

# CE103 Algorithms and Programming I

## Development Environments and Algorithm Basics

Author: Asst. Prof. Dr. Uğur CORUH

## Contents

0.1	CE103 Algorithms and Programming I . . . . .	3
0.2	Week-2 . . . . .	3
0.3	Algorithm Basics . . . . .	3
0.3.1	Flowgorithm (1) . . . . .	3
0.3.2	Pseudocode (1) . . . . .	7
0.3.3	Introduction to Analysis of Algorithms . . . . .	7
0.3.4	Programming Environment Setup and Configuration . . . . .	8
0.3.5	C / C++ Environment and Development . . . . .	8
0.3.6	VSCode (Install / Compile / Run / Debug) (1) . . . . .	27
0.3.7	Visual Studio Code Extension List (1) . . . . .	35
0.3.8	Visual Studio Code Extension List (2) . . . . .	36
0.3.9	Visual Studio Code Extension List (3) . . . . .	36
0.3.10	Visual Studio Code Extension List (4) . . . . .	36
0.3.11	Visual Studio Code Extension List (5) . . . . .	37
0.3.12	Visual Studio Code Extension List (6) . . . . .	37
0.3.13	Visual Studio Code Extension List (7) . . . . .	37
0.3.14	Visual Studio Community Edition (Install / Compile / Run / Debug) (1) . . . . .	38
0.3.15	Visual Studio Community Edition (Install / Compile / Run / Debug) (2) . . . . .	38
0.3.16	Visual Studio Community Edition (Install / Compile / Run / Debug) (3) . . . . .	39
0.3.17	Visual Studio Community Edition (Install / Compile / Run / Debug) (4) . . . . .	39
0.3.18	Visual Studio Community Edition (Install / Compile / Run / Debug) (5) . . . . .	40
0.3.19	Visual Studio Community Edition (Install / Compile / Run / Debug) (6) . . . . .	40
0.3.20	Visual Studio Community Edition (Install / Compile / Run / Debug) (7) . . . . .	41
0.3.21	Visual Studio Community Edition (Install / Compile / Run / Debug) (8) . . . . .	42
0.3.22	Visual Studio Community Edition (Install / Compile / Run / Debug) (9) . . . . .	43
0.3.23	Visual Studio Community Edition (Install / Compile / Run / Debug) (10) . . . . .	43
0.3.24	Visual Studio Community Edition (Install / Compile / Run / Debug) (11) . . . . .	44
0.3.25	Visual Studio Community Edition (Install / Compile / Run / Debug) (12) . . . . .	44
0.3.26	Visual Studio Community Edition (Install / Compile / Run / Debug) (13) . . . . .	44
0.3.27	Visual Studio Community Edition (Install / Compile / Run / Debug) (14) . . . . .	45
0.3.28	Visual Studio Community Edition (Install / Compile / Run / Debug) (15) . . . . .	45
0.3.29	Visual Studio Community Edition (Install / Compile / Run / Debug) (16) . . . . .	46
0.3.30	Visual Studio Community Edition (Install / Compile / Run / Debug) (17) . . . . .	46
0.3.31	Visual Studio Community Edition (Install / Compile / Run / Debug) (18) . . . . .	47
0.3.32	Visual Studio Community Edition (Install / Compile / Run / Debug) (19) . . . . .	48
0.3.33	Visual Studio Community Edition (Install / Compile / Run / Debug) (20) . . . . .	49
0.3.34	Visual Studio Community Edition (Install / Compile / Run / Debug) (21) . . . . .	50
0.3.35	Visual Studio Community Edition (Install / Compile / Run / Debug) (22) . . . . .	52
0.3.36	Visual Studio Community Edition (Install / Compile / Run / Debug) (23) . . . . .	54
0.3.37	MSYS2 . . . . .	61
1	JAVA Environment and Development . . . . .	112
1.0.1	JDK and JRE Setup (1) . . . . .	113

1.0.2	JDK and JRE Setup (2) . . . . .	113
1.0.3	System Environments and Paths for Java (1) . . . . .	115
1.0.4	System Environments and Paths for Java (2) . . . . .	115
1.0.5	System Environments and Paths for Java (3) . . . . .	116
1.0.6	Netbeans (Java) (1) . . . . .	116
1.0.7	Netbeans (Java) (2) . . . . .	117
1.0.8	Netbeans (Java) (3) . . . . .	118
1.0.9	Netbeans (Java) (4) . . . . .	119
1.0.10	Netbeans (Java) (5) . . . . .	119
1.0.11	Netbeans (Java) (6) . . . . .	120
1.0.12	Netbeans (Java) (7) . . . . .	121
1.0.13	Netbeans (Java) (8) . . . . .	121
1.0.14	Netbeans (Java) (9) . . . . .	122
1.0.15	Netbeans (Java) (10) . . . . .	122
1.0.16	Eclipse (Java) (1) . . . . .	122
1.0.17	Eclipse (Java) (2) . . . . .	124
1.0.18	Eclipse (Java) (3) . . . . .	125
1.0.19	Eclipse (Java) (4) . . . . .	126
1.0.20	Eclipse (Java) (5) . . . . .	127
1.0.21	Eclipse (Java) (6) . . . . .	127
1.0.22	Eclipse (Java) (7) . . . . .	128
1.0.23	Eclipse (Java) (8) . . . . .	129
1.0.24	IntelliJ Idea (JetBrains) (Java) . . . . .	129
1.0.25	VSCODE (Java) . . . . .	130
1.0.26	Notepad++ (Java) . . . . .	131
1.0.27	Cmake (Java) . . . . .	131
<b>2</b>	<b>C# Environment and Development</b> . . . . .	<b>131</b>
2.0.1	Visual Studio Community Edition (C#) . . . . .	131
2.0.2	Notepad++ (C#) . . . . .	132
2.0.3	Cmake (C#) . . . . .	132
2.0.4	Common Tools and Platforms . . . . .	132
2.0.5	Fatih Kalem . . . . .	132
2.0.6	Notepad++ (Notepad for Source Code) . . . . .	133
2.0.7	HxD (Hex Editor) . . . . .	134
2.0.8	Marktext (Markdown Syntax Editor) . . . . .	135
2.0.9	Cygwin (Linux environment for Windows) . . . . .	136
2.0.10	Dependency Walker (32-bit or 64-bit Windows module dependency checker) . . . . .	137
2.0.11	Doxygen (Code Documentation) . . . . .	138
2.0.12	Sonarlint (Code Quality and Code Security Extension) . . . . .	139
2.0.13	Codepen.io (online code sharing) . . . . .	139
2.0.14	Codepen.io (online code sharing) . . . . .	140
2.0.15	Codeshare.io (real-time code sharing) . . . . .	140
2.0.16	Codebeautify.org (online data conversion tools) . . . . .	141
2.0.17	AsciiFlow.com (ASCII drawing tool) . . . . .	141
2.0.18	Freemind (opensource mindmap application) . . . . .	142
2.0.19	PlantUML (software designer) . . . . .	143
2.0.20	Drawio (drawing tool) . . . . .	143
2.0.21	Putty (Remote Connection) . . . . .	144
2.1	Download file over SSH Protocol . . . . .	148
2.2	Upload file using SSH . . . . .	148
2.2.1	MobaXterm (Remote Connection) . . . . .	149
2.2.2	Teamviewer (Remote Connection) . . . . .	150
2.2.3	AnyDesk . . . . .	151
2.2.4	Paletton.com and Colorhunt.co (Color Chooser) . . . . .	152
2.2.5	Understand (Static Code Analysis) . . . . .	154
2.2.6	JD Project (Java Decomplier) . . . . .	156

2.2.7	Cutter (Multi-Platform Reverse Engineering Tool) . . . . .	157
2.2.8	IDA Pro / Freeware (Native Reverse Engineering Tool) . . . . .	158
2.2.9	IDA Pro / Freeware (Native Reverse Engineering Tool) . . . . .	158
2.2.10	IDA Pro / Freeware (Native Reverse Engineering Tool) . . . . .	159
2.2.11	Code Visualization (Python, C , C++ , Java) . . . . .	159
2.2.12	Assembly of C Code . . . . .	159
2.2.13	Mobile Device Screen Sharing for Demo . . . . .	160
2.2.14	Travis-CI . . . . .	160
2.2.15	AppVeyor . . . . .	160
2.2.16	Jenkins . . . . .	161
2.2.17	Jenkins . . . . .	161
2.2.18	Jenkins . . . . .	162
2.2.19	Vagrant . . . . .	162
2.2.20	Docker / Docker Compose / Kubernetes (1) . . . . .	163
2.2.21	Docker / Docker Compose / Kubernetes (2) . . . . .	163
2.2.22	Docker / Docker Compose / Kubernetes (3) . . . . .	164
2.2.23	Docker / Docker Compose / Kubernetes (4) . . . . .	164
2.2.24	Docker / Docker Compose / Kubernetes (5) . . . . .	165
2.2.25	Docker / Docker Compose / Kubernetes (6) . . . . .	166
2.2.26	NuGet Packages (1) . . . . .	166
2.2.27	NuGet Tools (2) . . . . .	167
2.3	Managing dependencies (3) . . . . .	168
2.4	Tracking references and restoring packages (4) . . . . .	168
2.4.1	SCV Cryptomanager . . . . .	169
2.4.2	Addario CryptoBench . . . . .	169
2.4.3	Raymond's MD5 & SHA Checksum Utility . . . . .	170
2.4.4	SlavaSoft HashCalc . . . . .	171
2.4.5	Portable PGP . . . . .	172
2.4.6	Online Programming Environments . . . . .	173

## List of Figures

## List of Tables

### 0.1 CE103 Algorithms and Programming I

#### 0.2 Week-2

0.2.0.1 Fall Semester, 2021-2022 Download DOC<sup>1</sup>, SLIDE<sup>2</sup>, PPTX<sup>3</sup>

---

### 0.3 Algorithm Basics

---

#### 0.3.1 Flowgorithm (1)

- <http://www.flowgorithm.org/>
- Flowgorithm - Documentation<sup>4</sup>
- <https://github.com/timoteponce/flowgorithm-examples>

---

<sup>1</sup>ce103-week-2-setup.en.md\_doc.pdf

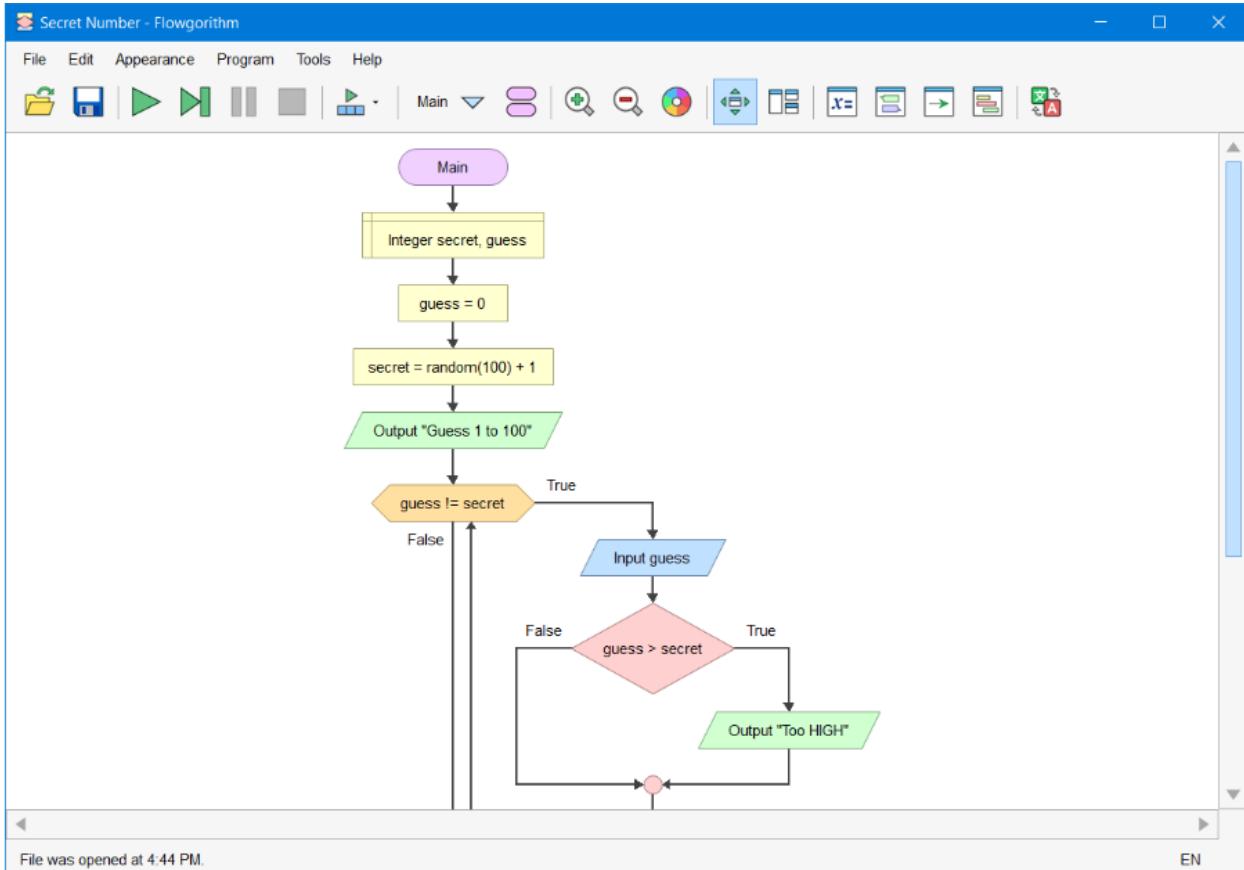
<sup>2</sup>ce103-week-2-setup.en.md\_slide.pdf

<sup>3</sup>ce103-week-2-setup.en.md\_slide.pptx

<sup>4</sup><http://www.flowgorithm.org/documentation/index.html>

### 0.3.1.1 Flowgorithm (2)

- Main Window



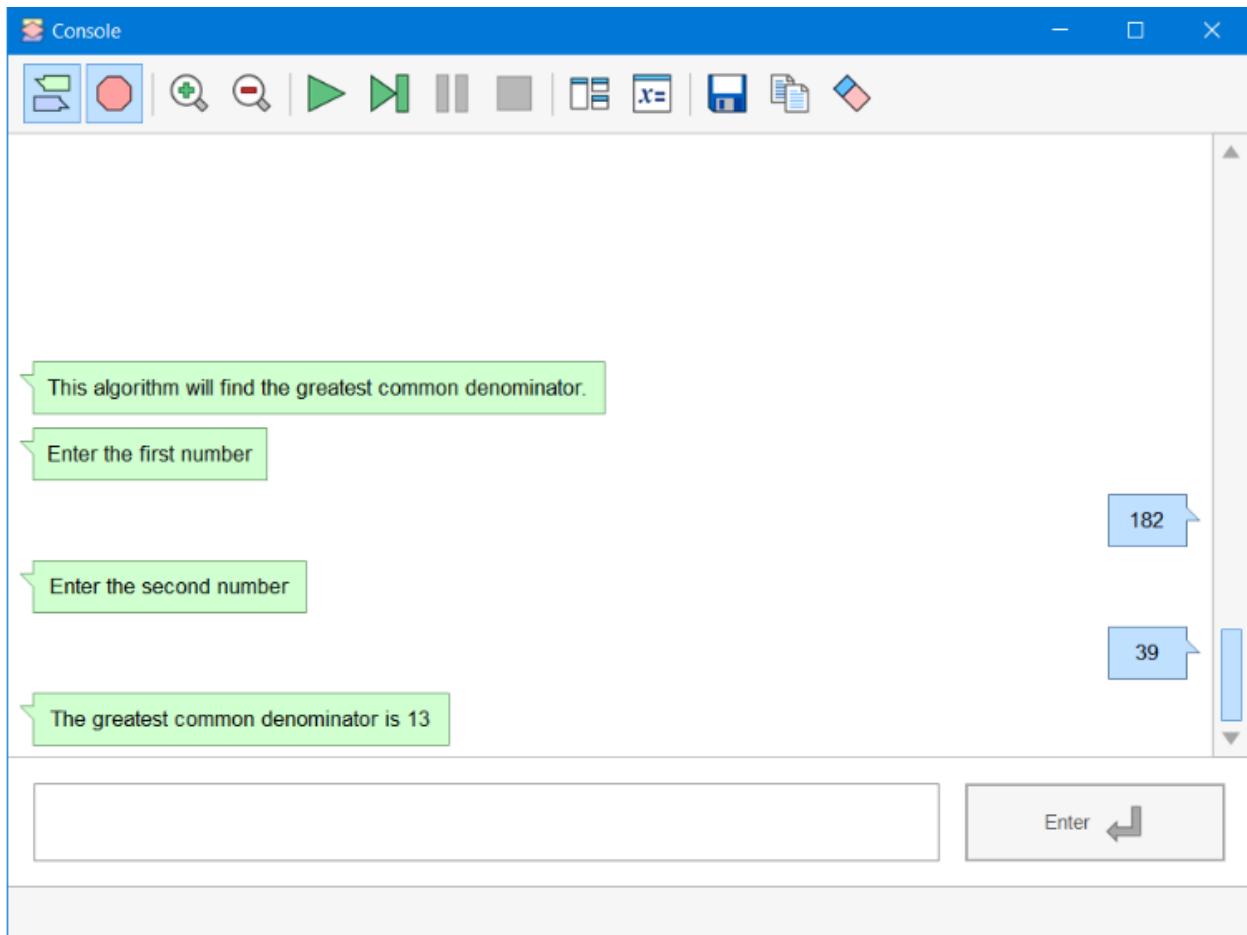
### 0.3.1.2 Flowgorithm (3)

- Console Window

– The classic method to interact with the computer is to use the “Console”. Flowgorithm attempts to make it look like a typical instant messenger window. The “chat bubbles” are color coded to match the Input and Output shapes used in the flowchart. If you don’t want to use the chat bubbles, you can also toggle between them and the classical plain text.

### 0.3.1.3 Flowgorithm (4)

- Console Window



---

#### 0.3.1.4 Flowgorithm (5)

- Source Code Viewer Window
    - The Source Code Viewer can convert your flowchart to several major programming languages. So, if you planning to learn a high-level language, then this feature should help you along the way.
- 

#### 0.3.1.5 Flowgorithm (6)

- Source Code Viewer Window

The screenshot shows a Java source code editor window titled "Source Code Viewer". The toolbar includes icons for Java, file operations, and search. The code editor displays a Java program with syntax highlighting:

```
7  public static void main(String[] args) {  
8      int secret, guess;  
9  
10     guess = 0;  
11     secret = random.nextInt(100) + 1;  
12     System.out.println("Guess 1 to 100");  
13     while (guess != secret) {  
14         guess = input.nextInt();  
15         if (guess > secret) {  
16             System.out.println("Too HIGH");  
17         }  
18         if (guess < secret) {  
19             System.out.println("Too LOW");  
20         }  
21     }  
22     System.out.println("Correct!");  
23 }  
24 }
```

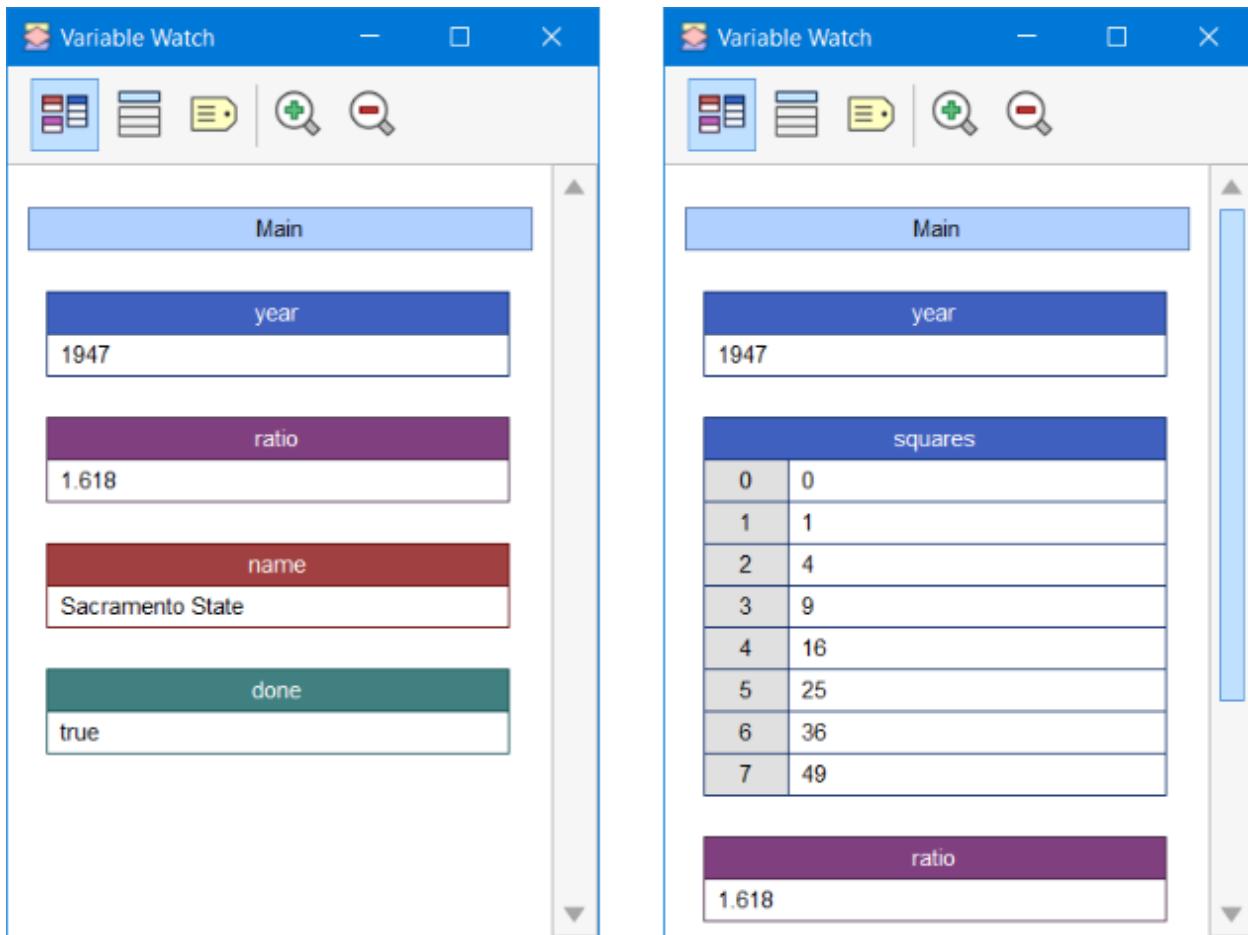
---

### 0.3.1.6 Flowgorithm (7)

- Variable Watch Window
    - The variable watch window is used to keep track of how your variables are changing as your program executes. Each variable is color-coded based on its data type. At a glance, you can tell exactly what type of data is being stored - and catch where you may want to use a different data type.
- 

### 0.3.1.7 Flowgorithm (8)

- Variable Watch Window



### 0.3.1.8 Flowgorithm (8)

- More Resources for Flowgorithm
    - Flowgorithm Tutorial - TestingDocs.com<sup>5</sup>
- 

### 0.3.2 Pseudocode (1)

- Algorithm design language
    - Pseudocode - Wikipedia<sup>6</sup>
    - Pseudocode Examples<sup>7</sup>
    - How to write a Pseudo Code? - GeeksforGeeks<sup>8</sup>
- 

### 0.3.3 Introduction to Analysis of Algorithms

- In this course, we will learn how to code with several development environments and next term we will see an analysis of algorithms in detail.
- This topic is covered in the following link:

<sup>5</sup><https://www.testingdocs.com/flowgorithm-flowchart-tutorial/>

<sup>6</sup><https://en.wikipedia.org/wiki/Pseudocode>

<sup>7</sup><https://www.unf.edu/~broggio/cop2221/2221pseu.htm>

<sup>8</sup><https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/>

- CE100 Introduction to Analysis of Algorithms<sup>9</sup>
- 

### 0.3.4 Programming Environment Setup and Configuration

- Programming life is not about only learning how to code. Mostly you need to use several code development environments and you need to learn how to use them efficiently.
- 

### 0.3.5 C / C++ Environment and Development

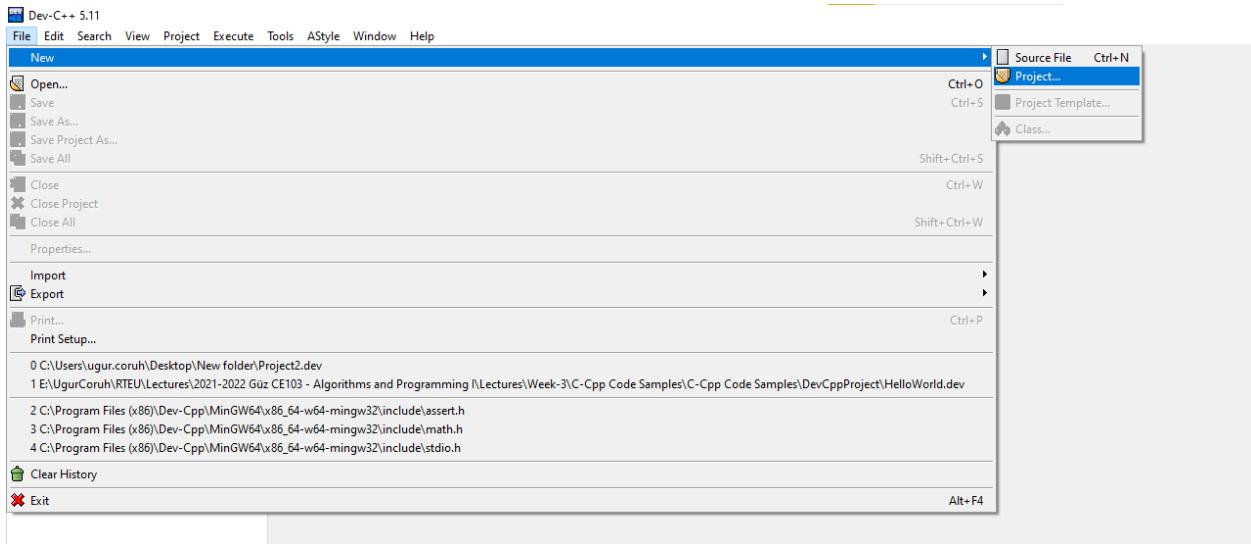
---

#### 0.3.5.1 DevCpp (Install / Compile / Run / Debug) (1)

- Download DevC++ IDE from the following link
    - <https://www.bloodshed.net/>
- 

##### 0.3.5.1.1 DevCpp (Install / Compile / Run / Debug) (2)

- Open DevC++ IDE for C Project Generation Open File->New->Project
- 



---

0.3.5.1.2 DevCpp (Install / Compile / Run / Debug) (3) Select **Console Application** from **Basic** tab and with **C Project** Option and write a project name such as “Hello” then press OK

Select a folder and save **Hello.dev** project file.

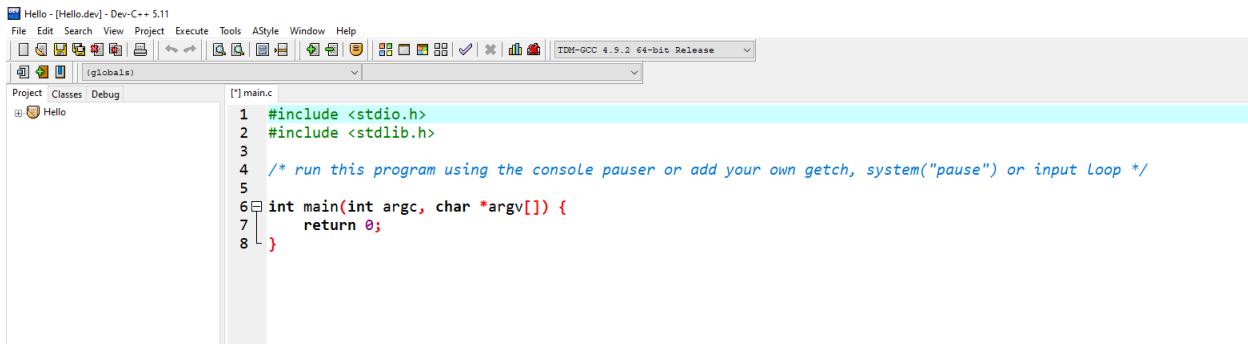
---

##### 0.3.5.1.3 DevCpp (Install / Compile / Run / Debug) (4)

- You will see a sample main with an empty body

---

<sup>9</sup><https://ucoruh.github.io/ce100-algorithms-and-programming-II/week-1/ce100-week-1-intro/>



#### 0.3.5.1.4 DevCpp (Install / Compile / Run / Debug) (5)

```
#include <stdio.h>
#include <stdlib.h>

/* run this program using the console pauser or add your own getch, system("pause") or input loop */
int main(int argc, char *argv[]) {
    retAdd 0;
}
```

#### 0.3.5.1.5 DevCpp (Install / Compile / Run / Debug) (6)

- Add the following line in the main function. This will write “Hello, World!” on the screen and then wait for a keypress to exit from the application

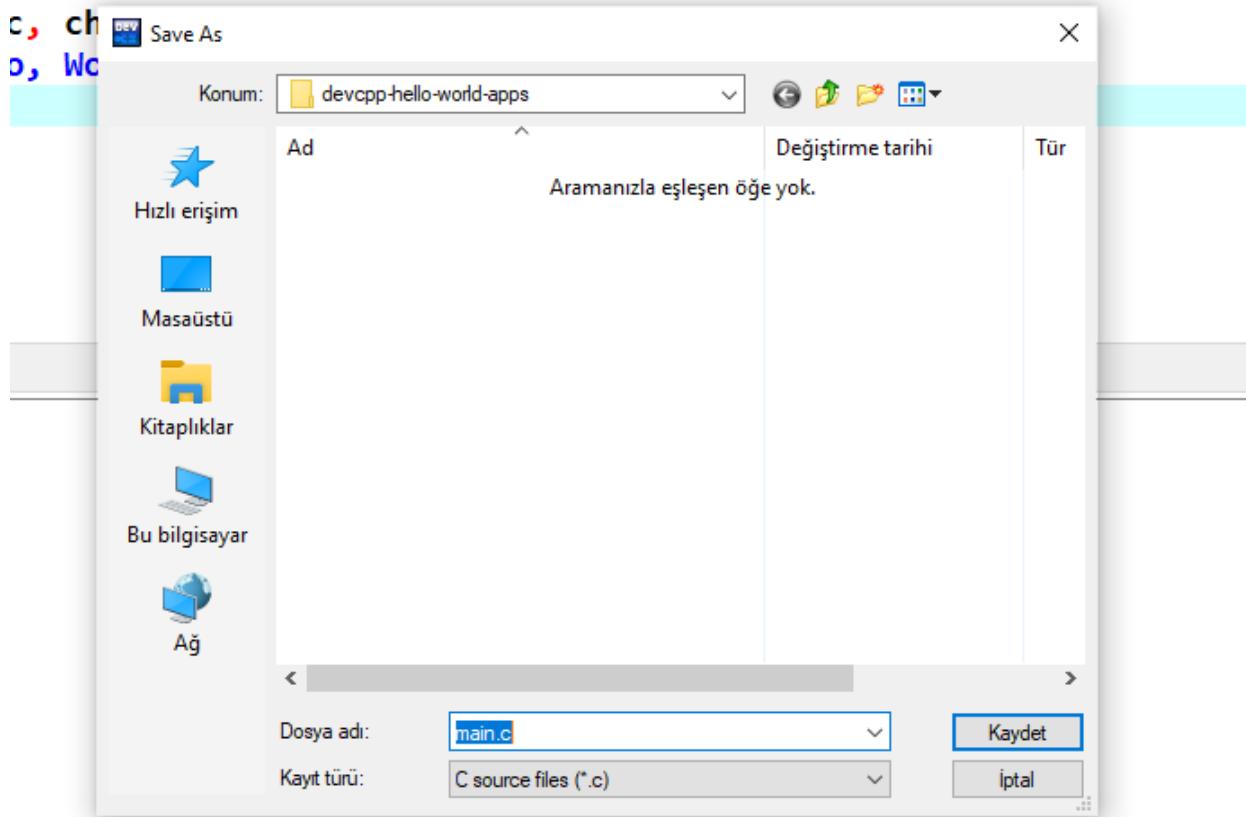
```
#include <stdio.h>
#include <stdlib.h>

/* run this program using the console pauser or add your own getch, system("pause") or input loop */

int main(int argc, char *argv[]) {
    printf("Hello, World!");
    getchar();
    return 0;
}
```

#### 0.3.5.1.6 DevCpp (Install / Compile / Run / Debug) (7)

- Then save the file



### 0.3.5.1.7 DevCpp (Install / Compile / Run / Debug) (8)

- Use from menu Execute->Compile F5 to generate Hello.exe

The screenshot shows the Dev-C++ IDE interface. The menu bar includes File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, and Help. The toolbar has icons for Open, Save, Run, Compile, and others. The project tree on the left shows a single project named 'Hello' with files 'Hello.cpp' and 'global.h'. The main code editor window contains the following C code:

```
stdio.h>
stdlib.h>

/* program using the console pauser or add your own getch, system("pause") or input Loop */

int argc, char *argv[] {
    ("Hello, World!");
}

0;
```

The status bar at the bottom displays the command line, build configuration (TDM-GCC 4.9.3 64-bit Release), and compilation results.

---

### 0.3.5.1.8 DevCpp (Install / Compile / Run / Debug) (9)

- You can find the generated Hello.exe path from Compile.log as follow. Check the Output Filename Compiling project changes...
- 

- Project Filename: E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programming I\Lecture

- Compiler Name: TDM-GCC 4.9.2 64-bit Release

Building makefile...

---

- Filename: E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programming I\Lectures\ce103

Processing makefile...

---

- Makefile Processor: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\mingw32-make.exe

- Command: mingw32-make.exe -f "E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programming I\Lecture"

gcc.exe -c main.c -o main.o -I"C:/Program Files (x86)/Dev-Cpp/MinGW64/include" -I"C:/Program Files (x86)/Dev-Cpp/MinGW64/include/c++/v2"

gcc.exe main.o -o Hello.exe -L"C:/Program Files (x86)/Dev-Cpp/MinGW64/lib" -L"C:/Program Files (x86)/Dev-Cpp/MinGW64/lib/c++/v2"

Compilation results...

---

- Errors: 0

- Warnings: 0

- Output Filename: E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programming I\Lecture

- Output Size: 128,103515625 KiB

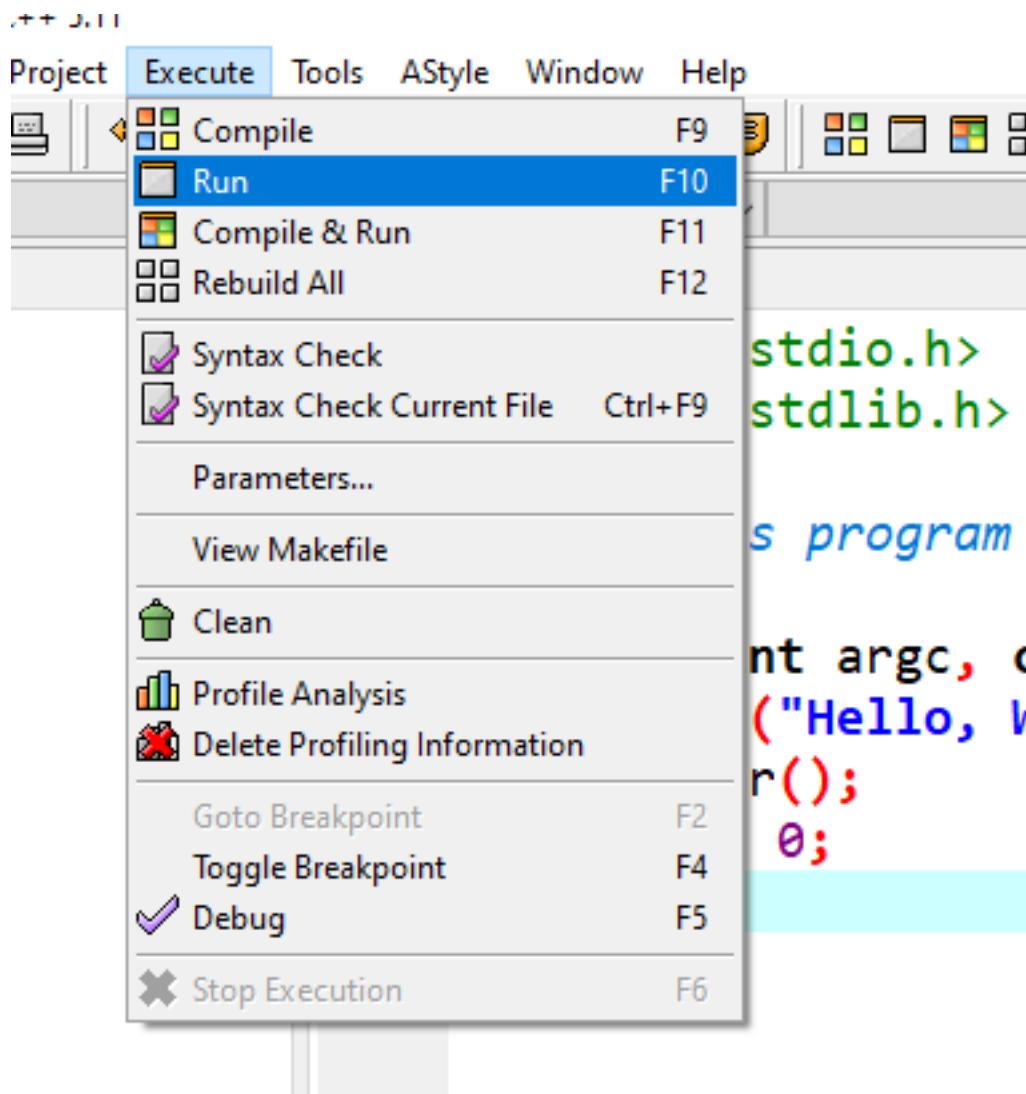
- Compilation Time: 2,13s

---

---

### 0.3.5.1.9 DevCpp (Install / Compile / Run / Debug) (10)

- Then you can run with Execute->Run F10 or Directly Compile&Run F11



**0.3.5.1.10 DevCpp (Install / Compile / Run / Debug) (11)** for debugging operations, just change the code and add more statements as follow

```

#include <stdio.h>
#include <stdlib.h>

/* run this program using the console pauser or add your getch, system(",pause") or input loop */

int main(int argc, char *argv[])
{
    printf("Hello, World! Step-1\n");
    printf("Hello, World! Step-2\n");
    printf("Hello, World! Step-3\n");
    printf("Hello, World! Step-4\n");
    printf("Hello, World! Step-5\n");
    printf("Hello, World! Step-6\n");

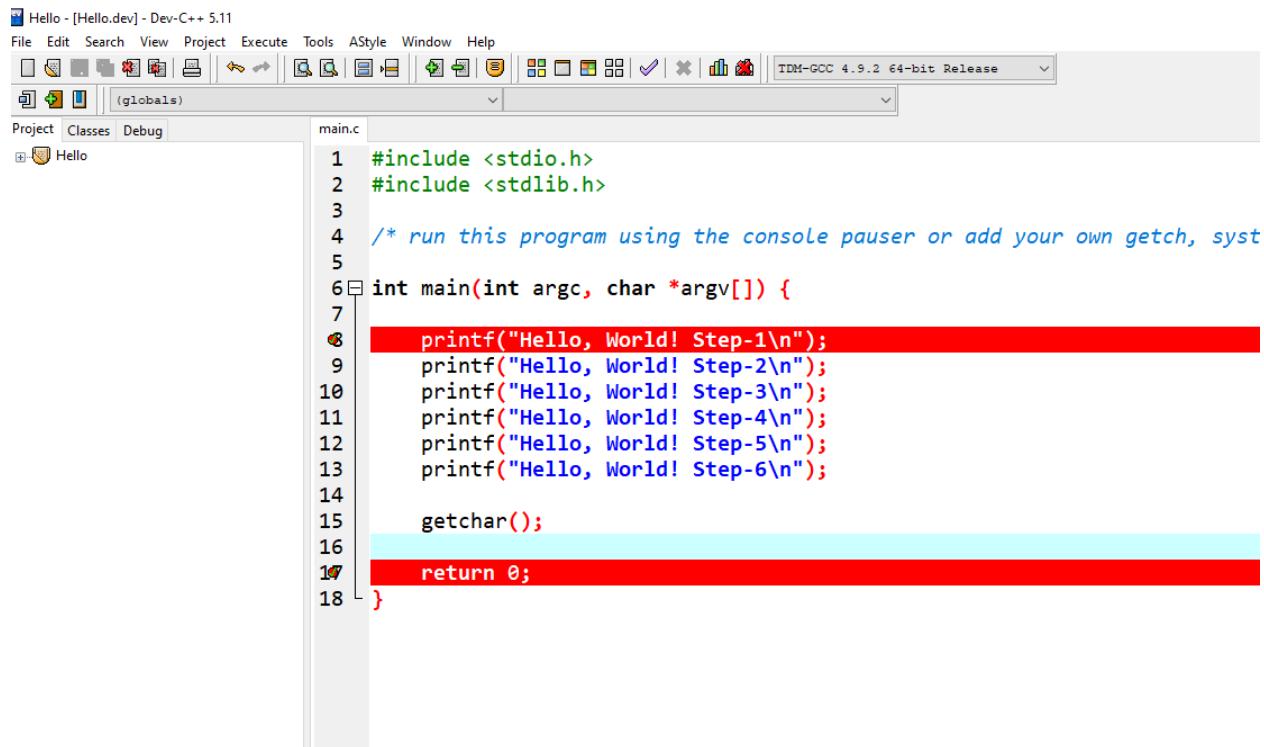
    getchar();

    return 0;
}

```

---

**0.3.5.1.11 DevCpp (Install / Compile / Run / Debug) (12)** Click on line numbers and add breakpoints for the debugger. This red point will be debugger stop points



The screenshot shows the Dev-C++ IDE interface. The menu bar includes File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, and Help. The toolbar contains various icons for file operations like Open, Save, and Build. The status bar indicates "TDM-GCC 4.9.2 64-bit Release". The project navigation bar shows "Hello" under "Hello". The code editor displays "main.c" with the following content:

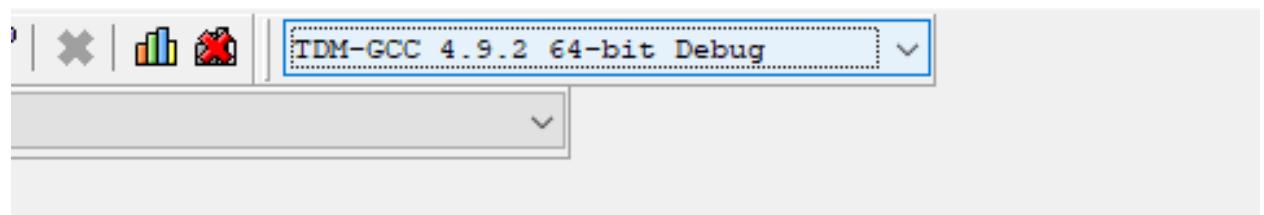
```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 /* run this program using the console pauser or add your own getch, syst
5
6 int main(int argc, char *argv[]) {
7
8     printf("Hello, World! Step-1\n");
9     printf("Hello, World! Step-2\n");
10    printf("Hello, World! Step-3\n");
11    printf("Hello, World! Step-4\n");
12    printf("Hello, World! Step-5\n");
13    printf("Hello, World! Step-6\n");
14
15    getchar();
16
17    return 0;
18 }
```

Line 8 is highlighted with a red background, indicating it is a breakpoint. Lines 17 and 18 are also highlighted with red backgrounds.

