int n) {  
    // TODO  
}
```

MulTable

```
class MulTable {  
public:  
    MulTable() {}  
    MulTable(int f, int t) : from_(f), to_(t) {}  
  
    void print(int n);  
private:  
    int from_ = 1;  
    int to_ = 10;  
};  
  
void MulTable::print(int n) {  
    // TODO  
}
```

Defaultní konstruktor

Inicializace datových položek

Defaultní hodnoty datových položek

(složitá) metoda napsaná mimo definici třídy

MulTable

Většinou v .hpp souboru

```
class MulTable {  
public:  
    MulTable() {}  
    MulTable(int f, int t) : from_(f), to_(t) {}  
  
    void print(int n);  
private:  
    int from_ = 1;  
    int to_ = 10;  
};
```

Defaultní konstruktor

Inicializace datových položek

Defaultní hodnoty datových položek

Většinou v .cpp souboru

```
void MulTable::print(int n) {  
    // TODO  
}
```

(složitá) metoda napsaná mimo definici třídy

Useful nástroje a zdroje informací

- **C++ reference:** <https://en.cppreference.com/w/>
 - Standard C++ převyprávěný pro programátory
- Clang nástroje
 - <https://clang.llvm.org/extra/clang-tidy/>
 - **cmake-format**: automatické formatování zdrojového kódu
 - **clang-tidy**: linter pro C++, označuje ne úplně jasné chyby
- **CMake** build systém
 - <https://cmake.org/>
 - Budeme používat na zápočtové programy
- **vcpkg**
 - Manager dependencí pro C++ (multiplatformní)
 - Můžete používat v zápočtových programech