



Software engineering

v2024

Lab 1: Programming with Modern C++ – Part I

Contents

<i>Exercise 1: Creating Projects using VS 2022.....</i>	<i>2</i>
<i>Exercise 2: Debugging in VS 2022.....</i>	<i>9</i>

Exercise 1: Creating Projects using VS 2022

In this exercise you will start an IDE (*Integrated Development Environment*) named the MS Visual Studio 2022 and using that tool you will create a *solution* and a *project* containing the C++ code.

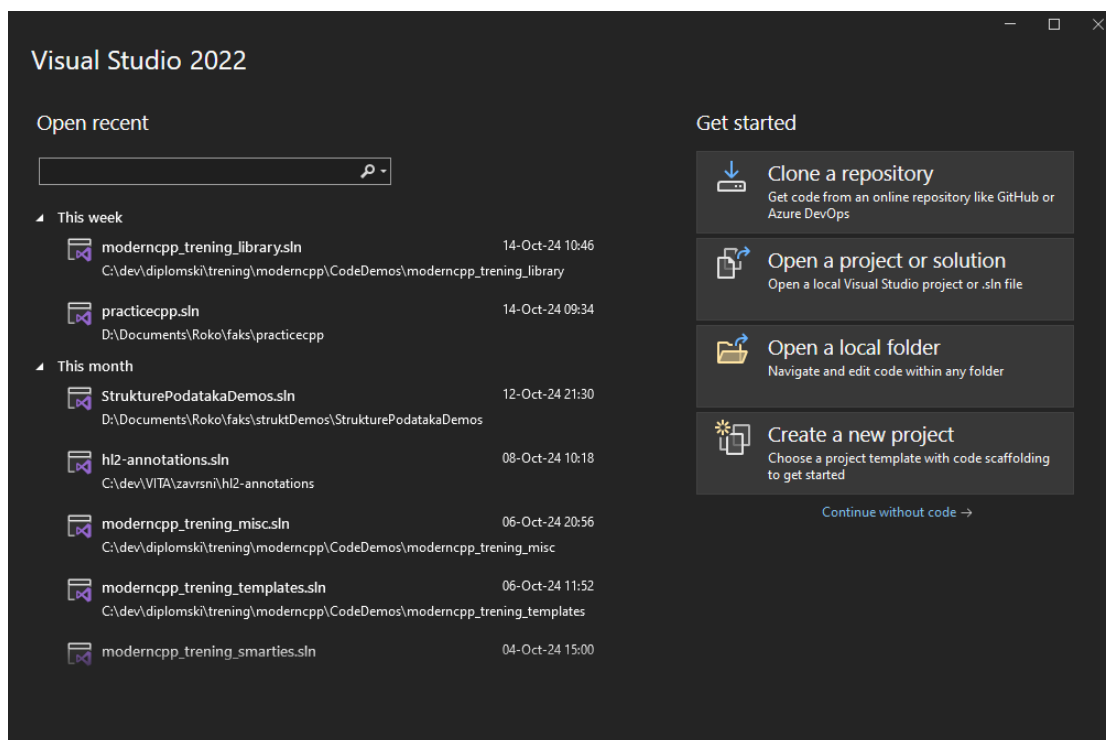
The VS 2022 family provides several project types. You will create a C++ *console project* which is compiled into an executable (an **.exe** file).

Related projects can be grouped into **solutions**, but you can hold each **project** in its own solution.

The **Solution Explorer** allows you to navigate between projects in a solution and see the individual files.