---

**0.3.5.1.12 DevCpp (Install / Compile / Run / Debug) (13)**

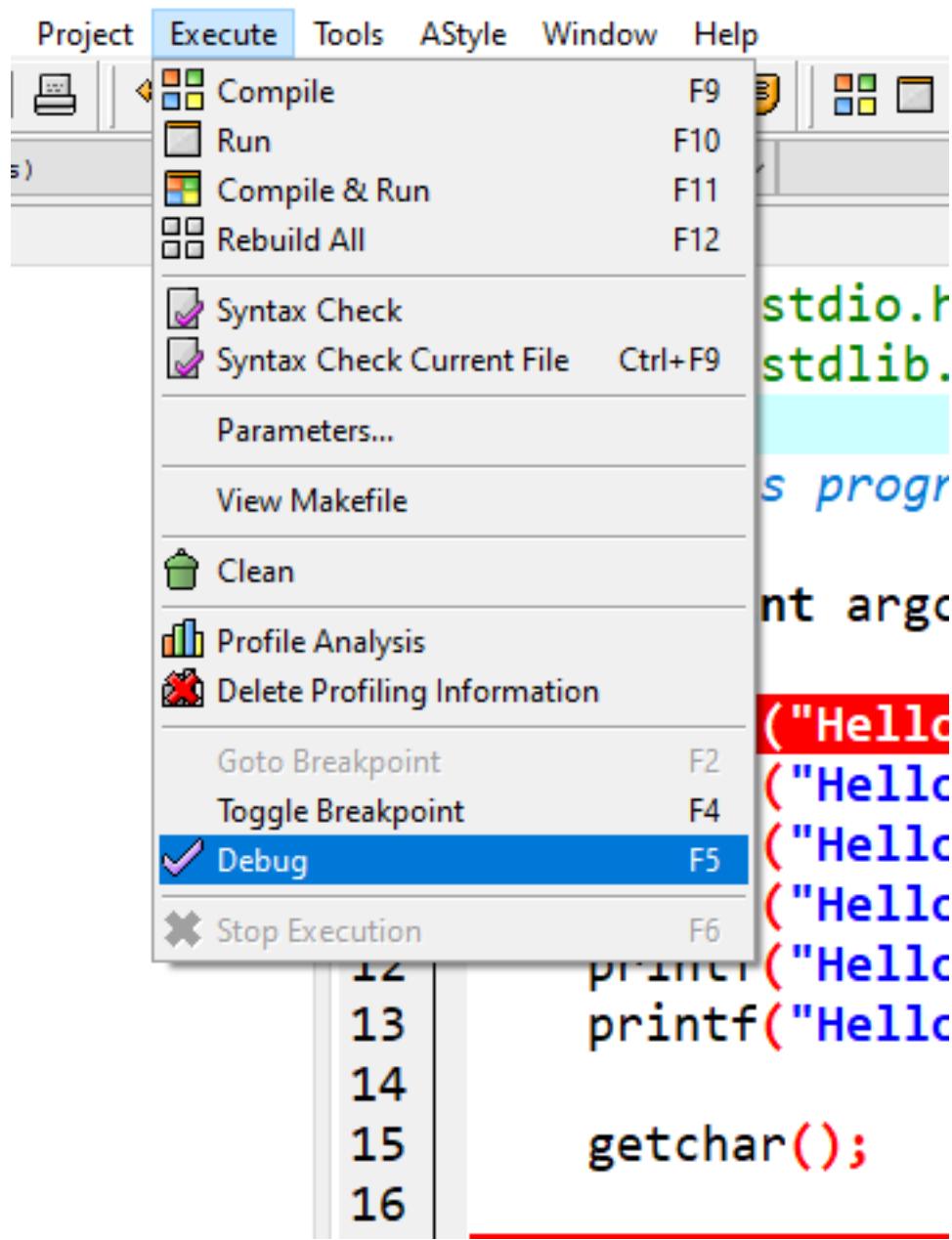
- In the menu section, select the compiler with debug option



---

**0.3.5.1.13 DevCpp (Install / Compile / Run / Debug) (14)**

- Compile application with debugging setting and in Execute Section use Debug F5 to start debugging



#### 0.3.5.1.14 DevCpp (Install / Compile / Run / Debug) (15)

- The debugger will stop at the breakpoint at the debug point (blue line)

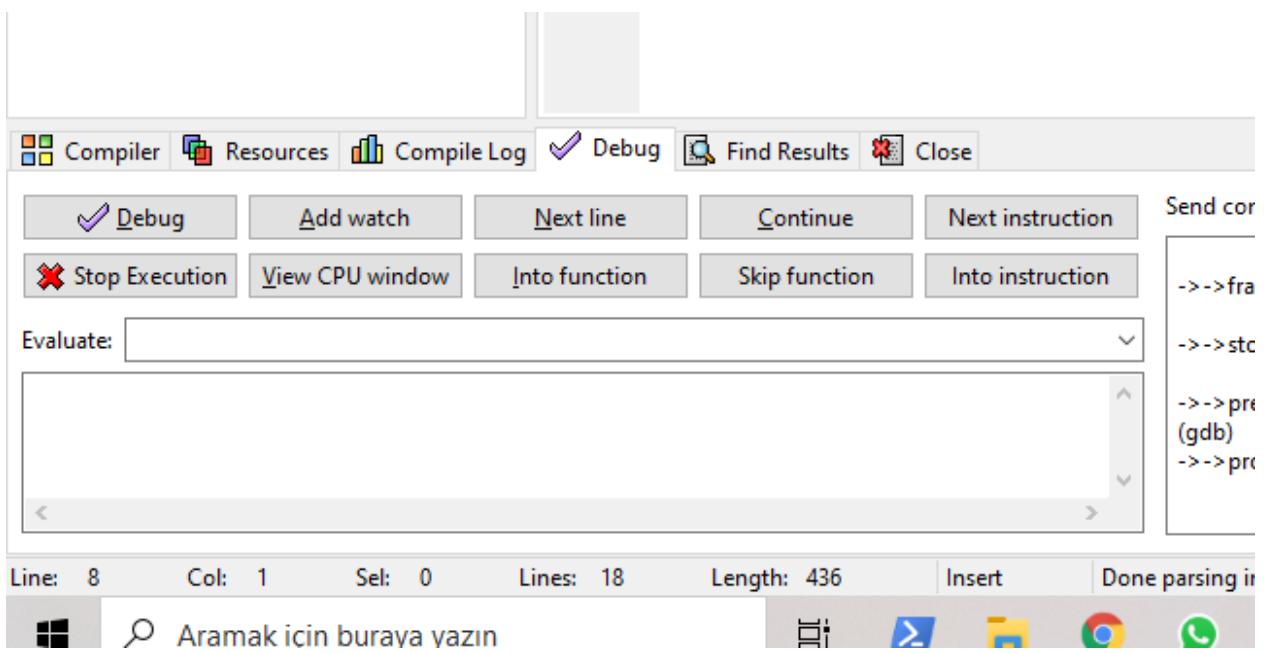
The screenshot shows the DevCpp IDE interface. The code editor window is open with the file 'main.c'. The code contains a series of printf statements and a getchar call. Lines 8 through 14 are highlighted in blue, and line 17 is highlighted in red. The code is as follows:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 /* run this program using the console pauser or add your own getch,
5
6 int main(int argc, char *argv[]) {
7
8     printf("Hello, World! Step-1\n");
9     printf("Hello, World! Step-2\n");
10    printf("Hello, World! Step-3\n");
11    printf("Hello, World! Step-4\n");
12    printf("Hello, World! Step-5\n");
13    printf("Hello, World! Step-6\n");
14
15    getchar();
16
17    return 0;
18 }
```

---

#### 0.3.5.1.15 DevCpp (Install / Compile / Run / Debug) (16)

- Moving to the next statement can be done via control buttons or shortcuts



---

#### 0.3.5.1.16 DevCpp (Install / Compile / Run / Debug) (17) -Press F8 to step-by-step continue

- Then go to **Project Options** -> **Compiler** -> **Linker** and set Generate debugging information to “yes”, and make sure you are not using any optimization options (they’re not good for debug mode). Also, check the Parameters tab, and make sure you don’t have any optimization options (like **-O2** or **-O3**, but **-O0** is ok because it means no optimization) or strip option (**-s**).
- 

#### **0.3.5.1.17 DevCpp (Install / Compile / Run / Debug) (18)**

- After that, do a full rebuild (**Ctrl-F11**), then set a breakpoint(s) where you want the debugger to stop (otherwise it will just run the program). To set a breakpoint on a line, just click on the gutter (the gray band on the left), or press **Ctrl-F5**.
- 

#### **0.3.5.1.18 DevCpp (Install / Compile / Run / Debug) (19)**

- Now you are ready to launch the debugger, by pressing F8 or clicking the debug button. If everything goes well, the program will start, and then stop at the first breakpoint. Then you can step through the code, entering function calls, by pressing **Shift-F7** or the “step into” button, or stepping over the function calls, by pressing F7 or the “next step” button. You can press **Ctrl-F7** or the “continue” button to continue execution till the next breakpoint. At any time, you can add or remove breakpoints.
- 

**0.3.5.1.19 DevCpp (Install / Compile / Run / Debug) (20)** When the program stopped at a breakpoint and you are stepping through the code, you can display the values of various variables in your program by putting your mouse over them, or you can display variables and expressions by pressing F4 or the “add watch” button and typing the expression.

---

#### **0.3.5.1.20 DevCpp (Install / Compile / Run / Debug) (21)** How do I debug using Dev-C++<sup>10</sup>

---

**0.3.5.2 Code Blocks (Install / Compile / Run / Debug) (1)** Download Code Blocks from the following link

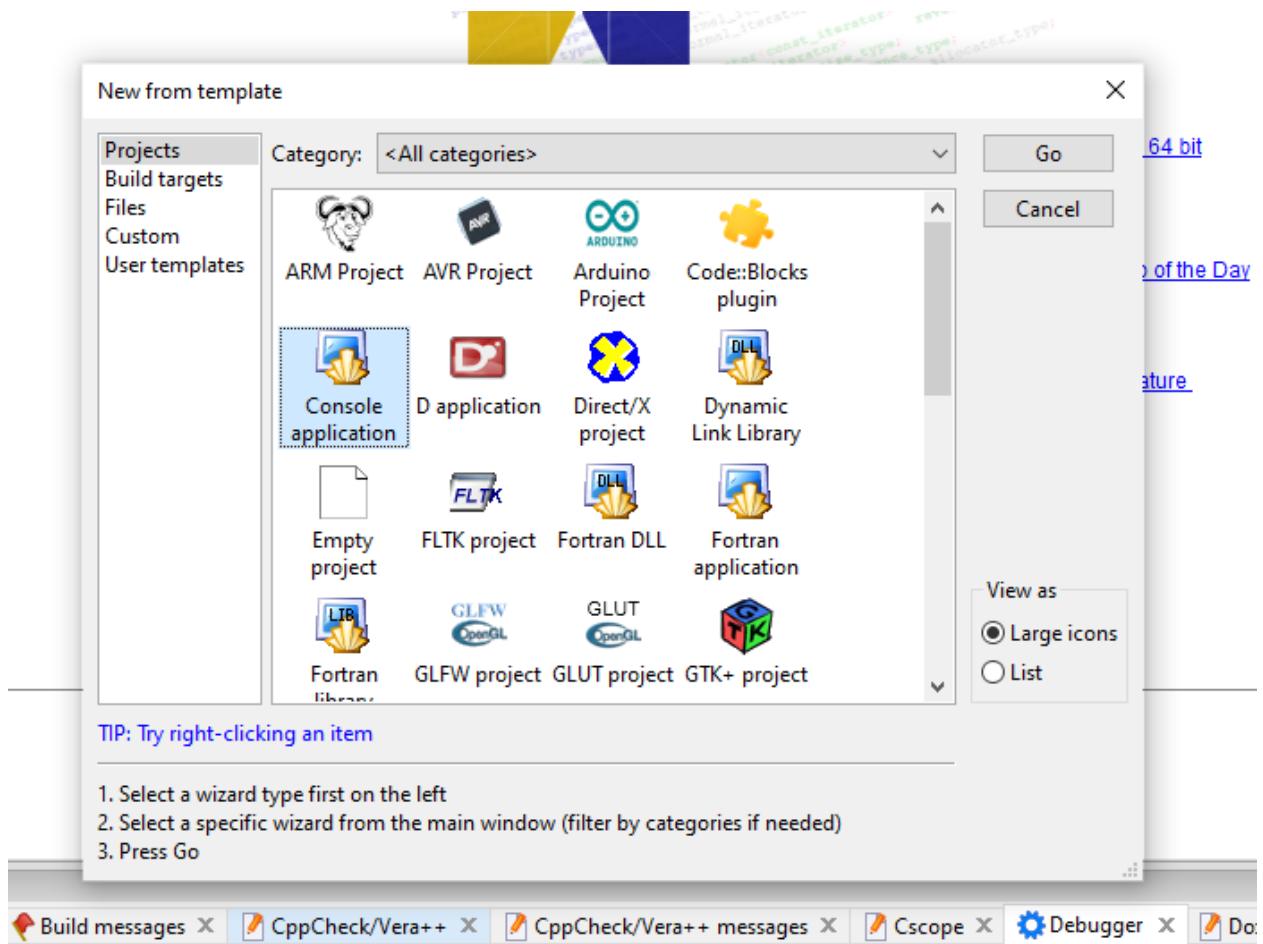
Binary releases - Code::Blocks<sup>11</sup>

---

**0.3.5.2.1 Code Blocks (Install / Compile / Run / Debug) (2)** Open Code Blocks and Select **File->New->Project**

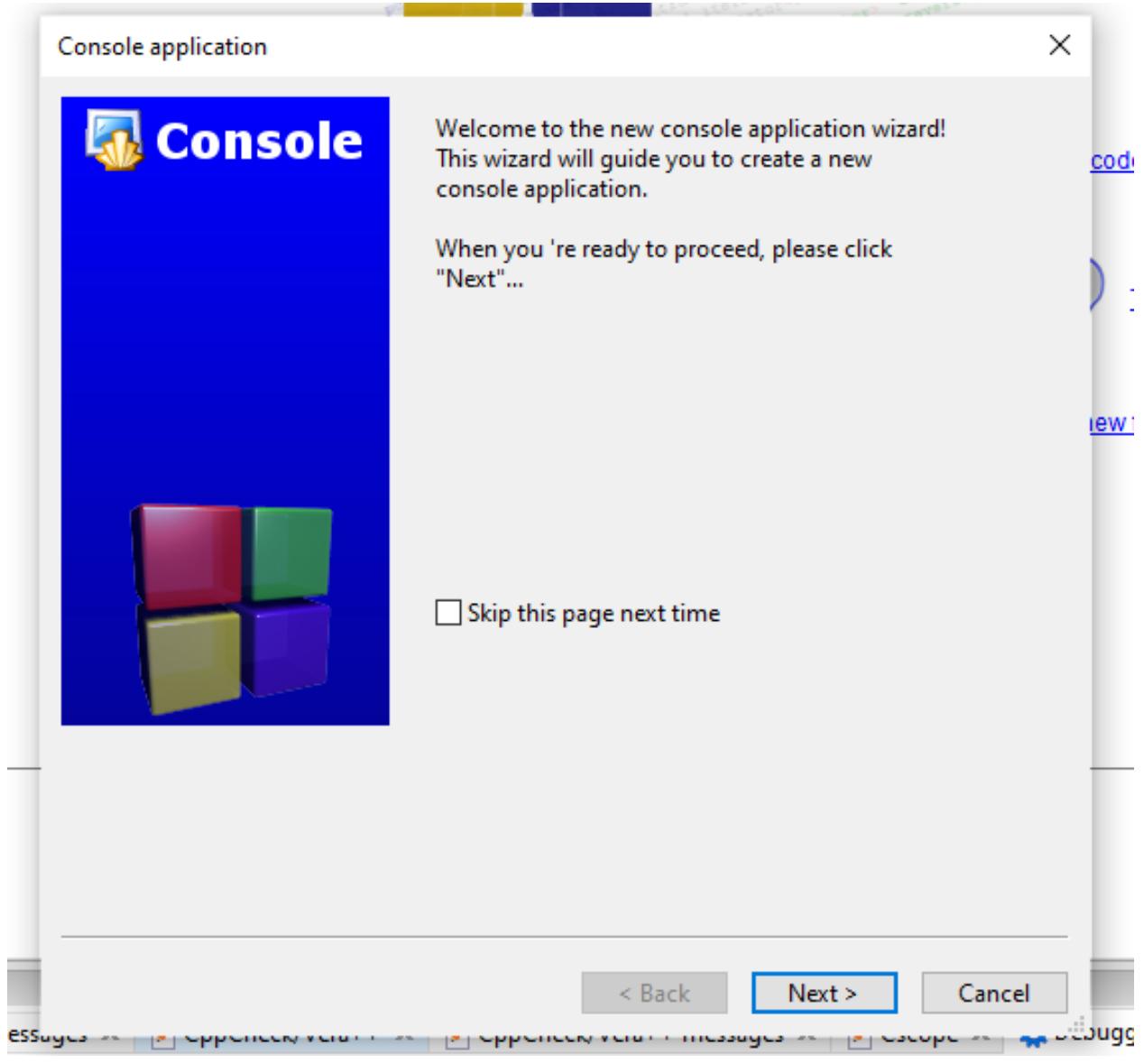
<sup>10</sup>[http://eilat.sci.brooklyn.cuny.edu/cis1\\_5/HowToDebug.htm](http://eilat.sci.brooklyn.cuny.edu/cis1_5/HowToDebug.htm)

<sup>11</sup><https://www.codeblocks.org/downloads/binaries/>

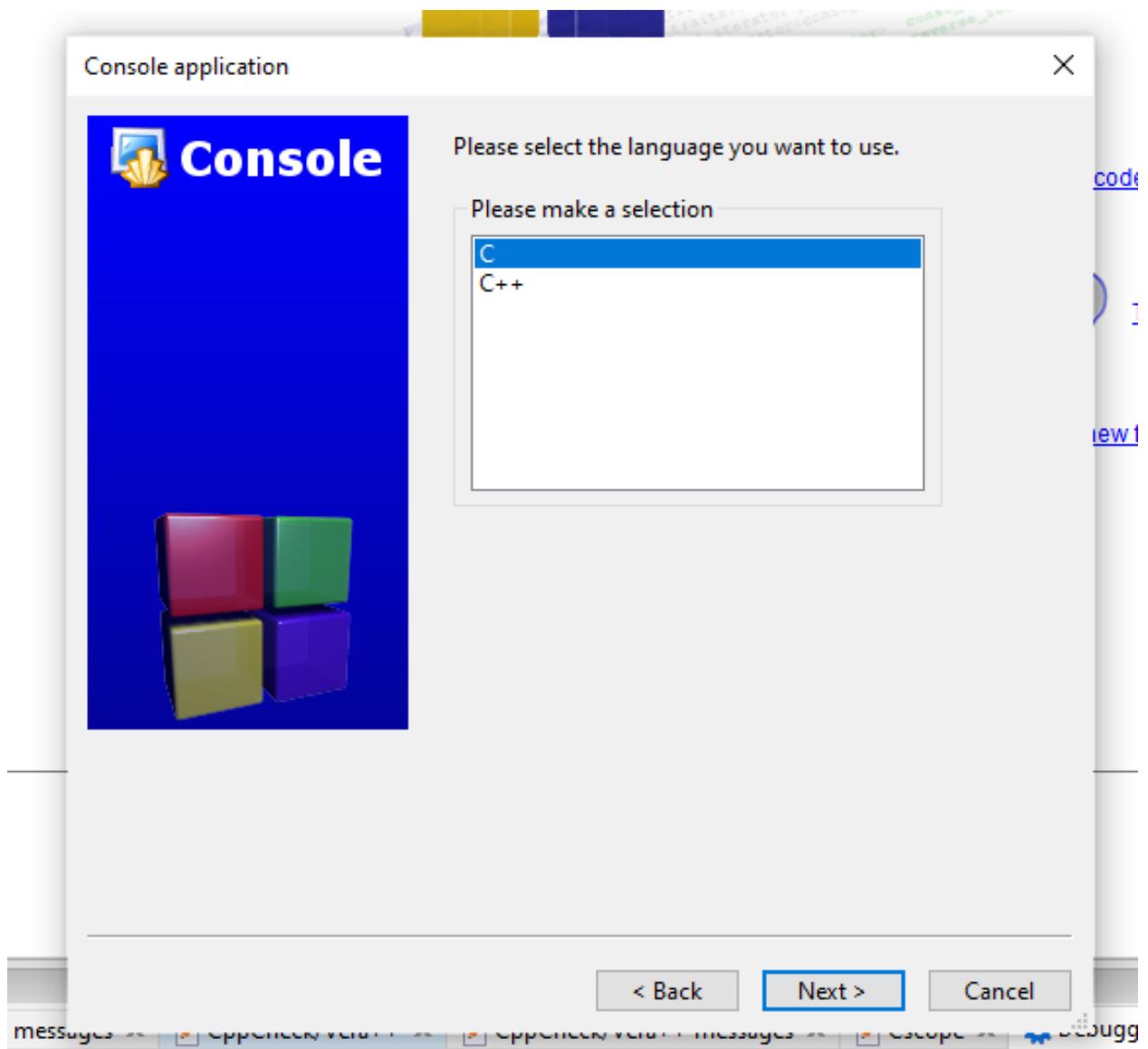


### 0.3.5.2.2 Code Blocks (Install / Compile / Run / Debug) (3) Select Console Application

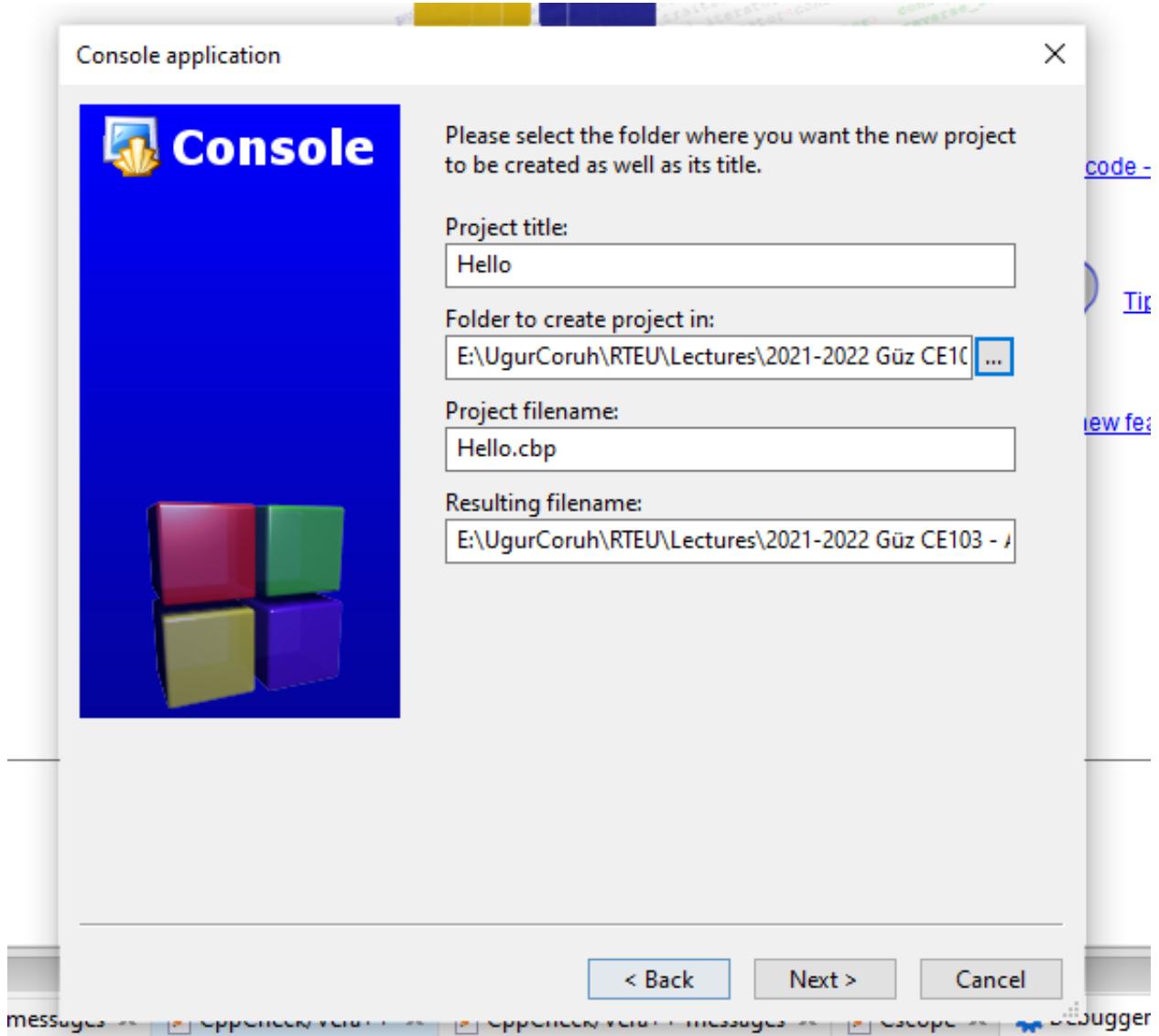
Click Next from Opening Window



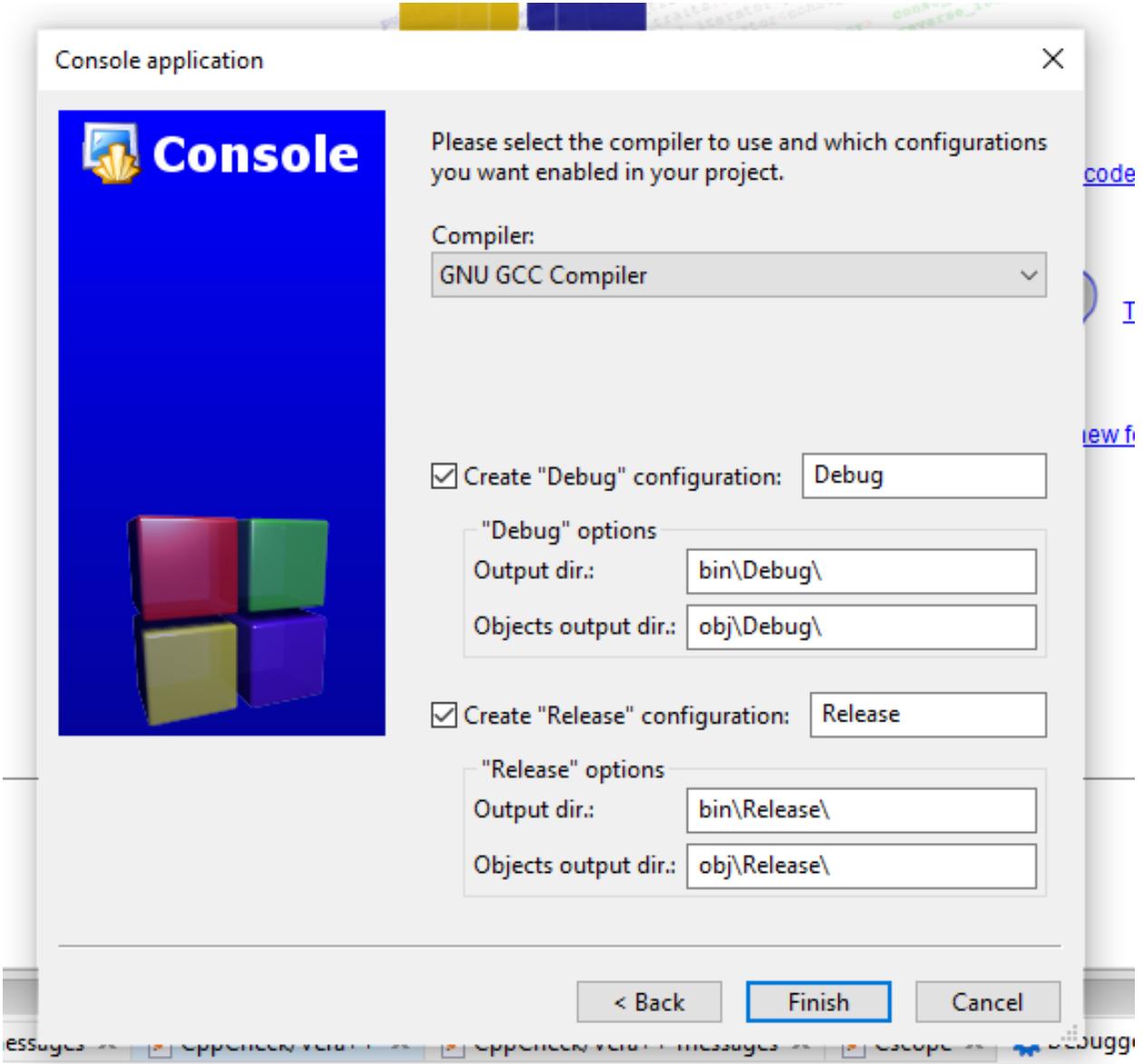
#### 0.3.5.2.3 Code Blocks (Install / Compile / Run / Debug) (4) Select C for Sample Project



**0.3.5.2.4 Code Blocks (Install / Compile / Run / Debug) (5)** Write a project name and title also set a project folder



**0.3.5.2.5 Code Blocks (Install / Compile / Run / Debug) (6)** Select a compiler for this project we selected GCC but you can select C compilers from the list. Set Debug and Release executable output folders.



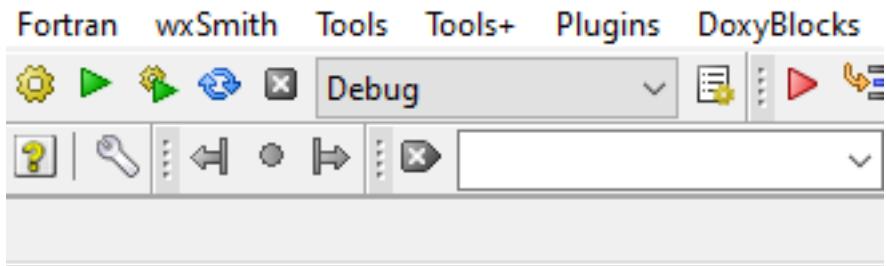
### 0.3.5.2.6 Code Blocks (Install / Compile / Run / Debug) (7)

- After this wizard, you will have the following code

```
#include <stdio.h>
#include <stdlib.h>

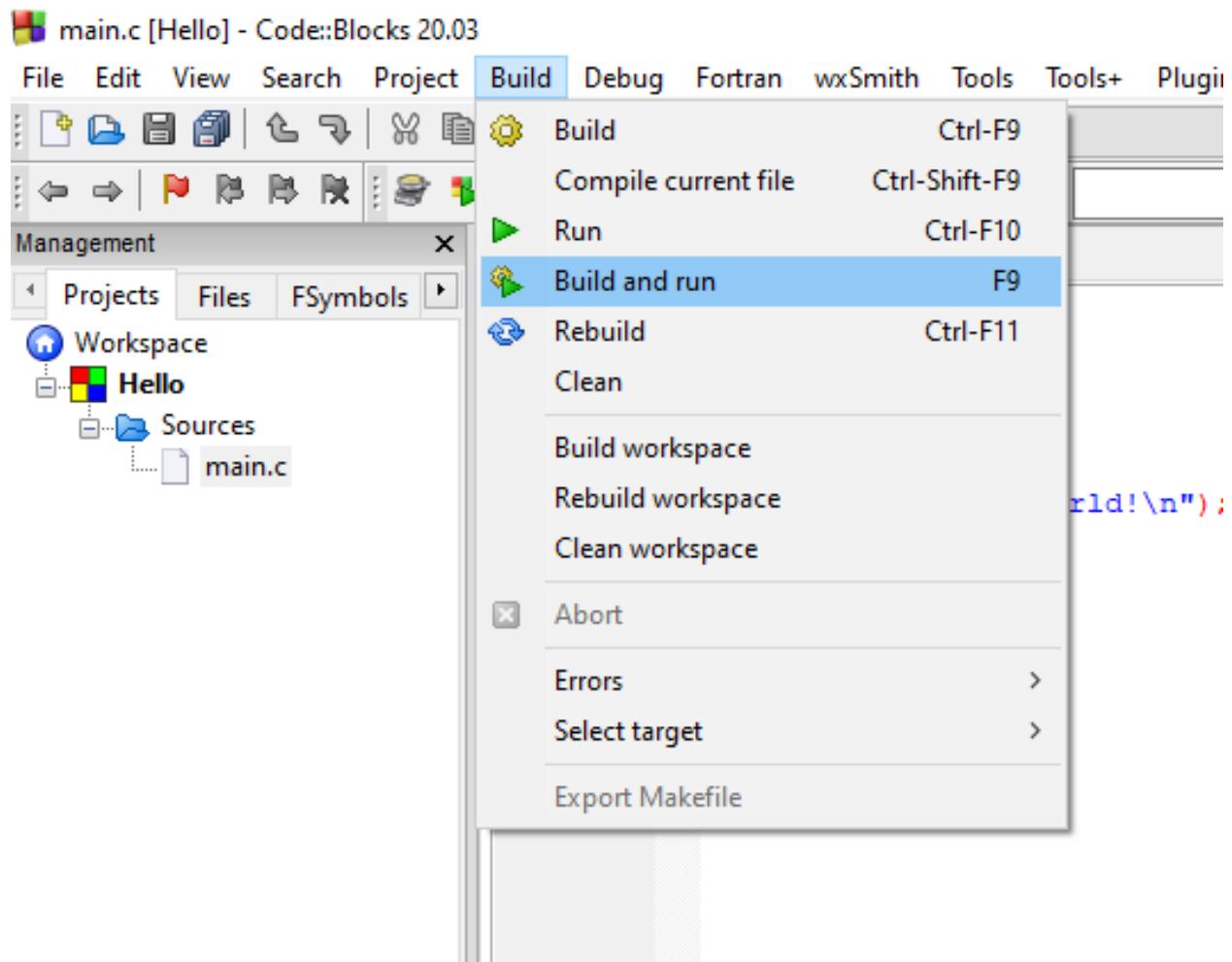
int main()
{
    printf("Hello world!\n");
    return 0;
}
```

### 0.3.5.2.7 Code Blocks (Install / Compile / Run / Debug) (8) Select Debug Build from the menu



---

#### 0.3.5.2.8 Code Blocks (Install / Compile / Run / Debug) (9) Run with Build and Run F9



---

#### 0.3.5.2.9 Code Blocks (Install / Compile / Run / Debug) (10)

- You should see the following output

```
"E:\UgurCoruh\RTEU\Lectures\2021-2022 G3z CE103 - Algorithms and Programming \Lectures
Hello world!
Process returned 0 (0x0)   execution time : 0.038 s
Press any key to continue.
```

---

#### 0.3.5.2.10 Code Blocks (Install / Compile / Run / Debug) (11)

- Add the following lines to your source code for debugging

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Hello world! Step-1\n");
    printf("Hello world! Step-2\n");
    printf("Hello world! Step-3\n");
    printf("Hello world! Step-4\n");
    printf("Hello world! Step-5\n");
    printf("Hello world! Step-6\n");
    return 0;
}
```

---

#### 0.3.5.2.11 Code Blocks (Install / Compile / Run / Debug) (12)

- and add breakpoints with F5 or mouse click

The screenshot shows the Code::Blocks 20.03 IDE interface. The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar has various icons for file operations like Open, Save, Find, and Run. The 'Management' panel on the left shows a 'Workspace' with a project named 'Hello' containing a 'Sources' folder and a file 'main.c'. The main code editor window displays the following C code:

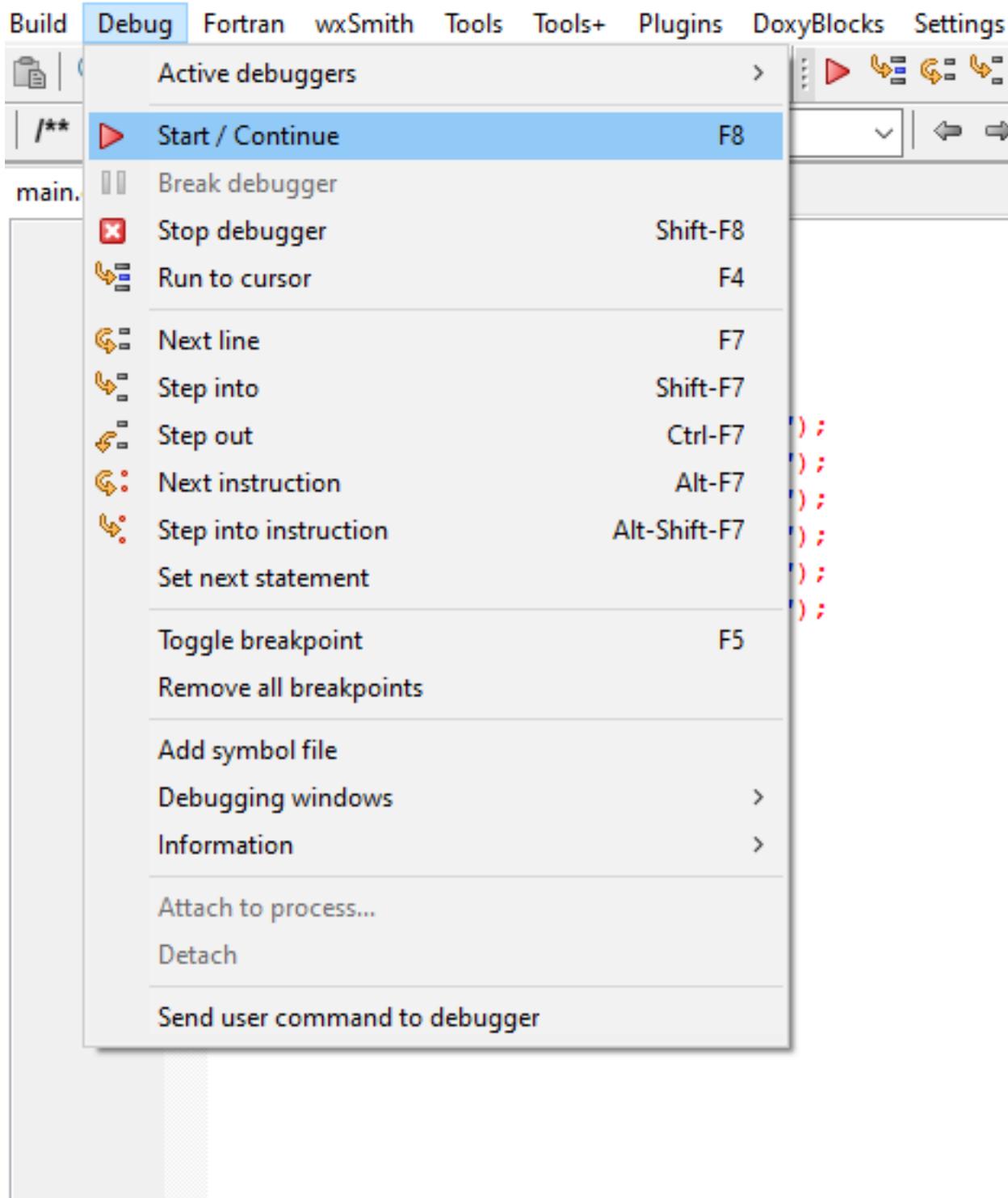
```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Hello world! Step-1\n");
    printf("Hello world! Step-2\n");
    printf("Hello world! Step-3\n");
    printf("Hello world! Step-4\n");
    printf("Hello world! Step-5\n");
    printf("Hello world! Step-6\n");
    return 0;
}
```

---

#### 0.3.5.2.12 Code Blocks (Install / Compile / Run / Debug) (13)

- select Debug->Start/Continue to start debugger



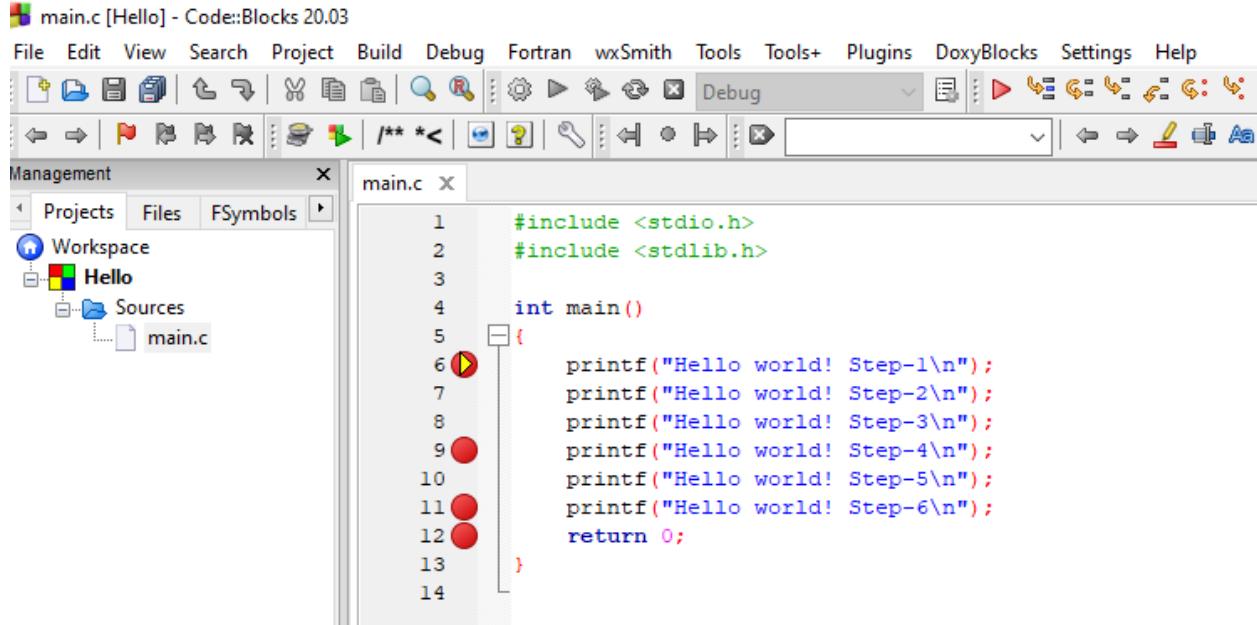
#### 0.3.5.2.13 Code Blocks (Install / Compile / Run / Debug) (14)

- If you see the following error this is related to long or turkish characters including the path. Just move the project to a shorter path and try again

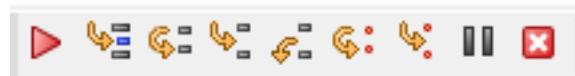
Setting breakpoints  
Debugger name and version: GNU gdb (GDB) 8.1

```
Starting the debugger failed: No executable specified, use `target exec'.  
Debugger finished with status 0
```

**0.3.5.2.14 Code Blocks (Install / Compile / Run / Debug) (15)** You will see the following yellow pointer for the debugger



**0.3.5.2.15 Code Blocks (Install / Compile / Run / Debug) (16)** You can use the following menu or shortcuts for step-by-step debugging.



**0.3.5.3 GCC/G++ Complier (MinGW) / Clang-cl (LLVM) (1)** Download and install MinGW or LLVM compiler (if you downloaded then skip this step)

- MinGW installer (clang)
    - Download MinGW-w64 - for 32 and 64-bit Windows from SourceForge.net<sup>12</sup>
  - If you have a problem try ‘Github“ builds
    - <https://github.com/niXman/mingw-builds-binaries/releases>
    - [https://github.com/niXman/mingw-builds-binaries/releases/download/12.2.0-rt\\_v10-rev0/x86\\_64-12.2.0-release-win32-seh-rt\\_v10-rev0.7z](https://github.com/niXman/mingw-builds-binaries/releases/download/12.2.0-rt_v10-rev0/x86_64-12.2.0-release-win32-seh-rt_v10-rev0.7z)
  - LLVM installer (gcc/g++)
  - Download LLVM releases<sup>13</sup>
    - Also use the following notes
      - \* <https://llvm.org/devmtg/2014-04/PDFs/Talks/clang-cl.pdf>

---

<sup>12</sup><https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/installer/mingw-w64-install.exe/download>

<sup>13</sup><https://releases.llvm.org/>

---

**0.3.5.4 GCC/G++ Complier (MinGW) / Clang-cl (LLVM) (2)** Open a console with “cmd” and test the following commands if commands are not recognized then set the system environment variable to add gcc and g++ executable paths to the path variable (add to both system and user path variable)

```
gcc --version
```

```
g++ --version
```

```
C:\Users\ugur.coruh>gcc --version
gcc (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
clang --version
```

---

### **0.3.5.5 GCC/G++ Complier (MinGW) / Clang-cl (LLVM) (3)**

- for gcc.exe, g++.exe and gdb.exe

```
C:\Program Files\mingw-w64\x86_64-8.1.0-win32-seh-rt_v6-rev0\mingw64\bin
```

- for clang.exe, lldb.exe

```
C:\Program Files\LLVM\bin
```

This folder path changes according to your setup

---

### **0.3.6 VSCode (Install / Compile / Run / Debug) (1)**

Download Visual Studio Code from the following link

Download Visual Studio Code - Mac, Linux, Windows<sup>14</sup>

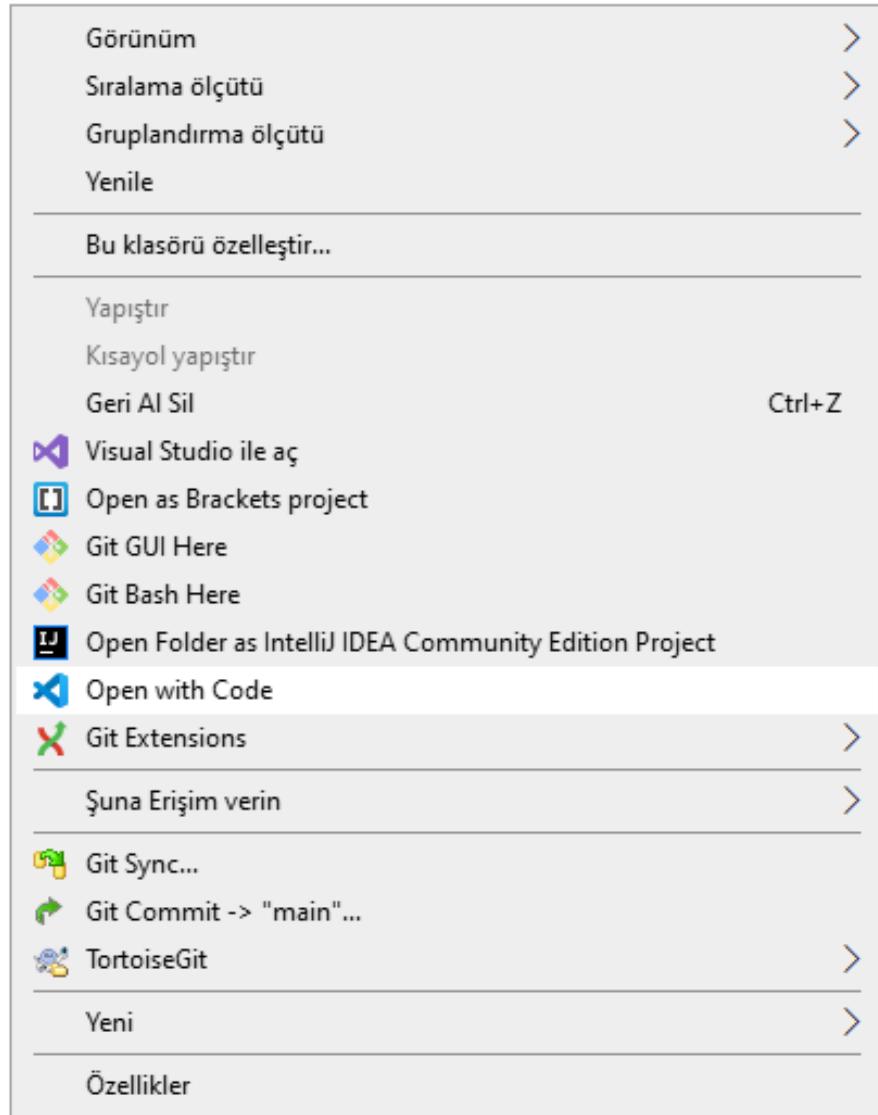
---

**0.3.6.1 VSCode (Install / Compile / Run / Debug) (2)** In this sample, you will find MinGW and LLVM compiler combinations for C and C++

Create a folder and enter this folder then open this folder with vscode by right click

---

<sup>14</sup><https://code.visualstudio.com/download>



### 0.3.6.2 VSCode (Install / Compile / Run / Debug) (3) or enter the folder via console

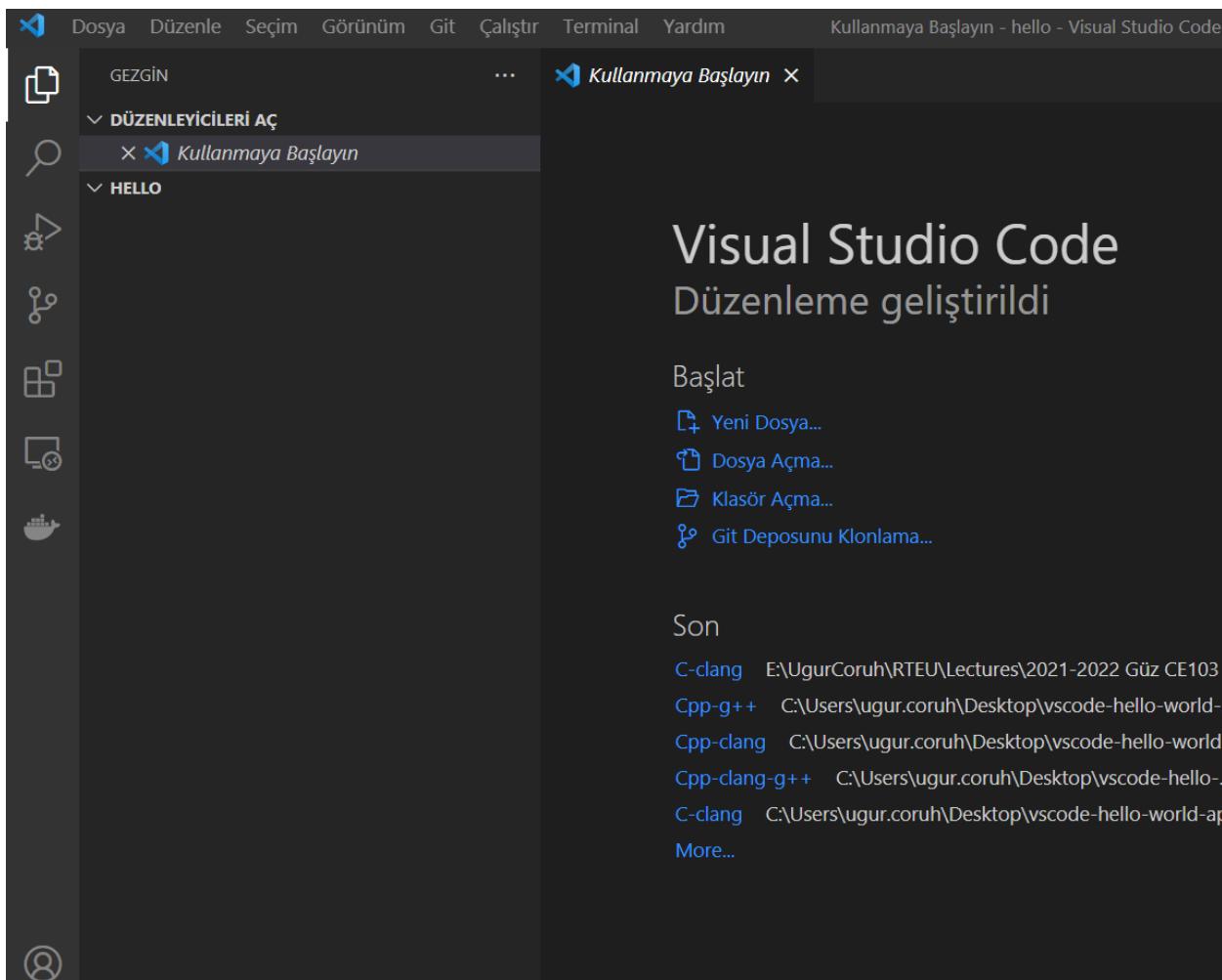
```
Güz CE:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programming I\Lectures\ce103-algorithms-and-programming-I\Week-2\vscode-hello-world-apps\C-clang>code .
```

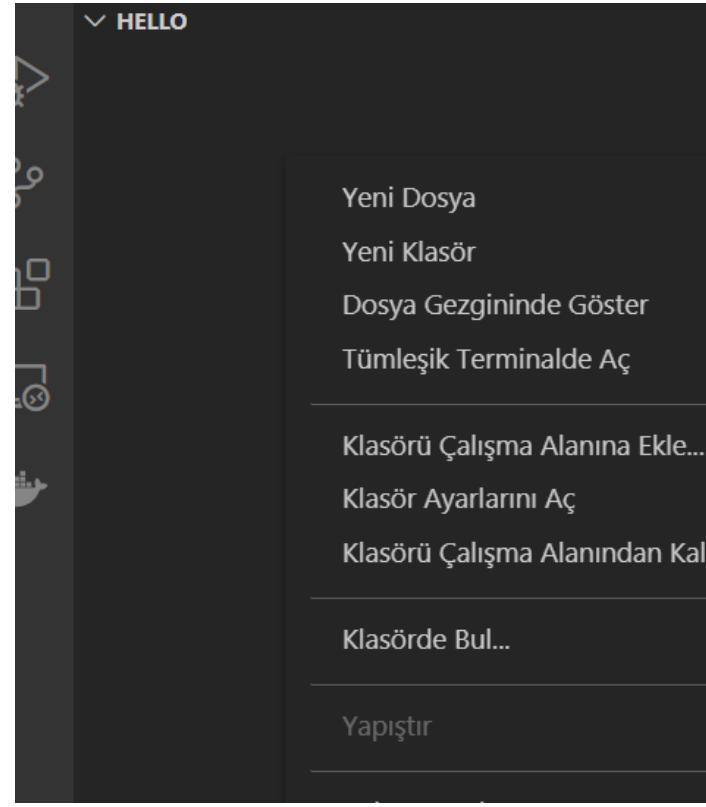
write

code .

### 0.3.6.3 VSCode (Install / Compile / Run / Debug) (4)

- This will open vscode for the current folder, (.) dot present current folder.
- You will see an empty folder in the right window





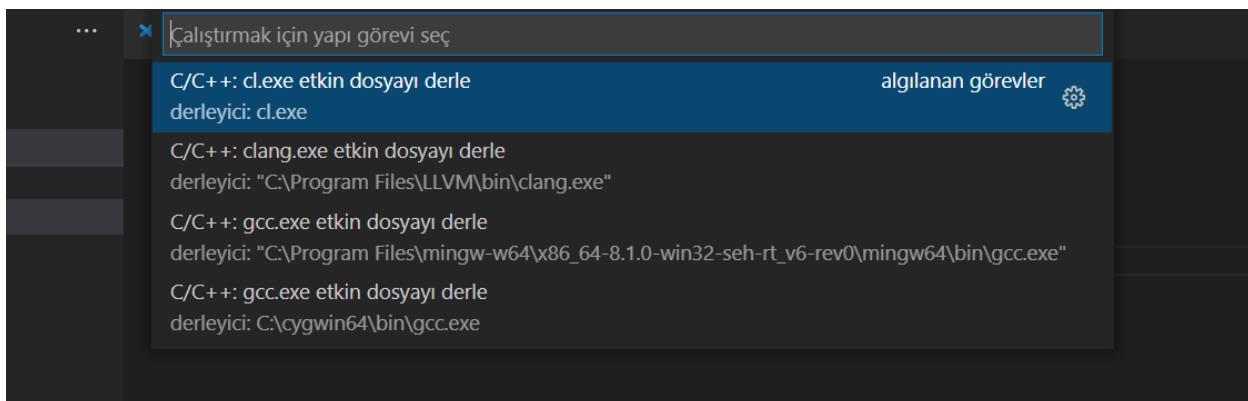
#### 0.3.6.4 VSCode (Install / Compile / Run / Debug) (5)

#### 0.3.6.5 VSCode (Install / Compile / Run / Debug) (6)

- Create a `hello.c` file and write the following content

```
#include <stdio.h>
int main() {
    // printf() displays the string inside quotation
    printf("Hello, World!");
    return 0;
}
```

#### 0.3.6.6 VSCode (Install / Compile / Run / Debug) (7) use CTRL+SHIFT+B (you should be on the source code section) to build a file



**0.3.6.7 VSCode (Install / Compile / Run / Debug) (8)** Select GCCor CLANGfor this sample we can use GCC

You will see the output generated ‘Hello.exe“

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with a tree view of files: 'GEZGİN', 'DÜZENLEYİCİLERİ AÇ' (which is expanded to show 'Kullanmaya Başlayın' and 'Hello.c'), 'HELLO' (which is also expanded to show 'Hello.c'), and 'Hello.exe'. The main editor area shows the 'Hello.c' file with the following code:

```
#include <stdio.h>
int main()
{
    // printf() displays the string inside quotation
    printf("Hello, World!");
    return 0;
}
```

Below the editor is the Terminal tab, which is active. It shows the command being run: 'Executing task: C/C++: gcc.exe etkin dosyayı derle <'. The terminal output shows the compilation process:

```
> Executing task: C/C++: gcc.exe etkin dosyayı derle <

Derleme başlatılıyor...
"C:\Program Files\mingw-w64\x86_64-8.1.0-win32-seh-rt_v6-rev0\mingw64\bin\gcc.exe" -fdiagnostics-color=auto -fmessage-length=0 "C:\Users\ugur.coruh\Desktop\hello\Hello.c" -o "C:\Users\ugur.coruh\Desktop\hello\Hello.exe"
Derleme başarıyla tamamlandı.
```

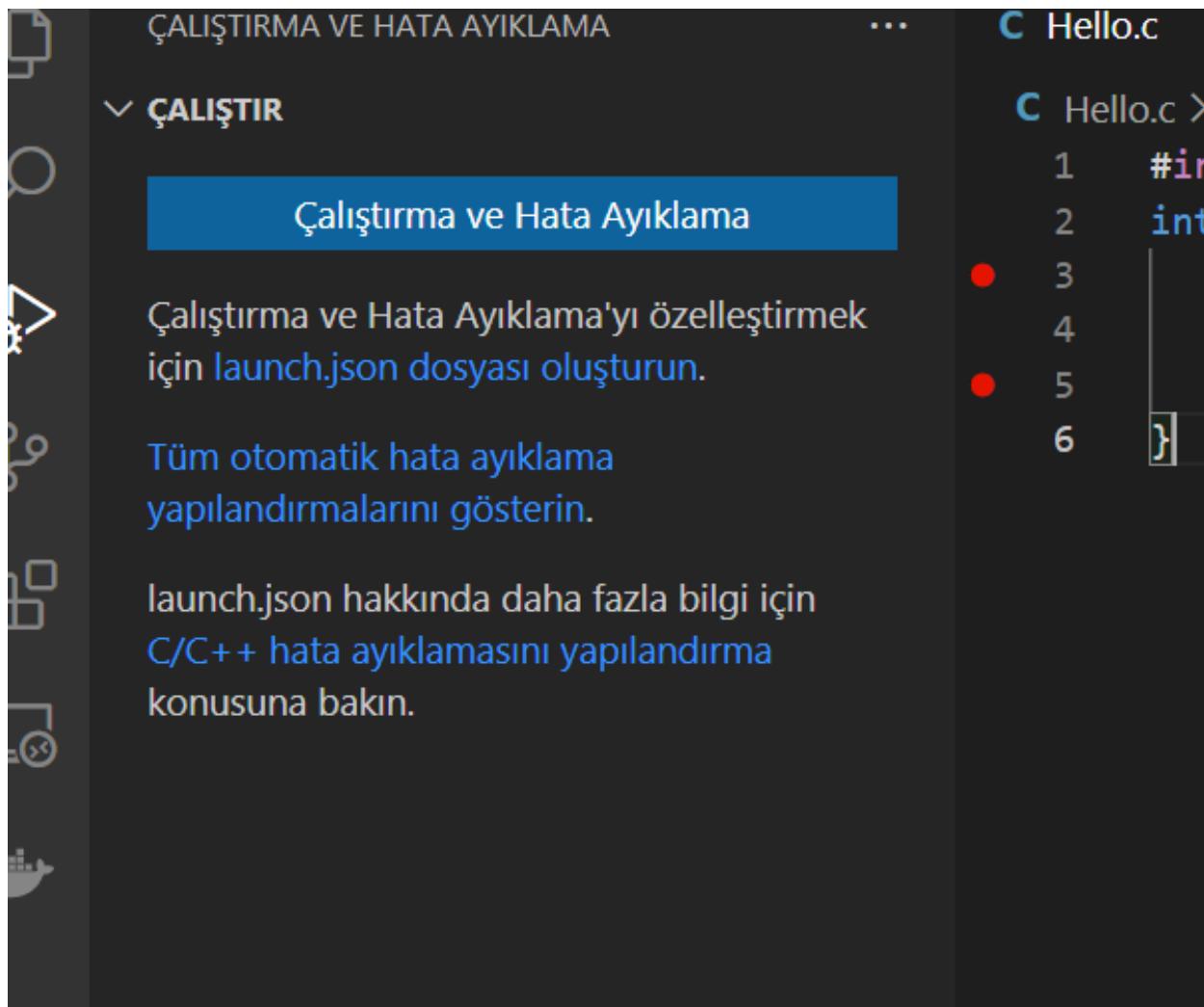
The terminal also includes a note: 'Terminal, görevler tarafından yeniden kullanılacak; kapatmak için bir tuşa basın.'

**0.3.6.8 VSCode (Install / Compile / Run / Debug) (9)** for debugging just put a breakpoint and build again

The screenshot shows the Visual Studio Code interface with the same file structure as the previous screenshot. The 'Hello.c' file in the editor has two red circular markers (breakpoints) placed on lines 3 and 5. The terminal output is identical to the previous screenshot, showing the successful compilation of the program.

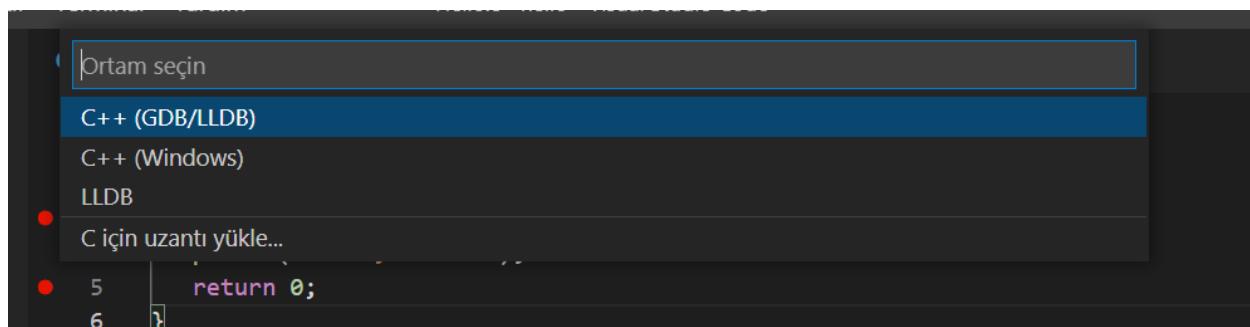
**0.3.6.9 VSCode (Install / Compile / Run / Debug) (10)**

- after building for debug press **CTRL+SHIFT+D** (you should be in the source code section)and in the right window select create **launch.json**



#### 0.3.6.10 VSCode (Install / Compile / Run / Debug) (11)

- from opening, window select C++ GDB/LLDB



#### 0.3.6.11 VSCode (Install / Compile / Run / Debug) (12)

- from the next opening, menu select mingw-w64 gcc.exe

İşler Terminal Yardım

Hello.c - hello - Visual Studio Code

Yapilandırma seçenek

- clang.exe - Etkin dosyayı derle ve dosyada hata ayıkla derleyici: "C:\Program Files\LLVM\bin\clang.exe"
- gcc.exe - Etkin dosyayı derle ve dosyada hata ayıkla derleyici: "C:\Program Files\mingw-w64\x86\_64-8...."
- gcc.exe - Etkin dosyayı derle ve dosyada hata ayıkla derleyici: C:\cygwin64\bin\gcc.exe

Varsayılan Yapılandırma

5      return 0;

6 }

**0.3.6.12 VSCode (Install / Compile / Run / Debug) (13)** this will run the debugger and you will see debug points activated

Dosya Düzenle Seçim Görünüm Git Çalıştır Terminal Yardım

ÇALIŞTIRMA V... D gcc.exe - Etkir ... C Hello.c X launch.json

DEĞİŞKENLER Locals Registers

IZLEME

C Hello.c > main()
1 #include <stdio.h>
2 int main() {
3 // printf() displays the string inside quotation
4 4 printf("Hello, World!");
5 | return 0;
6 }

**0.3.6.13 VSCode (Install / Compile / Run / Debug) (14)** then you can step-by-step debug your code.

the following task.json and launch.json automatically generated with your selections

{} launch.json .vscode

HELLO

.vscode

{} launch.json

{} tasks.json

C Hello.c

2  
3  
4  
5  
6

**0.3.6.14 VSCode (Install / Compile / Run / Debug) (15) launch.json**

```
{
  // Olası öznitelikler hakkında bilgi edinmek için IntelliSense kullanın.
  // Mevcut özniteliklerin açıklamalarını görüntülemek için üzerine gelin.
  // Daha fazla bilgi için şu adresi ziyaret edin: https://go.microsoft.com/fwlink/?LinkId=830387}
```

```

"version": "0.2.0",
"configurations": [
{
  "name": "gcc.exe - Etkin dosyayı derle ve dosyada hata ayıkla",
  "type": "cppdbg",
  "request": "launch",
  "program": "${fileDirname}\\${fileBasenameNoExtension}.exe",
  "args": [],
  "stopAtEntry": false,
  "cwd": "${fileDirname}",
  "environment": [],
  "externalConsole": false,
  "MIMode": "gdb",
  "miDebuggerPath": "C:\\Program Files\\mingw-w64\\x86_64-8.1.0-win32-seh-rt_v6-rev0\\mingw64\\bin\\",
  "setupCommands": [
    {
      "description": "gdb için düzgün yazdırmayı etkinleştir",
      "text": "-enable-pretty-printing",
      "ignoreFailures": true
    }
  ],
  "preLaunchTask": "C/C++: gcc.exe etkin dosyayı derle"
}
]
}

```

---

### 0.3.6.15 VSCode (Install / Compile / Run / Debug) (16) task.json

```

{
  "tasks": [
    {
      "type": "cppbuild",
      "label": "C/C++: gcc.exe etkin dosyayı derle",
      "command": "C:\\Program Files\\mingw-w64\\x86_64-8.1.0-win32-seh-rt_v6-rev0\\mingw64\\bin\\gcc.exe",
      "args": [
        "-fdiagnostics-color=always",
        "-g",
        "${file}",
        "-o",
        "${fileDirname}\\${fileBasenameNoExtension}.exe"
      ],
      "options": {
        "cwd": "${fileDirname}"
      },
      "problemMatcher": ["$gcc"],
      "group": {
        "kind": "build",
        "isDefault": true
      },
      "detail": "Hata Ayıklayıcısı tarafından oluşturulan görev."
    }
  ],
  "version": "2.0.0"
}

```

---

### **0.3.6.16 VSCode (Install / Compile / Run / Debug) (17)**

- You can do the same thing for other compilers and C++ source codes. LLVM does not support debugging on vscode now.

for C++ VsCode you can check the following links

- for Windows
    - <https://code.visualstudio.com/docs/cpp/config-mingw>
  - for Linux
    - <https://code.visualstudio.com/docs/cpp/config-linux>
  - for WSL
    - <https://code.visualstudio.com/docs/cpp/config-wsl>
- 

### **0.3.6.17 VSCode (Install / Compile / Run / Debug) (18)** in the launch file if you start debugging with F5

(you can select debugger with **CTRL+SHIFT+P** and then write **Debug** and Selecting **Configure Debugger Option**)

---

### **0.3.6.18 VSCode (Install / Compile / Run / Debug) (19)**

- the following line will be your debugging application path
  - if you start debugging with F5 in `Hello.c` file this will set `<Hello.c base path>/Hello.exe`
- 

### **0.3.6.19 VSCode (Install / Compile / Run / Debug) (20)** You should set this correct for both LLVM and GCC configuration in `launch.json`

```
"program": "${fileDirname}\\${fileBasenameNoExtension}.exe",
```

Also you should set your installed debugger paths

for GCC

```
"miDebuggerPath": "C:\\Program Files\\mingw-w64\\x86_64-8.1.0-win32-seh-rt_v6-rev0\\mingw64\\bin\\gdb.exe",
```

for LLVM

```
"miDebuggerPath": "C:\\Program Files\\LLVM\\bin\\lldb.exe",
```

for more details please check the sample source codes.

---

## **0.3.7 Visual Studio Code Extension List (1)**

My Extension List

- Listing Installed Extensions

```
PS C:\\Users\\ugur.coruh\\Desktop> code --list-extensions | % { "code --install-extension $_" }
```

Following topic can help you

How can you export the Visual Studio Code extension list? - Stack Overflow<sup>15</sup>

---

<sup>15</sup><https://stackoverflow.com/questions/35773299/how-can-you-export-the-visual-studio-code-extension-list>

### 0.3.8 Visual Studio Code Extension List (2)

```
code --install-extension 2gua.rainbow-brackets
code --install-extension aaron-bond.better-comments
code --install-extension abusaidm.html-snippets
code --install-extension ACharLuk.easy-cpp-projects
code --install-extension chris-noring.node-snippets
code --install-extension cschlosser.doxdocgen
code --install-extension csholmq.excel-to-markdown-table
code --install-extension DaChuiOpenSource.FreeMind
code --install-extension dannysteenman.cloudformation-yaml-snippets
code --install-extension Dart-Code.dart-code
code --install-extension Dart-Code.flutter
code --install-extension digized.umple
code --install-extension DotJoshJohnson.xml
code --install-extension DougFinke.vscode-pandoc
code --install-extension dzhavat.bracket-pair-toggler
code --install-extension esbenp.prettier-vscode
code --install-extension formulahendry.dotnet
code --install-extension franneck94.c-cpp-runner
code --install-extension gcc.
```

---

### 0.3.9 Visual Studio Code Extension List (3)

```
vscode-plugin-billionbottle
code --install-extension geeklearningio.graphviz-markdown-preview
code --install-extension geyao.html-snippets
code --install-extension GitHub.copilot-nightly
code --install-extension GrapeCity.gc-excelviewer
code --install-extension Ionide.Ionide-fsharp
code --install-extension ionut-botizan.vscode-cypher-ql
code --install-extension ipedrazas.kubernetes-snippets
code --install-extension JakeWilson.vscode-picture
code --install-extension James-Yu.latex-workshop
code --install-extension JasonMejane.base64viewer
code --install-extension jasonnutter.search-node-modules
code --install-extension jebbs.plantuml
code --install-extension jeff-hykin.better-cpp-syntax
code --install-extension Katacoda.vscode
code --install-extension KenDomino.Antlrvsix-vscode
code --install-extension l7ssha.tag-inserter
code --install-extension lolkush.quickstart
code --install-extension marp-team.marp-vscode
code --install-extension mindaro-dev.file-downloader
code --install-extension mindaro.mindaro
code --install-extension ms-azuretools.vscode-docker
code --install-extension MS-CEINTL.vscode-language-pack-tr
```

---

### 0.3.10 Visual Studio Code Extension List (4)

```
code --install-extension ms-dotnettools.csharp
code --install-extension ms-dotnettools.dotnet-interactive-vscode
code --install-extension ms-dotnettools.vscode-dotnet-pack
code --install-extension ms-dotnettools.vscode-dotnet-runtime
code --install-extension ms-kubernetes-tools.vscode-aks-tools
```

```
code --install-extension ms-kubernetes-tools.vscode-kubernetes-tools
code --install-extension ms-python.python
code --install-extension ms-python.vscode-pylance
code --install-extension ms-toolsai.jupyter
code --install-extension ms-toolsai.jupyter-keymap
code --install-extension ms-toolsai.jupyter-renderers
code --install-extension ms-vscode-remote.remote-containers
code --install-extension ms-vscode-remote.remote-ssh
code --install-extension ms-vscode-remote.remote-ssh-edit
code --install-extension ms-vscode-remote.remote-wsl
```

---

### 0.3.11 Visual Studio Code Extension List (5)

```
code --install-extension ms-vscode.azure-account
code --install-extension ms-vscode.brackets-keybindings
code --install-extension ms-vscode.brackets-pack
code --install-extension ms-vscode.cmake-tools
code --install-extension ms-vscode.cppTools
code --install-extension ms-vscode.cppTools-extension-pack
code --install-extension ms-vscode.cppTools-themes
code --install-extension ms-vscode.live-server
code --install-extension ms-vsliveshare.vsliveshare
code --install-extension oleg-shilo.cs-script
code --install-extension PascalReitermann93.vscode-yaml-sort
```

---

### 0.3.12 Visual Studio Code Extension List (6)

```
code --install-extension Pivotal.vscode-boot-dev-pack
code --install-extension Pivotal.vscode-concourse
code --install-extension Pivotal.vscode-manifest-yaml
code --install-extension Pivotal.vscode-spring-boot
code --install-extension PKief.material-icon-theme
code --install-extension platformio.platformio-ide
code --install-extension pranaygp.vscode-css-peek
code --install-extension redhat.fabric8-analytics
code --install-extension redhat.java
code --install-extension redhat.vscode-commons
code --install-extension redhat.vscode-xml
code --install-extension redhat.vscode-yaml
code --install-extension ritwickdey.LiveServer
code --install-extension sidthesloth.html5-boilerplate
code --install-extension TaodongWu.ejs-snippets
code --install-extension tht13.python
code --install-extension tomoki1207.pdf
code --install-extension twxs.cmake
code --install-extension vadimcn.vscode-lldb
```

---

### 0.3.13 Visual Studio Code Extension List (7)

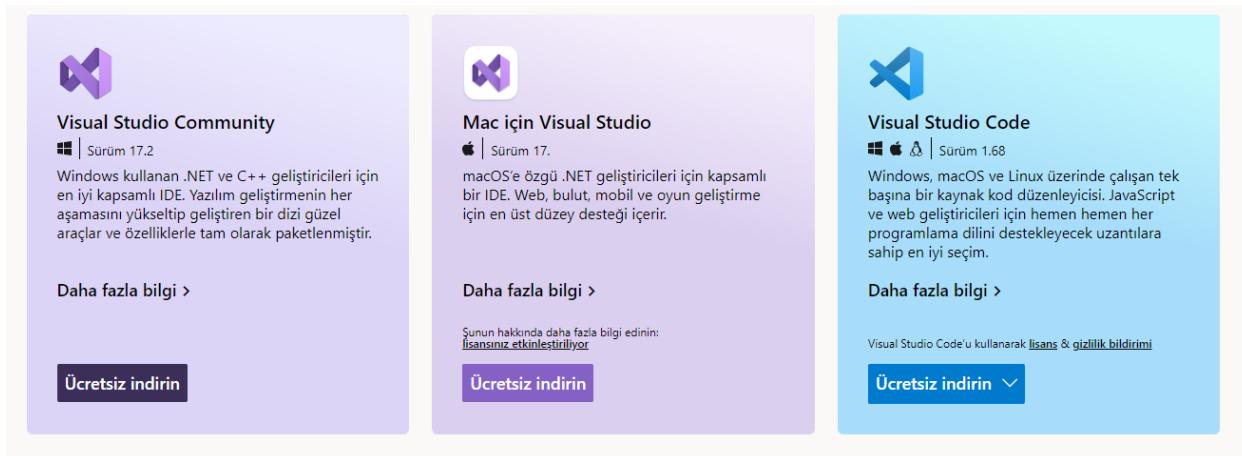
```
code --install-extension VisualStudioExptTeam.intellicode-api-usage-examples
code --install-extension VisualStudioExptTeam.vscodeintellicode
code --install-extension vscjava.vscode-java-debug
code --install-extension vscjava.vscode-java-dependency
```

```
code --install-extension vscjava.vscode-java-pack
code --install-extension vscjava.vscode-java-test
code --install-extension vscjava.vscode-maven
code --install-extension vscjava.vscode-spring-boot-dashboard
code --install-extension vscjava.vscode-spring-initializr
code --install-extension walkme.HTML5-extension-pack
code --install-extension webfreak.debug
code --install-extension well-ar.plantuml
code --install-extension wildboar asn1
code --install-extension Zignd.html-css-class-completion
```

### 0.3.14 Visual Studio Community Edition (Install / Compile / Run / Debug) (1)

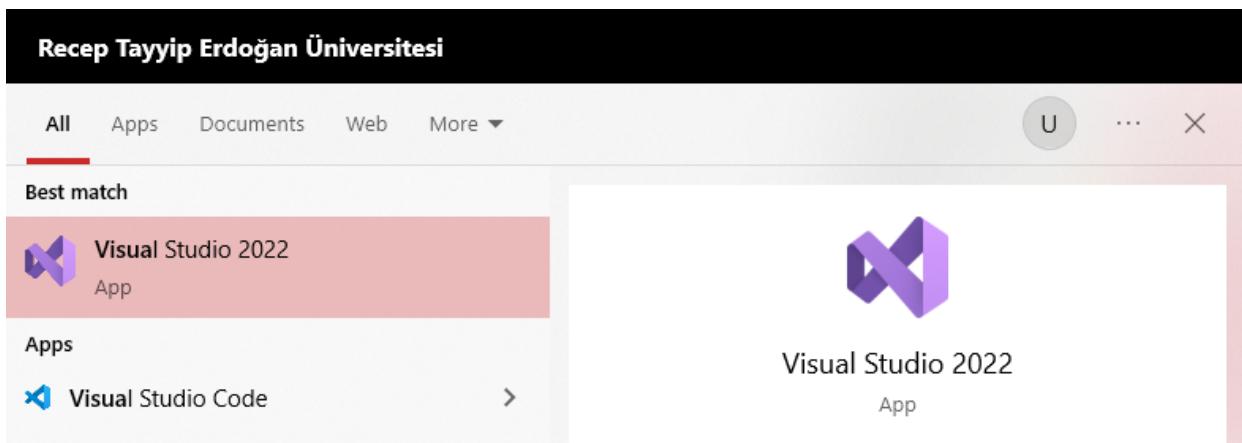
- Download and install Visual Studio Community Edition
- Select All Development Environments from Installer.

Ücretsiz Geliştirici Yazılımları ve Hizmetleri - Visual Studio<sup>16</sup>



### 0.3.15 Visual Studio Community Edition (Install / Compile / Run / Debug) (2)

- After installation open Visual Studio2022<sup>17</sup> from the menu.



<sup>16</sup><https://visualstudio.microsoft.com/tr/free-developer-offers/>

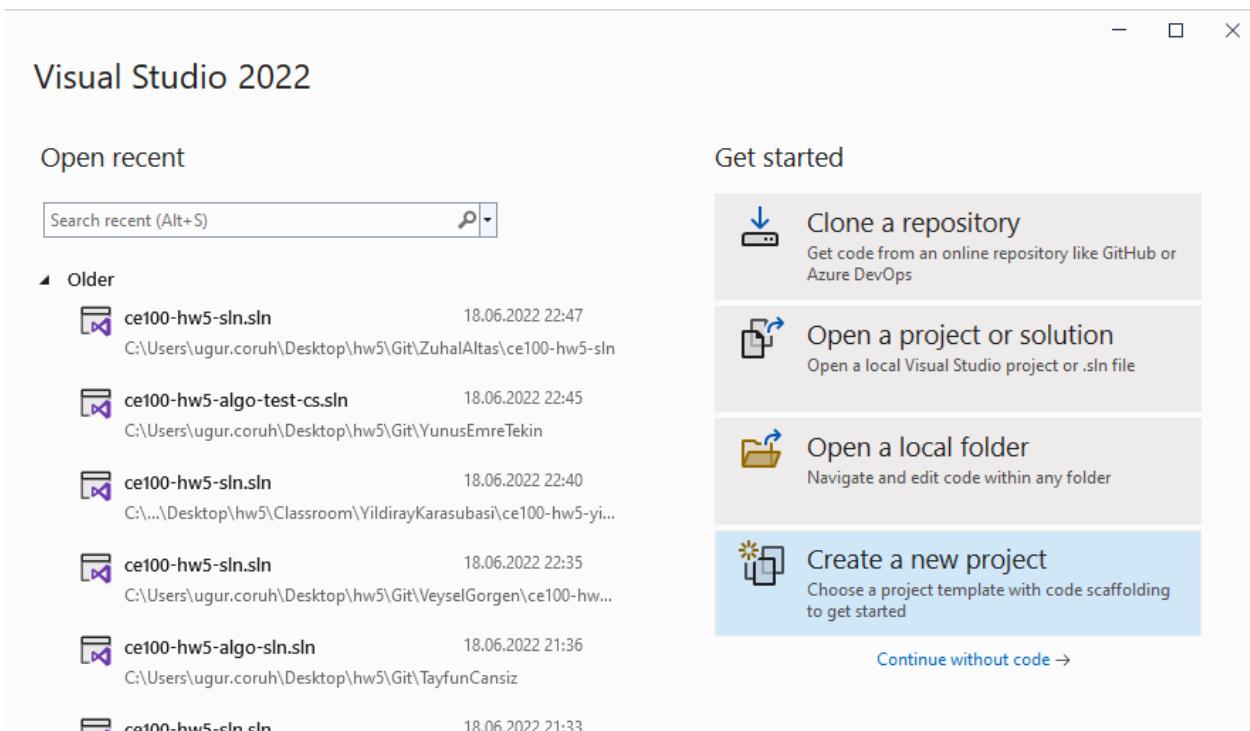
### 0.3.16 Visual Studio Community Edition (Install / Compile / Run / Debug) (3)

- The application will start...



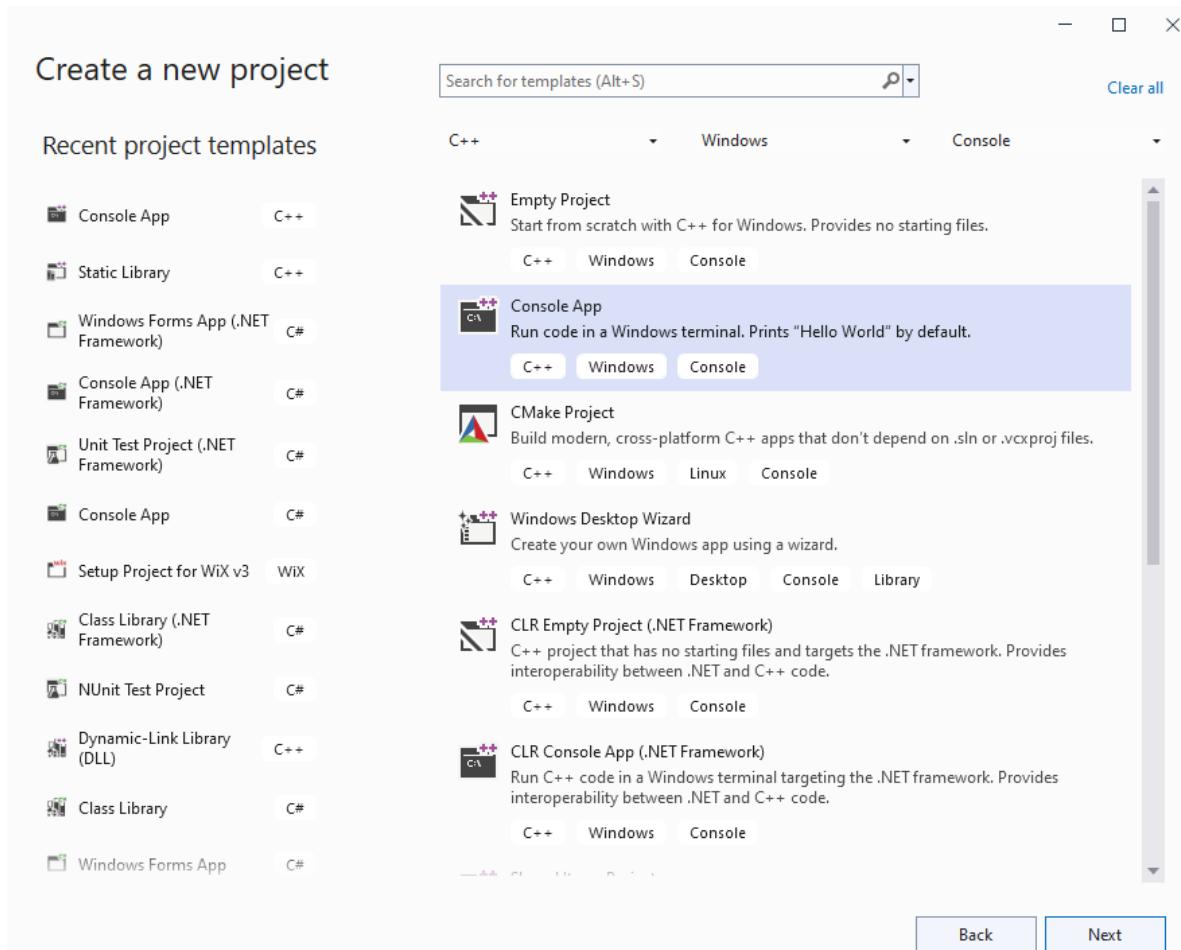
### 0.3.17 Visual Studio Community Edition (Install / Compile / Run / Debug) (4)

- From Opening Window Select Create a new project



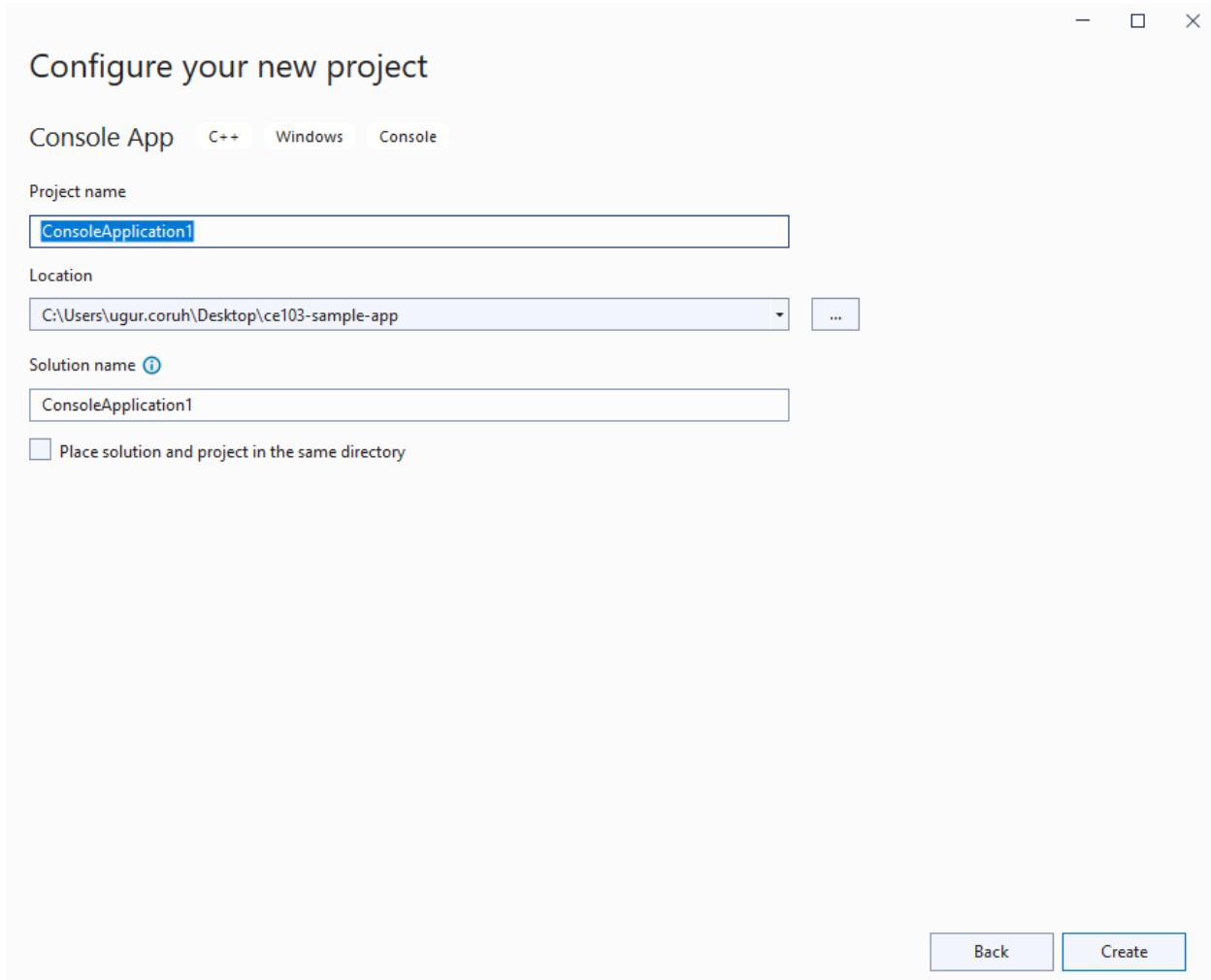
### 0.3.18 Visual Studio Community Edition (Install / Compile / Run / Debug) (5)

- There will be several options, you can review them.
- Select Windows, C++, Console Application from Combobox.
- Select Console Application



### 0.3.19 Visual Studio Community Edition (Install / Compile / Run / Debug) (6)

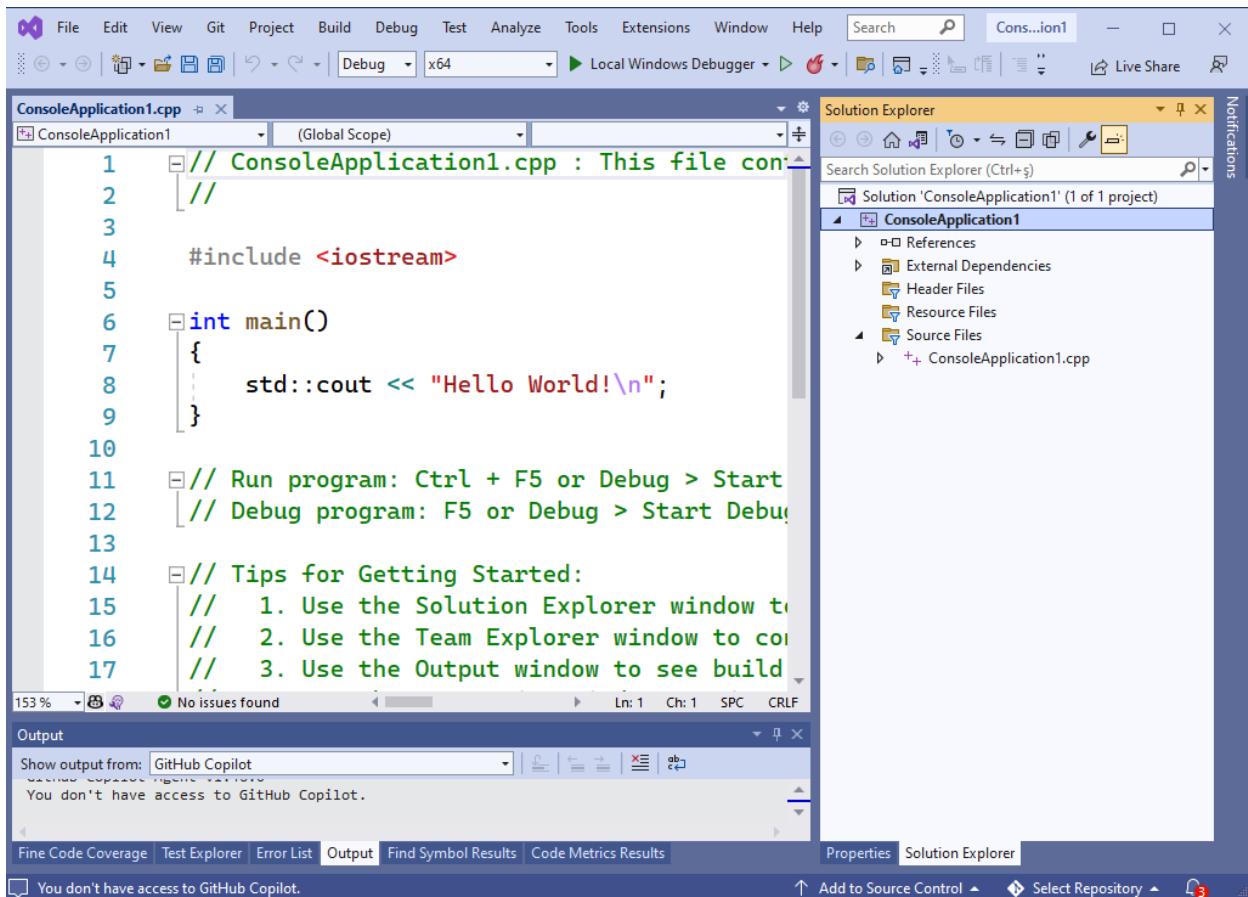
- Give a solution and project name.
- Select save location



---

### 0.3.20 Visual Studio Community Edition (Install / Compile / Run / Debug) (7)

- You will have C++ basic Hello World application.



### 0.3.21 Visual Studio Community Edition (Install / Compile / Run / Debug) (8)

- You will have C++ basic Hello World application.

```
// ConsoleApplication1.cpp : This file contains the 'main' function. Program execution begins and ends
// 

#include <iostream>

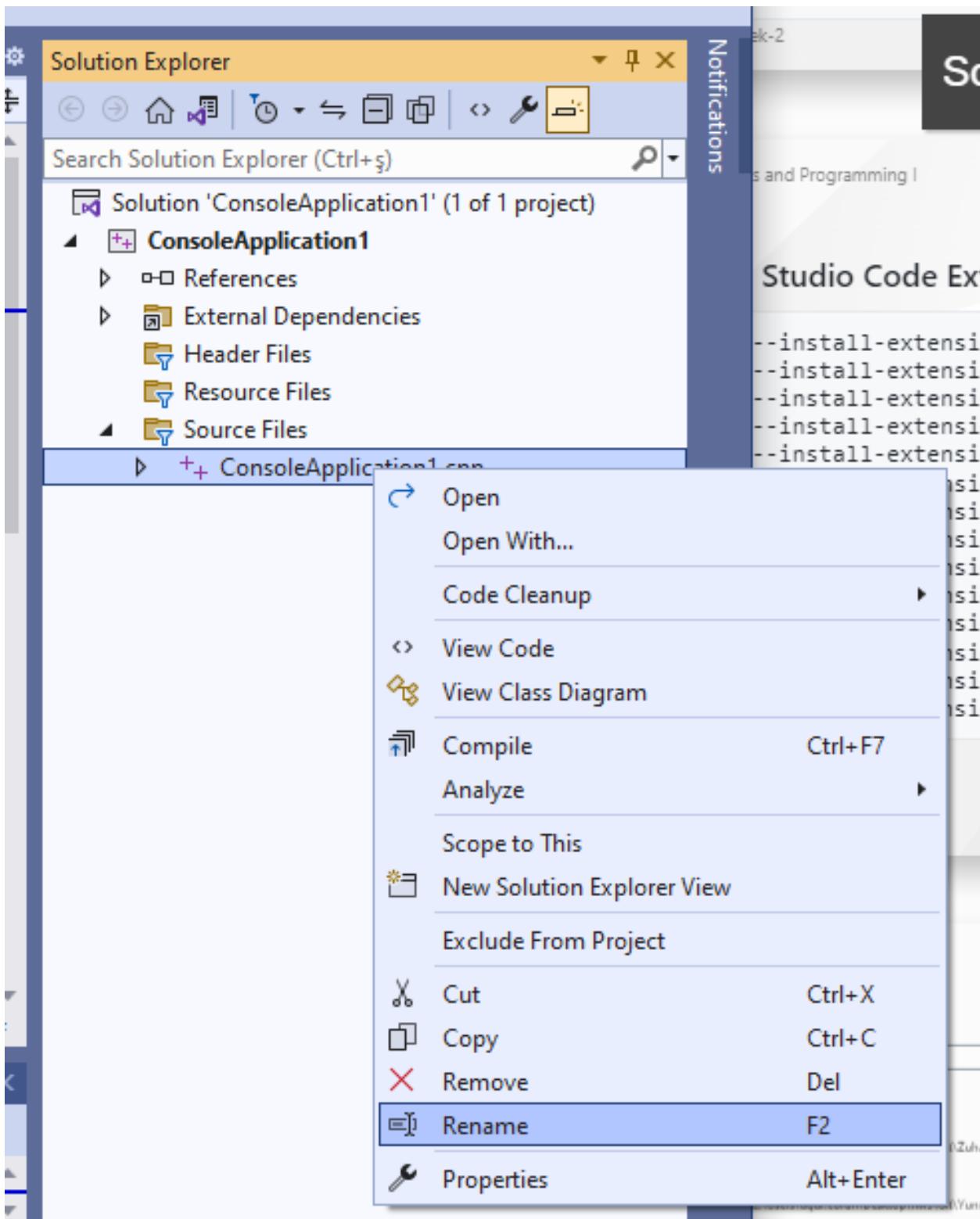
int main()
{
    std::cout << "Hello World!\n";
}

// Run program: Ctrl + F5 or Debug > Start Without Debugging menu
// Debug program: F5 or Debug > Start Debugging menu

// Tips for Getting Started:
// 1. Use the Solution Explorer window to add/manage files
// 2. Use the Team Explorer window to connect to source control
// 3. Use the Output window to see build output and other messages
// 4. Use the Error List window to view errors
// 5. Go to Project > Add New Item to create new code files, or Project > Add Existing Item to add existing code
// 6. In the future, to open this project again, go to File > Open > Project and select the .sln file
```

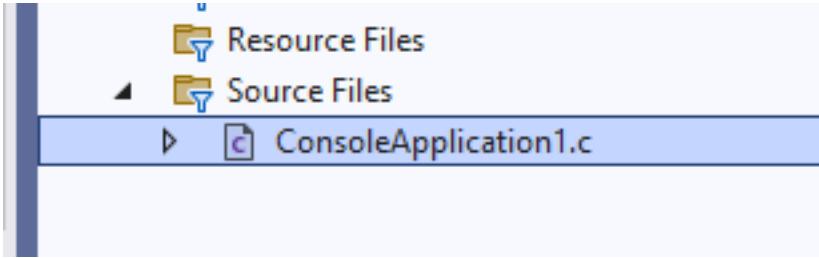
### 0.3.22 Visual Studio Community Edition (Install / Compile / Run / Debug) (9)

- We need to rename the file extension to c from cpp



### 0.3.23 Visual Studio Community Edition (Install / Compile / Run / Debug) (10)

- If you compile the source C compiler will be used.



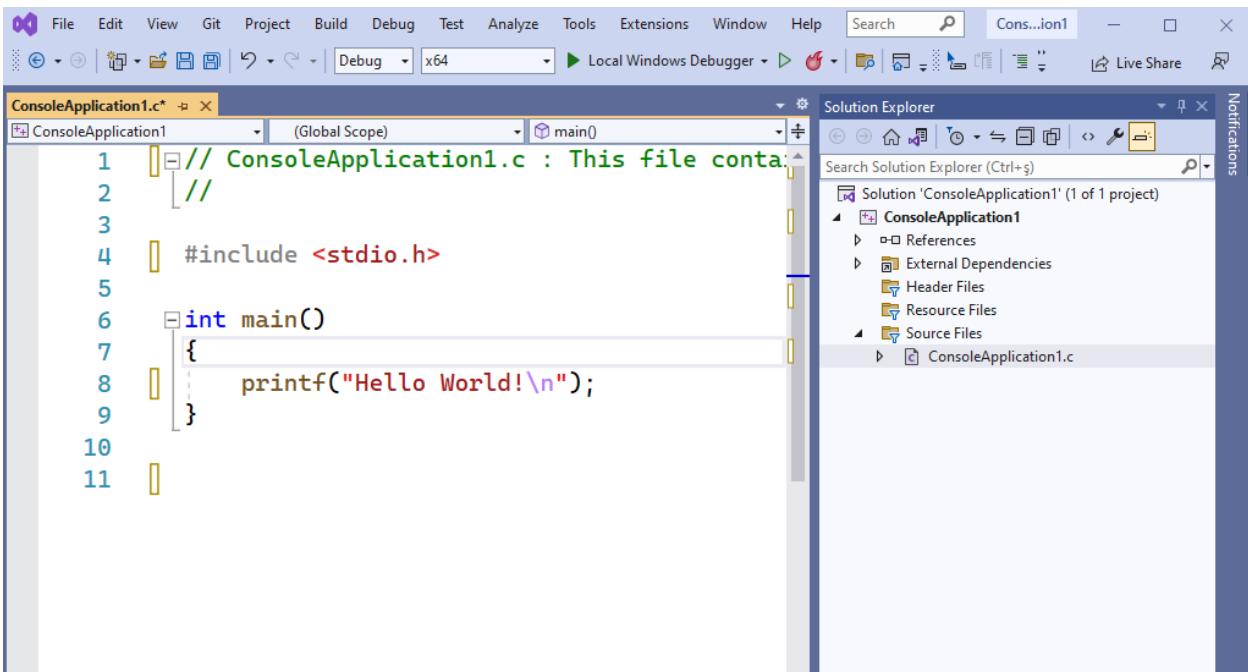
### 0.3.24 Visual Studio Community Edition (Install / Compile / Run / Debug) (11)

- We need to update our source for C as follows

```
// ConsoleApplication1.c : This file contains the 'main' function. Program execution begins and ends there.
//  
  
#include <stdio.h>  
  
int main(){  
    printf("Hello World!\n");  
}
```

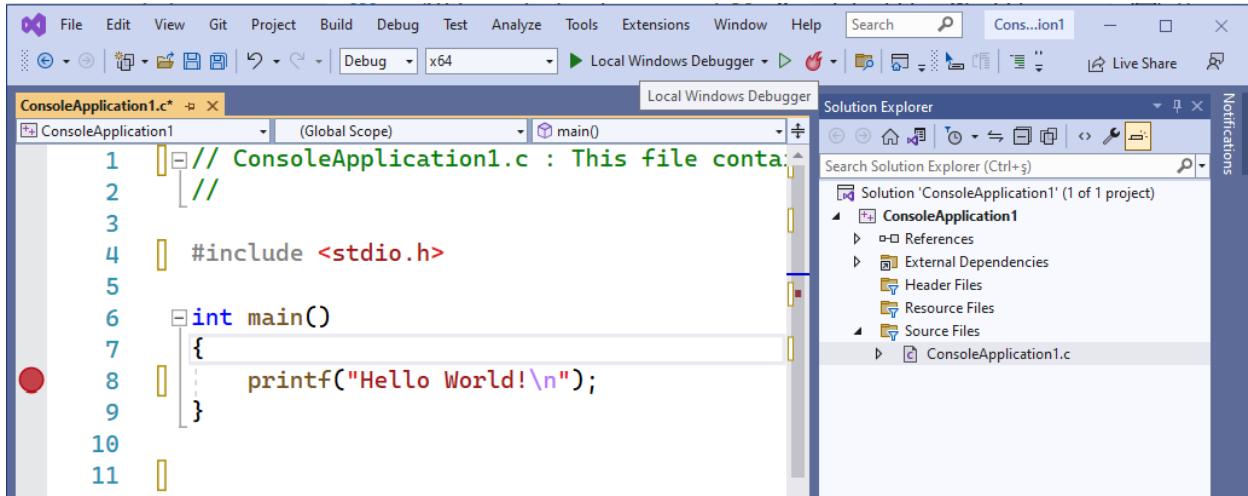
### 0.3.25 Visual Studio Community Edition (Install / Compile / Run / Debug) (12)

- We need to update our source for C as follows



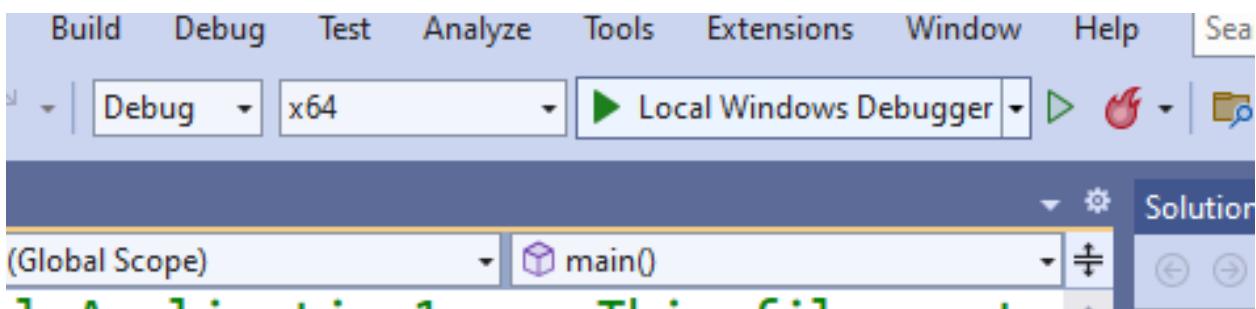
### 0.3.26 Visual Studio Community Edition (Install / Compile / Run / Debug) (13)

- Put a breakpoint by clicking the following location. Breakpoints will be stop points for our debugging operations.



### 0.3.27 Visual Studio Community Edition (Install / Compile / Run / Debug) (14)

- Then select Debug configuration and according to your operating system select x64 or x86 configuration and click Local Windows Debugger



### 0.3.28 Visual Studio Community Edition (Install / Compile / Run / Debug) (15)

- Update your source code as follow

```
// ConsoleApplication1.c : This file contains the 'main' function. Program execution begins and ends there.

#include <stdio.h>

int sum(int input1, int input2);

int main(){
    int number = 5;
    printf("Before Increment : %d\n", number);
    number = sum(number, 5);
    printf("After Increment : %d\n", number);
}

int sum(int input1, int input2){
    return input1 + input2;
}
```

### 0.3.29 Visual Studio Community Edition (Install / Compile / Run / Debug) (16)

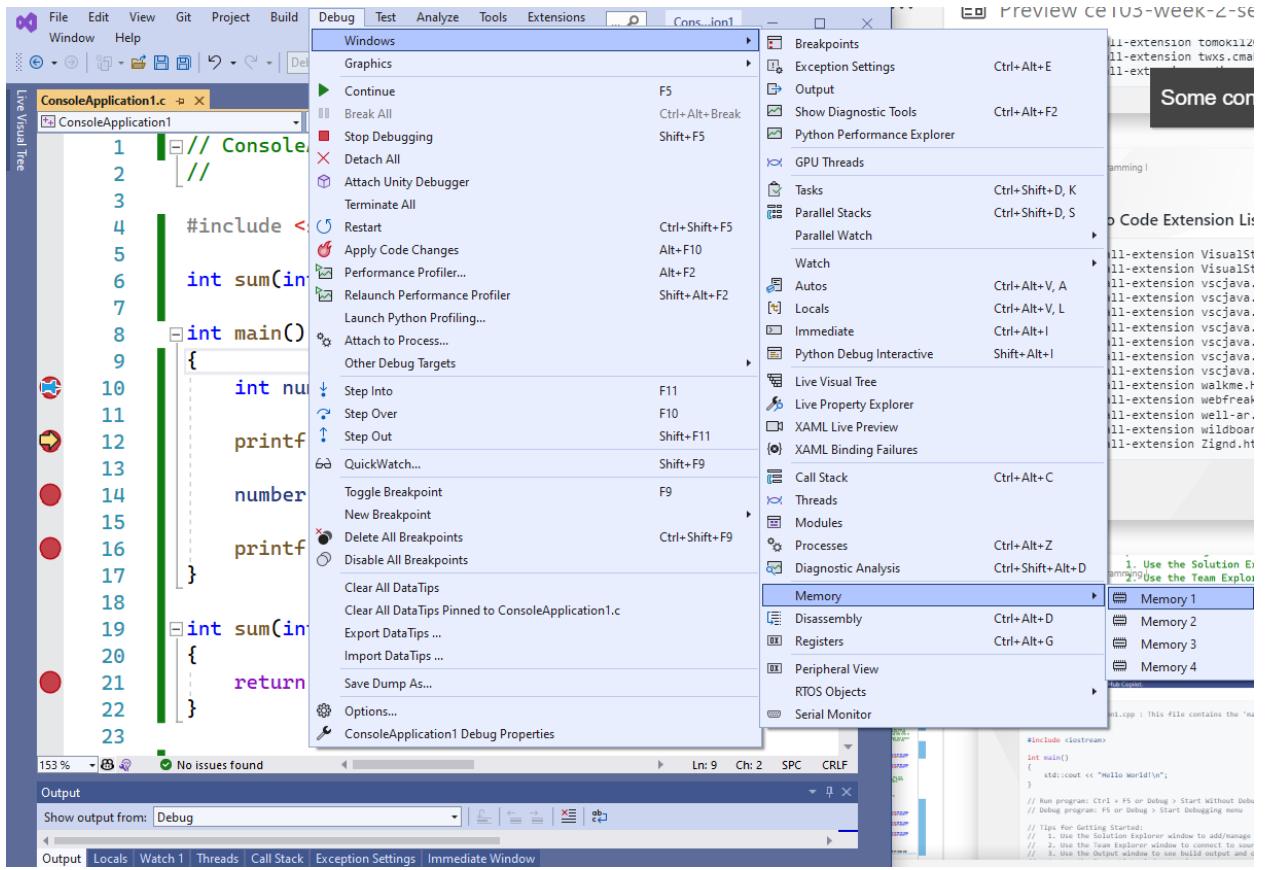
- Put the following breakpoints and run the debugger. On number, the variable pins the variable to see its value in real-time.