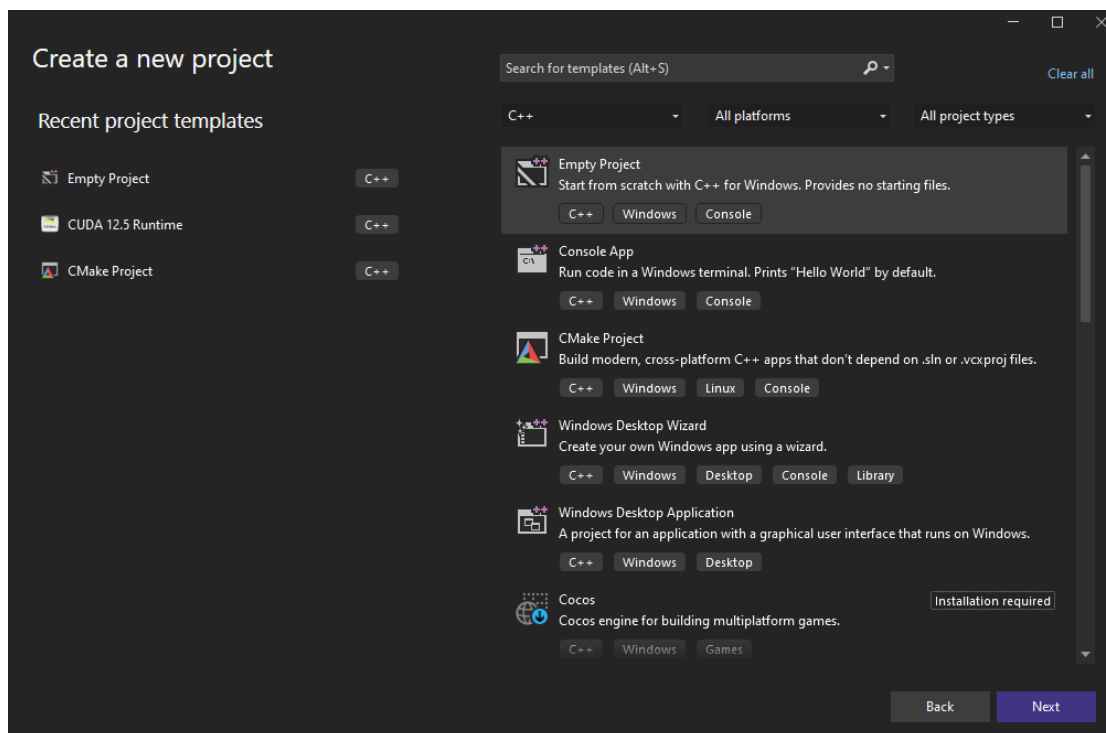
Start the VS 2022 and create project

1. Start Microsoft Visual Studio 2022.
2. When Visual Studio is running, you will see window like this:

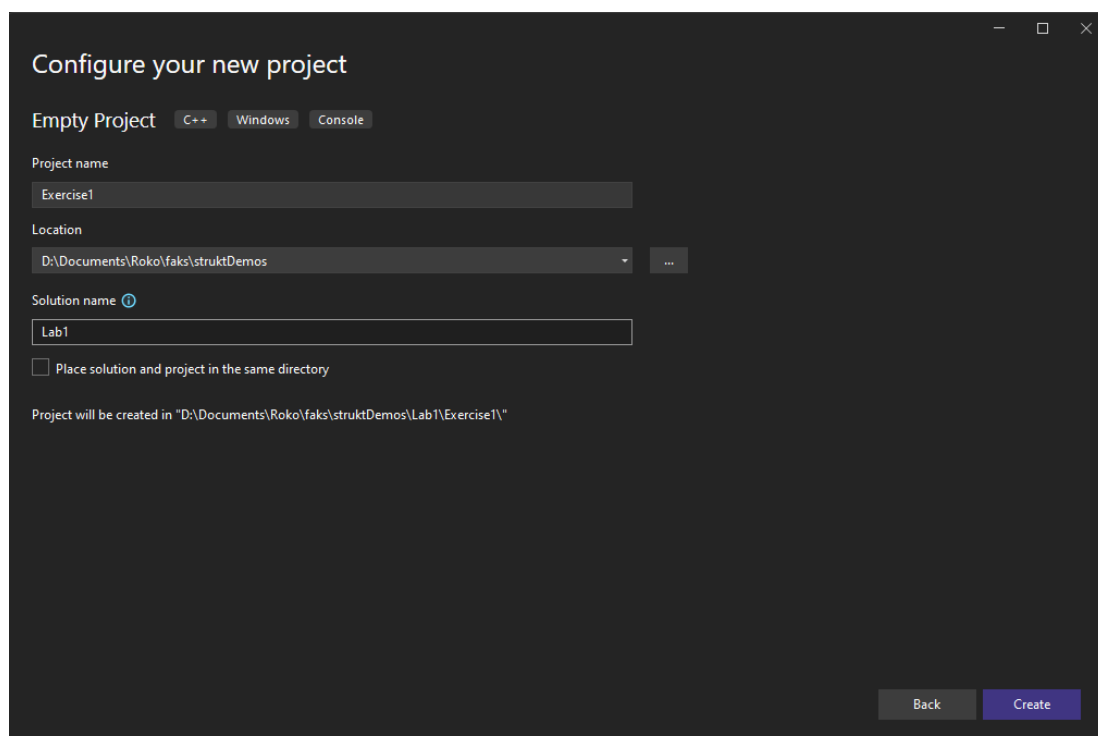


3. Select **Create a new project**

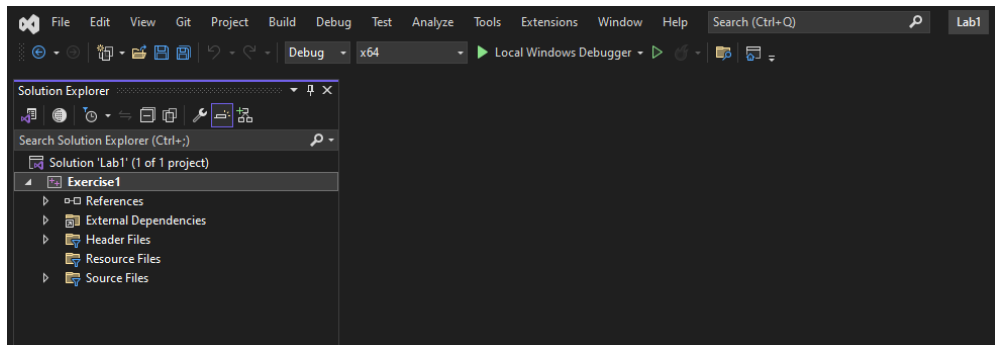
4. In the **New Project** dialog, select the **Empty Project** template which has the **C++**, **Windows** and **Console** tags, as shown below:



5. Enter information about the project (1) the **project name** (2) the **project location** (3) the **solution name**, as shown below:



- Click **Create** and the newly created empty project will appear in the **Solution Explorer**, as shown below:

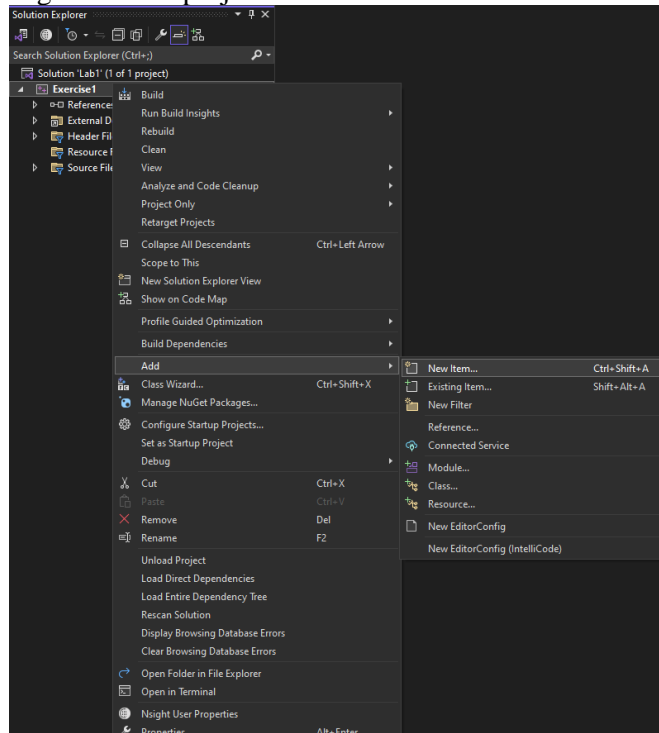


NOTE: Solution Explorer by default is on right side of Visual Studio editor

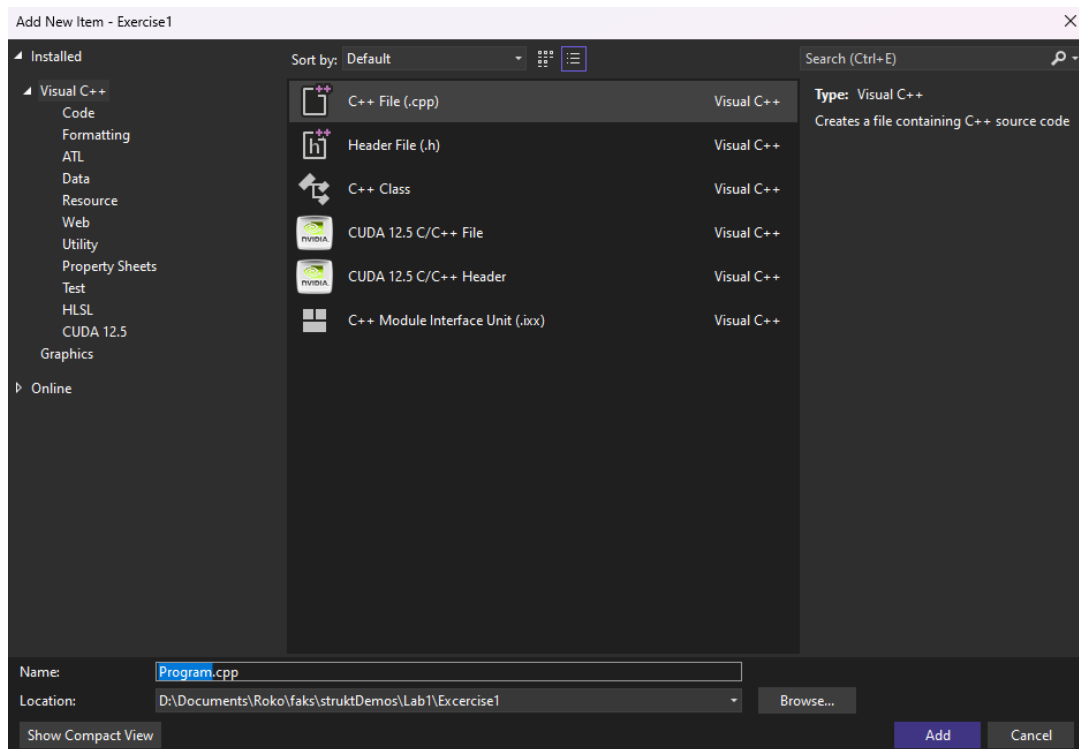
- If you close **Solution Explorer**, you can open it by clicking on **View > Solution Explorer** (or via shortcut: **Ctrl+Alt+L**).

Add C++ file to project

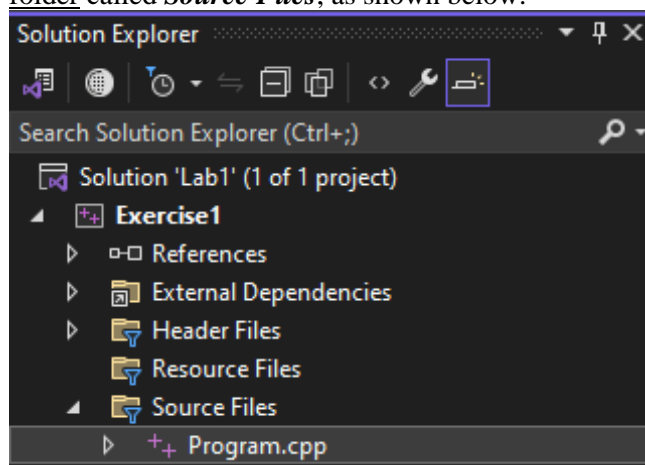
- Right click on project name *Exercise1* then **Add > New Item...**



- Now it will pop up wizard dialog for adding items, select **C++ File (.cpp)** then in **Name** section name item **Program.cpp**, as shown below:



- Add** then **Program.cpp** will shown in **Solution Explorer** under project in virtual folder called **Source Files**, as shown below:



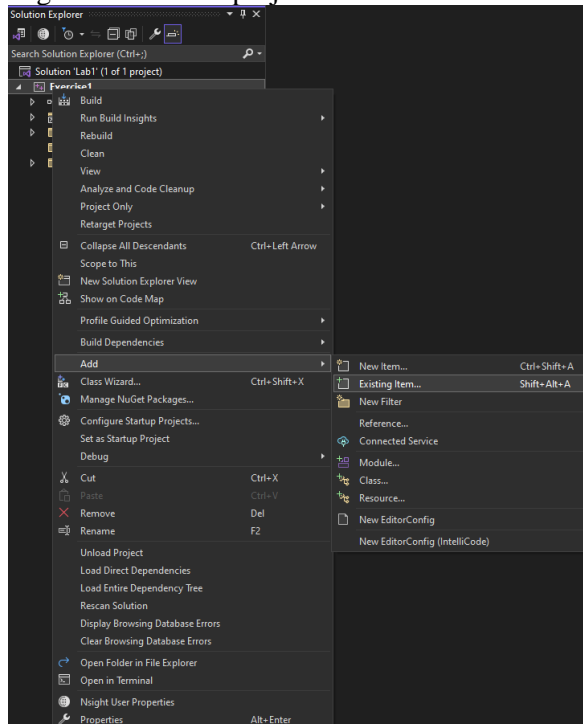
- Double click on Program.cpp for opening it, now you can start editing and writing code.
- For adding other files such header (.h) process is same, only you need select corresponding extension in 2. step.
- To delete file right click on file then **Remove > Delete**.