The screenshot shows the Visual Studio Community Edition interface with a C program named `ConsoleApplication1.c`. The code defines a `main()` function that increments a variable `number` and prints its value before and after the increment. A green vertical line indicates a breakpoint is set on the assignment statement `number = sum(number, 5);`. A tooltip for the variable `number` shows the value `5`. The `Output` window shows the command `Show output from: Debug`.

```
// ConsoleApplication1.c : This file contains the 'main' function.//  
#include <stdio.h>  
int sum(int input1, int input2);  
int main()  
{  
    int number = 5; // Breakpoint is here  
    printf("Before Increment : %d\n", number);  
    number = sum(number, 5);  
    printf("After Increment : %d\n", number);  
}  
int sum(int input1, int input2)  
{  
    return input1 + input2;  
}
```

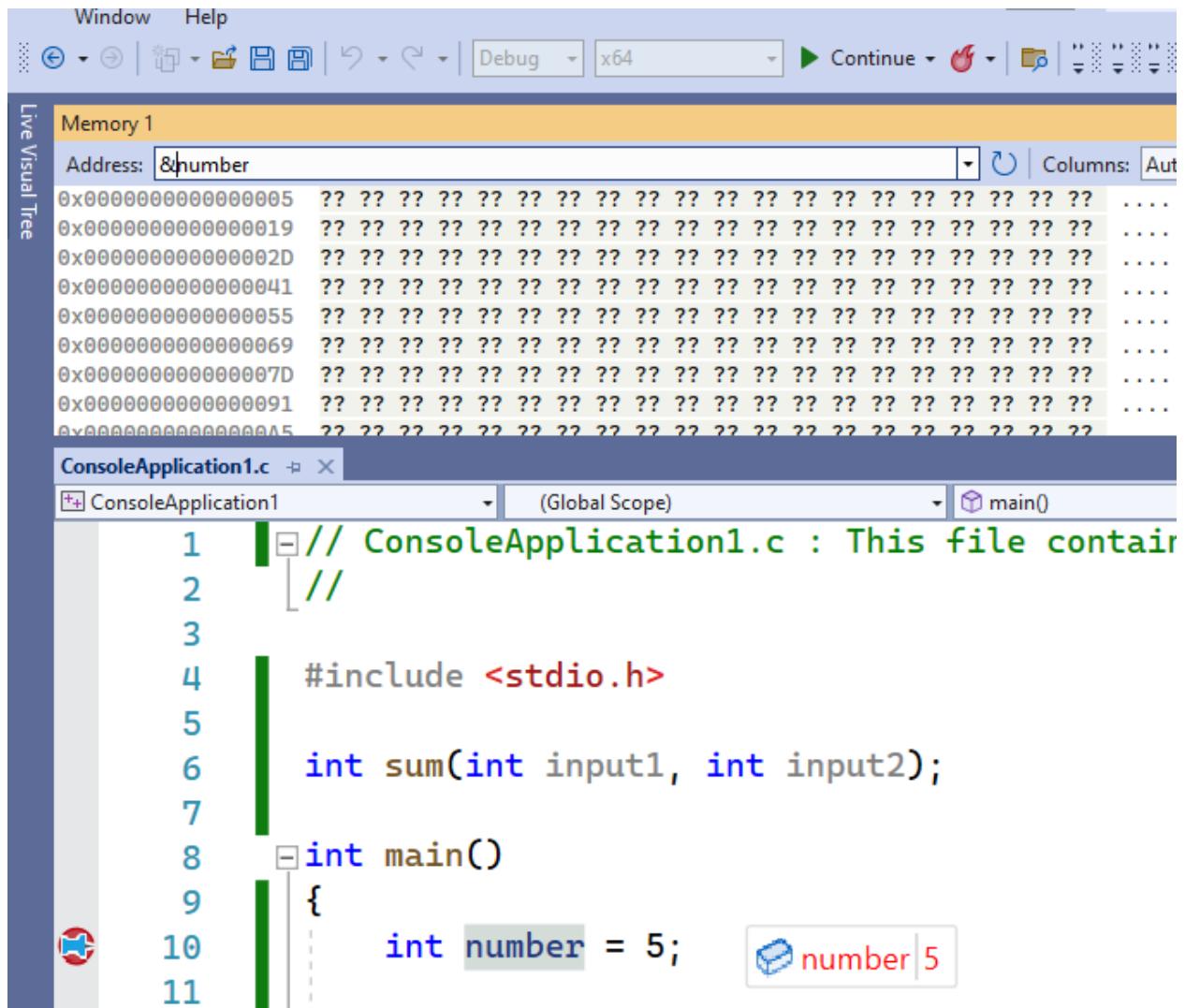
### 0.3.30 Visual Studio Community Edition (Install / Compile / Run / Debug) (17)

- Open Debug->Windows->Memory->Memory1 to see value in memory



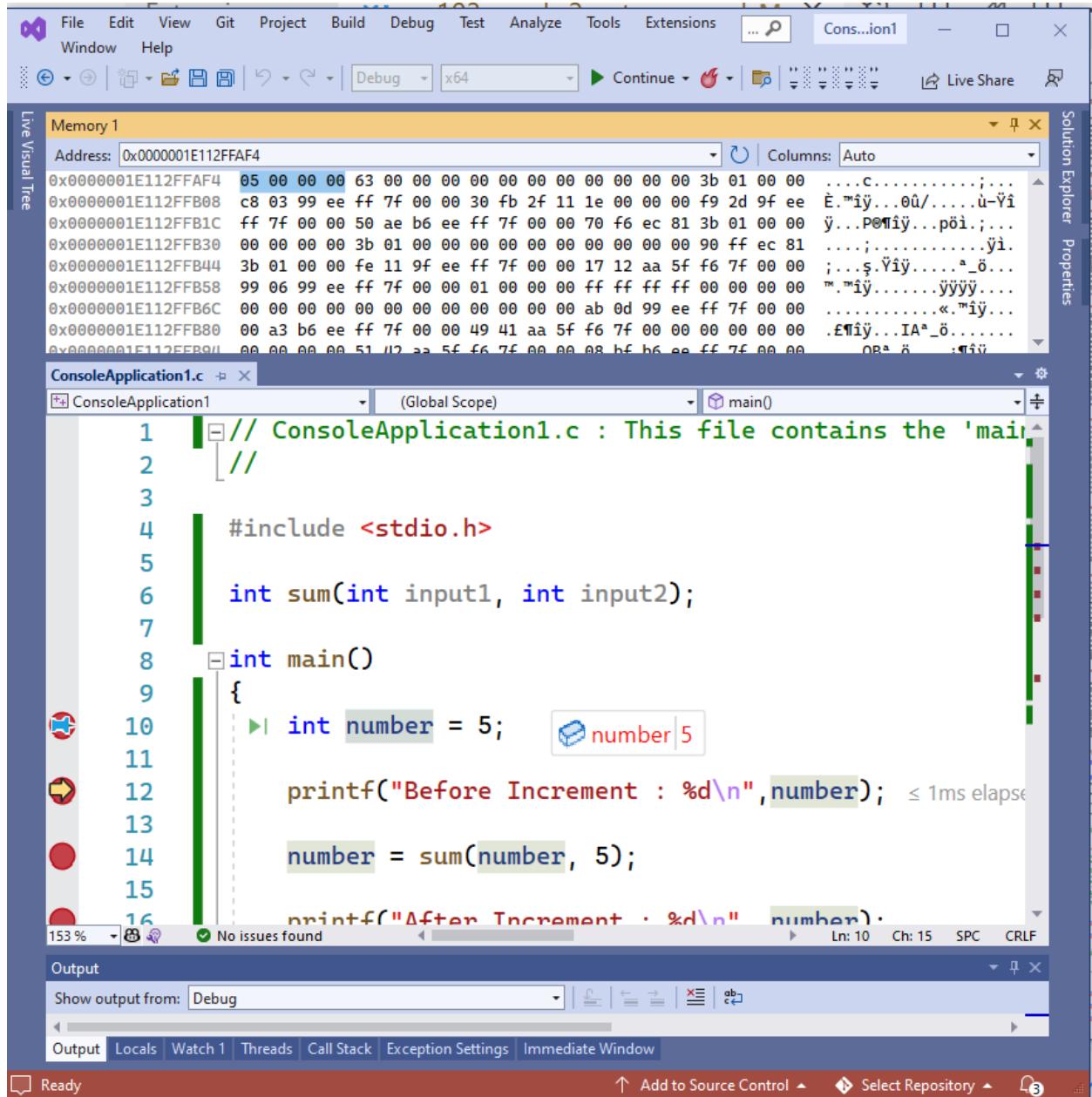
### 0.3.31 Visual Studio Community Edition (Install / Compile / Run / Debug) (18)

- In the memory window copy variable name (number) with address operator (&) and then (&number) press enter.



### 0.3.32 Visual Studio Community Edition (Install / Compile / Run / Debug) (19)

- You can see its value in memory with hexadecimal form (05 00 00 00)



### 0.3.33 Visual Studio Community Edition (Install / Compile / Run / Debug) (20)

- If you change value with pinned control your memory value and your current value will be updated. 20 in hexadecimal 0x14 (integer is 4 bytes length for this reason memory shows 14 00 00 00)

The screenshot shows the Visual Studio Community Edition interface. In the top left, there's a code editor with the following C code:

```
int main()
{
    int number = 5; // number 20
    printf("Before Increment : %d\n", number);
}
```

To the right of the code editor is a status bar showing the variable 'number' with a value of '20'. Below the code editor is a 'Memory 1' window. On the far left of this window is a vertical toolbar labeled 'Live Visual Tree'. The main area of the window is titled 'Memory 1' and shows memory starting at address 0x0000001E112FFAF4. The memory dump is as follows:

Address	Value
0x0000001E112FFAF4	14 00 00 00 63 00 00
0x0000001E112FFB08	c8 03 99 ee ff 7f 00
0x0000001E112FFB1C	ff 7f 00 00 50 ae b6
0x0000001E112FFB30	00 00 00 00 3b 01 00
0x0000001E112FFB44	3b 01 00 00 fe 11 9f

### 0.3.34 Visual Studio Community Edition (Install / Compile / Run / Debug) (21)

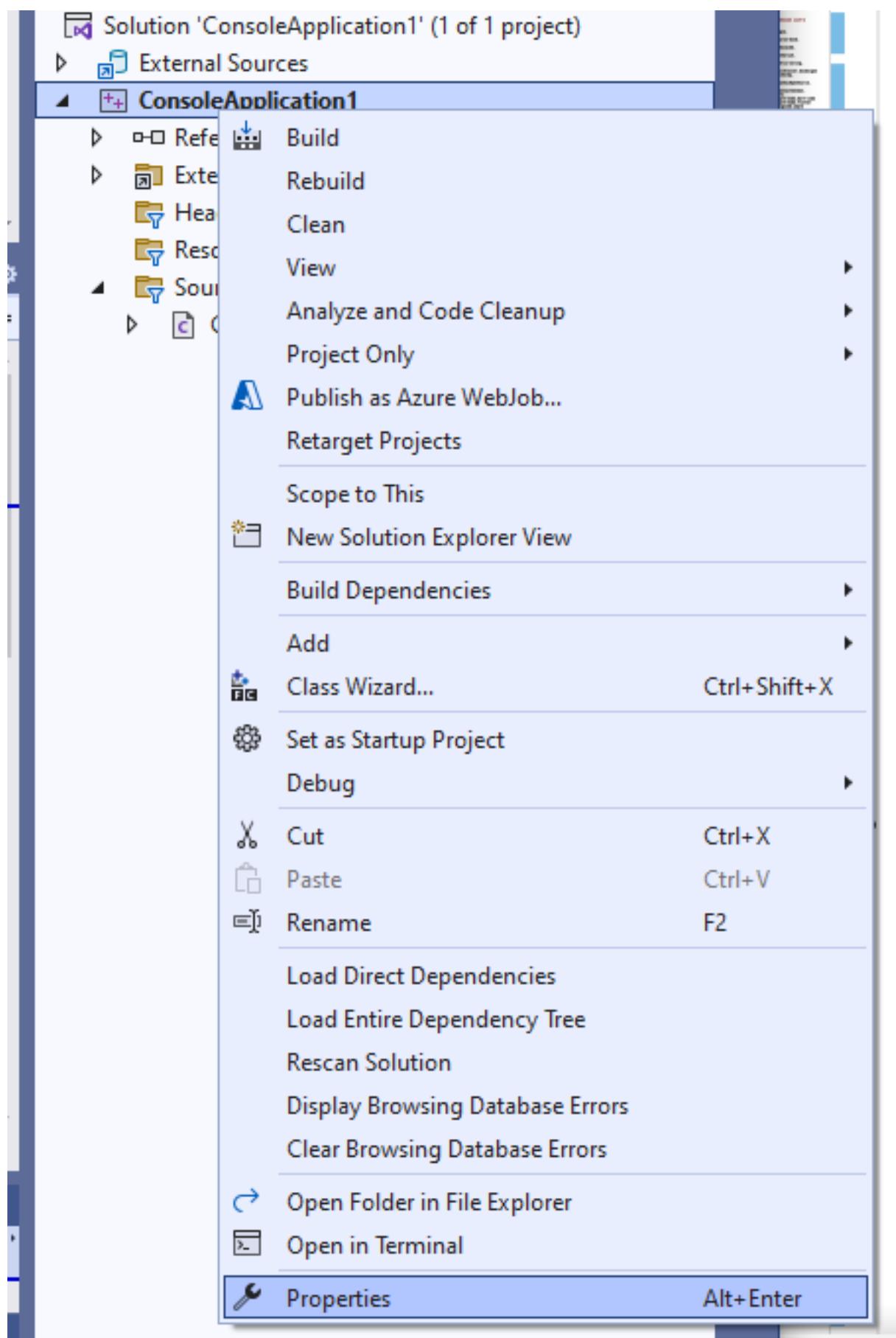
- If you close some windows such as solution explorer, and properties windows you can open them from the View menu.

View		
Code		
 Solution Explorer		
 Git Changes	Ctrl+0, Ctrl+G	
 Git Repository	Ctrl+0, Ctrl+R	
 Team Explorer	Ctrl+ç, Ctrl+M	
 Server Explorer		
 SQL Server Object Explorer	Ctrl+ç, Ctrl+S	
 Test Explorer		
 Cookiecutter Explorer		
<hr/>		
 Call Hierarchy		
 Class View	Ctrl+Shift+C	
 Code Definition Window	Ctrl+ç, D	
 Object Browser	Ctrl+Alt+J	
<hr/>		
 Error List	Ctrl+ç, E	
 Output	Ctrl+Alt+O	
 Task List	Ctrl+ç, T	
 Toolbox	Ctrl+Alt+X	
 Notifications		
 Terminal	Ctrl+"	
<hr/>		
Other Windows		
<hr/>		
Toolbars		
<hr/>		
 Full Screen	Shift+Alt+Enter	
 All Windows	Shift+Alt+M	
<hr/>		
 Navigate Backward	Ctrl+-	
 Navigate Forward	Ctrl+Shift+-	
<hr/>		
Next Task		
<hr/>		
Previous Task		
<hr/>		
 Properties Window	F4	
 Property Pages	Shift+F4	

---

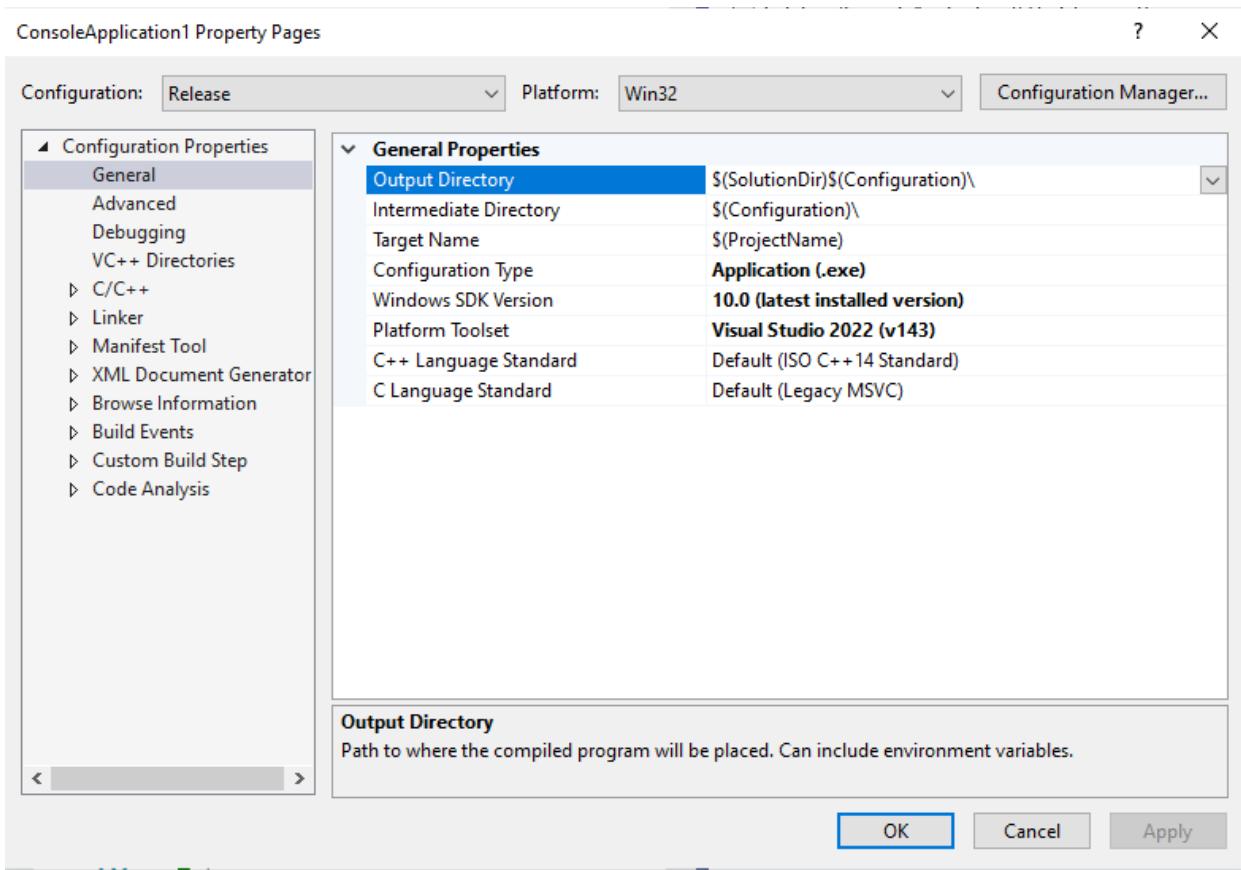
### **0.3.35 Visual Studio Community Edition (Install / Compile / Run / Debug) (22)**

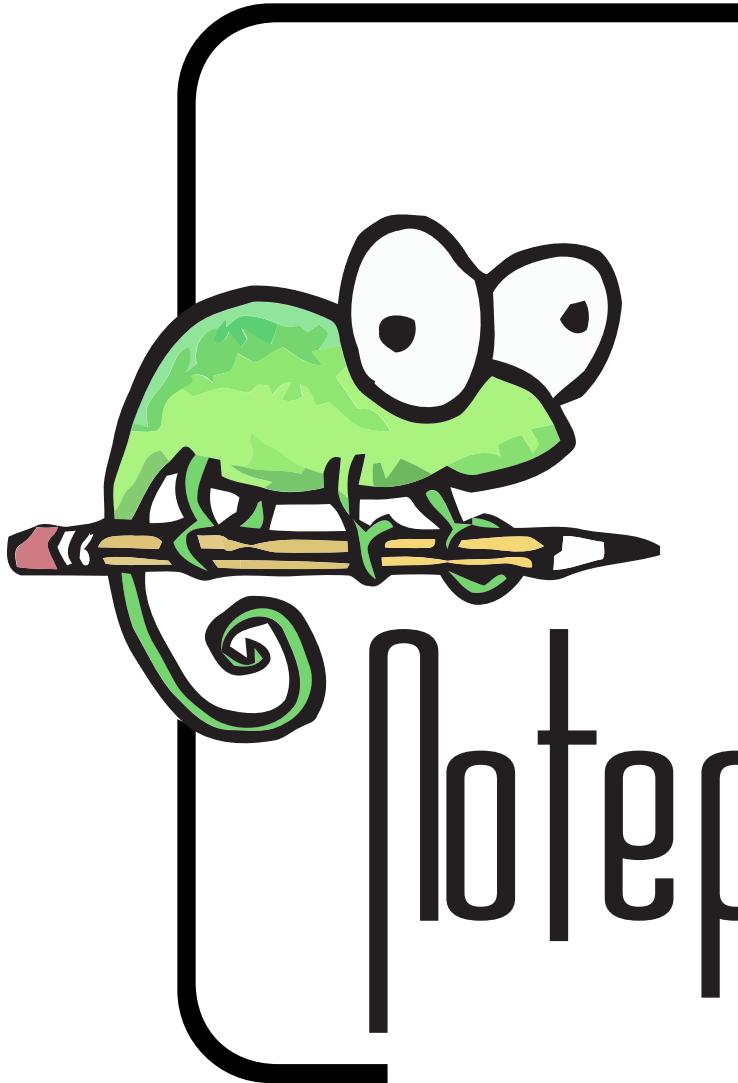
- Solution and Projects have several configurations for each setup such as Release - x86, Release-x64, Debug- x86, and Debug-x64 you need to configure all of them for your requirements. You can access configurations by right-clicking to project and then selecting properties.



### 0.3.36 Visual Studio Community Edition (Install / Compile / Run / Debug) (23)

- Project properties has several settings





#### 0.3.36.1 Notepad++ (Install / Compile ) (1)

- Please download Notepad++ from the following link
    - Downloads | Notepad++<sup>17</sup>
- 

#### 0.3.36.2 Notepad++ (Install / Compile ) (2) Download and install MinGW or LLVM compiler (if you downloaded then skip this step)

##### MinGW installer (gcc / g++)

- A complete runtime environment for gcc
    - <https://sourceforge.net/projects/mingw-w64/>
    - <https://sourceforge.net/projects/mingw-w64/files/latest/download>
  - w64devkit is a portable C and C++ development kit for x64 (and x86) Windows.
    - <https://www.mingw-w64.org/downloads/#w64devkit>
  - Also, see the following notes
    - <https://www.hanshq.net/building-gcc.html>
- 

#### 0.3.36.3 Notepad++ (Install / Compile ) (3) LLVM installer (clang)

- Download

---

<sup>17</sup><https://notepad-plus-plus.org/downloads/>

- <https://releases.llvm.org/>
  - Also, use the following notes
    - <https://llvm.org/devmtg/2014-04/PDFs/Talks/clang-cl.pdf>
    - <https://www.hanshq.net/clang-plugin-example.html>
- 

**0.3.36.4 Notepad++ (Install / Compile ) (4)** Open a console with “cmd” and test the following commands if commands are not recognized then set the system environment variable to add `gcc` and `g++` executable paths to the path variable (add to both system and user path variable)

```
gcc --version
```

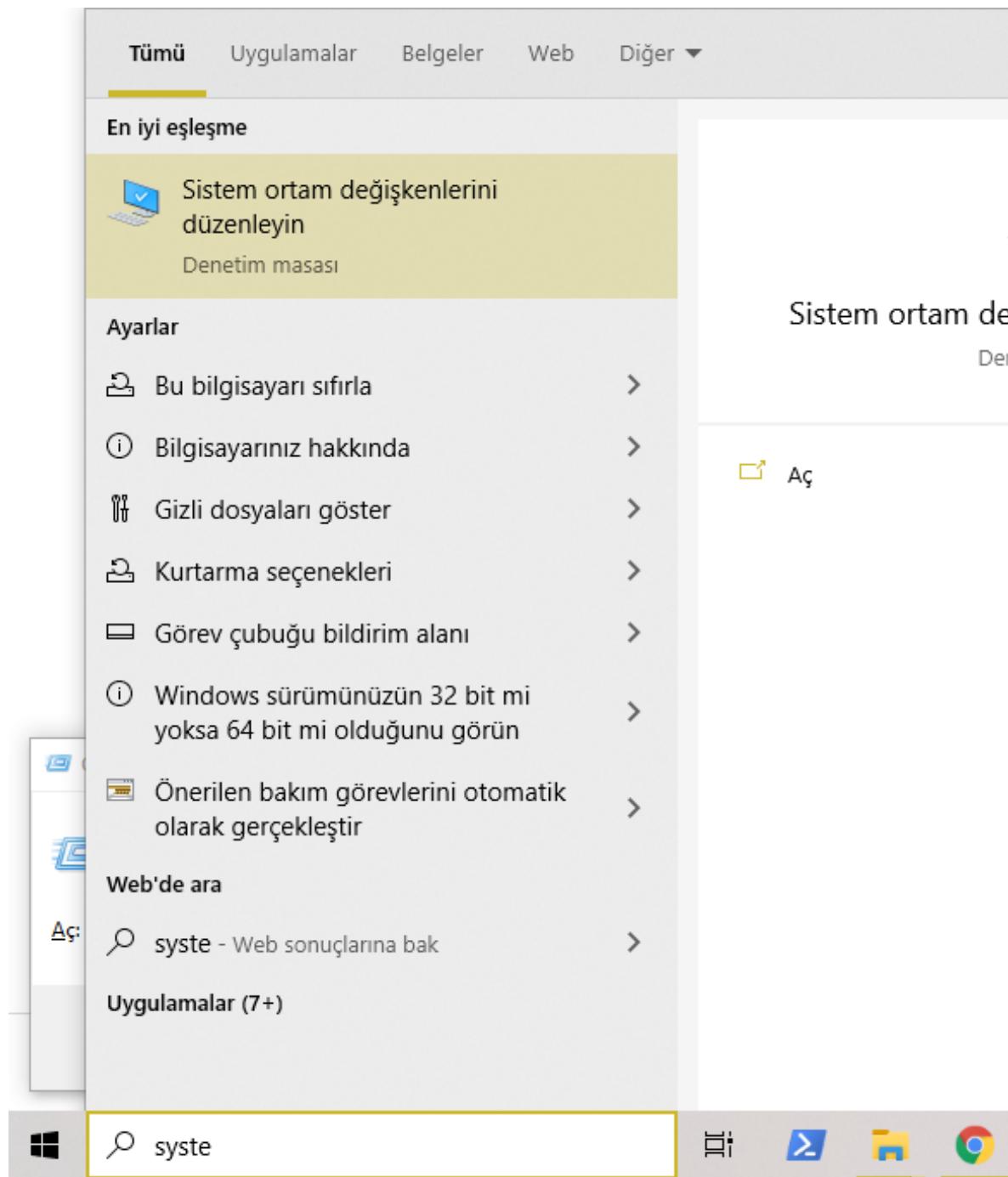
```
g++ --version
```

```
C:\Users\ugur.coruh>gcc --version
gcc (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

---

#### **0.3.36.5 Notepad++ (Install / Compile ) (5)**

- Open system environments to update path variable for `gcc/g++` and `clang`



#### 0.3.36.6 Notepad++ (Install / Compile ) (6)

- Open “Environment Variables”

## Sistem Özellikleri

Bilgisayar Adı Donanım Gelişmiş Sistem Koruması Uzak

Bu değişiklıkların çoğu için Yönetici olarak oturum açmanız gereklidir.

### Performans

Görsel efektler, işlemci zamanlaması, bellek kullanımı ve sanal bellek

Ayarlar...

### Kullanıcı Profilleri

Oturum açmanızla ilgili masaüstü ayarları

Ayarlar...

### Başlangıç ve Kurtarma

Sistem başlangıcı, sistem hataları ve hata ayıklama bilgisi

Ayarlar...

Ortam Değişkenleri...

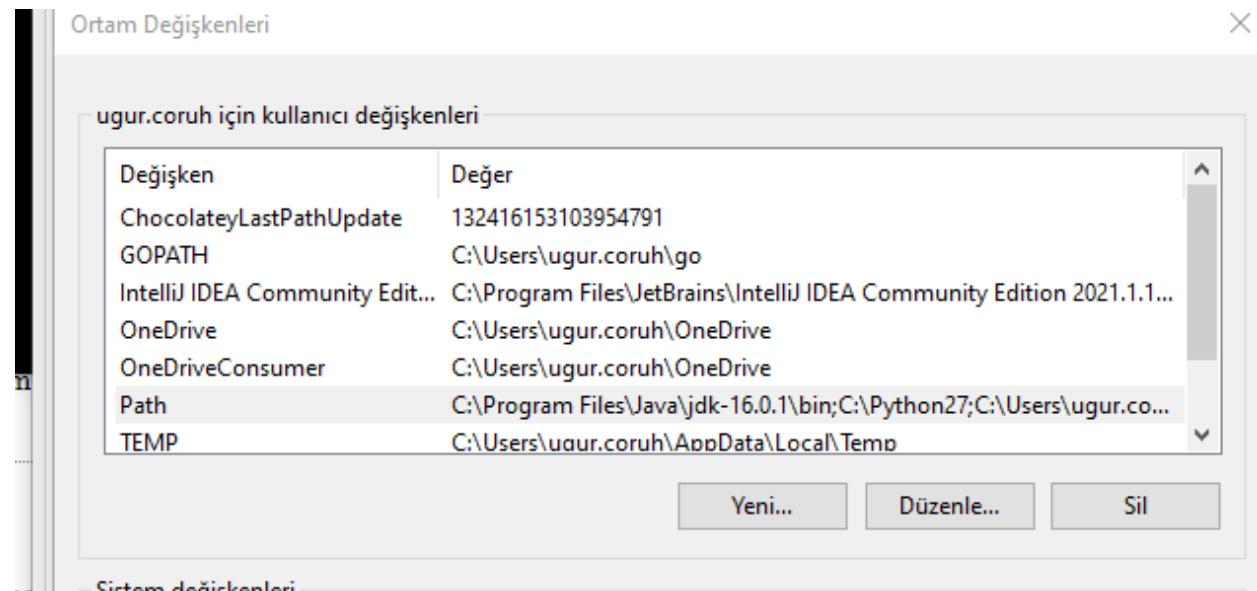
Tamam

İptal

Uygula

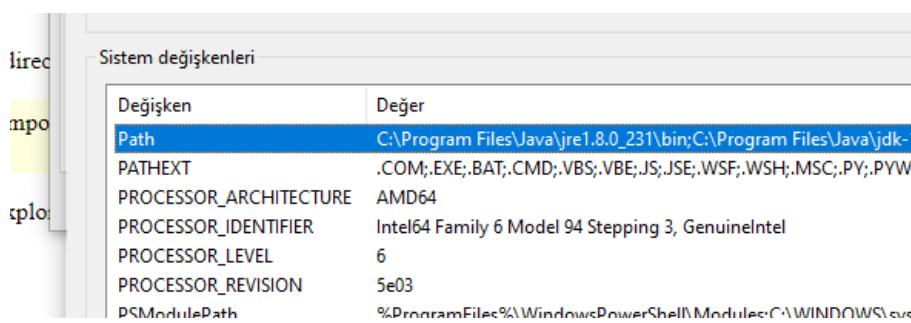
### 0.3.36.7 Notepad++ (Install / Compile ) (7)

- Select path variable from user section.



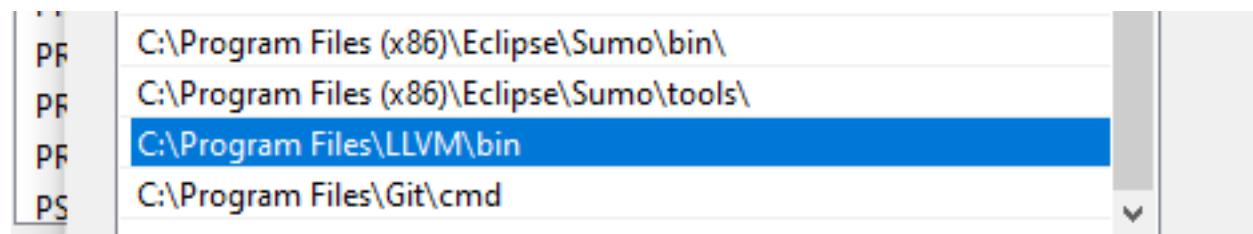
### 0.3.36.8 Notepad++ (Install / Compile ) (8)

- Select path variable from system section.



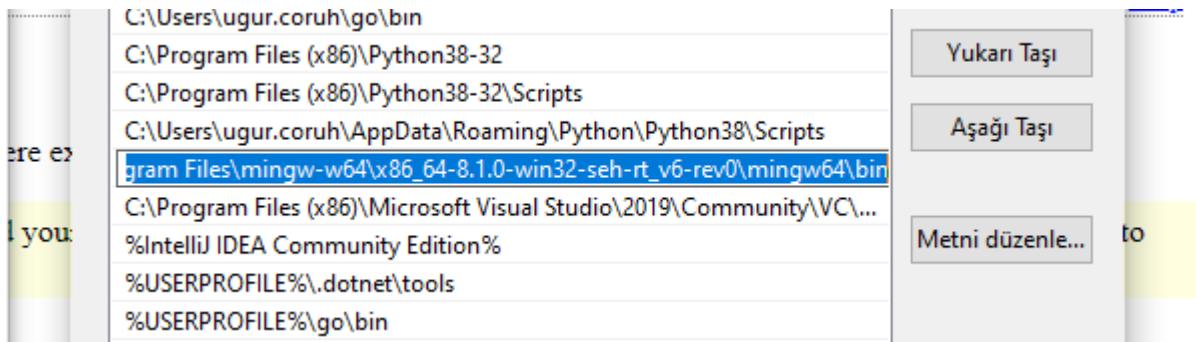
### 0.3.36.9 Notepad++ (Install / Compile ) (9)

- Update variables add MinGW and LLVM to path gcc.exe g++.exe clang.exe will be in bin folders. Then we can run this commands from command line.



### 0.3.36.10 Notepad++ (Install / Compile ) (9)

- Update variables add MinGW and LLVM to path gcc.exe g++.exe clang.exe will be in bin folders. Then we can run this commands from command line.



### 0.3.36.11 Notepad++ (Install / Compile ) (10)

- for gcc.exe, g++.exe and gdb.exe

C:\Program Files\mingw-w64\x86\_64-8.1.0-win32-seh-rt\_v6-rev0\mingw64\bin

### 0.3.36.12 Notepad++ (Install / Compile ) (11)

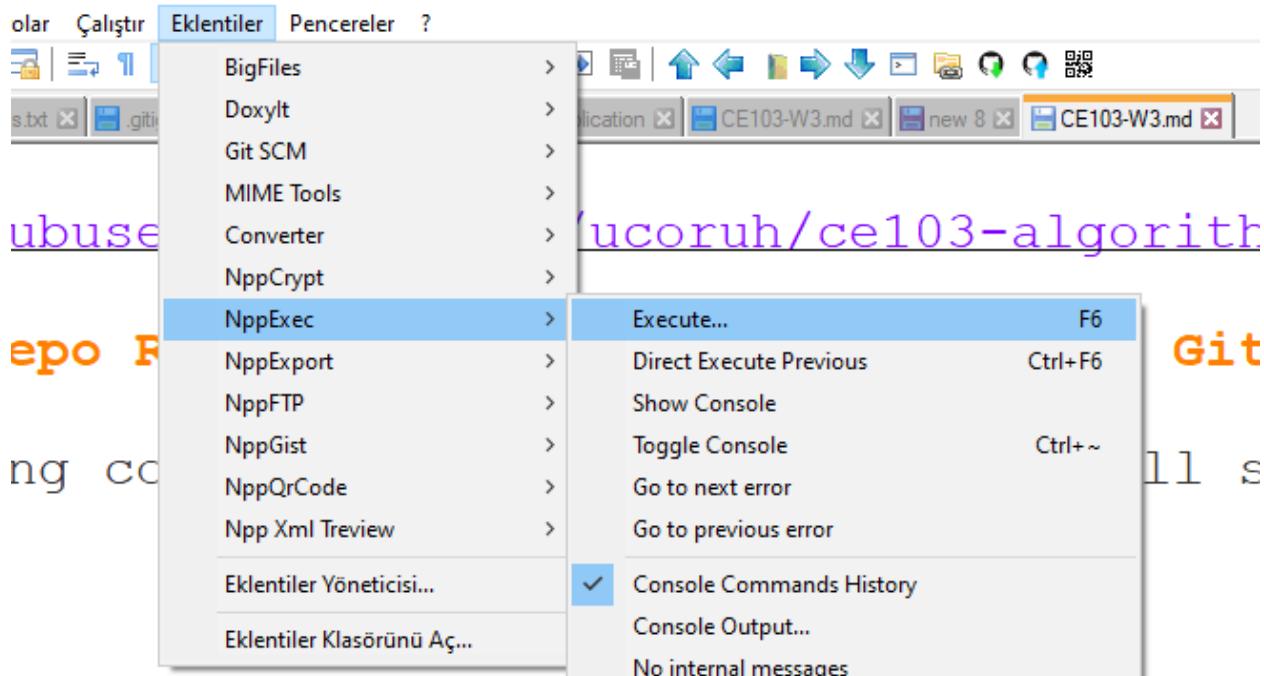
- for clang.exe , lldb.exe we will use the following path

C:\Program Files\LLVM\bin

### 0.3.36.13 Notepad++ (Install / Compile ) (12)

- This folder paths changes according to your setup
- Open NppExec extension (install from extension manager if not exist)

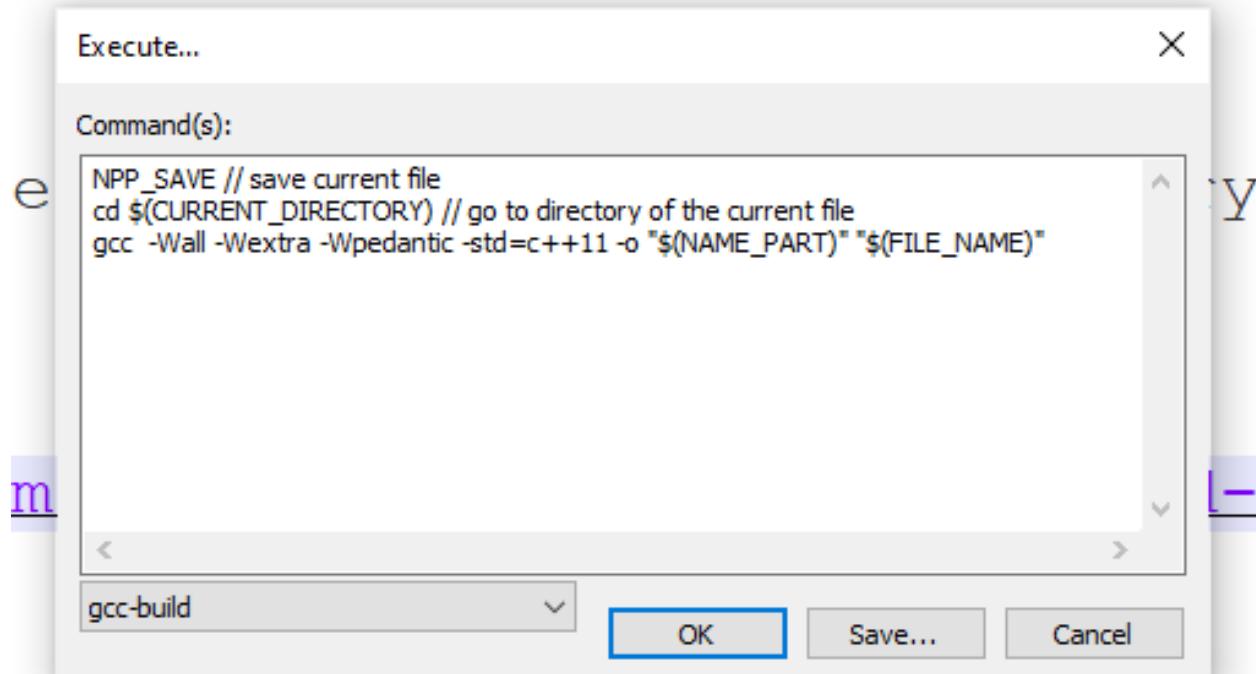
ming\n\lectures\ce103-algorithms-and-programming-\n\Week-3\CE103-W3.md - Notepad++



#### 0.3.36.14 Notepad++ (Install / Compile ) (13)

- Write the following commands in the box

```
NPP_SAVE // save current file  
cd $(CURRENT_DIRECTORY) // go to directory of the current file  
gcc -Wall -Wextra -Wpedantic -std=c++11 -o "$(NAME_PART)" "$(FILE_NAME)"
```



---

#### 0.3.36.15 Notepad++ (Install / Compile ) (14)

- Save the script as gcc-build and for more information check the following link
  - You can modify or add multiple scripts for another task.
- 

### 0.3.37 MSYS2

- Software Distribution and Building Platform for Windows

<https://www.msys2.org/>

---

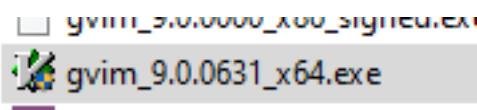
#### 0.3.37.1 Vi/Vim (C/C++) for Windows (1)

- Vim is a command-line editor for programming
  - Use the following links to download Vim for Windows
    - <https://github.com/vim/vim-win32-installer/releases>
    - download : vim online<sup>18</sup>
- 

<sup>18</sup><https://www.vim.org/download.php>

### 0.3.37.2 Vi/Vim (C/C++) for Windows (2)

- Run setup to install the application on your computer.



Vim 9.0 (x64) Setup

### Welcome to Vim 9.0

Setup will guide you through the process.

It is recommended that you close all other applications before starting Setup. This will prevent Vim from overwriting relevant system files without having to restart your computer.

Click Next to continue.



### 0.3.37.3 Vi/Vim (C/C++) for Windows (3)

### 0.3.37.4 Vi/Vim (C/C++) for Windows (4)

- Installation steps.



### License Agreement

Please review the license terms before installing Vim 9.0 (x64).

Press Page Down to see the rest of the agreement.

| For Vim version 9.0. Last change: 2022 Mar 02

VIM REFERENCE MANUAL by Bram Moolenaar

#### SUMMARY

Vim is Charityware. You can use and copy it as much as you like, but you are encouraged to make a donation for needy children in Uganda. Please see [|kcc|](#) below or visit the ICCF web site, available at these URLs:

If you accept the terms of the agreement, click the check box below. You must accept the agreement to install Vim 9.0 (x64). Click Next to continue.

I accept the terms of the License Agreement

Nullsoft Install System v3.04

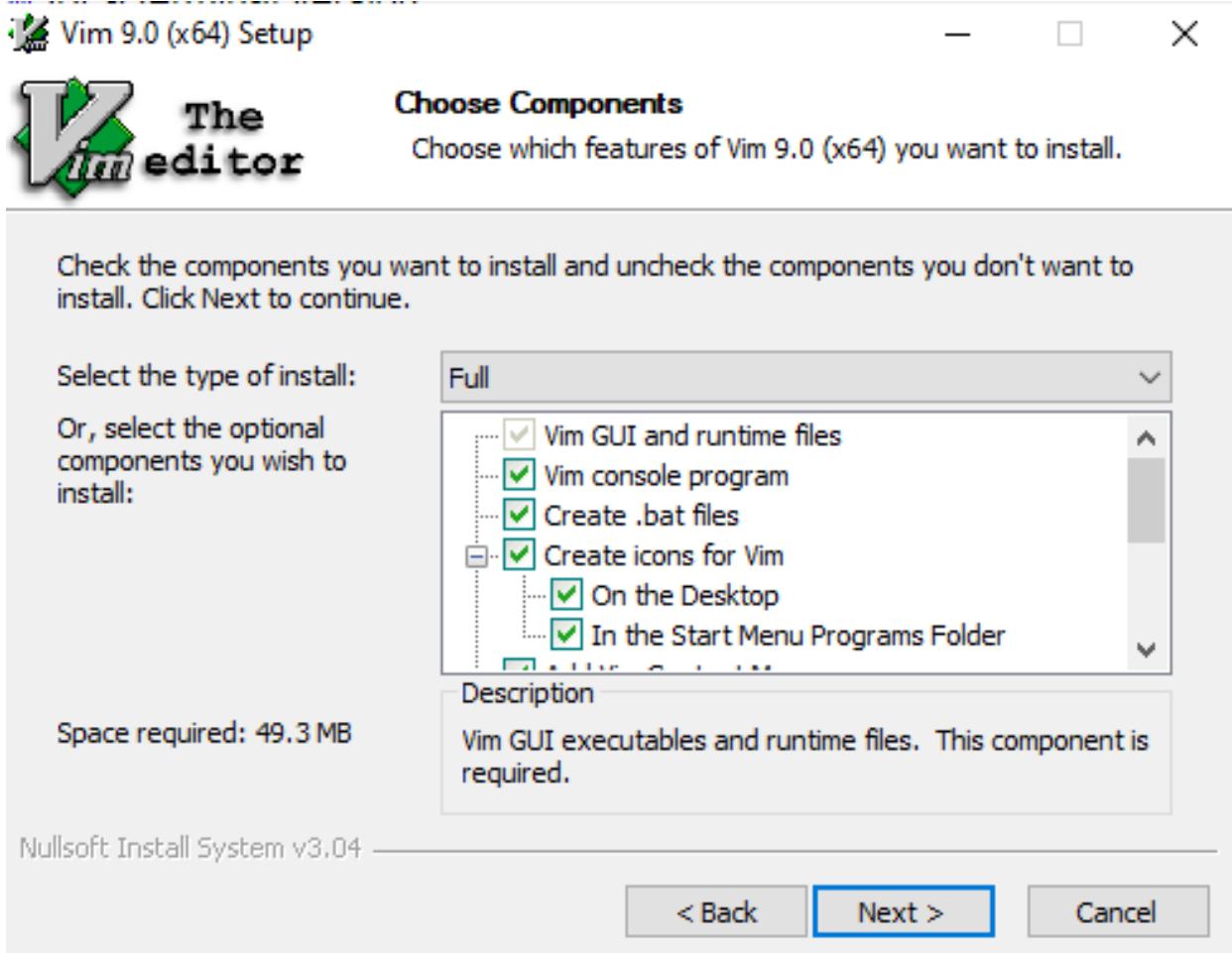
< Back

Next >

Cancel

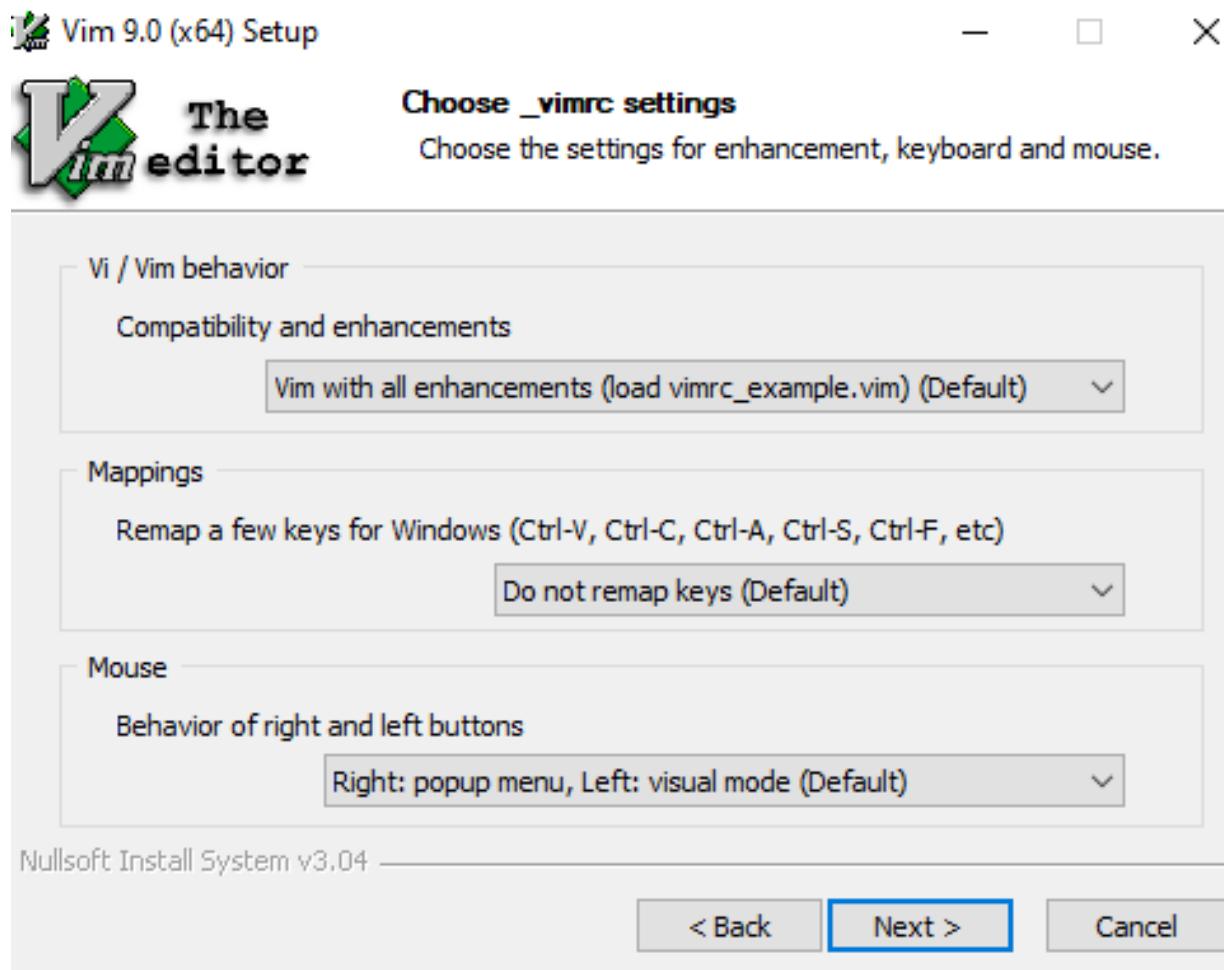
#### 0.3.37.5 Vi/Vim (C/C++) for Windows (5)

- Installation steps.



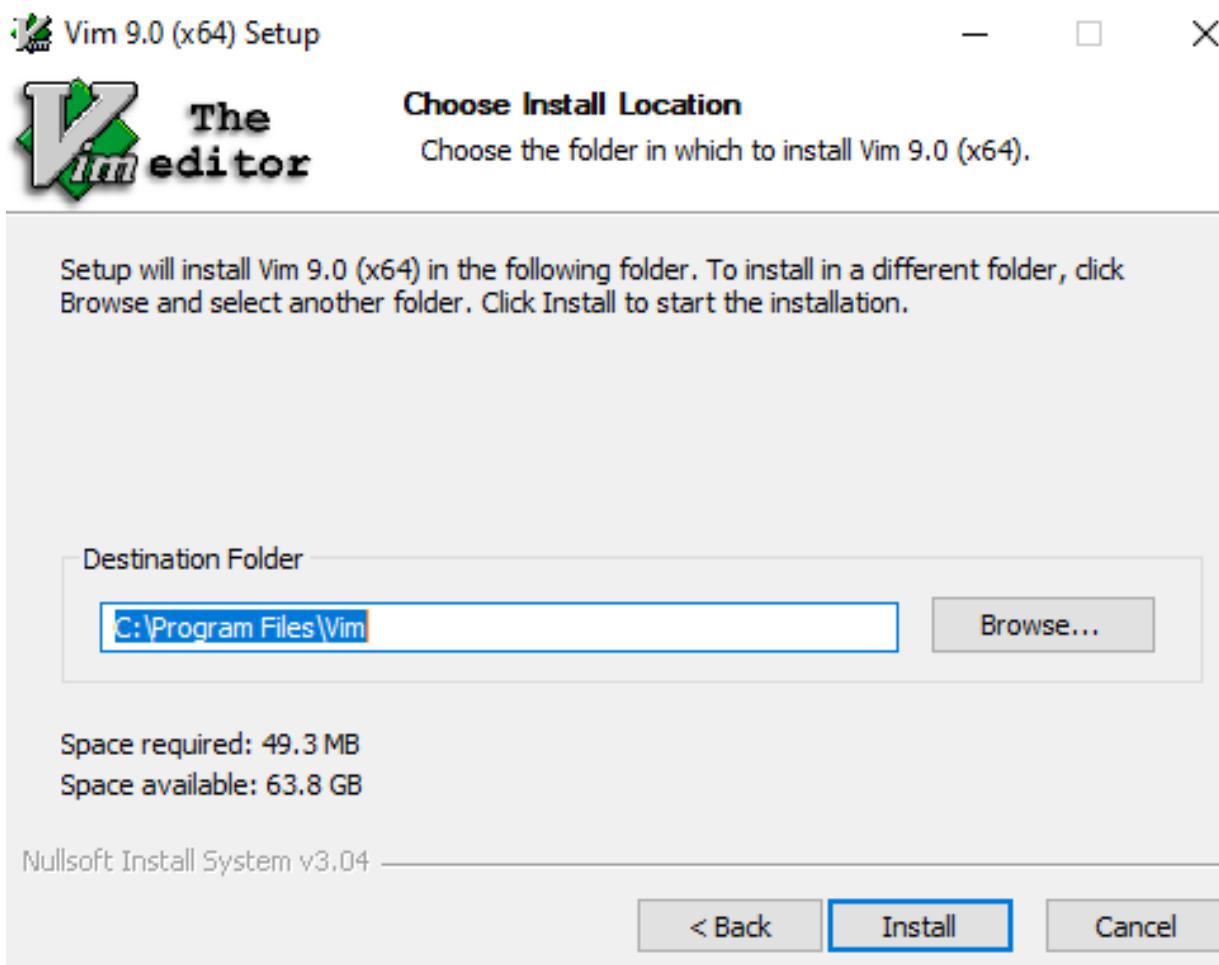
### 0.3.37.6 Vi/Vim (C/C++) for Windows (6)

- Installation steps.



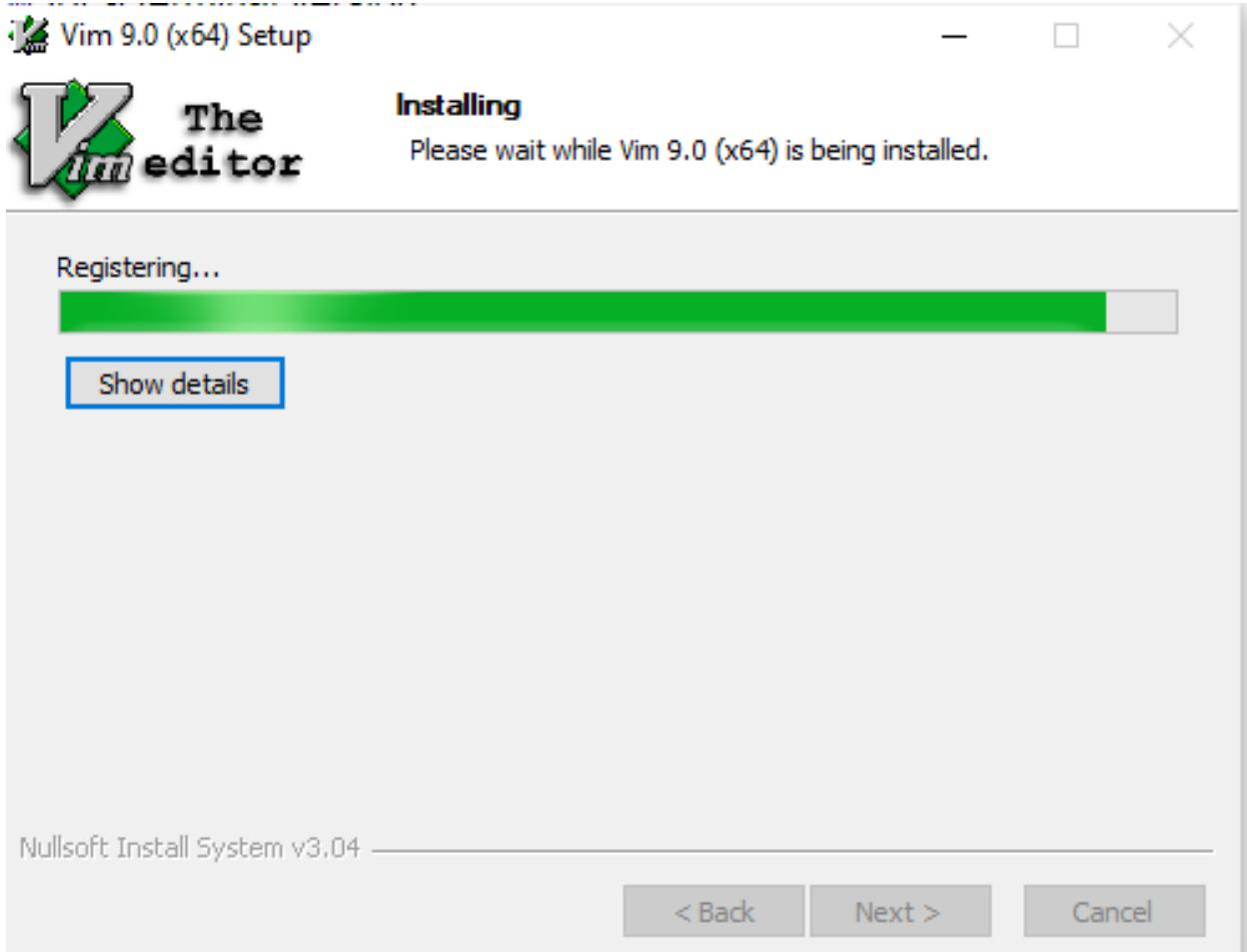
### 0.3.37.7 Vi/Vim (C/C++) for Windows (7)

- Installation steps.



#### 0.3.37.8 Vi/Vim (C/C++) for Windows (8)

- Installation steps.



### 0.3.37.9 Vi/Vim (C/C++) for Windows (9)

- Installation steps.



## Completing Vim 9.0 (x64) Setup

Vim 9.0 (x64) has been installed on your computer.

Click Finish to close Setup.

Show README after installation finished

< Back

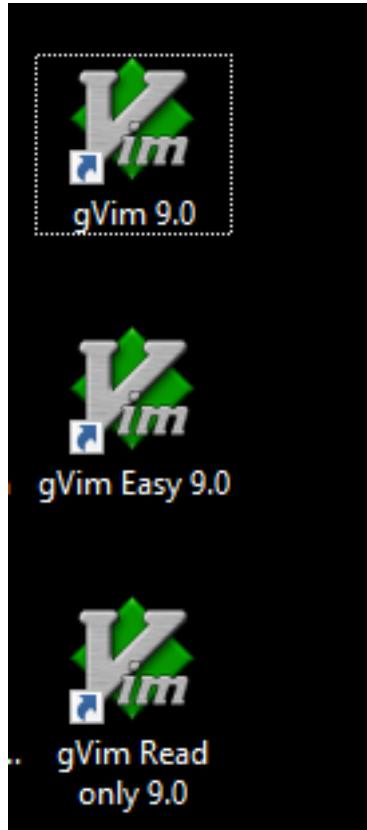
Finish

Cancel

---

### 0.3.37.10 Vi/Vim (C/C++) for Windows (10)

- Generated shortcuts on your desktop



---

#### 0.3.37.11 Vi/Vim (C/C++) for Windows (11)

- Run `vim hello.c` on your command-line to open a c file with vim editor.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ugur.coruh>cd Desktop

C:\Users\ugur.coruh\Desktop>mkdir vim-sample-project

C:\Users\ugur.coruh\Desktop>cd vim-sample-project

C:\Users\ugur.coruh\Desktop\vim-sample-project>dir
Volume in drive C is Windows
Volume Serial Number is 8C3C-8F8C

Directory of C:\Users\ugur.coruh\Desktop\vim-sample-project

01.10.2022  15:36    <DIR>      .
01.10.2022  15:36    <DIR>      ..
          0 File(s)           0 bytes
          2 Dir(s)  68.409.643.008 bytes free

C:\Users\ugur.coruh\Desktop\vim-sample-project>vim hello.c
```

---

### 0.3.37.12 Vi/Vim (C/C++) for Windows (12)

- You will have the following editor.
- Use INSERT to change edit options.

A screenshot of the Vim text editor window. The title bar says "hello.c + (~\Desktop\vim-sample-project) - VIM". The code in the buffer is:

```
#include <stdio.h>
int main()
{
    char name[20];
    printf("Enter Name:");
    scanf("%s",name);
    printf("Your name is %s",name);
    return 0;
}
```

The status bar at the bottom shows "EKLE -- Windows (C:) 9,2 Tüm Belge".

---

### 0.3.37.13 Vi/Vim (C/C++) for Windows (13)

- Sample source code

```
#include <stdio.h>
int main()
{
    char name[20];
    printf("Enter Name:");
    scanf("%s",name);
    printf("Your name is %s",name);
    return 0;
}
```

---

### 0.3.37.14 Vi/Vim (C/C++) for Windows (14)

- Write source code
- Press the Esc button to enter command mode
- Then type :wq. It will save the file and exit from Vim
  - w: write
  - q: quit

```
hello.c + (~\Desktop\vim-sample-project) - VIM
#include <stdio.h>
int main()
{
    char name[20];
    printf("Enter Name:");
    scanf("%s",name);
    printf("Your name is %s",name);
    return 0;
}
:wq
```

---

#### 0.3.37.15 Vi/Vim (C/C++) for Windows (15)

- compile source code with gcc
- link the objects and
- run executable

```
C:\Users\ugur.coruh\Desktop\vim-sample-project>gcc -c hello.c -o hello.o
C:\Users\ugur.coruh\Desktop\vim-sample-project>gcc hello.o -o hello.exe
C:\Users\ugur.coruh\Desktop\vim-sample-project>hello.exe
Enter Name:Ugur Coruh
Your name is Ugur
C:\Users\ugur.coruh\Desktop\vim-sample-project>
```

---

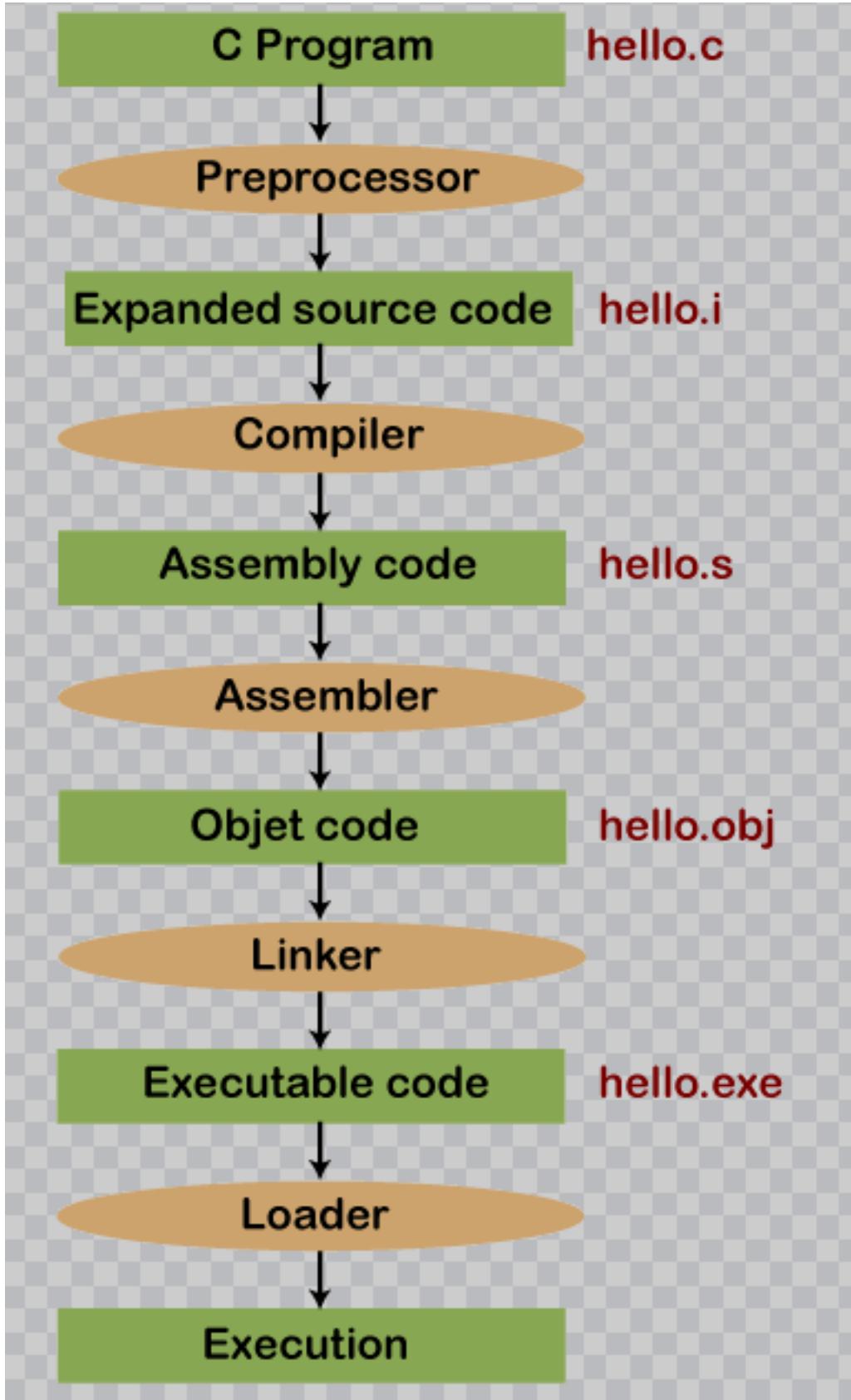
#### 0.3.37.16 Vi/Vim (C/C++) for Windows (17)

- In the folder, you can find your executable. hello.exe

Name	Date modified
.hello.c.swp	1.10.2022 15:40
.hello.c.un~	1.10.2022 15:47
a.exe	1.10.2022 15:48
hello.c	1.10.2022 15:47
hello.exe	1.10.2022 15:49
hello.o	1.10.2022 15:49

### 0.3.37.17 Vi/Vim (C/C++) for Windows (16)

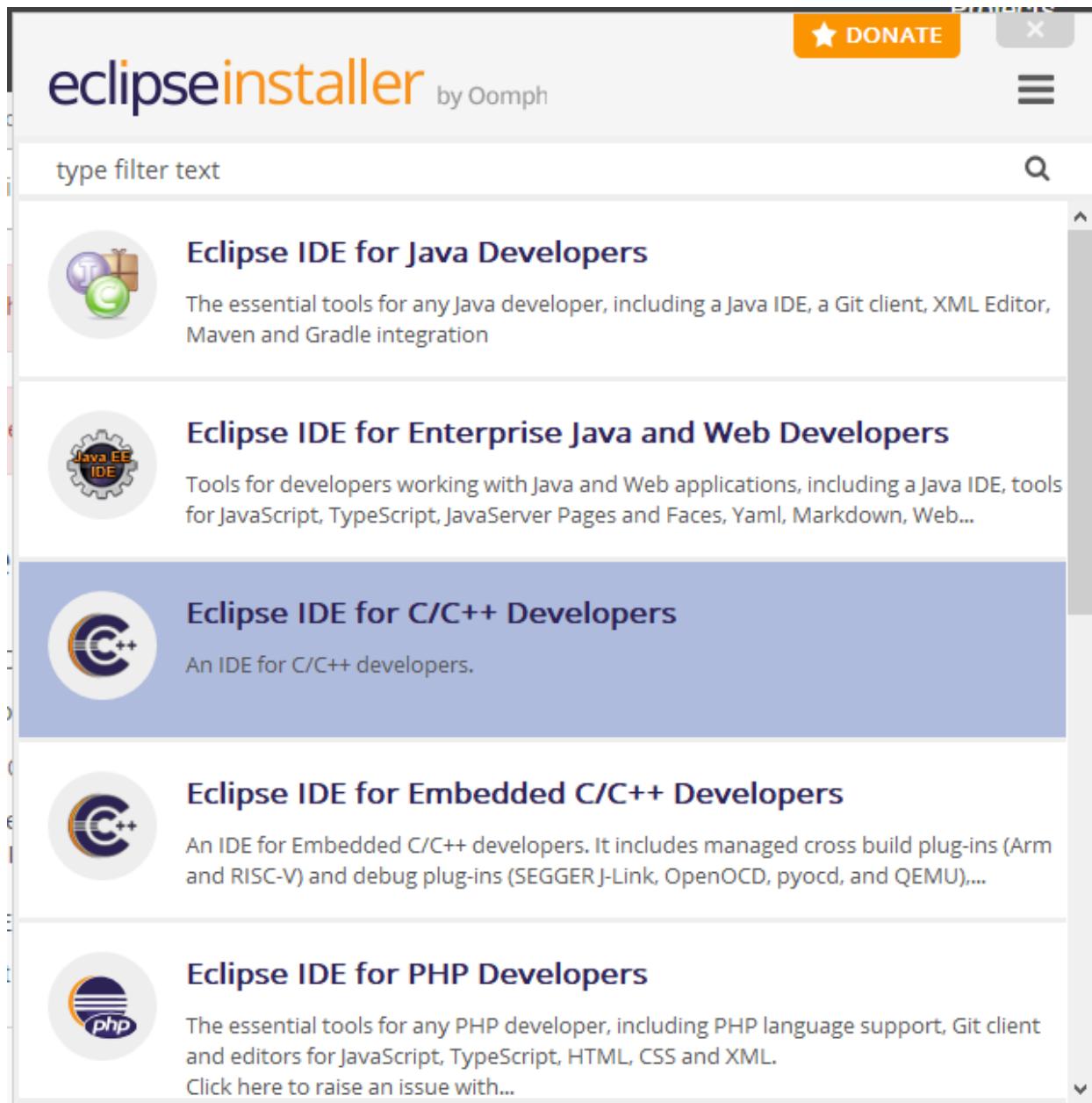
- compile, link and execute flow will be as follow;



---

0.3.37.18 Eclipse (C/C++) - Compile Only / Debugging Has Problem (1)

- Download and install Eclipse IDE from the following link
  - Eclipse IDE for C/C++ Developers | Eclipse Packages<sup>19</sup>
- Run Installer
- Select Eclipse IDE for C/C++ Developers



The screenshot shows the homepage of the **eclipseinstaller** website, which is a repository for Eclipse IDE packages. The page features a search bar at the top with a magnifying glass icon. Below the search bar, there's a "DONATE" button with a star icon. The main content area displays five different Eclipse IDE packages:

- Eclipse IDE for Java Developers**: Described as "The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration". It has a logo featuring a purple 'J' and a green 'C'.
- Eclipse IDE for Enterprise Java and Web Developers**: Described as "Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web...". It has a logo featuring a gear with "Java EE IDE" text.
- Eclipse IDE for C/C++ Developers**: Described as "An IDE for C/C++ developers.". It has a logo featuring a blue 'C' with a plus sign.
- Eclipse IDE for Embedded C/C++ Developers**: Described as "An IDE for Embedded C/C++ developers. It includes managed cross build plug-ins (Arm and RISC-V) and debug plug-ins (SEGGER J-Link, OpenOCD, pyocd, and QEMU),...". It has a logo featuring a blue 'C' with a plus sign.
- Eclipse IDE for PHP Developers**: Described as "The essential tools for any PHP developer, including PHP language support, Git client and editors for JavaScript, TypeScript, HTML, CSS and XML.". It has a logo featuring a blue 'P' and 'hp' text.

#### 0.3.37.19 Eclipse (C/C++) - Compile Only / Debugging Has Problem (2)

- Select Java Version and Installation Path

<sup>19</sup><https://www.eclipse.org/downloads/packages/release/kepler/sr2/eclipse-ide-cc-developers>

# eclipseinstaller

by Oomph

[DONATE](#)



## Eclipse IDE for C/C++ Developers

[details](#)

An IDE for C/C++ developers.

**Java 17+ VM**

`ps://download.eclipse.org/justj/jres/17/updates/release/latest` ▾



**Installation Folder**

`C:\Users\ugur.coruh\eclipse\cpp-2022-09`



**create start menu entry**

**create desktop shortcut**

**INSTALL**

**BACK**

### 0.3.37.20 Eclipse (C/C++) - Compile Only / Debugging Has Problem (3)

- After installation you can LAUNCH eclipse IDE

[DONATE](#)

# eclipseinstaller

by Oomph



## Eclipse IDE for C/C++ Developers

[details](#)

An IDE for C/C++ developers.

**Java 17+ VM**`ps://download.eclipse.org/justj/jres/17/updates/release/latest`**Installation Folder**`C:\Users\ugur.coruh\eclipse\cpp-2022-09` **create start menu entry** **create desktop shortcut****► LAUNCH**[show readme file](#)[open in system explorer](#) [BACK](#)

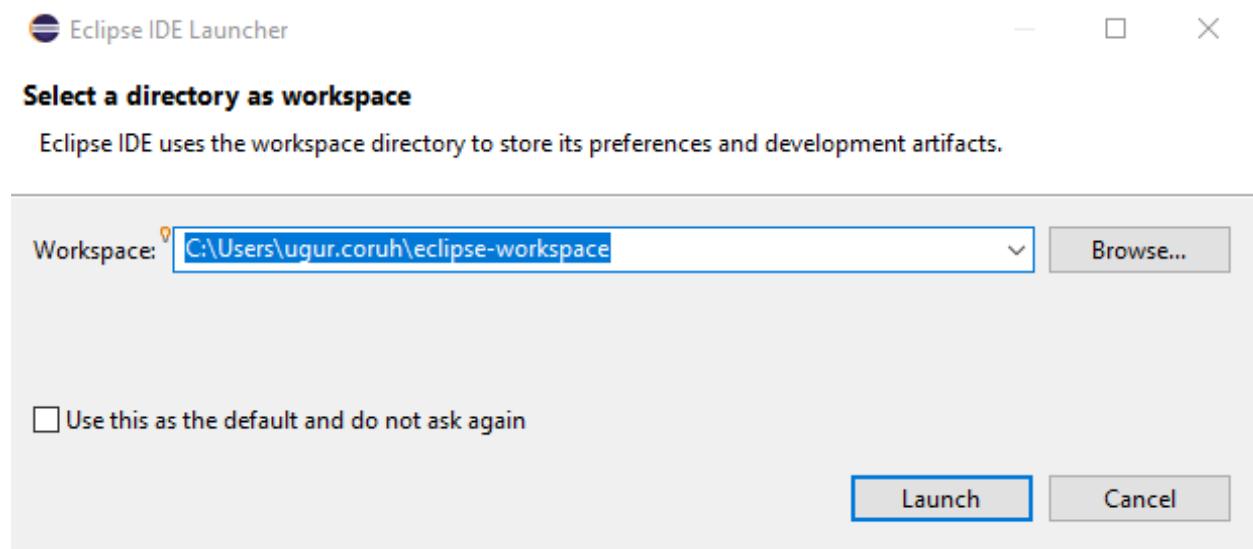


#### 0.3.37.21 Eclipse (C/C++) - Compile Only / Debugging Has Problem (4)

---

#### 0.3.37.22 Eclipse (C/C++) - Compile Only / Debugging Has Problem (5)

- Select a workspace that your project will be saved



---

#### 0.3.37.23 Eclipse (C/C++) - Compile Only / Debugging Has Problem (6)

- You can find your installation under your user folder

File Share View

This PC > Windows (C:) > Users > ugur.coruh > eclipse >

	Name	Date modified	Type	Size
	committers-2021-03	16.05.2021 19:06	File folder	
	cpp-2021-03	16.05.2021 14:44	File folder	
	cpp-2022-09	1.10.2022 19:48	File folder	
	dsl-2021-03	16.05.2021 19:14	File folder	
	embedcpp-2021-03	16.05.2021 18:56	File folder	
	java-2021-03	16.05.2021 10:17	File folder	
	java-2021-09	24.10.2021 02:19	File folder	
	java-2021-12	30.01.2022 22:40	File folder	
	java-2022-03	5.04.2022 15:12	File folder	
	java-2022-032	15.06.2022 01:43	File folder	
	jee-2021-03	16.05.2021 14:09	File folder	
	jee-2021-09	24.10.2021 11:39	File folder	
	modeling-2021-03	16.05.2021 19:22	File folder	
	parallel-2021-03	16.05.2021 19:35	File folder	
	php-2021-03	16.05.2021 19:01	File folder	
	rcp-2021-03	16.05.2021 19:32	File folder	
	scout-2021-03	16.05.2021 19:54	File folder	

### 0.3.37.24 Eclipse (C/C++) - Compile Only / Debugging Has Problem (7)

- You can create shortcut to desktop for your working eclipse version.

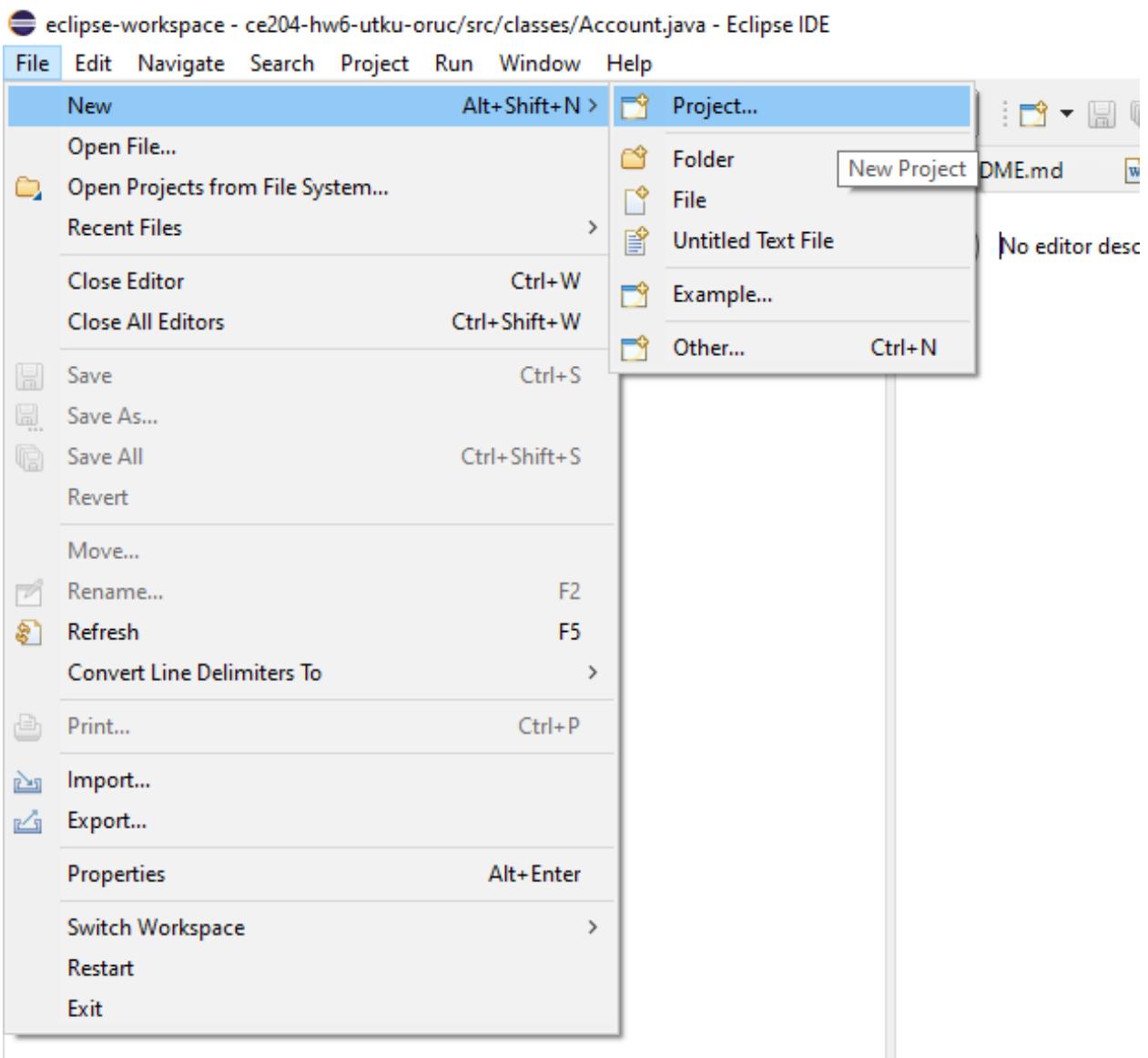
Share View Application Tools

This PC > Windows (C:) > Users > ugur.coruh > eclipse > cpp-2022-09 > eclipse >

	Name	Date modified	Type	Size
	configuration	1.10.2022 20:08	File folder	
	dropins	1.10.2022 19:53	File folder	
	plugins	1.10.2022 19:53	File folder	
	readme	1.10.2022 19:53	File folder	
	.eclipseproduct	31.08.2022 22:06	ECLIPSEPRODUCT...	1 KB
	eclipse.exe	31.08.2022 23:04	Application	519 KB
	eclipse.ini	1.10.2022 19:53	Configuration sett...	2 KB
	eclipsec.exe	31.08.2022 23:04	Application	231 KB
	notice.html	22.08.2022 21:08	Chrome HTML Do...	10 KB

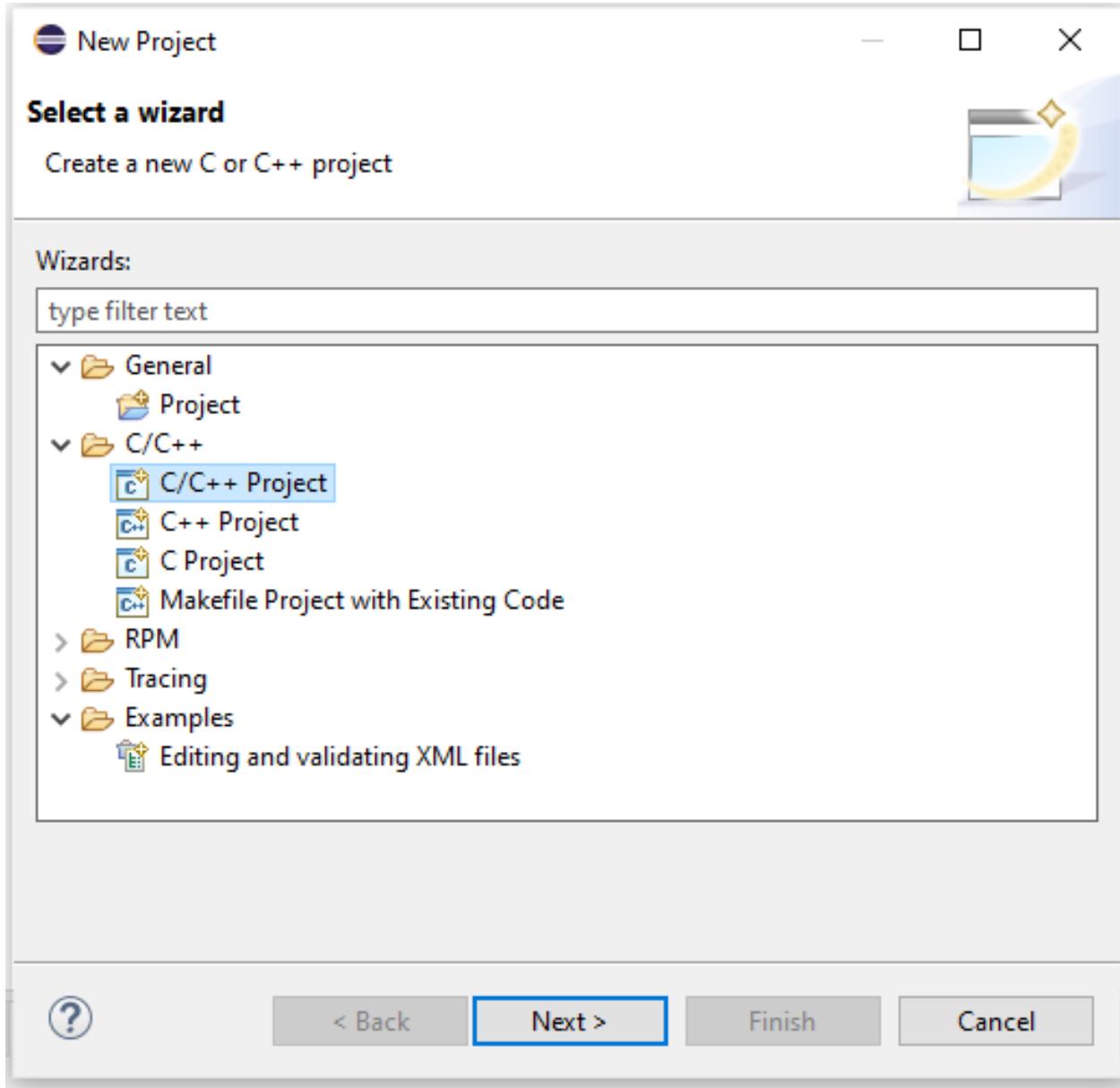
### 0.3.37.25 Eclipse (C/C++) - Compile Only / Debugging Has Problem (8)

- File -> New -> Project



#### 0.3.37.26 Eclipse (C/C++) - Compile Only / Debugging Has Problem (9)

- Select C/C++ Project



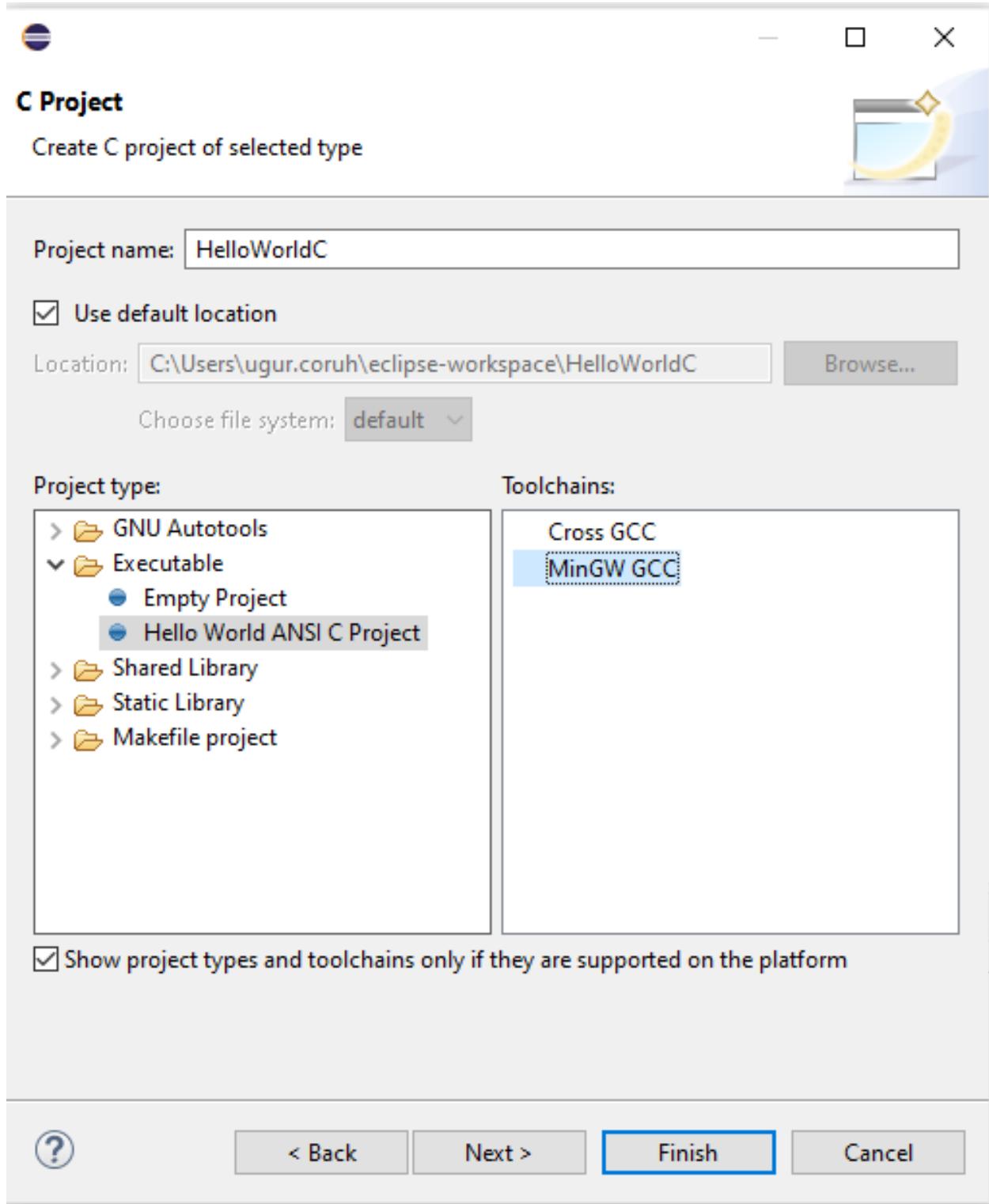
#### 0.3.37.27 Eclipse (C/C++) - Compile Only / Debugging Has Problem (10)

- Select C Managed Build, Eclipse CDT will do job for us.



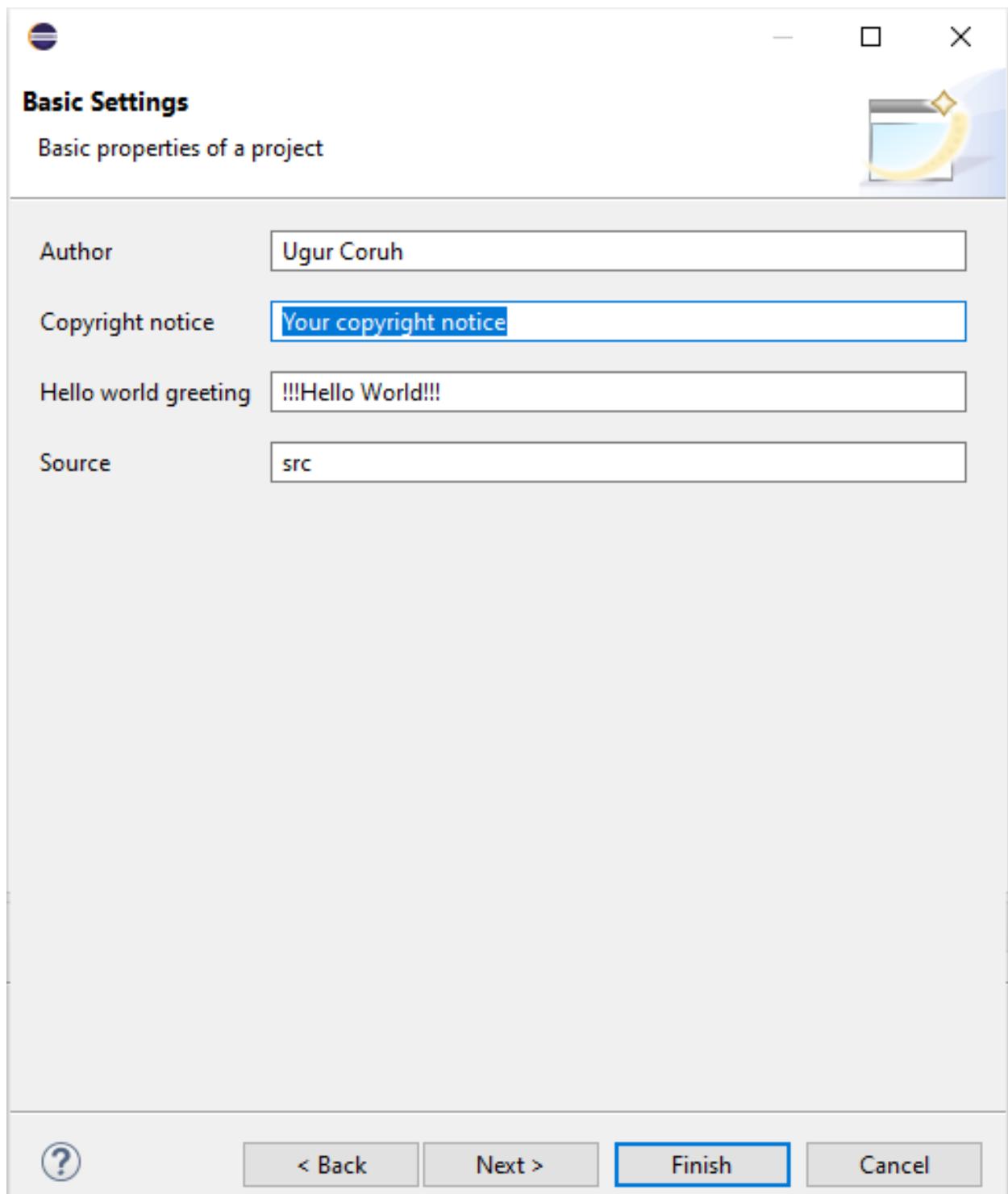
### 0.3.37.28 Eclipse (C/C++) - Compile Only / Debugging Has Problem (11)

- Give project name and select a basic template executable with MinGW GCC.



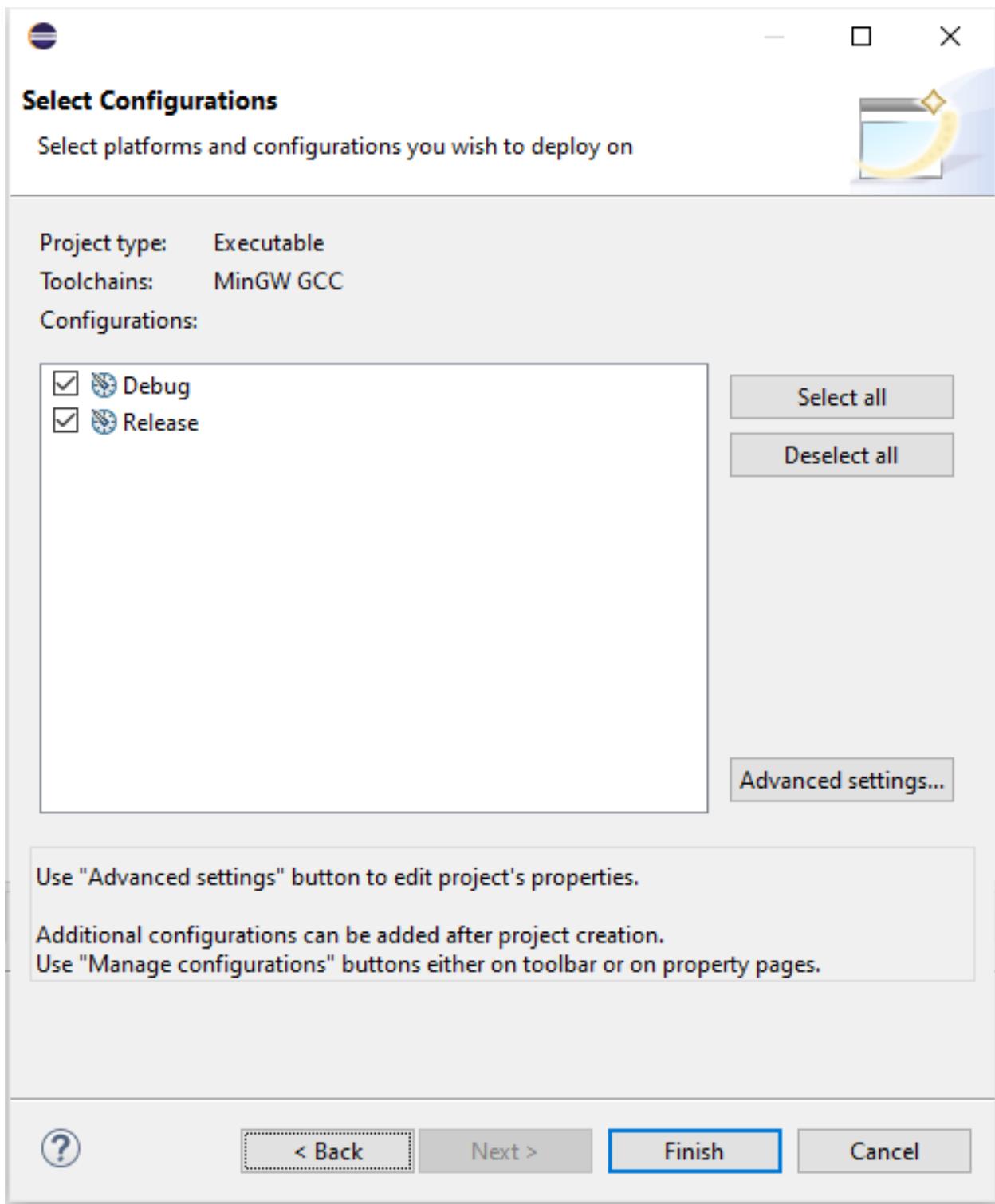
#### 0.3.37.29 Eclipse (C/C++) - Compile Only / Debugging Has Problem (12)

- Configura Basic Settings



### 0.3.37.30 Eclipse (C/C++) - Compile Only / Debugging Has Problem (13)

- There are default Debug and Release configurations you can add your customized configurations from Advanced Settings.