Add existing C++ files to project

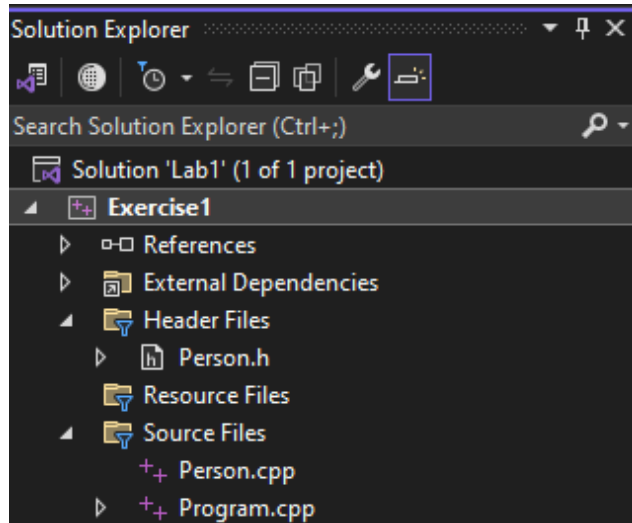
For these labs, you will receive source code for each exercise. Source code will be in the **src** folder, separate from the Visual Studio solution. For each lab you will create new solution, and for each exercise you will add new project to the lab's solution. Instructions for creating a new solution can be found in *Start the VS 2022 and create*

project section, and adding new project is explained in *Add new project to solution* section.

1. Right click on project name *Exercise1* then **Add > Existing Item...**



2. Then browse to the location of **src** folder and select all the files inside the folder.
3. Click **Add**, then all files will appear in **Solutions Explorer**, as shown below:

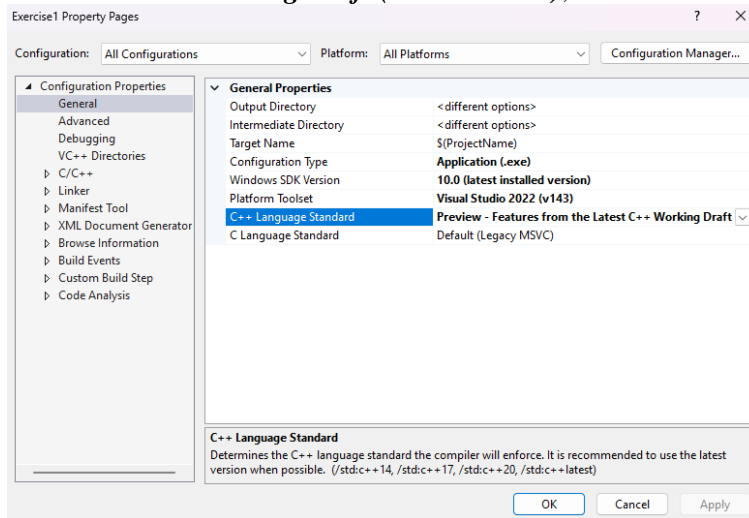


Configure project to latest C++ standard

In this labs, we will use latest features of C++ available in Visual C++ compiler. Latest standard is C++23 (but not all features are implemented in VC++)

1. Right click on name of project *Exercise1* then click on *Properties*.

- Under **General** tab, select **C++ Language Standard** to *Preview – Features from the Latest C++ Working Draft (/std:c++latest)*, as shown below:



- Click *Ok*.

Build and run project

- Go to **Debug > Start Without Debugging** (or shortcut: Ctrl+F5) – this will build and run project, which is console app:

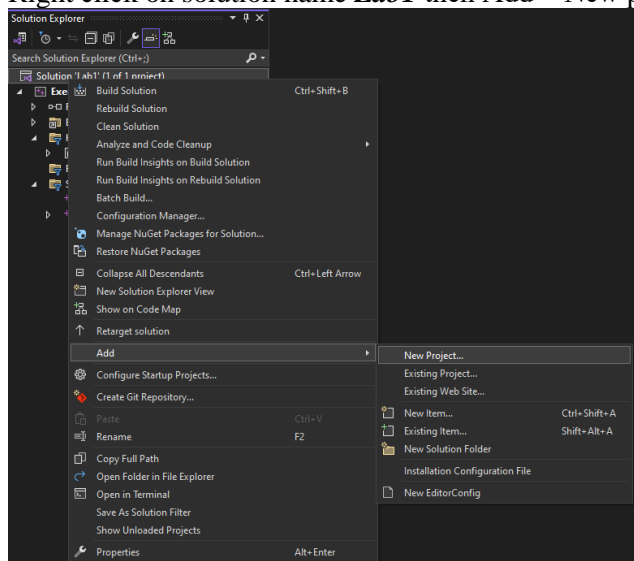
```

Microsoft Visual Studio Debug Console
Constructor (Ivo) called
Program started
Constructor (Ana) called
Constructor (Mate) called
Ana Anic is born in 1990
Mate Matic is born in 2001
Number: 5
Program ended
Destructor (Mate) called
Destructor (Ana) called
Destructor (Ivo) called

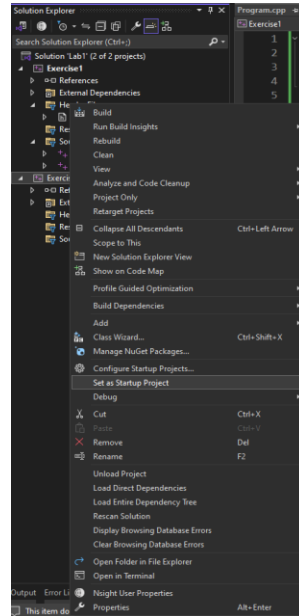
D:\Documents\Roko\faks\struktDemos\Lab1\64\Debug\Exercise1.exe (process 13676) exited with code 0 (0x0).
Press any key to close this window . . .
    
```

Add new project to the solution

- Right click on solution name *Lab1* then Add > New project...



2. In the **New Project** dialog, select the **Empty Project** template which has the **C++**, **Windows** and **Console** tags, then **Next**.
3. Set project name to *Exercise2*.
4. Click **Create**.
5. Add files of exercise 2 and configure latest C++ standard for project.
6. To run this project you need setup it as startup project, right click on project name *Exercise2* > **Set as Startup Project**



Exercise 2: Debugging in VS 2022

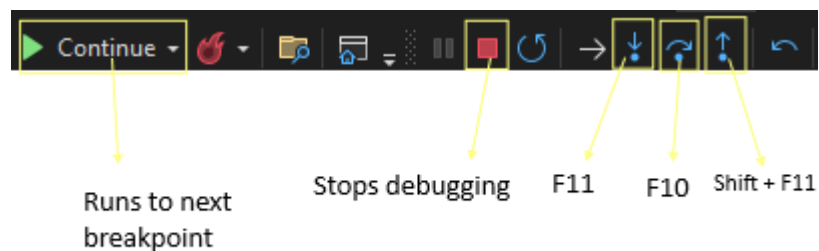
Debugging is the process of identifying and resolving errors or bugs in your code. While writing and running code may seem straightforward, software often contains hidden issues that can cause unexpected behavior. Debugging allows you to find and fix these issues efficiently.

In VS 2022, with debugging tools you can:

- **Pause the program** at any point using breakpoints
- **Step through the code** line by line to see how each instruction is executed
- **Inspect variables** to monitor how values change during execution of program

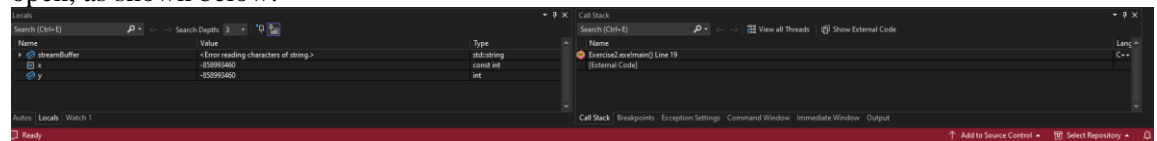
Debug the Project

1. **F9** - sets a breakpoint on selected line
2. **F5** - starts the debugging
3. **F10** – steps over the code
4. **F11** - steps into the code (function)
5. **Shift + F11** – Step out (from function)
6. **Ctrl + F10** – runs to the cursor
7. **Continue** – runs to the next breakpoint, if no other breakpoint runs to the end of program



Watch locals and stack

1. When you start debugging both window of **Locals** and **Call Stack** should be open, as shown below:



2. If they are not open, when you start debugging
 - (1) Locals: **Debug > Windows > Locals**
 - (2) Call Stack: **Debug > Windows > Call Stack**

Exercise 2

1. Setup Exercise 2 project, instructions are under *Add new project to the solution* section in Exercise 1 instructions.
2. Set breakpoints on lines specified at top of main in comments.
3. Follow instructions in code comments.