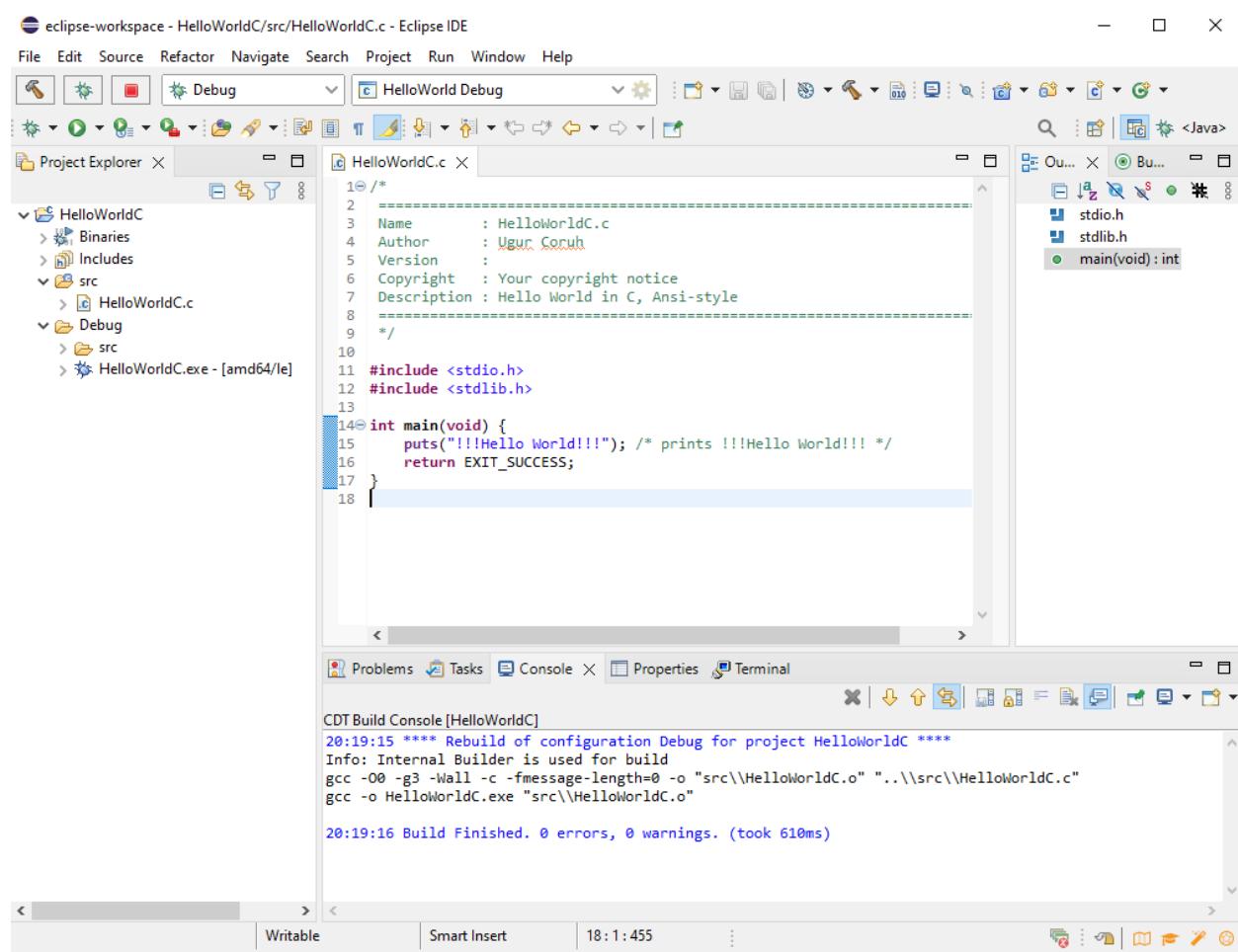


### 0.3.37.31 Eclipse (C/C++) - Compile Only / Debugging Has Problem (14)

- Project settings will be C Select Debug/Release configuration and then Build Application Project->Build All (Ctrl+B)
- HelloWorldC.exe will be generated

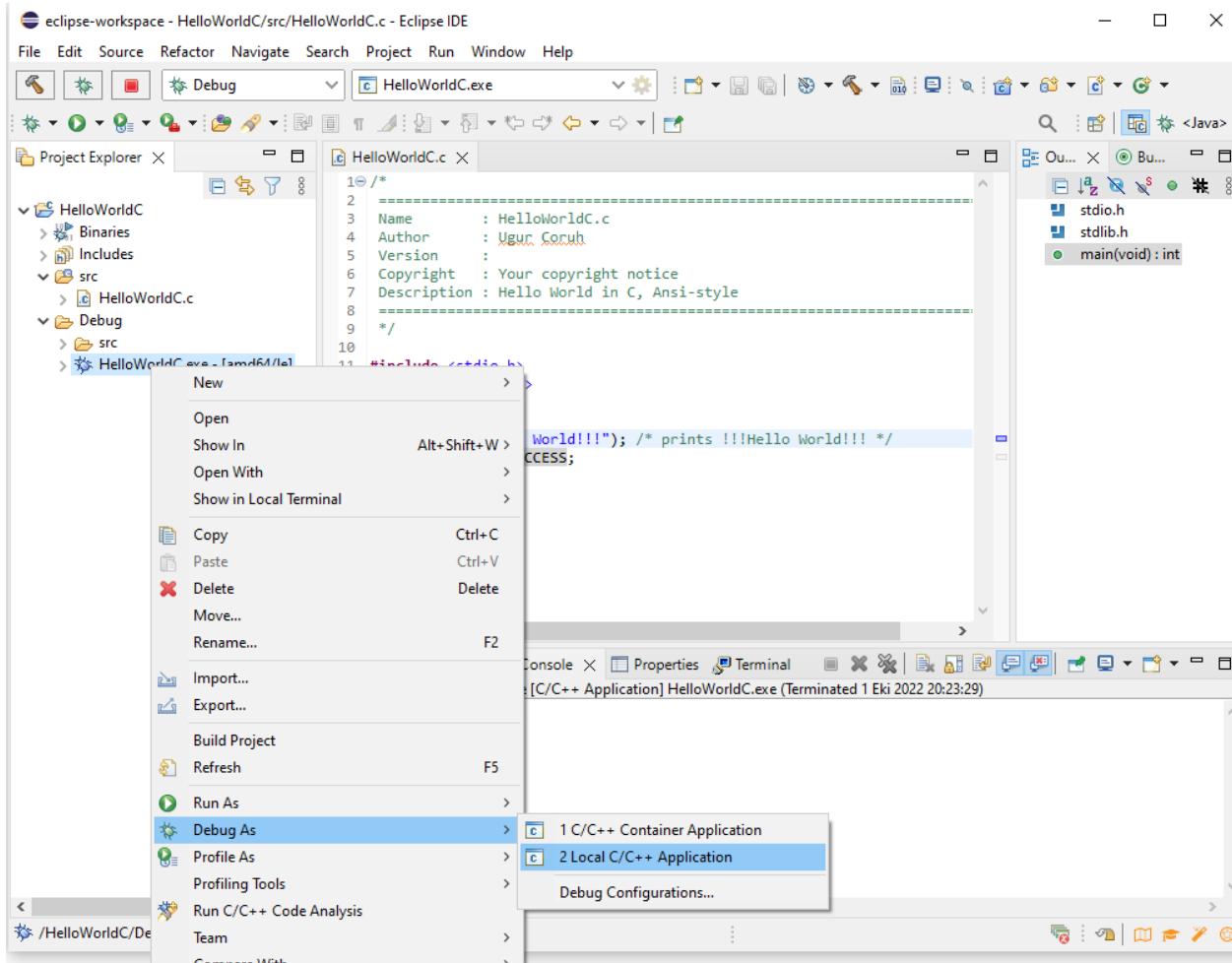
```
gcc -O0 -g3 -Wall -c -fmessage-length=0 -o "src\\HelloWorldC.o" "...\\src\\HelloWorldC.c"
```

```
gcc -o HelloWorldC.exe "src\\HelloWorldC.o"
```



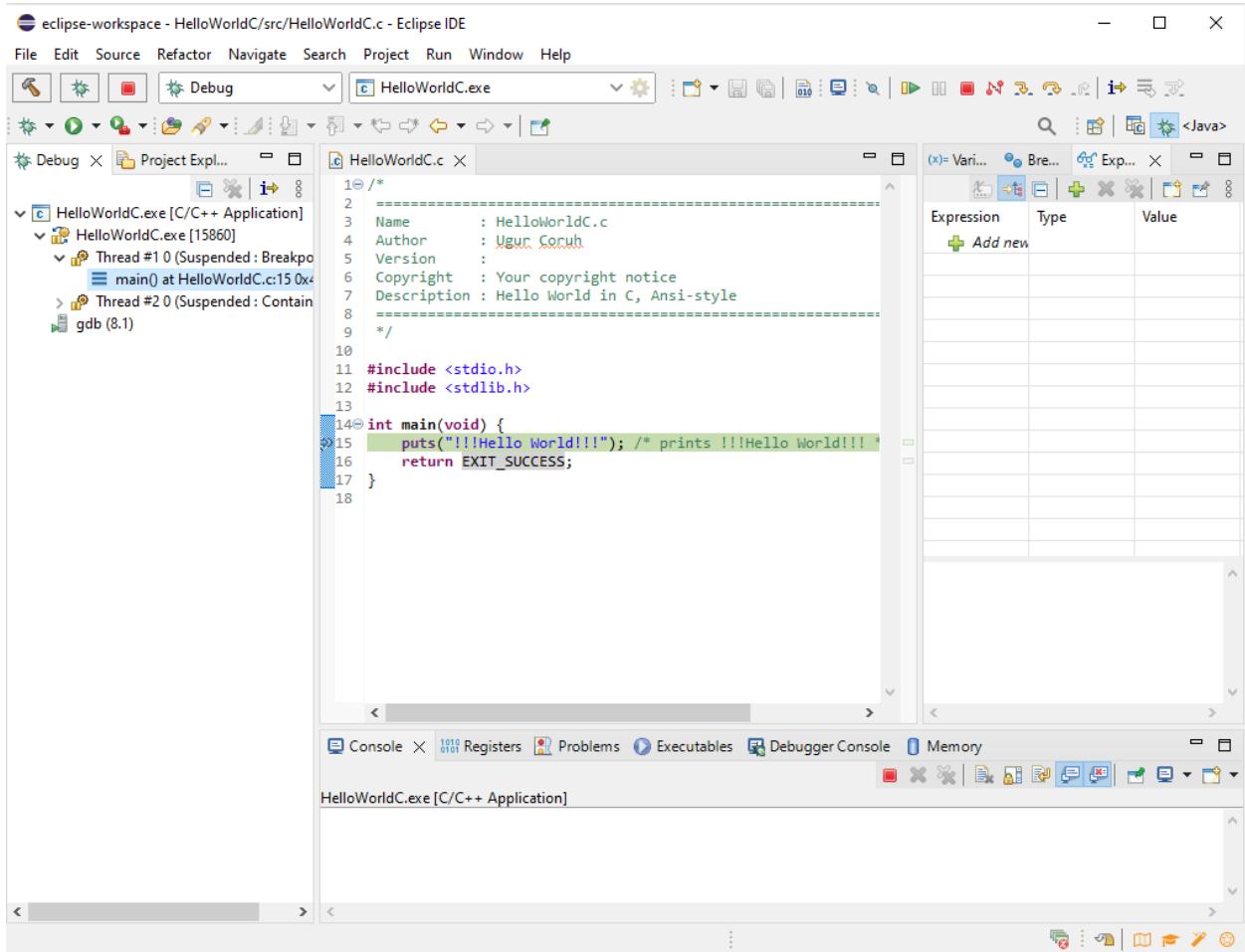
### 0.3.37.32 Eclipse (C/C++) - Compile Only / Debugging Has Problem (15)

- Before build if you want to debug application select debug configuration, put your breakpoints and then Build application again.
- Right click the generated executable Debug As -> Local C/C++ Application



### 0.3.37.33 Eclipse (C/C++) - Compile Only / Debugging Has Problem (16)

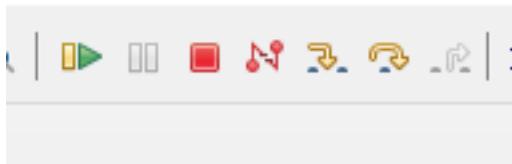
- Debugger will start and stop at breakpoint as follow.



---

### 0.3.37.34 Eclipse (C/C++) - Compile Only / Debugging Has Problem (16)

- Check debug control shortcuts and use them



---

### 0.3.37.35 Eclipse (C/C++) - Compile Only / Debugging Has Problem (17)

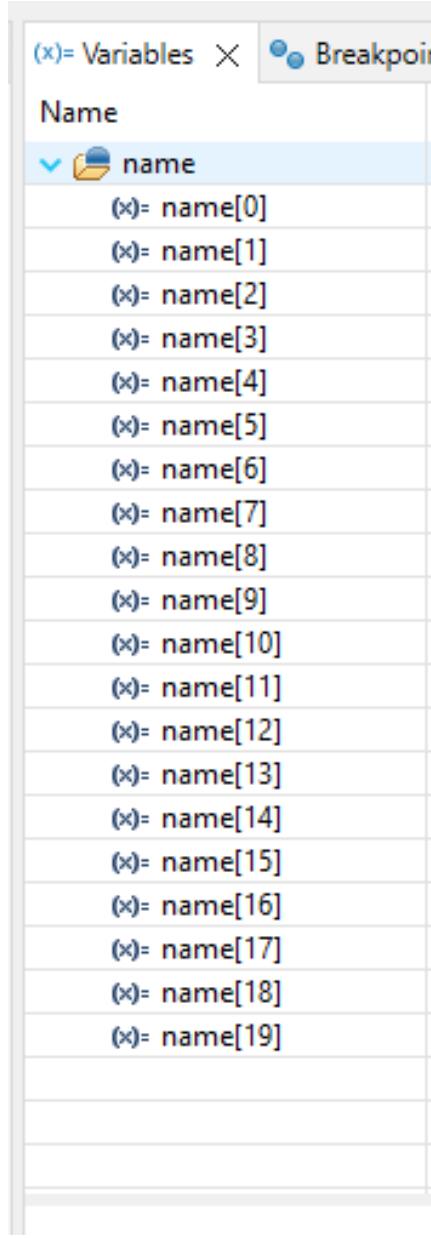
- To watch variables use Expressions and Variables

Variables Breakpoints Expressions

The screenshot shows the 'Expressions' window in the Visual Studio debugger. The window title is 'Expressions'. Below the title bar are several icons: a magnifying glass, a plus sign, a minus sign, a double arrow, a square, a cross, and a file icon. The main area displays a table with three columns: 'Expression', 'Type', and 'Value'. The first row shows a collapsed node for 'name' of type 'char [20]' with a value of '0x61fe00'. Subsequent rows show individual elements of the array, each labeled '(\*)= name char' followed by its ASCII value and character representation. The values are: 8 '\b', 0 '\0', 0 '\0', 0 '\0', 0 '\0', 0 '\0', 0 '\0', 72 'H', 0 '\0', 0 '\0', 0 '\0', 0 '\0', 0 '\0', 0 '\0', 0 '\0', 112 'p', 38 '&', 108 'l', and 0 '\0'. A green plus sign icon with the text 'Add new' is located at the bottom left of the table.

Expression	Type	Value
name	char [20]	0x61fe00
(*)= name char		8 '\b'
(*)= name char		0 '\0'
(*)= name char		0 '\0'
(*)= name char		0 '\0'
(*)= name char		0 '\0'
(*)= name char		0 '\0'
(*)= name char		0 '\0'
(*)= name char		72 'H'
(*)= name char		0 '\0'
(*)= name char		0 '\0'
(*)= name char		0 '\0'
(*)= name char		0 '\0'
(*)= name char		0 '\0'
(*)= name char		0 '\0'
(*)= name char		112 'p'
(*)= name char		38 '&'
(*)= name char		108 'l'
(*)= name char		0 '\0'

**Add new**



#### 0.3.37.36 Eclipse (C/C++) - Compile Only / Debugging Has Problem (18)

---

#### 0.3.37.37 Eclipse (C/C++) - Compile Only / Debugging Has Problem (19)

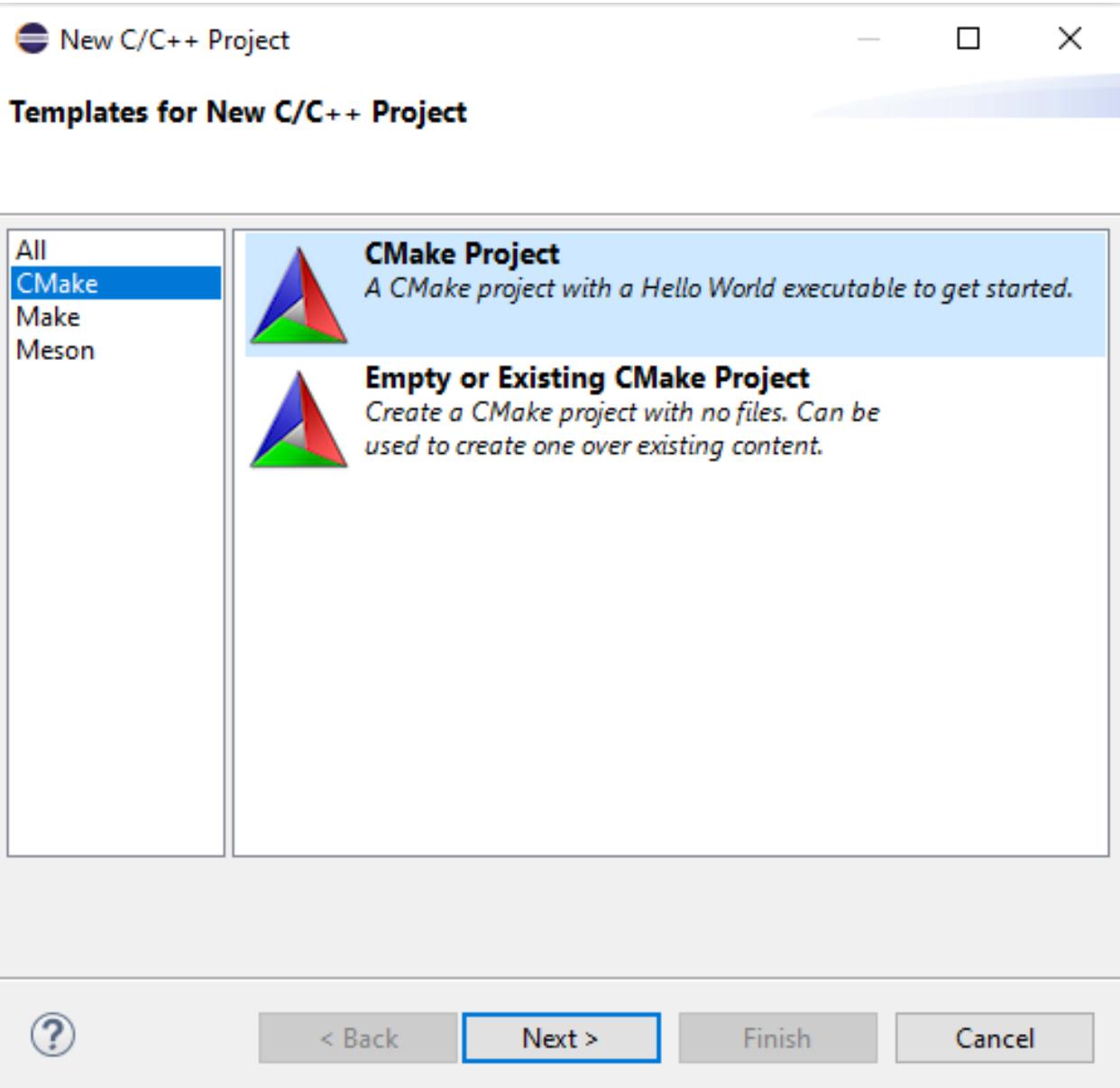
- for more visit eclipse webpage
    - Effective Techniques for Debugging C & C++ | The Eclipse Foundation<sup>20</sup>
    - Help - Eclipse IDE<sup>21</sup>
- 

#### 0.3.37.38 Eclipse (C/C++) - Compile Only / Debugging Has Problem (20)

- Generate CMAKE project from new Project and Select CMake Project Template

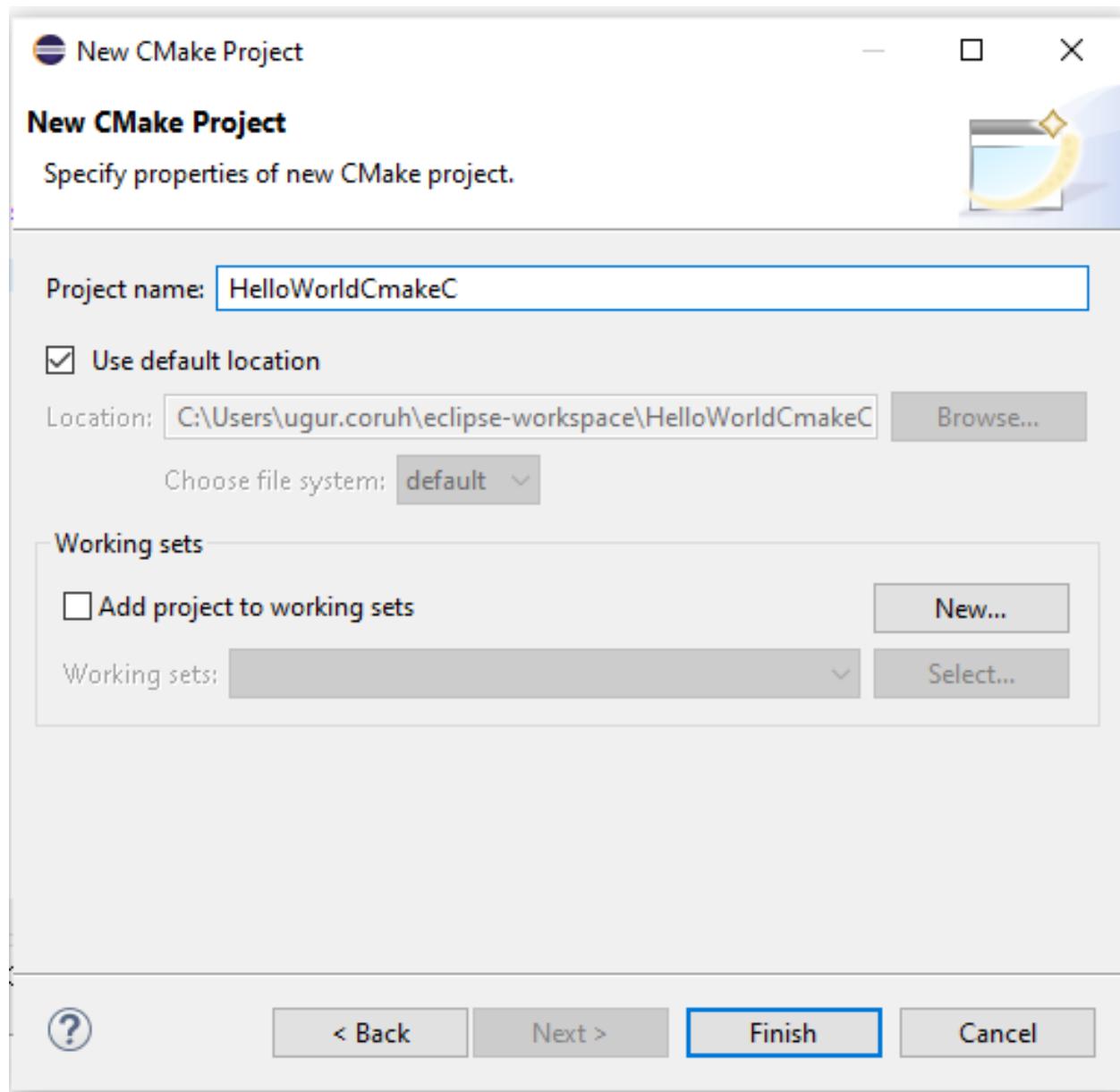
<sup>20</sup>[https://www.eclipse.org/community/eclipse\\_newsletter/2017/april/article2.php](https://www.eclipse.org/community/eclipse_newsletter/2017/april/article2.php)

<sup>21</sup>[https://rtist.helldoc.com/help/index.jsp?topic=%2Forg.eclipse.cdt.doc.user%2Fgetting\\_started%2Fcdt\\_w\\_debug.htm](https://rtist.helldoc.com/help/index.jsp?topic=%2Forg.eclipse.cdt.doc.user%2Fgetting_started%2Fcdt_w_debug.htm)



#### 0.3.37.39 Eclipse (C/C++) - Compile Only / Debugging Has Problem (21)

- Give project name



### 0.3.37.40 Eclipse (C/C++) - Compile Only / Debugging Has Problem (22)

- This will generate default C++ Hello World project

eclipse-workspace - HelloWorldCmakeC/HelloWorldCmakeC.cpp - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Debug Local

Project Explorer X

HelloWorldC  
HelloWorldCmakeC  
build  
HelloWorldCmakeC.cpp  
iostream  
std  
main(int, char\*\*) : int  
CMakeLists.txt

HelloWorldC.c

```

1 #include <iostream>
2 using namespace std;
3
4 int main(int argc, char **argv) {
5     cout << "Hello world";
6     return 0;
7 }

```

#### 0.3.37.41 Eclipse (C/C++) - Compile Only / Debugging Has Problem (23)

- Build Project

eclipse-workspace - HelloWorldCmakeC/HelloWorldCmakeC.cpp - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Debug Local

Project Explorer X

HelloWorldC  
HelloWorldCmakeC  
build  
HelloWorldCmakeC.cpp  
iostream  
std  
main(int, char\*\*) : int  
CMakeLists.txt

Build All Ctrl+B

Build Configurations >

Build Project >

Build Working Set >

Clean...

Build Automatically

Build Targets >

C/C++ Index >

Properties

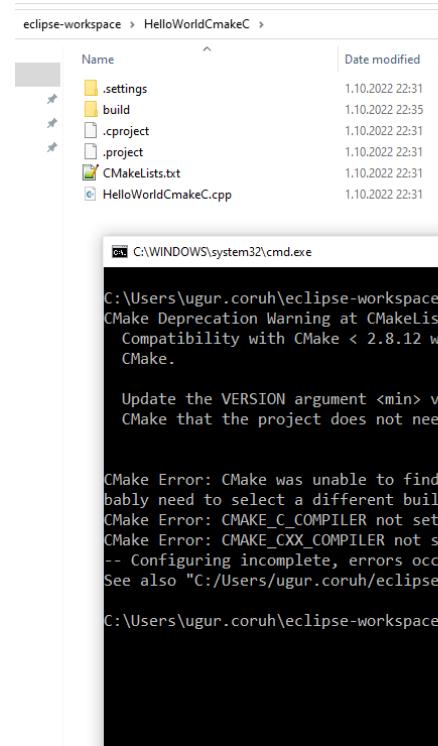
#### 0.3.37.42 Eclipse (C/C++) - Compile Only / Debugging Has Problem (24)

- It will give following errors, for missing configurations. This errors are generated by CMAKE
- Then clean and rebuild again.

Errors occurred during the build.  
 Errors running builder 'CDT Core Builder' on project 'HelloWorldCmakeC'.  
 Resource '/HelloWorldCmakeC/build/cmake.debug.win32.x86\_64/compile\_commands.json' does not exist.

```
Resource '/HelloWorldCmakeC/build/cmake.debug.win32.x86_64/compile_commands.json' does not exist.  
Resource '/HelloWorldCmakeC/build/cmake.debug.win32.x86_64/compile_commands.json' does not exist.  
Resource '/HelloWorldCmakeC/build/cmake.debug.win32.x86_64/compile_commands.json' does not exist.
```

---

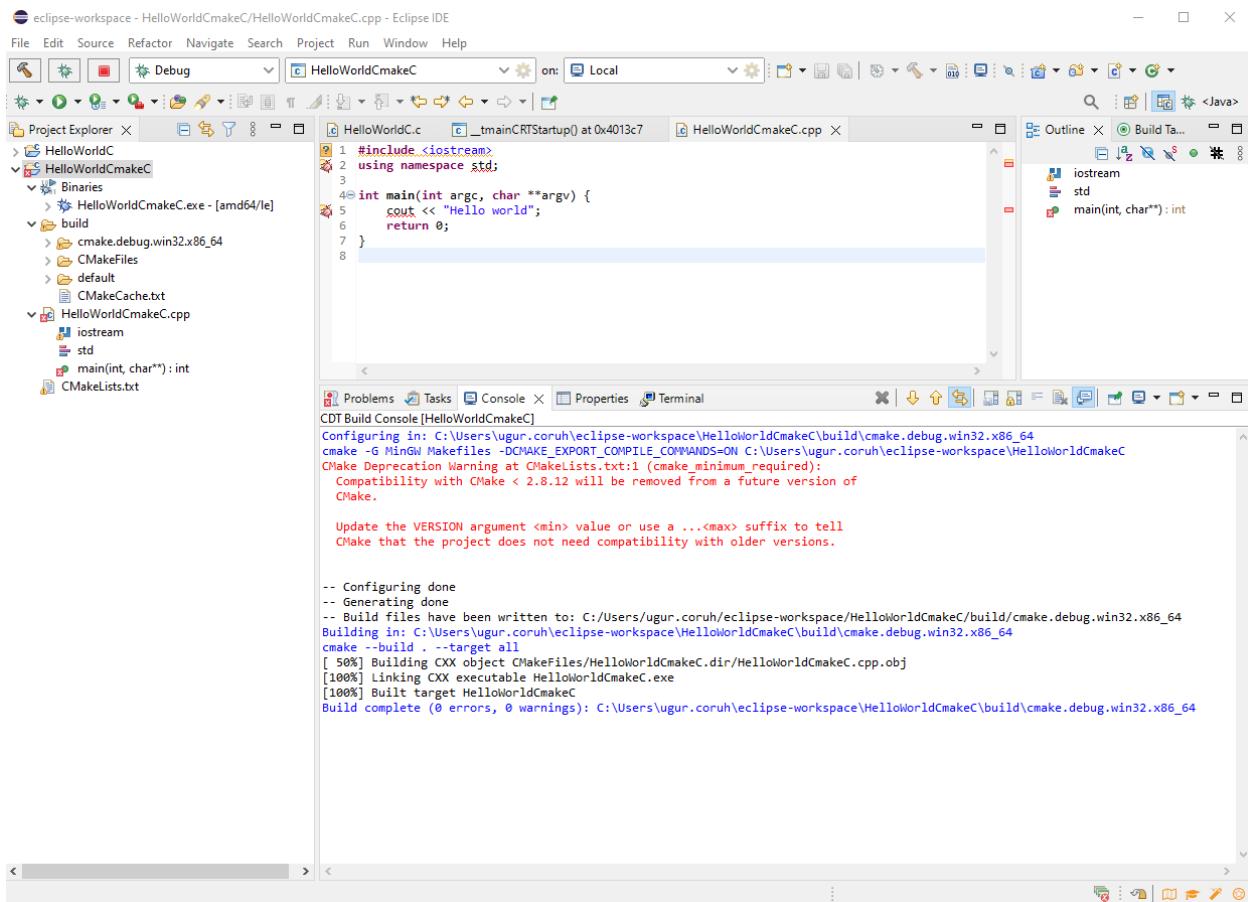


#### 0.3.37.43 Eclipse (C/C++) - Compile Only / Debugging Has Problem (25)

---

#### 0.3.37.44 Eclipse (C/C++) - Compile Only / Debugging Has Problem (26)

- After this operation first Clean project from Project menu and then Build All again



### 0.3.37.45 Eclipse (C/C++) - Compile Only / Debugging Has Problem (27)

- Eclipse with CMake project on windows<sup>22</sup>
- JV - Science and stuff.<sup>23</sup>

<sup>22</sup><https://yairgadelov.me/eclipse-with-cmake-project-on-windows/>

<sup>23</sup><https://jvgomez.github.io/pages/how-to-configure-a-cc-project-with-eclipse-and-cmake.html>

eclipseinstalled



Eclipse IDE for E

An [IDE for Embedded](#)

Java 17+ VM

ps://do

Installation Folder

C:\User

cr

cr

BACK

#### 0.3.37.46 Eclipse (C/C++) - Compile Only / Debugging Has Problem (28)

# eclipseinstaller



Eclipse IDE for E

An IDE for Embedded

Java 17+ VM

ps://do

Installation Folder

C:\User

cr

cr

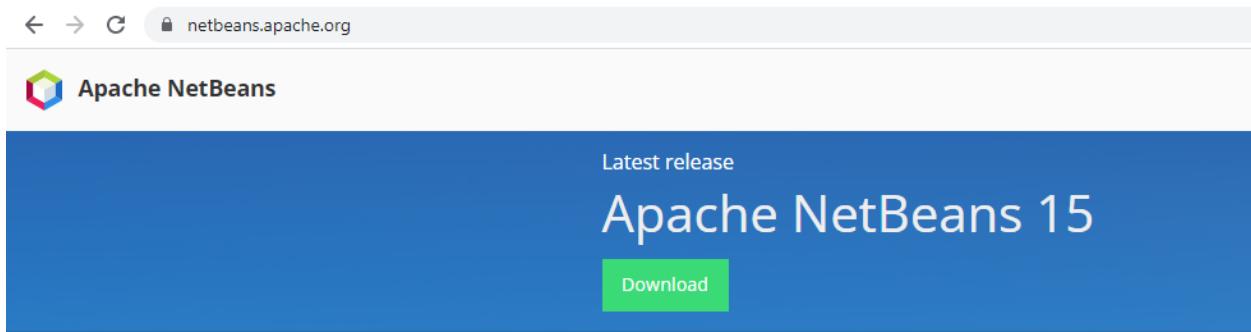
◀ BACK

## 0.3.37.47 Eclipse (C/C++) - Compile Only / Debugging Has Problem (29)

### 0.3.37.48 Netbeans (C/C++) - Manuel Build/Clean/Run Command Setting Not Good Option for C/C++ Development (1)

- <https://netbeans.apache.org/>
- C and C++ Tutorials<sup>24</sup>

<sup>24</sup><https://netbeans.apache.org/kb/docs/cnd/>



#### 0.3.37.49 Netbeans (C/C++) - Manuel Build/Clean/Run Command Setting Not Good Op-

Apache NetBeans IDE Installer

#### Welcome to the Apache NetBeans IDE 15

The installer will install the NetBeans IDE with the following packs.  
Click Customize to select the packs and runtimes to install.

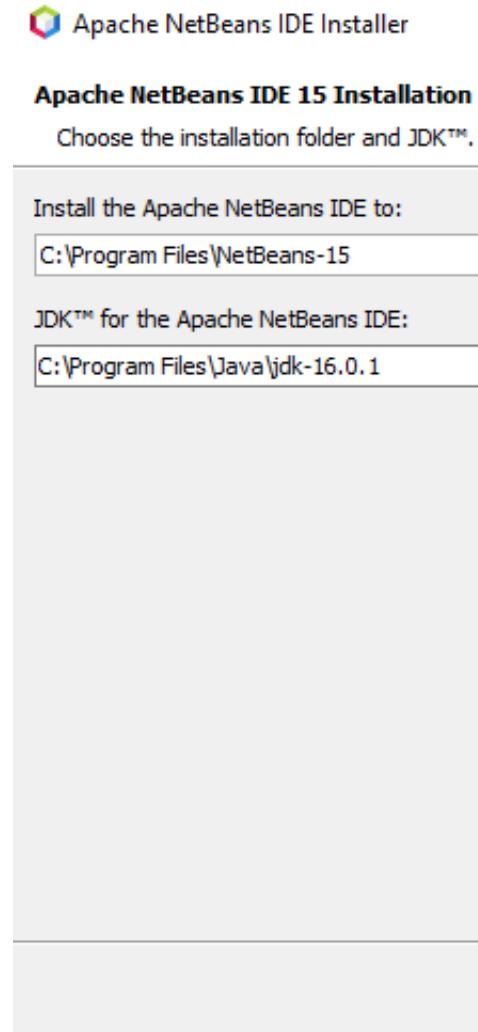
Base IDE  
Java SE  
Java EE  
HTML5/JavaScript  
PHP



Customize...

tion for C/C++ Development (2)

0.3.37.50 Netbeans (C/C++) - Manuel Build/Clean/Run Command Setting Not Good Op-



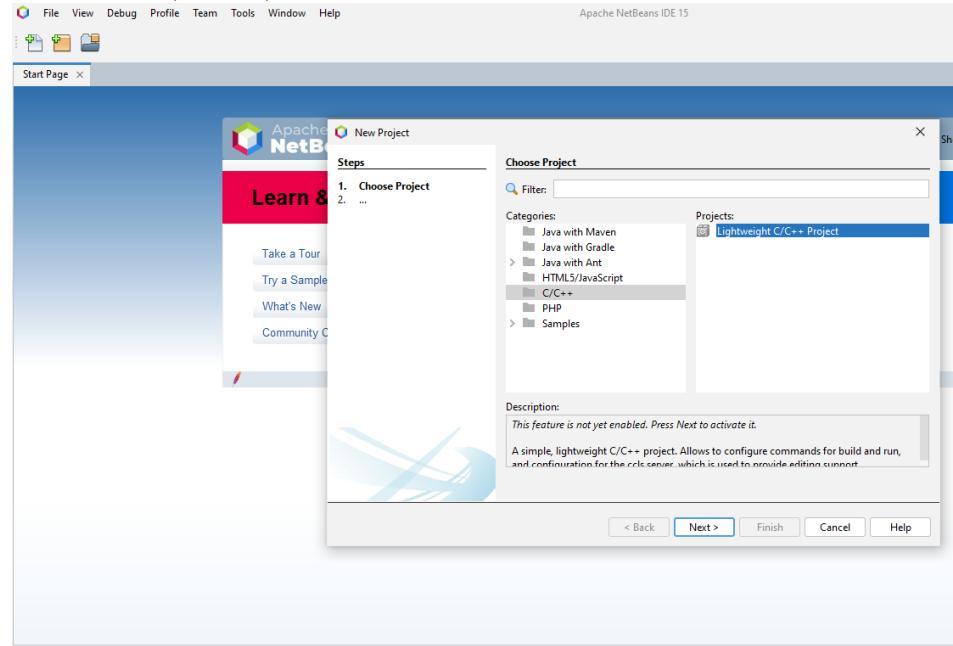
tion for C/C++ Development (3)

0.3.37.51 Netbeans (C/C++) - Manuel Build/Clean/Run Command Setting Not Good Op-



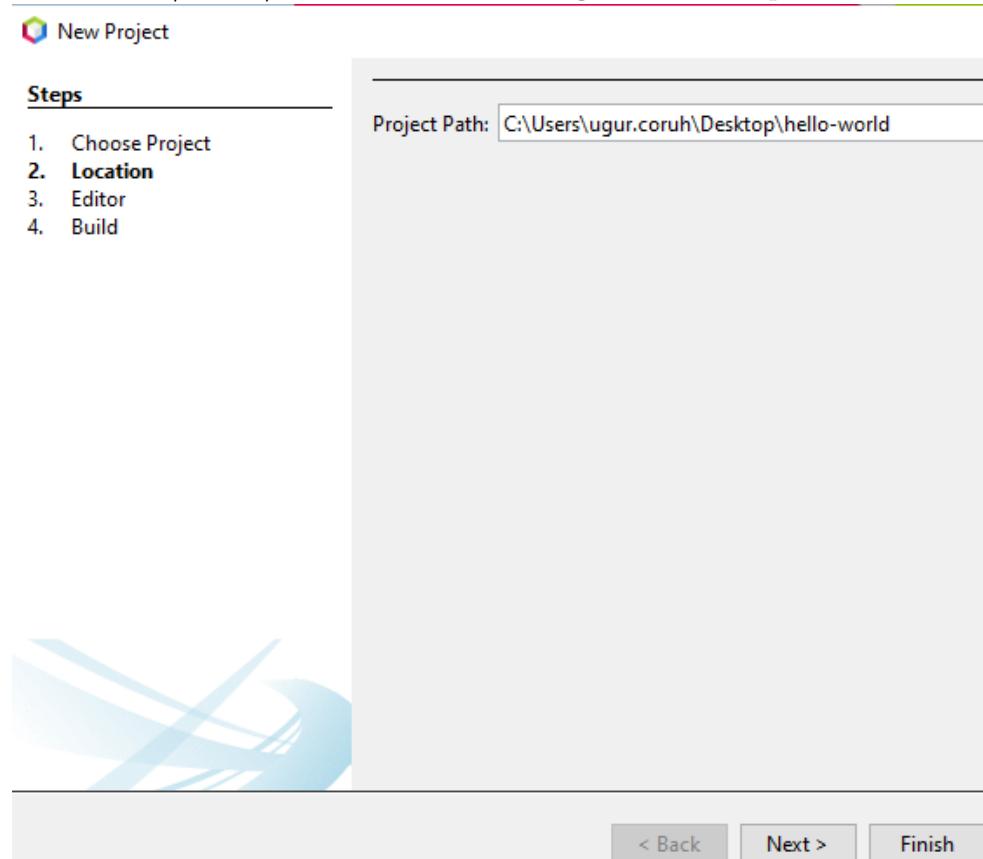
tion for C/C++ Development (4)

### 0.3.37.52 Netbeans (C/C++) - Manuel Build/Clean/Run Command Setting Not Good Option for C/C++ Development (5)



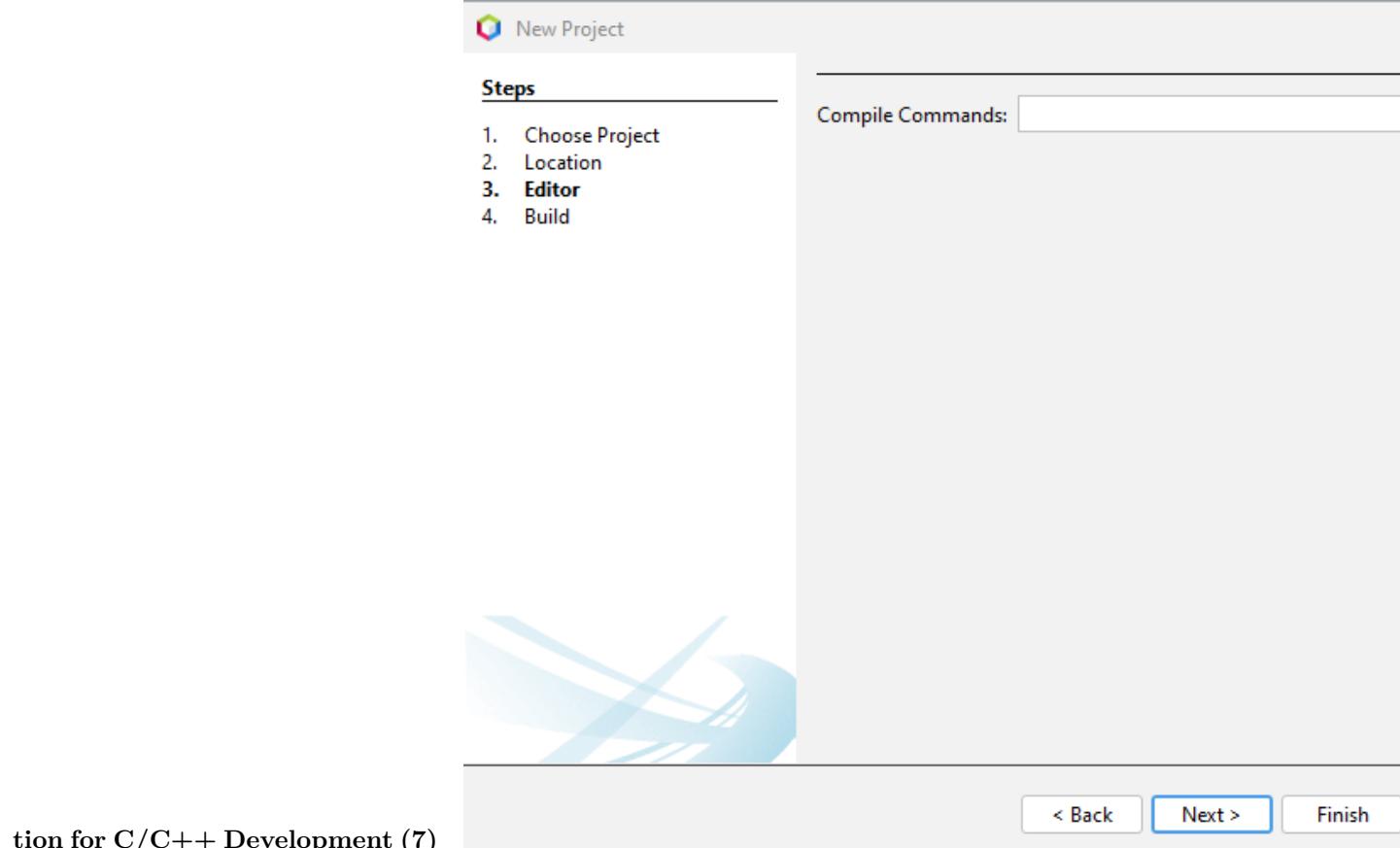
tion for C/C++ Development (5)

### 0.3.37.53 Netbeans (C/C++) - Manuel Build/Clean/Run Command Setting Not Good Option for C/C++ Development (6)



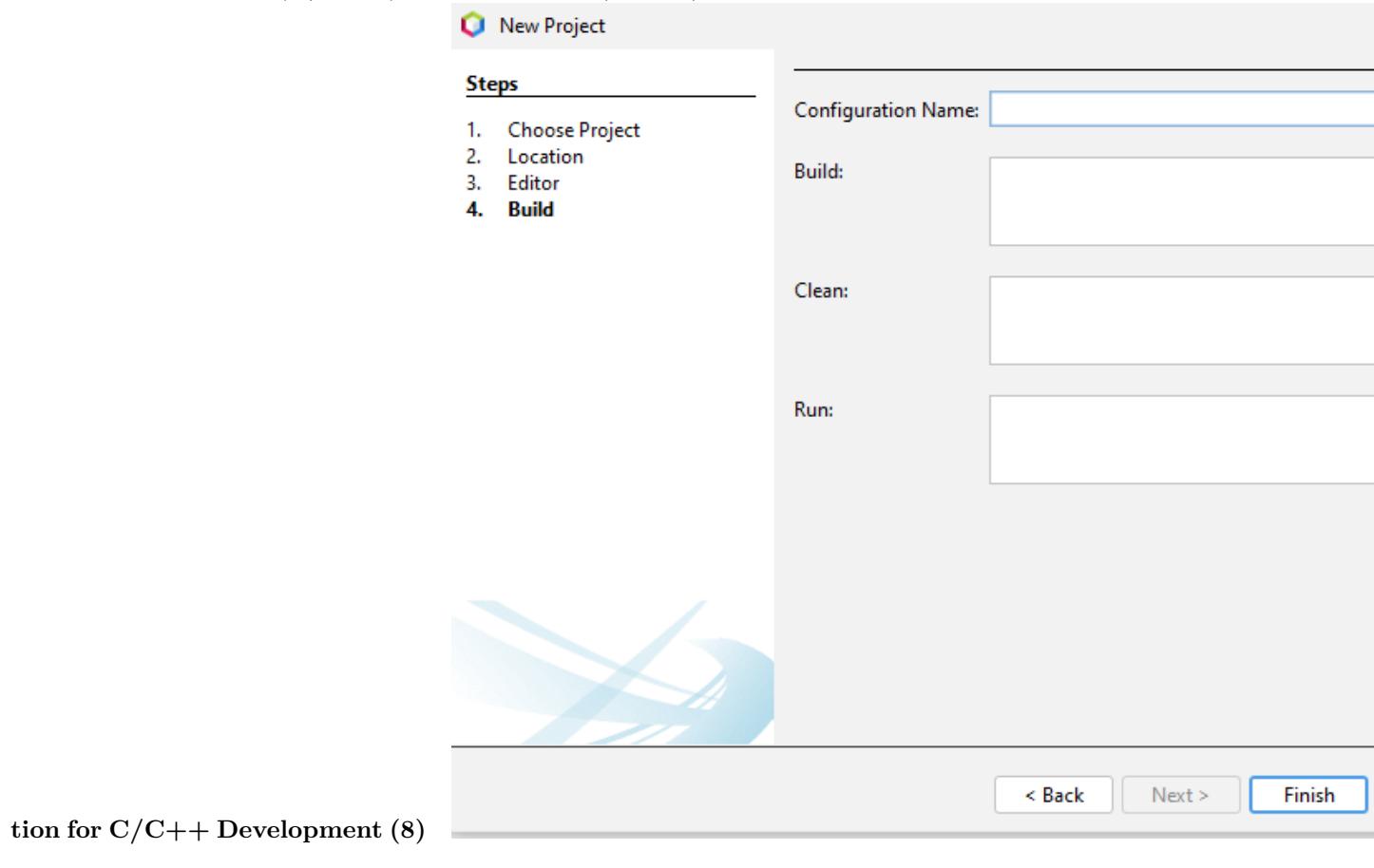
tion for C/C++ Development (6)

### 0.3.37.54 Netbeans (C/C++) - Manuel Build/Clean/Run Command Setting Not Good Op-



tion for C/C++ Development (7)

### 0.3.37.55 Netbeans (C/C++) - Manuel Build/Clean/Run Command Setting Not Good Op-



### tion for C/C++ Development (8)

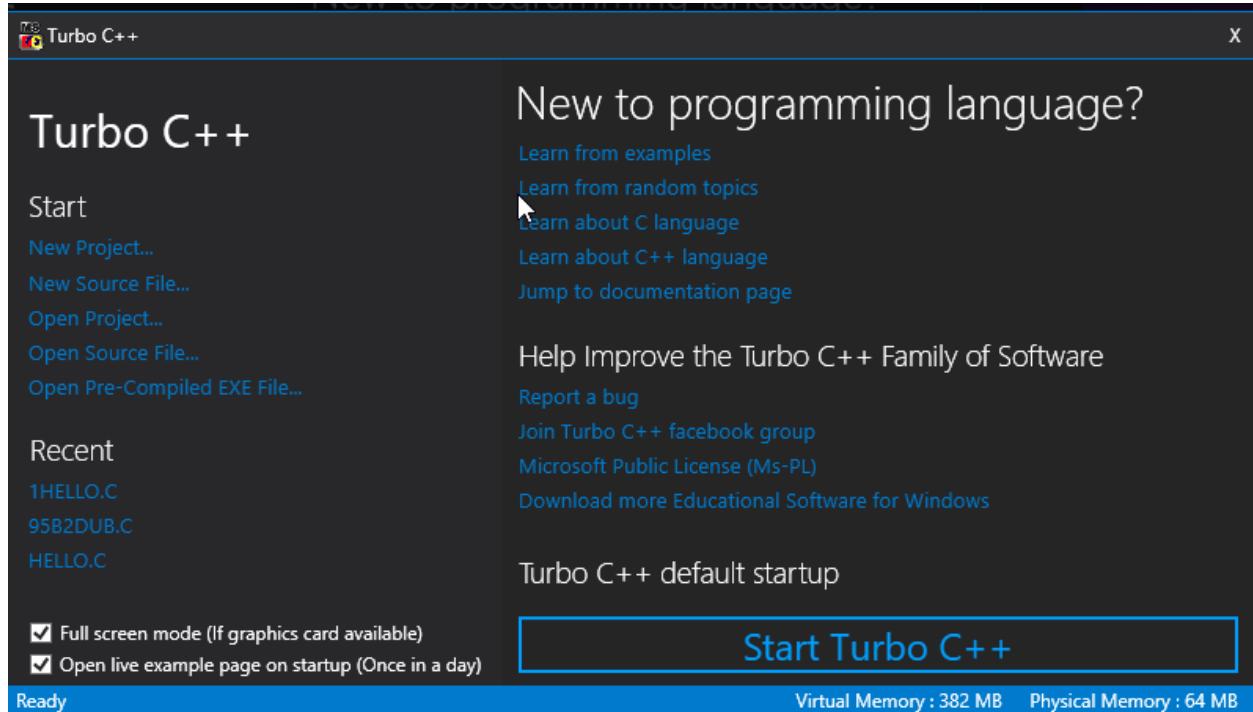
#### 0.3.37.56 Turbo C/C++ (1) Download Turbo.C.3.2.zip<sup>25</sup>

- Download Turbo C++ for Windows 7, 8, 8.1, 10 and Windows 11 (32-64 bit) with full/window screen mode and many more extra features<sup>26</sup>
- Turbo C++ Shortcuts - C Programming Language Tutorials<sup>27</sup>

<sup>25</sup>[files/Turbo.C.3.2.zip](#)

<sup>26</sup><https://developerinsider.co/download-turbo-c-for-windows-7-8-8-1-and-windows-10-32-64-bit-full-screen/>

<sup>27</sup><https://developerinsider.co/turbo-c-shortcuts/>



The screenshot shows the code editor window of the Turbo C++ IDE. The file path is \PROJECTS\HELLOW~1.CPP. The code displayed is:

```
#include<iostream>

using namespace std;
main()
{
    cout<<"hello world c++";
}
```

The status bar at the bottom shows the time as 6:31. The keyboard shortcut bar at the bottom includes F1 Help, F2 Save, F3 Open, Alt-F9 Compile, F9 Make, and F10 Menu.

### 0.3.37.57 Turbo C/C++ (2)

0.3.37.58 Cmake (C++/C) (1) CMake (<http://www.cmake.org/>) is a program which generates the Makefiles used by Make.

### 0.3.37.59 Cmake (C++/C) (2) Why use CMake ?

- Eases **Make** use
    - but the same way of thinking
    - generate the **Makefile**
  - Separate the compilation from the sources
  - Multi-platforms
  - Very flexible
- 

### 0.3.37.60 Cmake (C++/C) (3)

- Check if the libraries/programs are available on your system
  - File generator (**configure\_file**)
  - Calling programs or scripts (**doxygen**)
  - One of the new standards
- 

### 0.3.37.61 Cmake (C++/C) (4) (Download and Install) use the following link for download

Download | CMake<sup>28</sup>

---

### 0.3.37.62 Cmake (C++/C) (5) (WSL and Linux Environment) Hello world with CMake<sup>29</sup>

#### 0.3.37.63 Cmake (C++/C) (6) (Windows Environment) main.c

```
#include <stdio.h>
int main()
{
    char name[20];
    printf("Enter name: ");
    scanf("%s", name);
    printf("Your name is %s.", name);
    return 0;
}
```

CMakeLists.txt

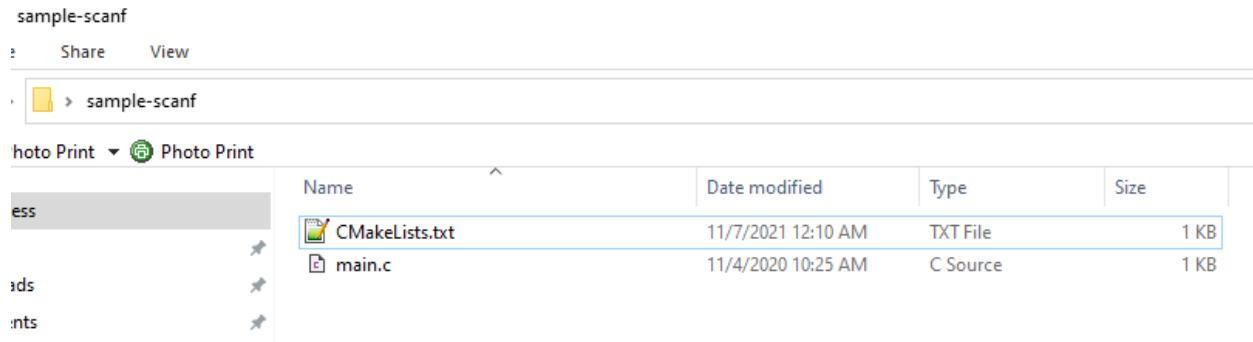
```
cmake_minimum_required(VERSION 3.7.2)
project(scanf-sample)
add_executable(scanf-sample main.c)
```

---

### 0.3.37.64 Cmake (C++/C) (7) (Windows Environment) put main.c and CMakeLists.txt file in sample-scanf folder and from command line

<sup>28</sup><https://cmake.org/download/>

<sup>29</sup>[https://lappweb.in2p3.fr/~paubert/ASTERICS\\_HPC/2-2-100.html](https://lappweb.in2p3.fr/~paubert/ASTERICS_HPC/2-2-100.html)



run the following cmake command with dot (.) to create solution file for c project

```
C:\Users\ugur.coruh\Desktop\sample-scanf>cmake .
```

---

**0.3.37.65 Cmake (C++/C) (8) (Windows Environment)** I have Visual Studio 2022 Community Edition Installed on My Computer, for these reason build tools are selected for visual studio environment and the following outputs are generated

```
C:\Users\ugur.coruh\Desktop\sample-scanf>cmake .
-- Building for: Visual Studio 17 2022
-- Selecting Windows SDK version 10.0.22000.0 to target Windows 10.0.19043.
-- The C compiler identification is MSVC 19.30.30704.0
-- The CXX compiler identification is MSVC 19.30.30704.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: C:/Program Files/Microsoft Visual Studio/2022/Community/VC/Tools/MSVC/
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: C:/Program Files/Microsoft Visual Studio/2022/Community/VC/Tools/MSVC/
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: C:/Users/ugur.coruh/Desktop/sample-scanf

C:\Users\ugur.coruh\Desktop\sample-scanf>
```

---

**0.3.37.66 Cmake (C++/C) (9) (Windows Environment)** also following files are generated

sample-scanf

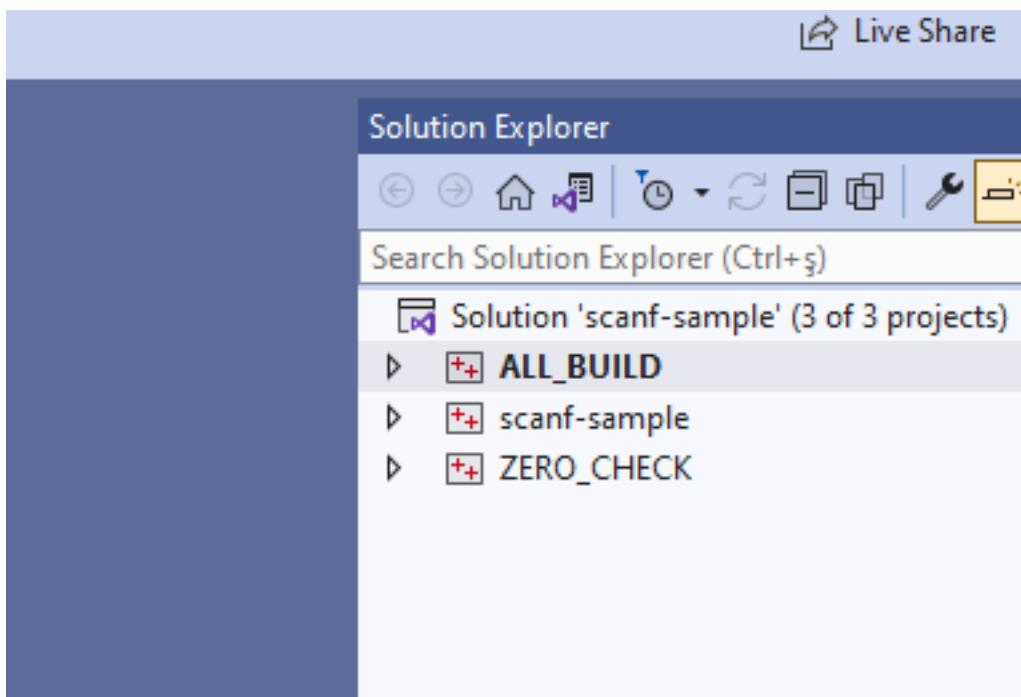
Share View

sample-scanf

Photo Print ▾ Photo Print

	Name	Date modified	Type	Size
ess	CMakeFiles	11/7/2021 12:43 AM	File folder	
ads	ALL_BUILD.vcxproj	11/7/2021 12:43 AM	VC++ Project	41 KB
nts	ALL_BUILD.vcxproj.filters	11/7/2021 12:43 AM	VC++ Project Filte...	1 KB
020-HWS	cmake_install.cmake	11/7/2021 12:43 AM	CMake Source File	2 KB
	CMakeCache.txt	11/7/2021 12:43 AM	TXT File	14 KB
	CMakeLists.txt	11/7/2021 12:10 AM	TXT File	1 KB
scanf	main.c	11/4/2020 10:25 AM	C Source	1 KB
- Personal	scanf-sample.sln	11/7/2021 12:43 AM	Visual Studio Solu...	4 KB
	scanf-sample.vcxproj	11/7/2021 12:43 AM	VC++ Project	50 KB
	scanf-sample.vcxproj.filters	11/7/2021 12:43 AM	VC++ Project Filte...	1 KB
	ZERO_CHECK.vcxproj	11/7/2021 12:43 AM	VC++ Project	41 KB
	ZERO_CHECK.vcxproj.filters	11/7/2021 12:43 AM	VC++ Project Filte...	1 KB

**0.3.37.67 Cmake (C++/C) (10) (Windows Environment)** if we open scanf-sample.sln file we will have automated generated project files



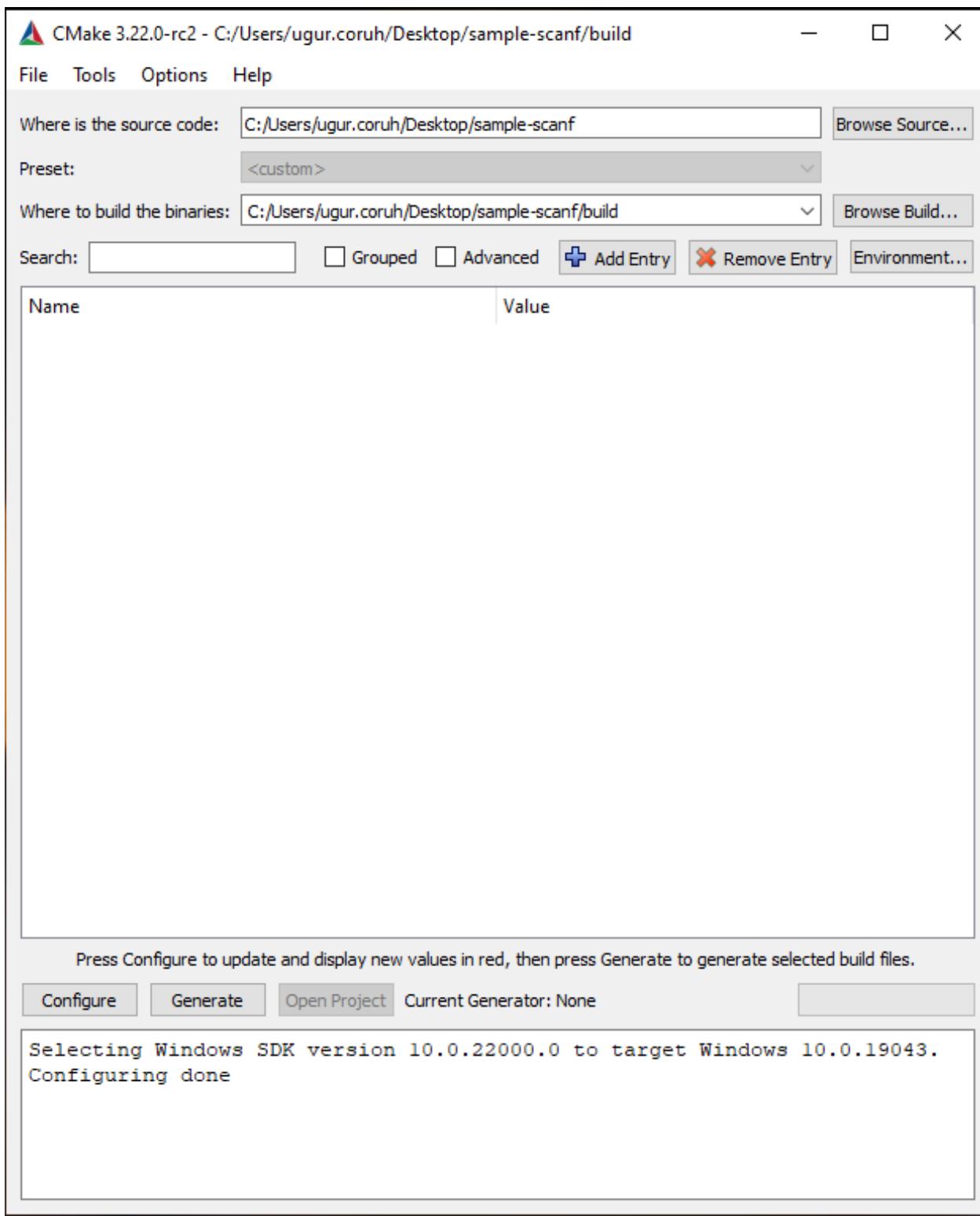
**0.3.37.68 Cmake (C++/C) (11) (Windows Environment)** you can make scanf-sample with startup project with right click and then run on visual studio.

if you want to configure for another build tool you can use Cmake-GUI installed with setup on your computer



---

**0.3.37.69 Cmake (C++/C) (12) (Windows Environment)** Open GUI and Select *File-> Delete Cache*



**0.3.37.70 Cmake (C++/C) (13) (Windows Environment)** then you can click “Configure” to select build tool

? X



Specify the generator for this project

Visual Studio 17 2022

Optional platform for generator (if empty, generator uses: x64)

Optional toolset to use (argument to -T)

- Use default native compilers
- Specify native compilers
- Specify toolchain file for cross-compiling
- Specify options for cross-compiling

Finish

Cancel



Specify the generator for this project

- Visual Studio 17 2022
- Visual Studio 17 2022
- Visual Studio 16 2019
- Visual Studio 15 2017
- Visual Studio 14 2015
- Visual Studio 12 2013
- Visual Studio 11 2012
- Visual Studio 10 2010
- Visual Studio 9 2008
- Borland Makefiles
- NMake Makefiles
- Specify native compilers
- Specify toolchain file for cross-compiling
- Specify options for cross-compiling

### 0.3.37.71 Cmake (C++/C) (14) (Windows Environment)

0.3.37.72 Cmake (C++/C) (15) (Windows Environment) if you click “Configure” twice it will generate the visual studio solution in build folder

for more detailed examples that include also docker and travis-ci sample you can check the following repo  
GitHub - ttroy50/cmake-examples: Useful CMake Examples<sup>30</sup>

### 0.3.37.73 Make (1) Sample

hello.c

```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

### 0.3.37.74 Make (2) Makefile

```
# This is the default target, which will be built when
# you invoke make
```

<sup>30</sup><https://github.com/ttroy50/cmake-examples>

```

.PHONY: all
all: hello

# This rule tells make how to build hello from hello.cpp
hello: hello.c
    g++ -o hello hello.c

# This rule tells make to copy hello to the binaries subdirectory,
# creating it if necessary
.PHONY: install
install:
    mkdir -p binaries
    cp -p hello binaries

# This rule tells make to delete hello and hello.o
.PHONY: clean
clean:
    rm -f hello

```

---

### 0.3.37.75 Make (3) compile.bat

make all .

will create hello.exe

check hello-make sample

---

s-and-programming-l > Week-2 > hello-make

---

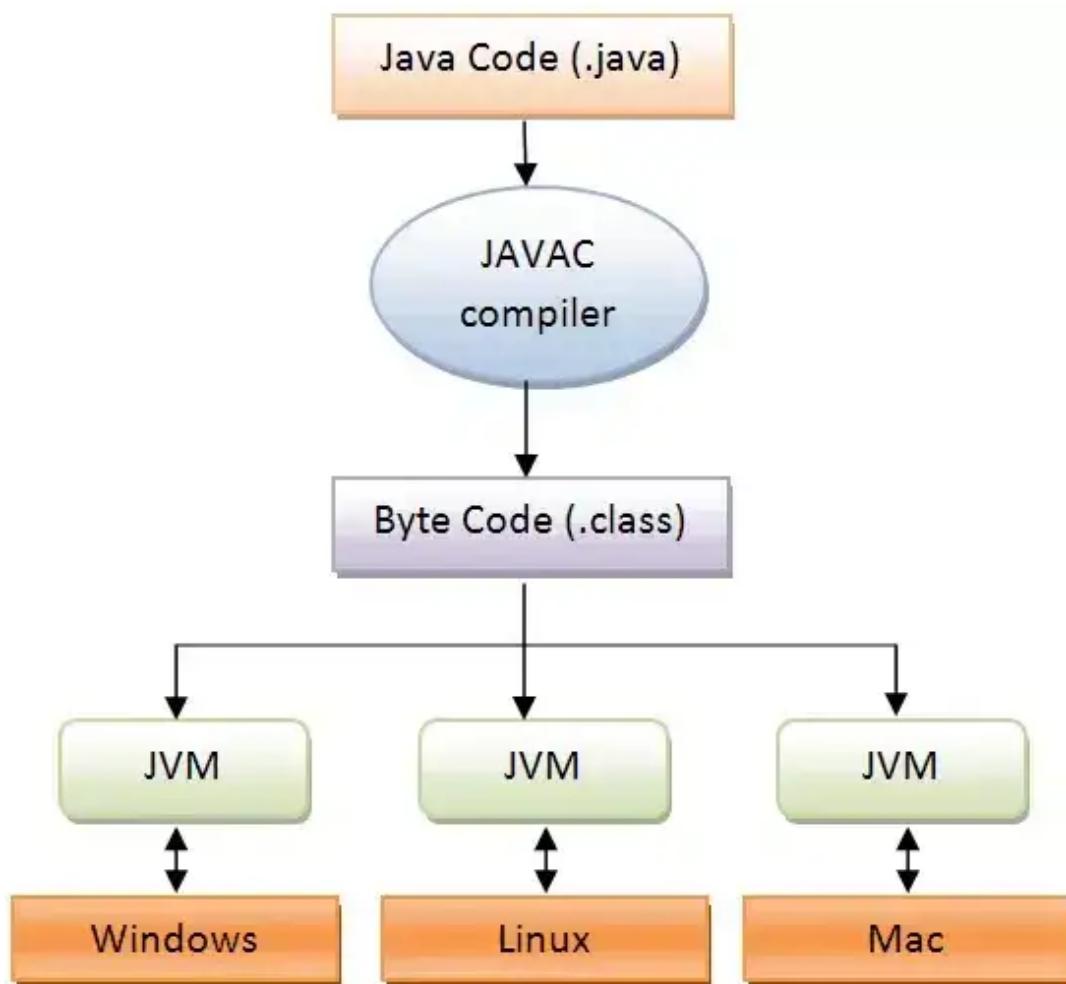
Print

	Name	Date mod
	compile.bat	11/7/2021
	hello.c	11/2/2021
	Makefile	11/2/2021

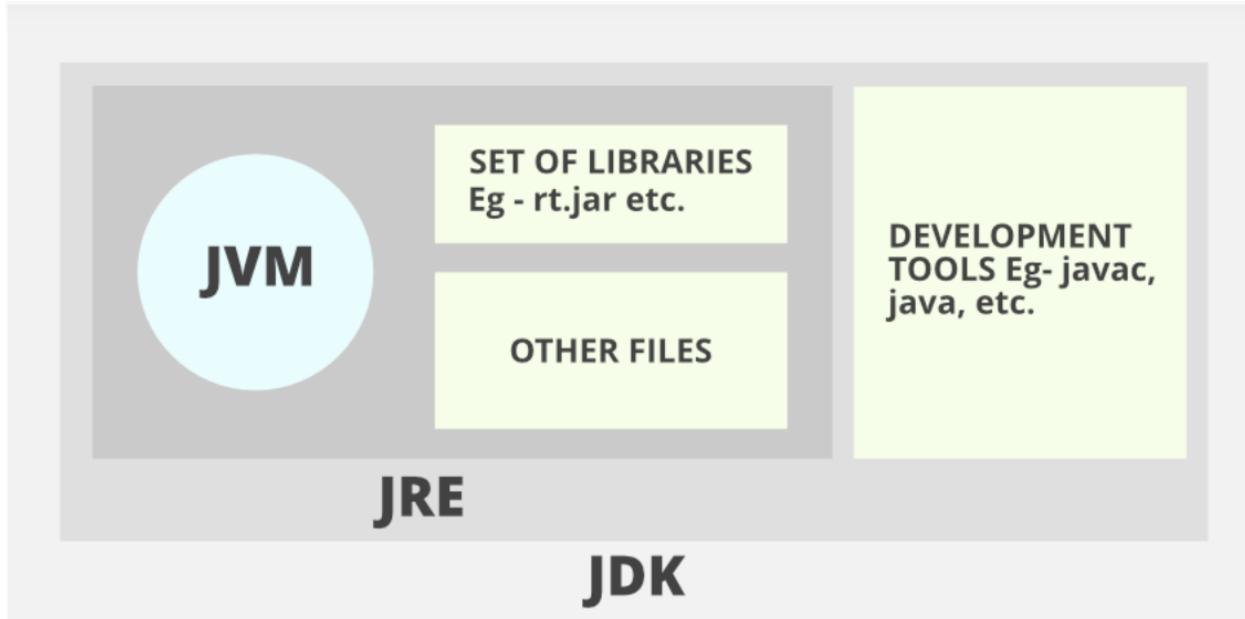
### 0.3.37.76 Make (4)

---

## 1 JAVA Environment and Development



### 1.0.1 JDK and JRE Setup (1)



### 1.0.2 JDK and JRE Setup (2)

- **JDK** (Java Development Kit) is a Kit that provides the environment to **develop and execute(run)** the Java program. JDK is a kit(or package) that includes two things
  - Development Tools(to provide an environment to develop your java programs)
  - JRE (to execute your java program).
- **JRE** (Java Runtime Environment) is an installation package that provides an environment to **only run(not develop)** the java program(or application)onto your machine. JRE is only used by those who only want to run Java programs that are end-users of your system.
- **JVM (Java Virtual Machine)<sup>31</sup>** is a very important part of both JDK and JRE because it is contained or inbuilt in both. Whatever Java program you run using JRE or JDK goes into JVM and JVM is responsible for executing the java program line by line, hence it is also known as an **i\*\*\*nterpreter\*\*\*<sup>32</sup>**.

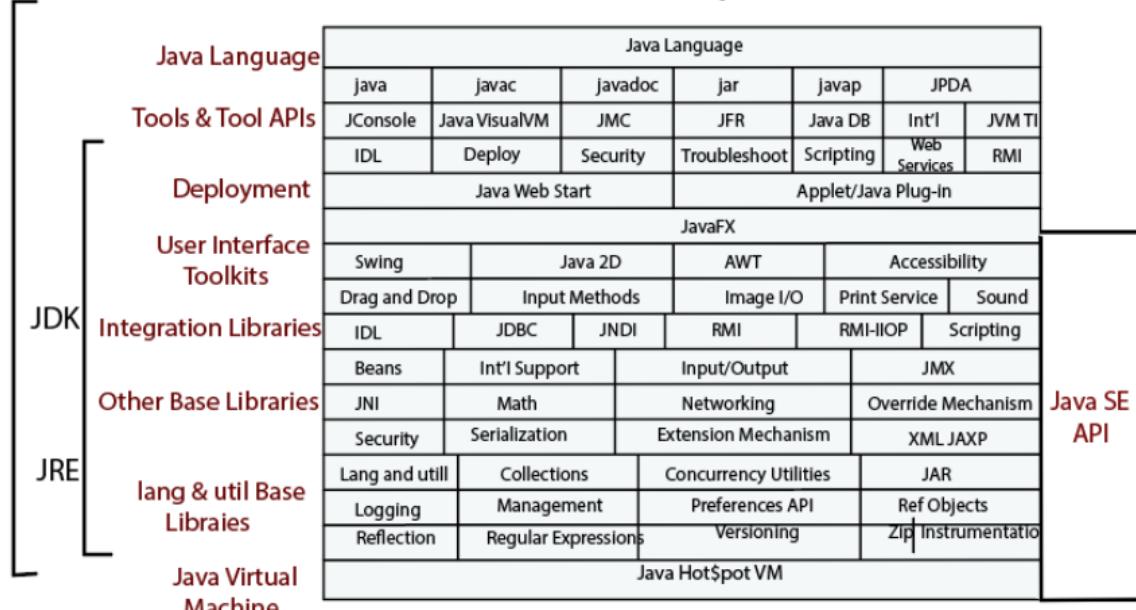
- 
- Difference between JDK, JRE, JVM - TutorialAndExample<sup>33</sup>

<sup>31</sup><https://www.geeksforgeeks.org/jvm-works-jvm-architecture/>

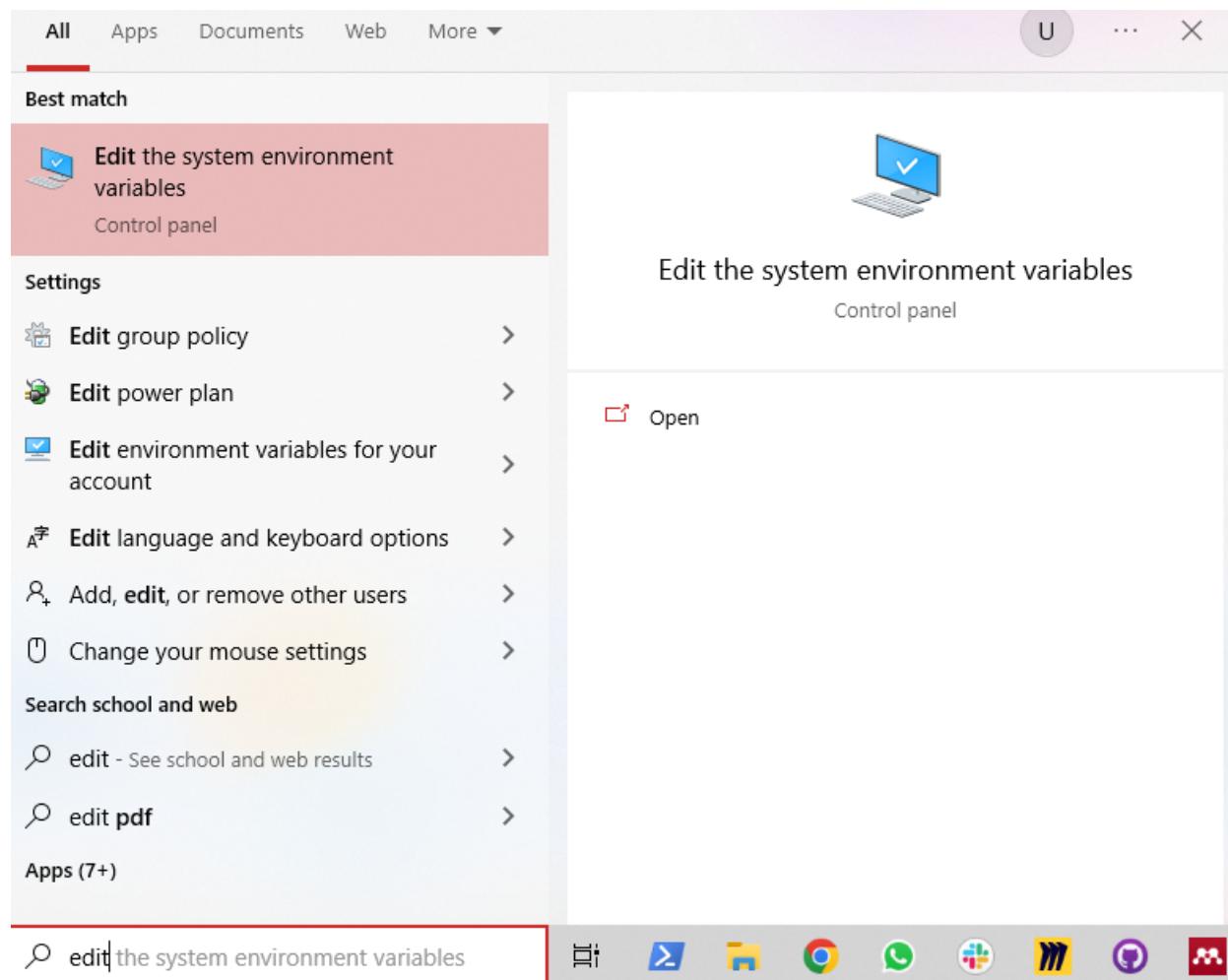
<sup>32</sup><https://www.geeksforgeeks.org/compiler-vs-interpreter-2/>

<sup>33</sup><https://www.tutorialandexample.com/difference-between-jdk-jre-jvm>

# JDK(Java Development Kit)

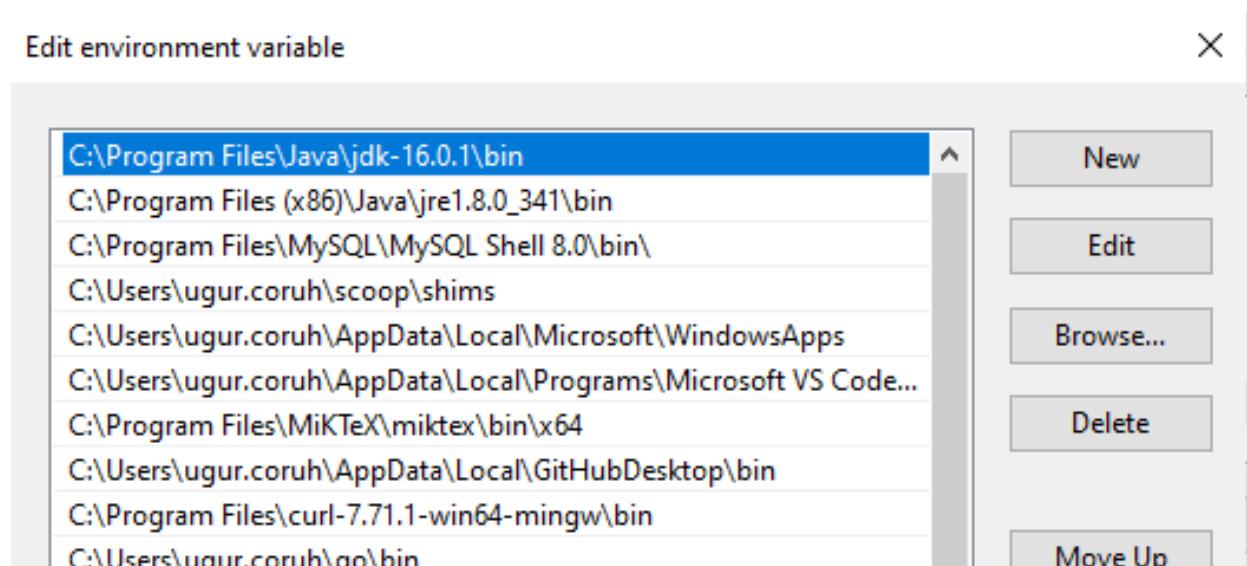


### 1.0.3 System Environments and Paths for Java (1)



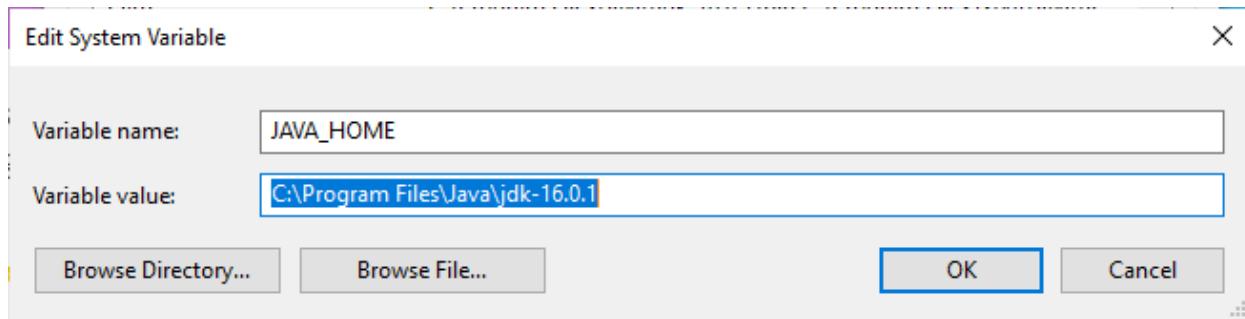
### 1.0.4 System Environments and Paths for Java (2)

- Select path variable (JDK should be set there)



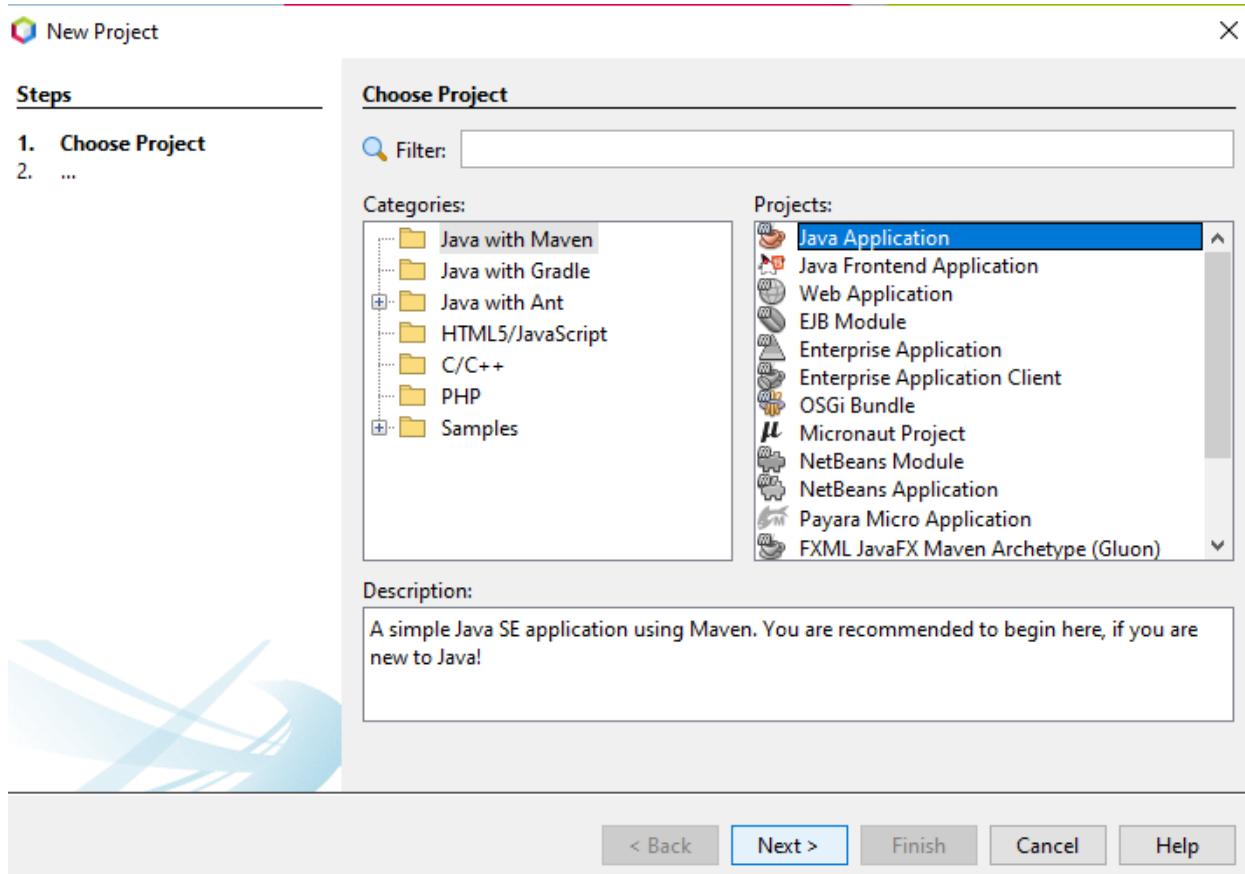
### 1.0.5 System Environments and Paths for Java (3)

- JAVA\_HOME also should be set

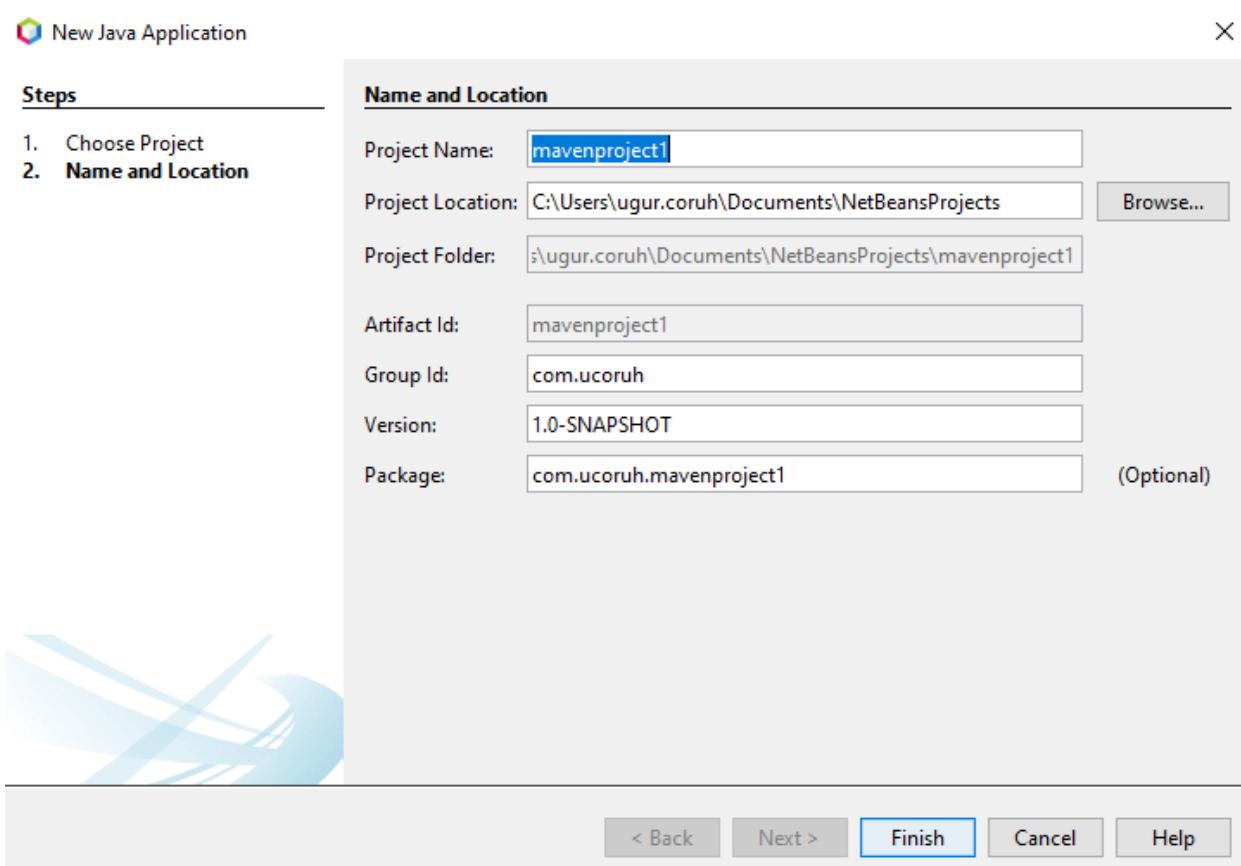


### 1.0.6 Netbeans (Java) (1)

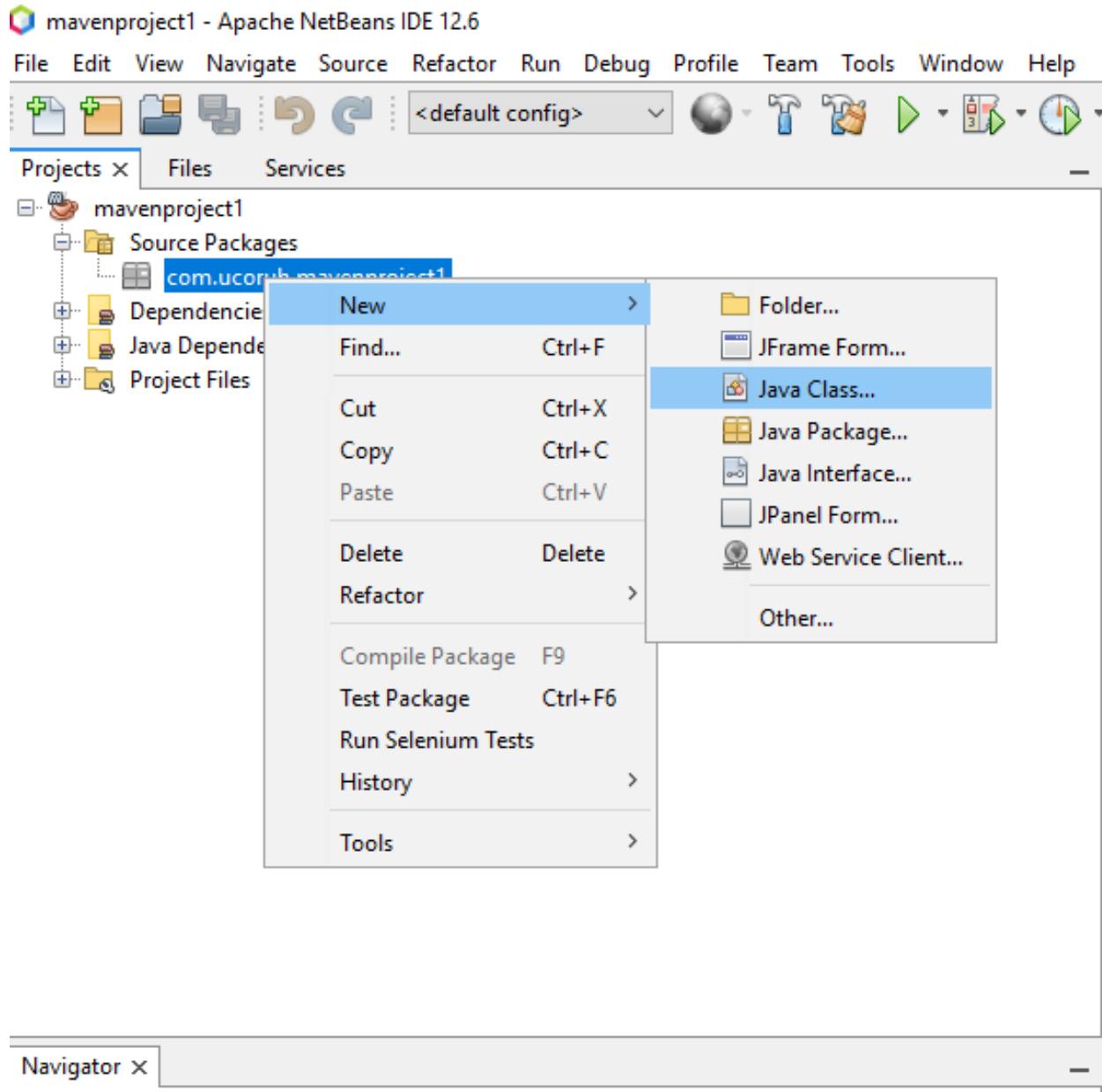
- Open New Project -> Java Project



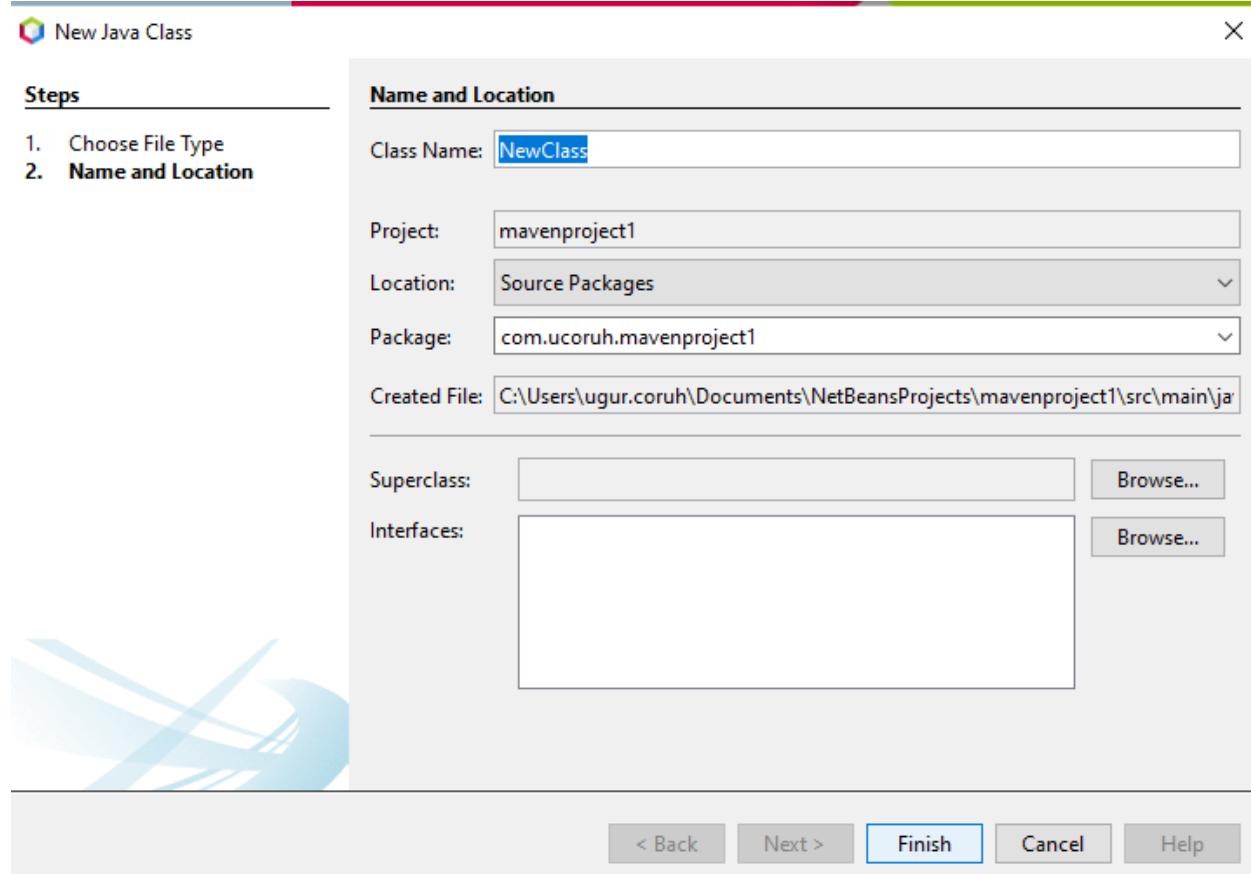
### 1.0.7 Netbeans (Java) (2)



### 1.0.8 Netbeans (Java) (3)



### 1.0.9 Netbeans (Java) (4)



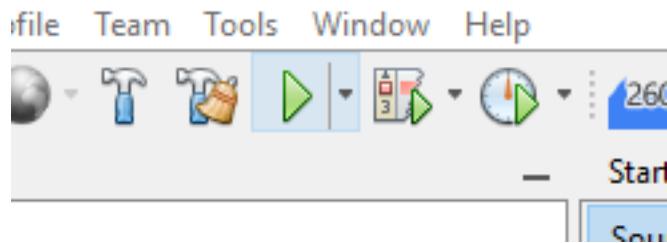
### 1.0.10 Netbeans (Java) (5)

The screenshot shows the Netbeans IDE with the 'NewClass.java' file open. The code editor displays the following Java code:

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package com.ucoruh.mavenproject1;
6
7  /**
8  * @author ugur.coruh
9  */
10 public class NewClass {
11
12 }
13
14 }
```

### 1.0.11 Netbeans (Java) (6)

Update code and run



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.ucoruh.mavenproject1;

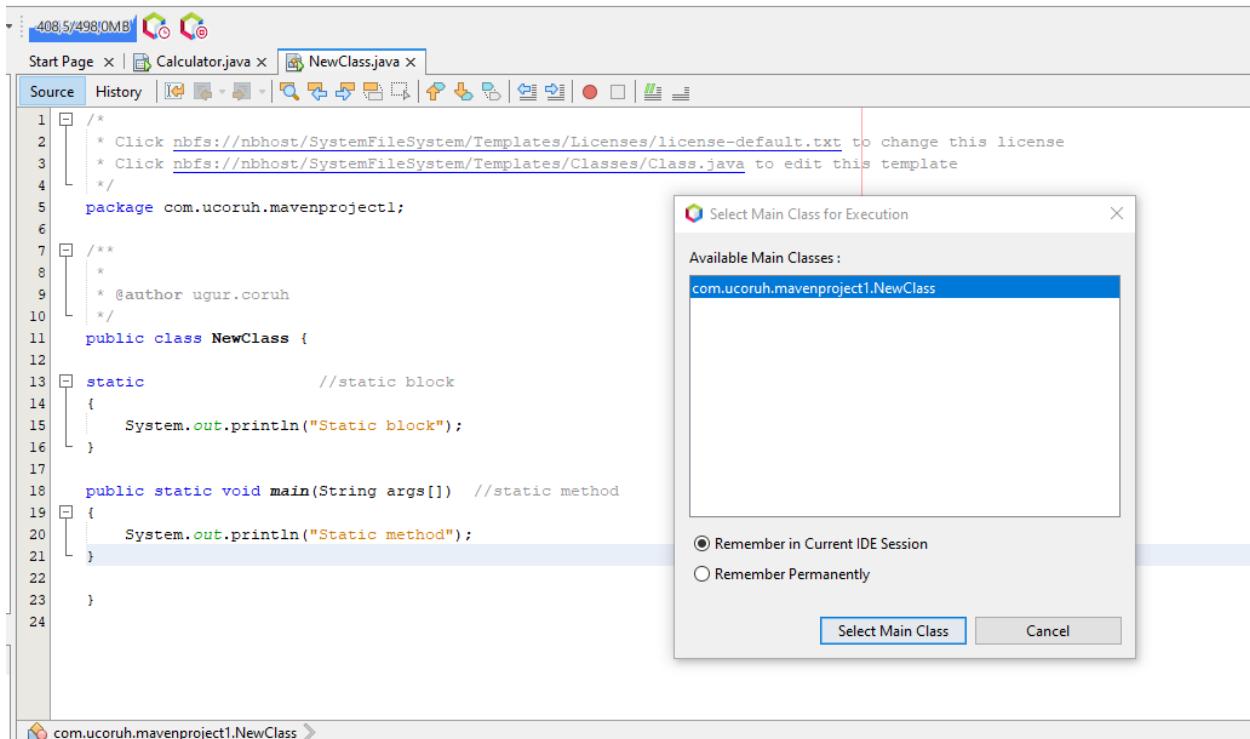
/**
 *
 * @author ugur.coruh
 */
public class NewClass {

    static //static block
    {
        System.out.println("Static block");
    }

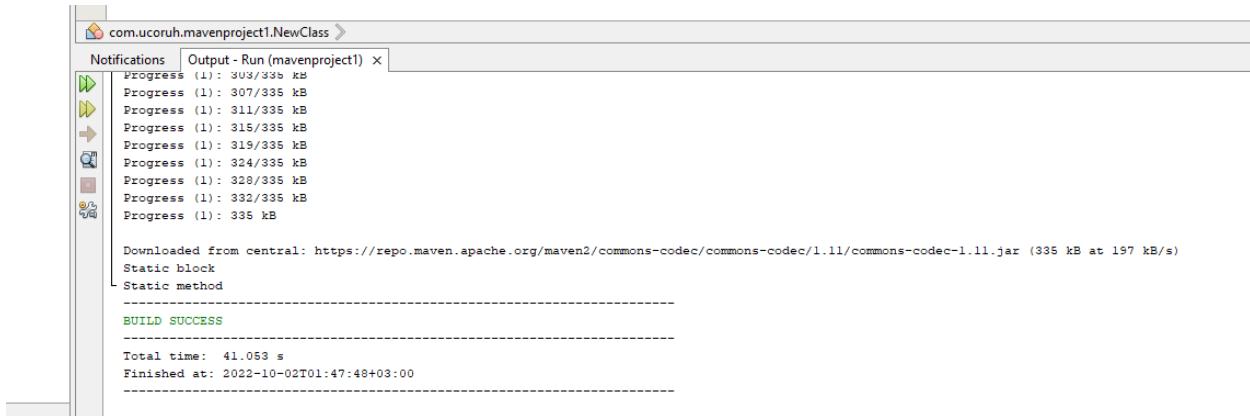
    public static void main(String args[]) //static method
    {
        System.out.println("Static method");
    }
}
```

---

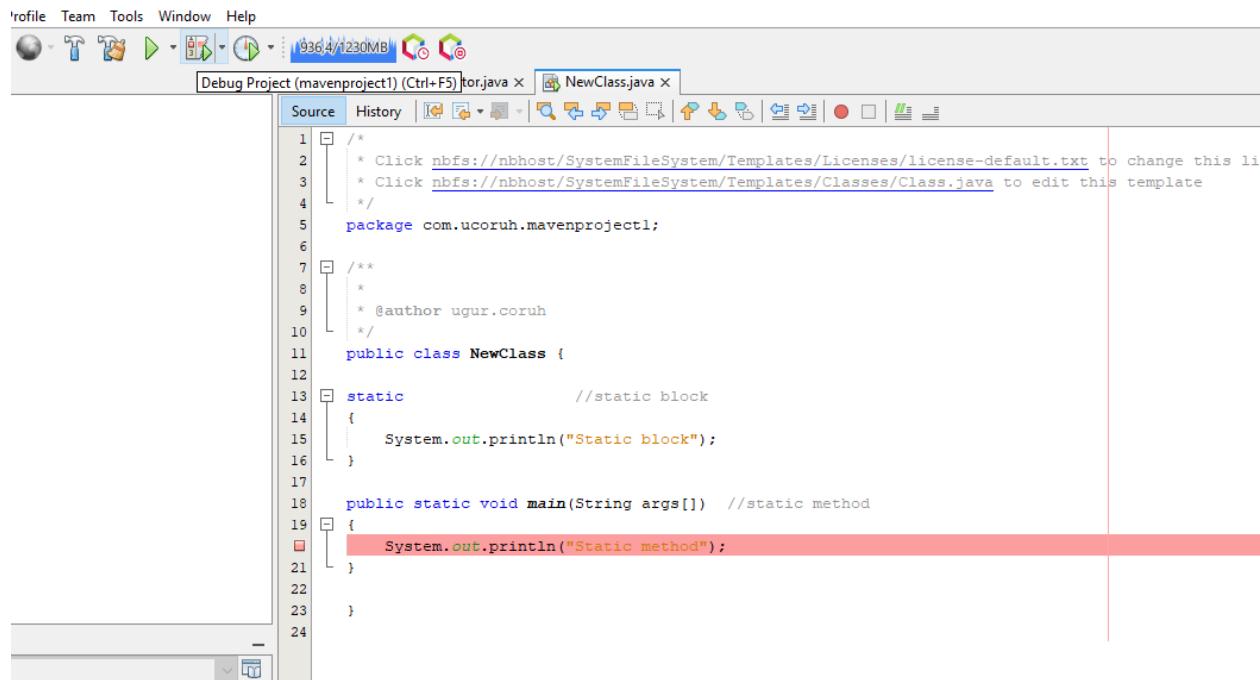
### 1.0.12 Netbeans (Java) (7)



### 1.0.13 Netbeans (Java) (8)



### 1.0.14 Netbeans (Java) (9)



The screenshot shows the Netbeans IDE interface with the following details:

- Toolbar:** Standard Netbeans toolbar with icons for file operations, project management, and toolbars.
- Project Bar:** Shows "Debug Project (mavenproject1)" and "NewClass.java x".
- Source Tab:** Active tab showing the Java code for `NewClass.java`. The code includes a static block and a main method.
- Code:**

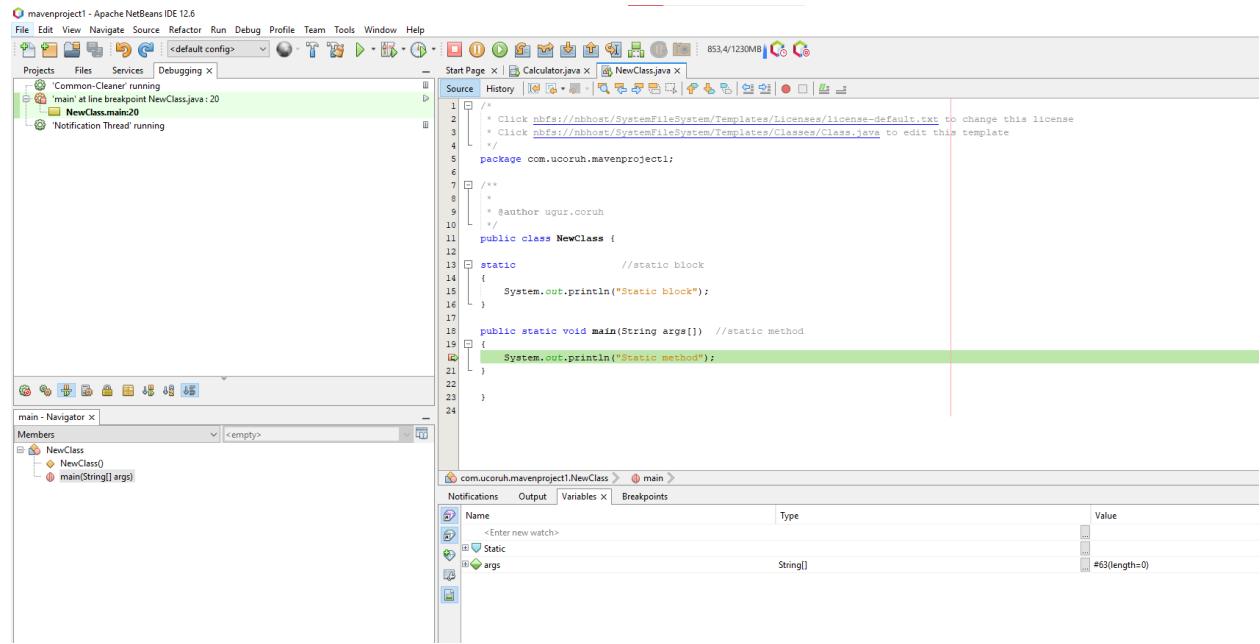
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.ucoruh.mavenproject1;

/**
 *
 * @author ugur.coruh
 */
public class NewClass {

    static //static block
    {
        System.out.println("Static block");
    }

    public static void main(String args[]) //static method
    {
        System.out.println("Static method");
    }
}
```

### 1.0.15 Netbeans (Java) (10)



The screenshot shows the Netbeans IDE interface with the following details:

- Toolbar:** Standard Netbeans toolbar with icons for file operations, project management, and toolbars.
- Project Bar:** Shows "Common Classes running", "main at breakpoint NewClass.java:20", and "Notification Thread running".
- Source Tab:** Active tab showing the Java code for `NewClass.java`. The code includes a static block and a main method.
- Code:**

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.ucoruh.mavenproject1;

/**
 *
 * @author ugur.coruh
 */
public class NewClass {

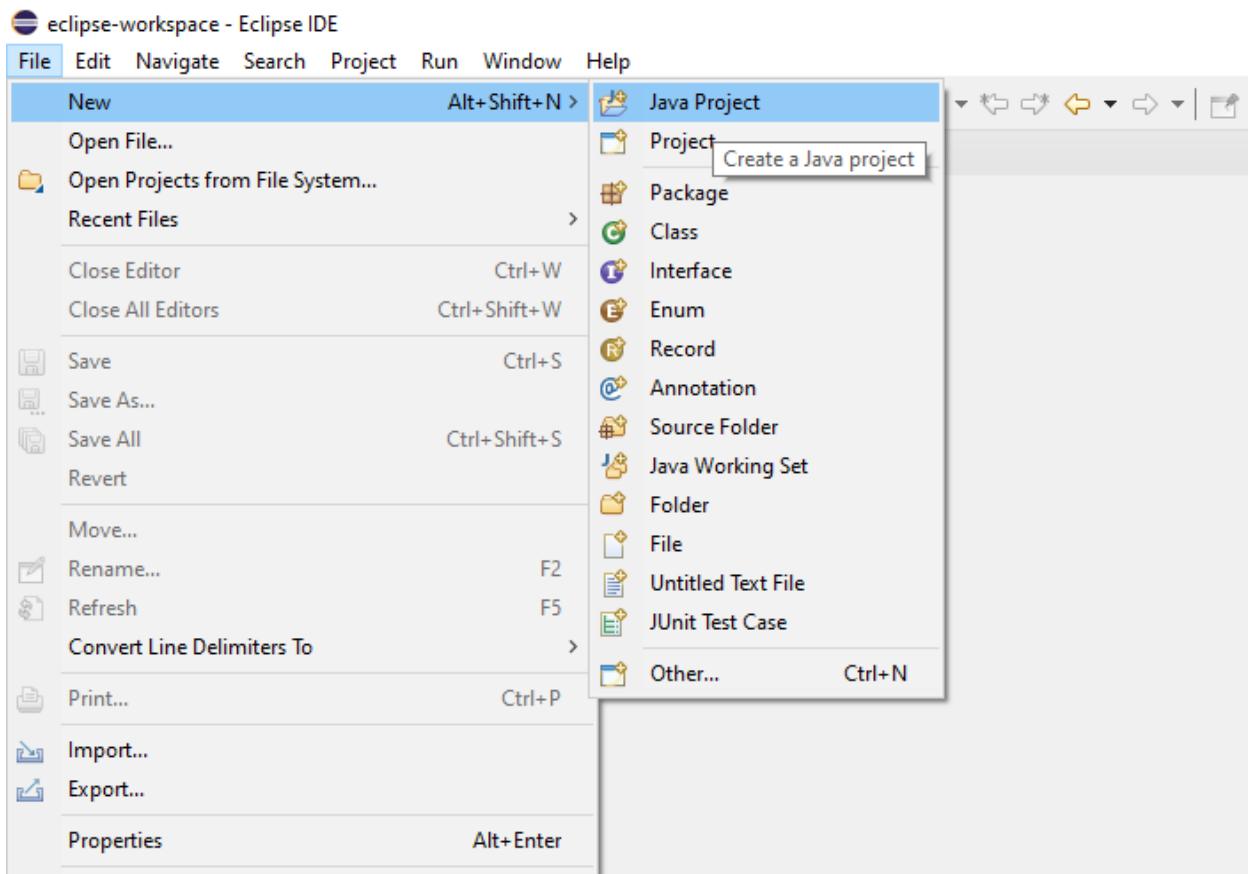
    static //static block
    {
        System.out.println("Static block");
    }

    public static void main(String args[]) //static method
    {
        System.out.println("Static method");
    }
}
```
- Bottom Panel:** Shows the Navigator, Members, and a Breakpoints table.

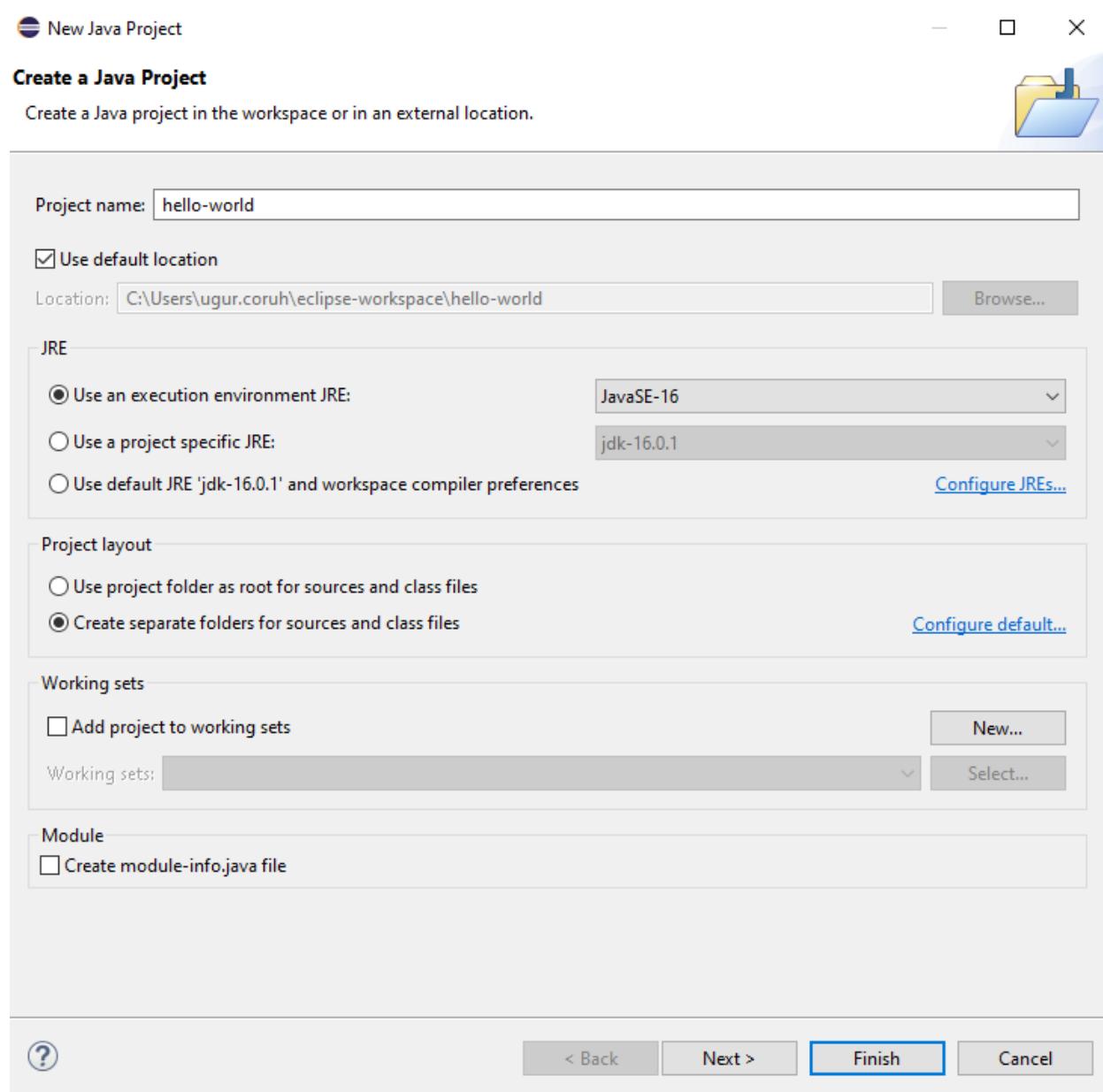
Name	Type	Value
<Enter new watch>		
Static	String[]	#63(length=0)
args		

### 1.0.16 Eclipse (Java) (1)

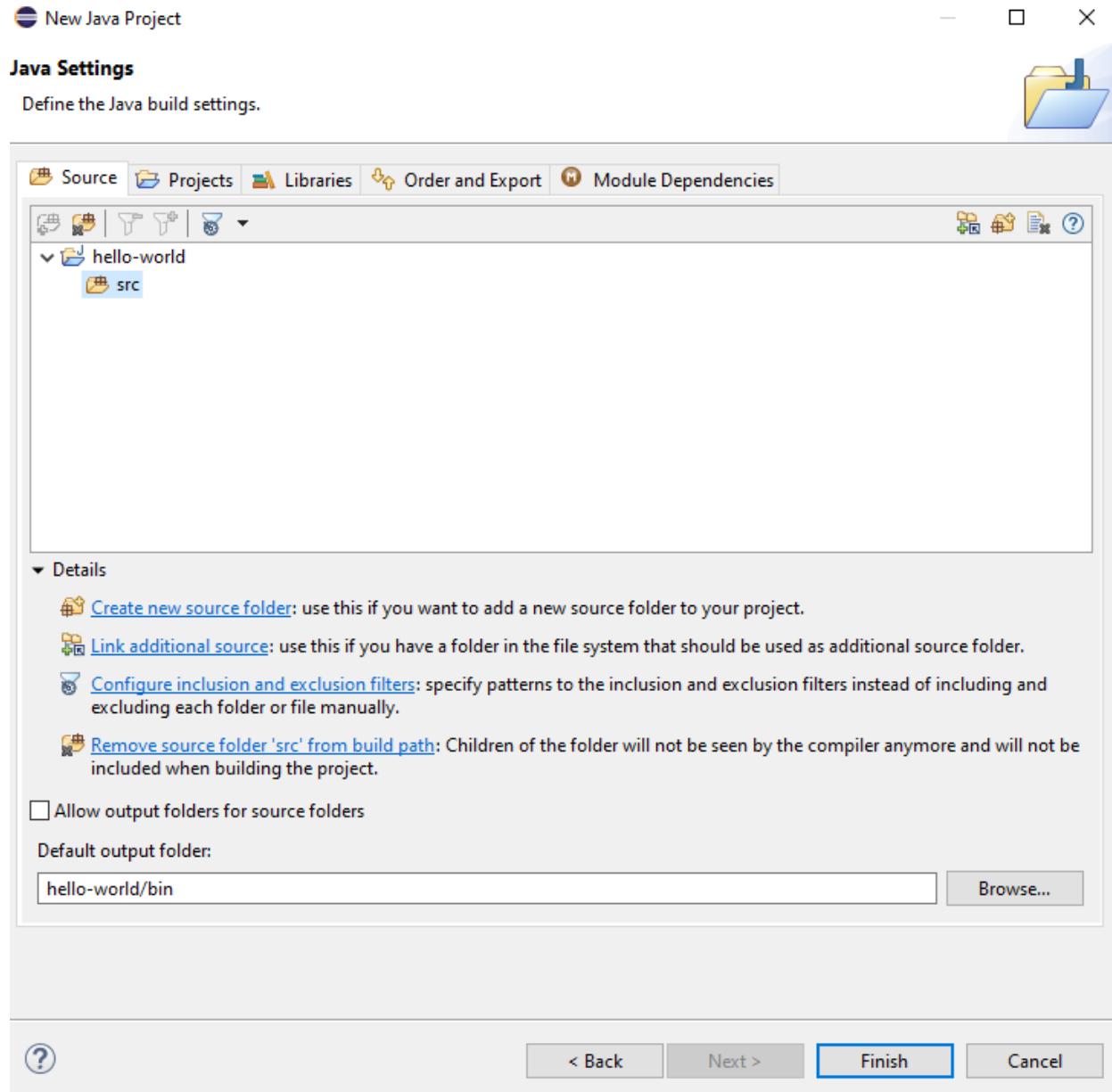
- Select File -> New Project



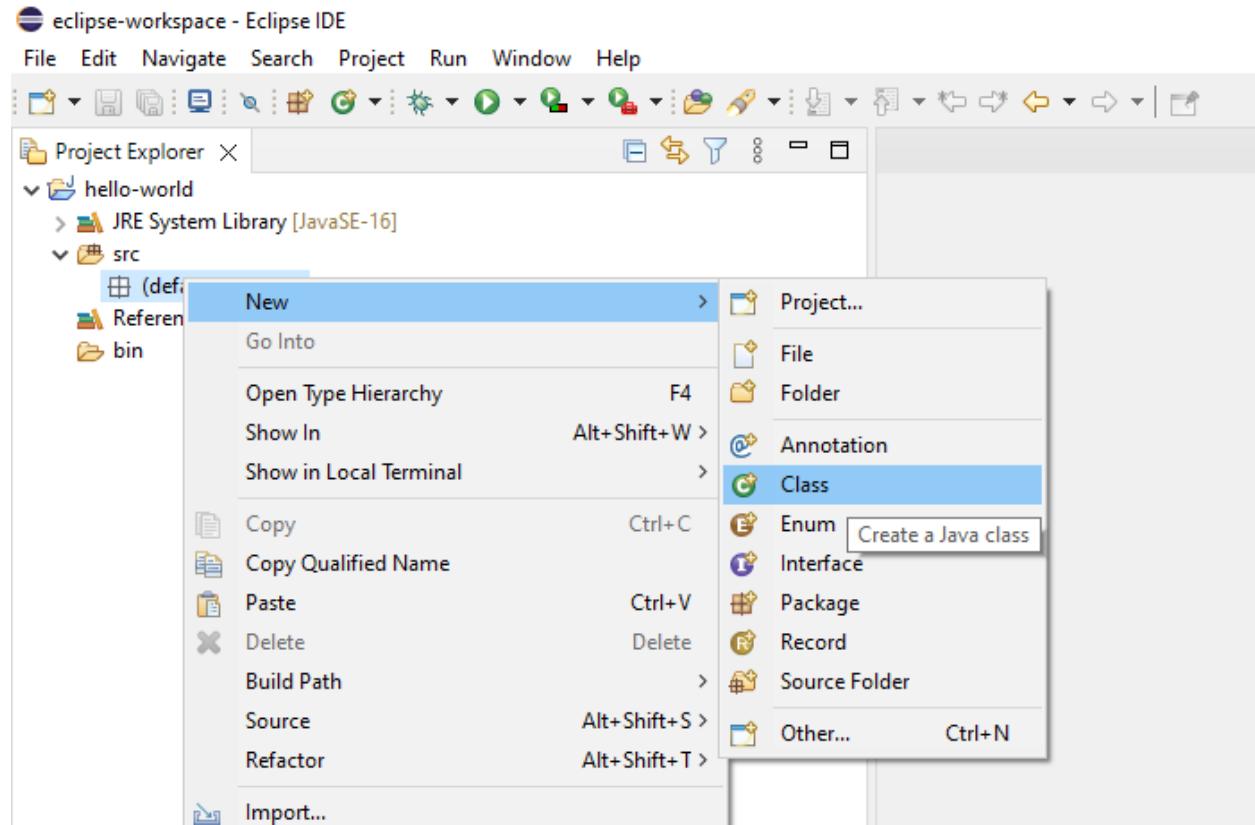
### 1.0.17 Eclipse (Java) (2)



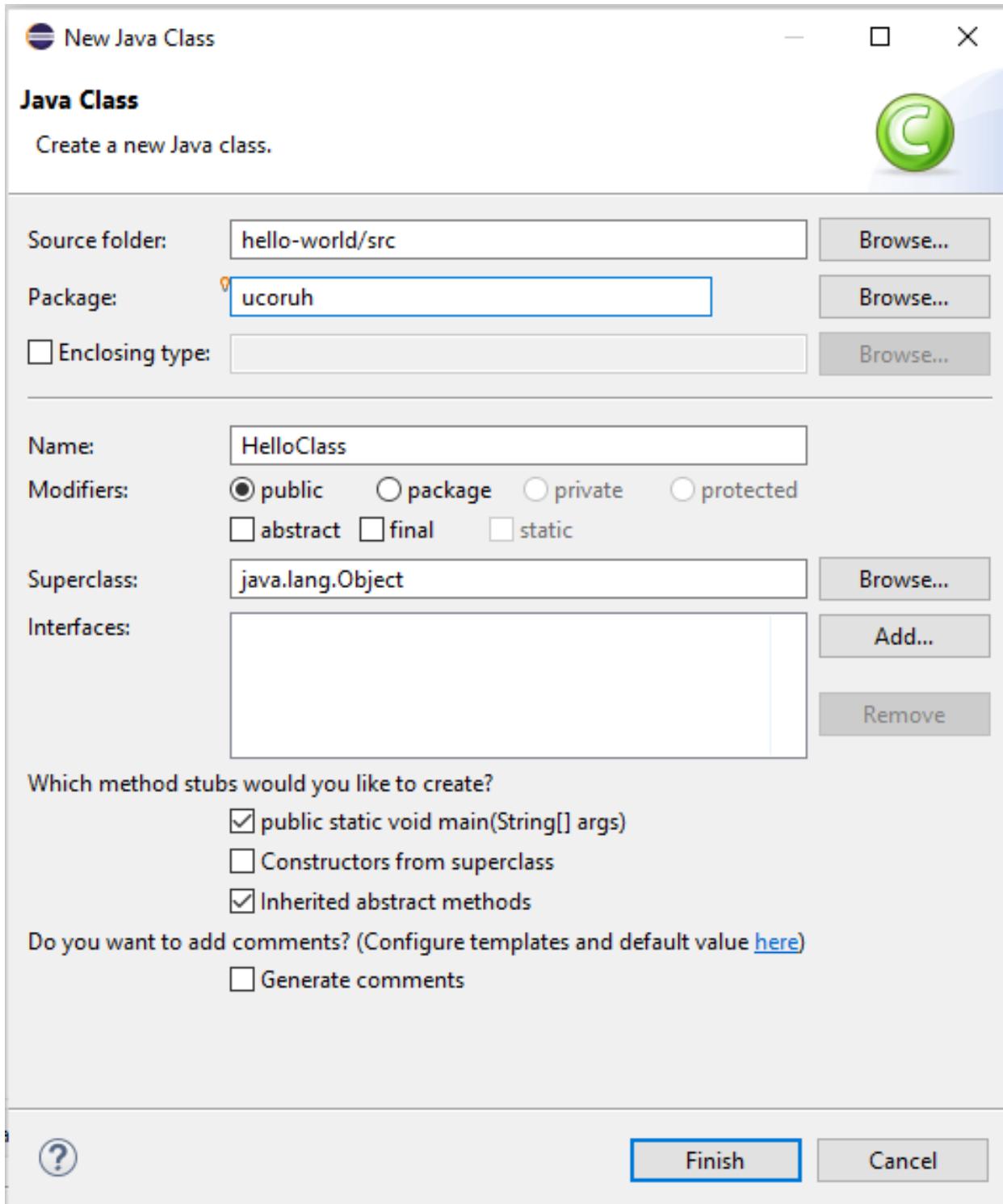
### 1.0.18 Eclipse (Java) (3)



### 1.0.19 Eclipse (Java) (4)



### 1.0.20 Eclipse (Java) (5)



### 1.0.21 Eclipse (Java) (6)

- Update source

```
package ucoruh;
```

```
public class HelloClass {
```

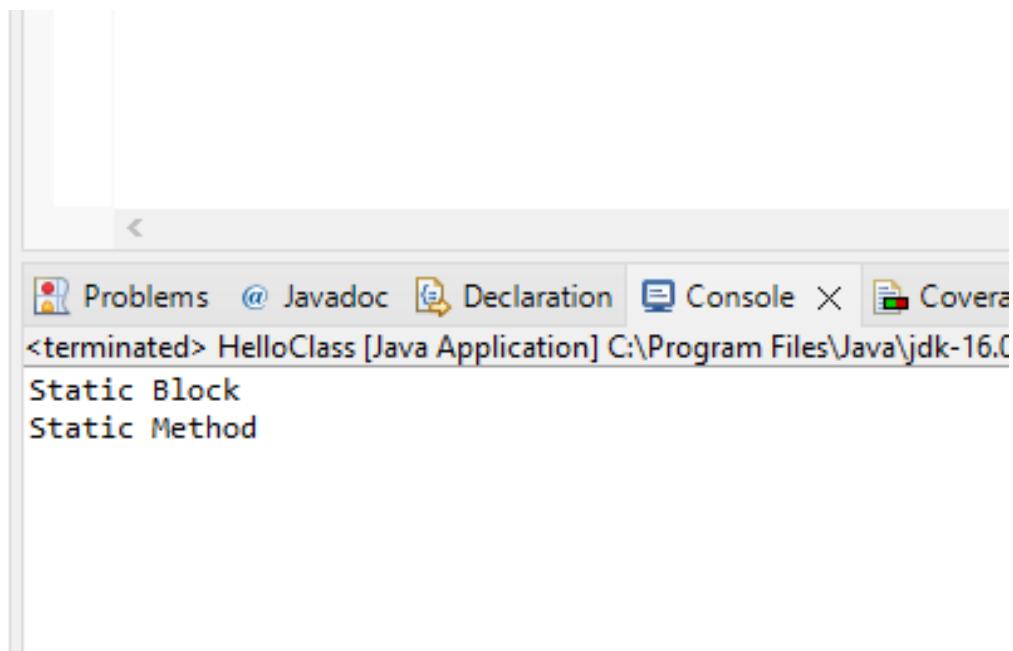
```
static {
    System.out.println("Static Block");
}

public static void main(String[] args) {
    // TODO Auto-generated method stub
    System.out.println("Static Method");

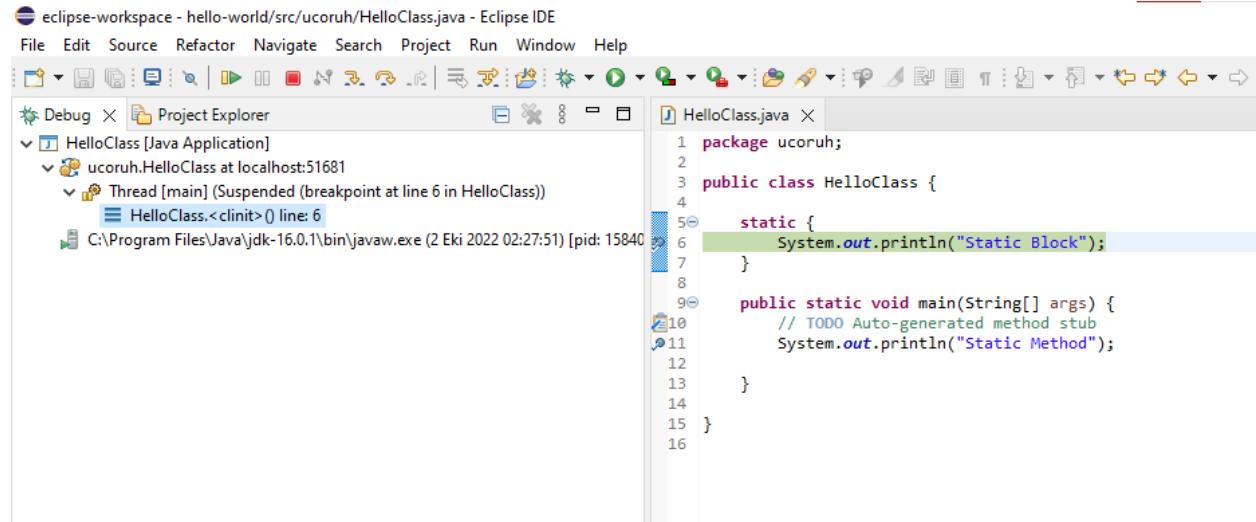
}
```

---

#### 1.0.22 Eclipse (Java) (7)



### 1.0.23 Eclipse (Java) (8)



The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The left sidebar has a 'Debug' tab and a 'Project Explorer' tab, with 'HelloClass [Java Application]' selected. Under 'HelloClass', there is a 'ucoruh.HelloClass at localhost:51681' entry with a 'Thread [main] (Suspended (breakpoint at line 6 in HelloClass))' sub-item. The right side is a code editor for 'HelloClass.java' with the following content:

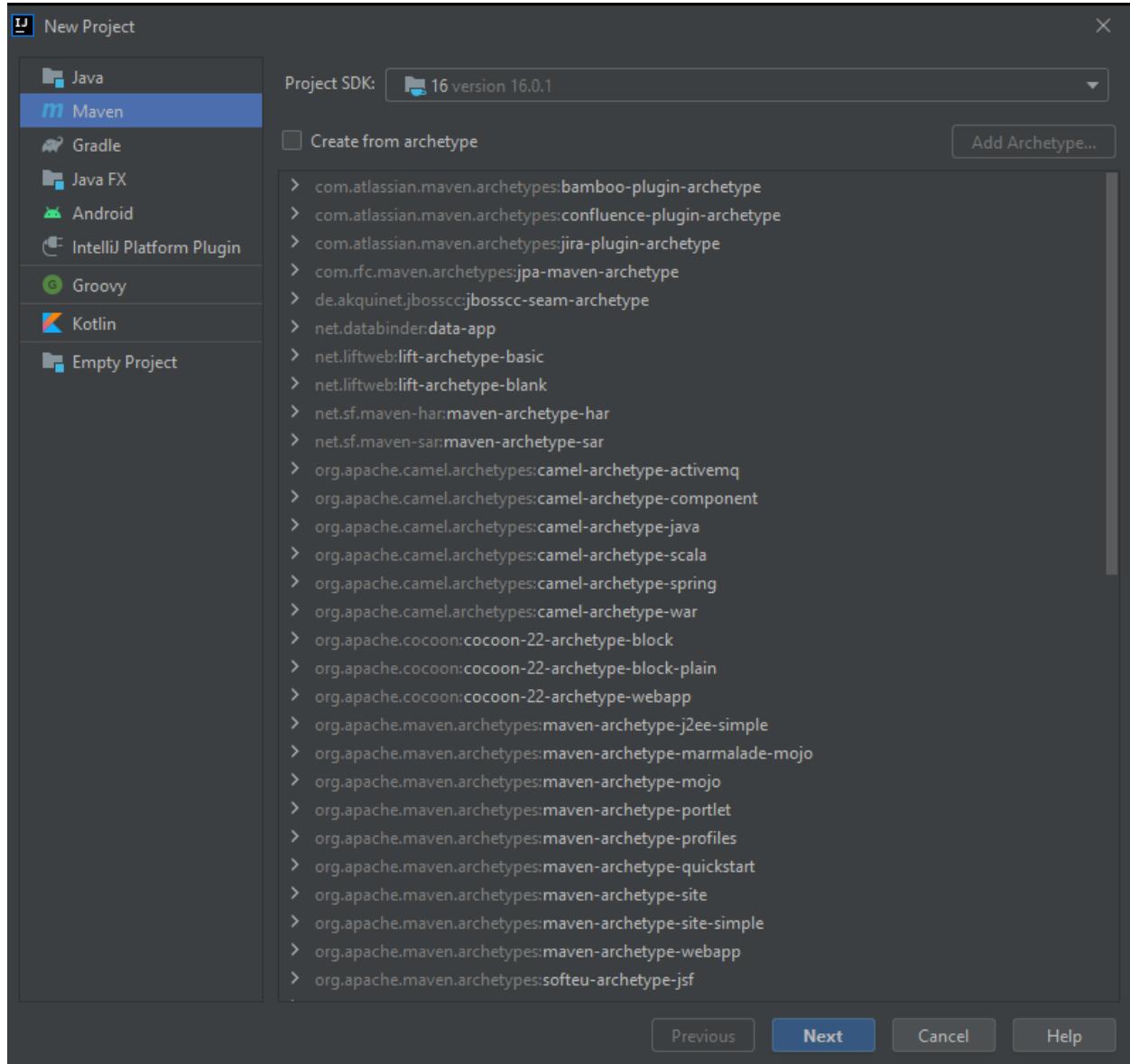
```
1 package ucoruh;
2
3 public class HelloClass {
4
5     static {
6         System.out.println("Static Block");
7     }
8
9
10    public static void main(String[] args) {
11        // TODO Auto-generated method stub
12        System.out.println("Static Method");
13    }
14
15 }
16
```

A blue vertical bar highlights line 6, indicating it is the current line of execution.

### 1.0.24 IntelliJ Idea (JetBrains) (Java)

- Download IntelliJ IDEA: The Capable & Ergonomic Java IDE by JetBrains<sup>34</sup>
  - Select Community Version or Student Ultimate Version

<sup>34</sup><https://www.jetbrains.com/idea/download/#section=windows>



### 1.0.25 VSCode (Java)

- Java Extension Run&Debug Java Files

The screenshot shows a Java file named `HelloClass.java` in the Visual Studio Code editor. The code contains a static block and a main method. The editor has a sidebar on the left with various icons and sections like RUN and DEBUG. Below the editor is a bottom navigation bar with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, JUPYTER, and others. The status bar at the bottom shows file paths and other system information.

```

1 package ucoruh;
2
3 public class HelloClass {
4
5     static {
6         System.out.println("Static Block");
7     }
8
9     public static void main(String[] args) {
10        // TODO Auto-generated method stub
11        System.out.println("Static Method");
12    }
13
14 }
15

```

### 1.0.26 Notepad++ (Java)

- How to Compile and Run Java Programs Using Notepad++<sup>35</sup>

---

### 1.0.27 Cmake (Java)

- UseJava — CMake 3.24.2 Documentation<sup>36</sup>
- GitHub - ptitpoulpe/cmake-swig-java-example: An example of combining cmake, swig and java<sup>37</sup>

---

## 2 C# Environment and Development

---

### 2.0.1 Visual Studio Community Edition (C#)

//TODO//

<sup>35</sup><https://www.wikihow.com/Compile-and-Run-Java-Program-by-Notepad>

<sup>36</sup><https://cmake.org/cmake/help/latest/module/UseJava.html>

<sup>37</sup><https://github.com/ptitpoulpe/cmake-swig-java-example>



---

## 2.0.2 Notepad++ (C#)

- This use command-line utilities for csharp, nppeexec should be configured for this utility.
- Compiling/Executing a C# Source File in Command Prompt - Stack Overflow<sup>38</sup>

```
c:\windows\Microsoft.NET\Framework\v3.5\  
c:\windows\Microsoft.NET\Framework\v3.5\bin\csc.exe  
    /t:exe /out:MyApplication.exe MyApplication.cs ...
```

---

## 2.0.3 Cmake (C#)

- GitHub - crud89/DotNetWithCMake: Your swiss army knife for creating .NET assemblies with CMake and integrating unmanaged code.<sup>39</sup>

---

## 2.0.4 Common Tools and Platforms

---

## 2.0.5 Fatih Kalem



[https://cdnvideo.eba.gov.tr/fatihkalem/fatihkalem\\_portable.zip](https://cdnvideo.eba.gov.tr/fatihkalem/fatihkalem_portable.zip)

[https://cdnvideo.eba.gov.tr/fatihkalem/fatihkalem\\_setup.exe](https://cdnvideo.eba.gov.tr/fatihkalem/fatihkalem_setup.exe)

---

<sup>38</sup><https://stackoverflow.com/questions/553143/compiling-executing-a-c-sharp-source-file-in-command-prompt>

<sup>39</sup><https://github.com/crud89/DotNetWithCMake>



---

## 2.0.6 Notepad++ (Notepad for Source Code)

Downloads | Notepad++<sup>40</sup>

---

<sup>40</sup><https://notepad-plus-plus.org/downloads/>

The screenshot shows the Notepad++ interface with a single file open: "notepad4ever.cpp". The code contains a simple infinite loop:

```
1 #include <GPL.h>
2 #include <free_software.h>
3
4 void notepad4ever()
5 {
6     while (true)
7     {
8         NotePad++;
9     }
10 }
```

---

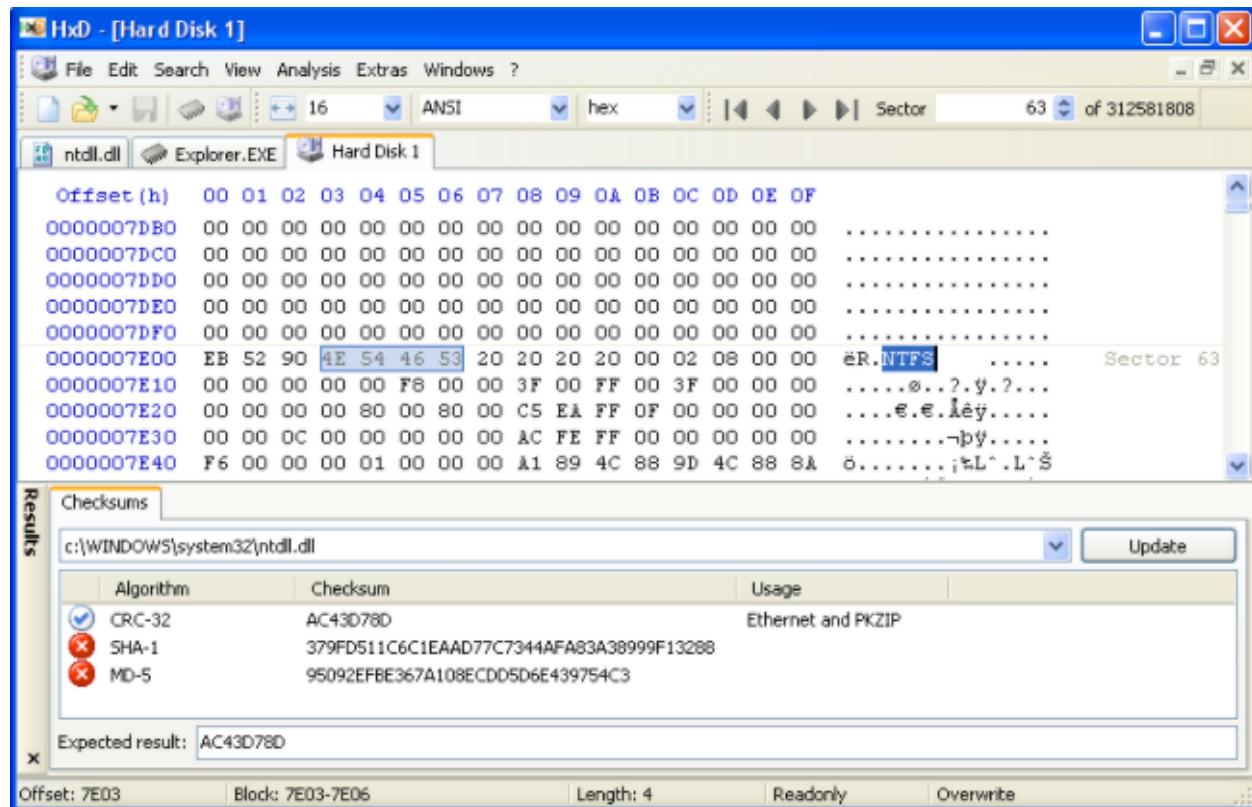
## 2.0.7 HxD (Hex Editor)



HxD - Freeware Hex Editor and Disk Editor | mh-nexus<sup>41</sup>

---

<sup>41</sup><https://mh-nexus.de/en/hxd/>



## 2.0.8 Marktext (Markdown Syntax Editor)



- <https://marktext.app/>
- <https://github.com/marktext/marktext/releases>
- Download latest version
  - <https://github.com/marktext/marktext/releases/tag/v0.17.1>

An h1 header

Paragraphs are separated by a blank line.

2nd paragraph. *italic*, **bold**, and `monospace`. Itemized lists look like:

- this one
- that one
- the other one

Note that --- not considering the asterisk --- the actual text content starts at 4-columns in.

Block quotes are written like so.

They can span multiple paragraphs,

B I ~~S~~ ~~H~~ ~~X~~ ~~P~~ ~~U~~ ~~C~~

Use 3 dashes for an em-dash. Use 2 dashes for ranges (ex., "it's all in chapters 12–14"). Three dots ... will be converted to an ellipsis. Unicode is supported. ☺

An h2 header

Here's a numbered list:

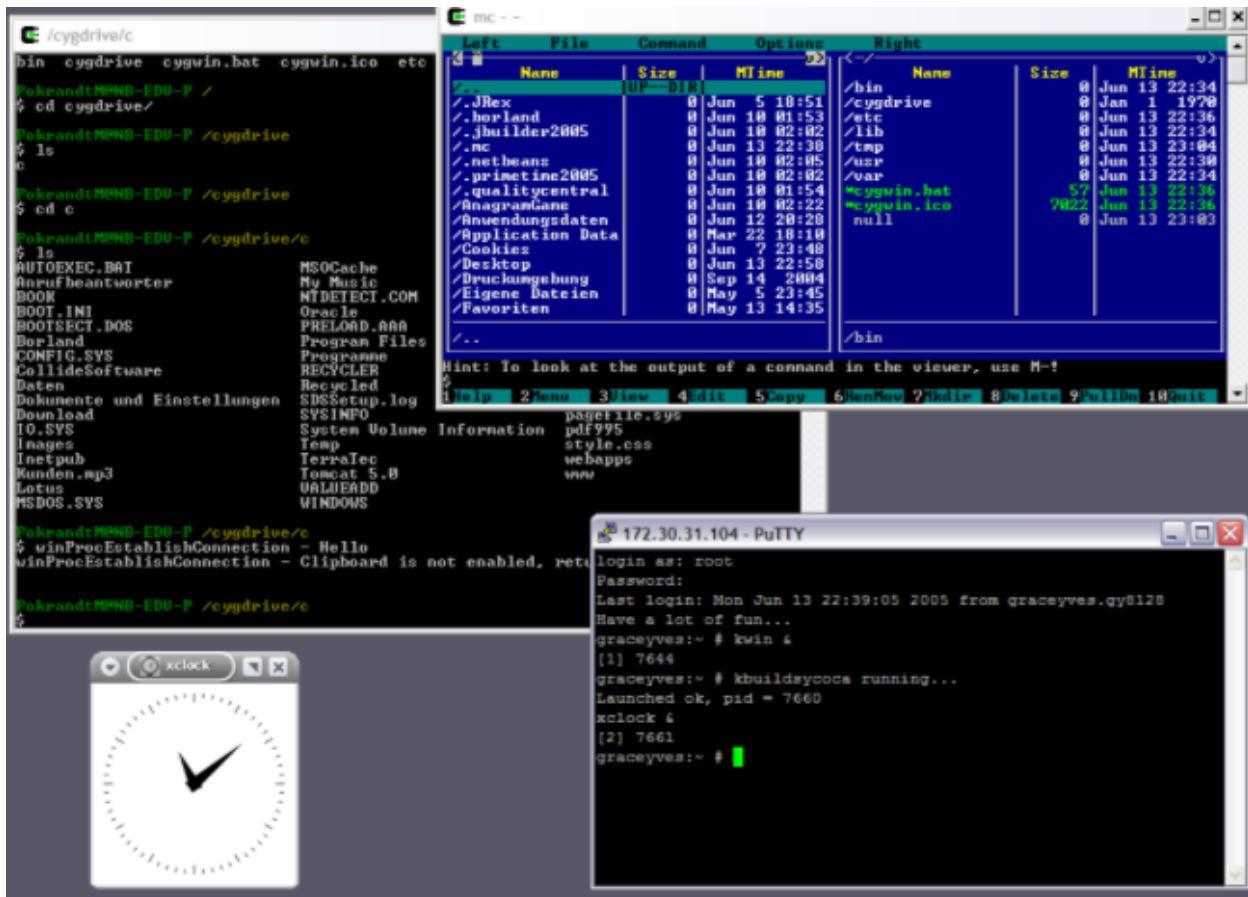
1. first item
2. second item
3. third item

---

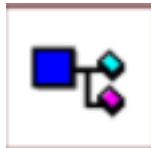
## 2.0.9 Cygwin (Linux environment for Windows)



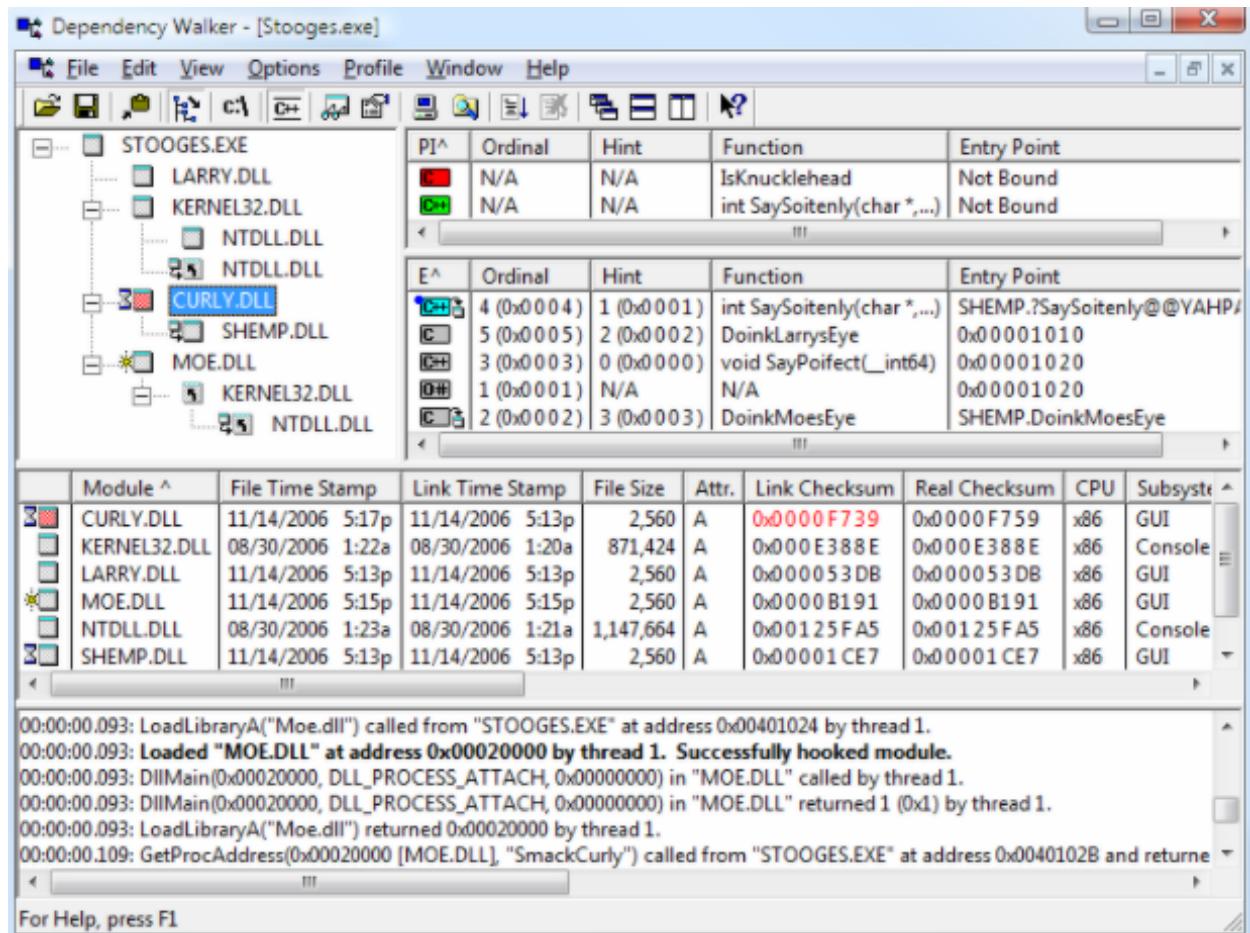
- <https://www.cygwin.com/>



## 2.0.10 Dependency Walker (32-bit or 64-bit Windows module dependency checker)



- <https://www.dependencywalker.com/>



## 2.0.11 Doxygen (Code Documentation)



Doxxygen: Doxygen<sup>42</sup>

<sup>42</sup><https://www.doxygen.nl/index.html>

**ACME Project** 1.0

Main Page Classes

Class List Class Index Class Members

Search

Public Member Functions | Public Attributes | List of all members

ACME Project  
Classes  
Class List  
acme  
ACMESmartphone  
App  
Dynamite  
Class Index  
Class Members

**acme.ACMESmartphone Class Reference**

**Public Member Functions**

ACMESmartphone (double model, String license)  
String `findRoadRunner` (String city, String state) throws IOException  
void `zapRoadRunner` (int voltage) throws IOException

**Public Attributes**

String `LongLat` = "Longitude = 39.2334, Latitude = 41.4899"

**Detailed Description**

Works like a regular smartphone but also tracks roadrunners.

The ACME Smartphone can perform similar functions as other smartphones, such as making phone calls, sending

Generated on Sun Sep 27 2015 09:05:27 for ACME Project by **doxygen** 1.6.7

## 2.0.12 Sonarlint (Code Quality and Code Security Extension)



<https://www.sonarlint.org/>

---

## 2.0.13 Codepen.io (online code sharing)



- <https://codepen.io/>
- CodePen is a social development environment. At its heart, it allows you to write code in the browser, and see the results of it as you build.
- A useful and liberating online code editor for developers of any skill, and particularly empowering for people learning to code. We focus primarily on front-end languages like HTML, CSS, JavaScript, and preprocessing syntaxes that turn into those things

The screenshot shows a Codepen.io project titled "SVG Stroke Animation" by Toshiyuki TAKAHASHI. The project URL is [codepen.io/gau/pen/ZWZKxO](https://codepen.io/gau/pen/ZWZKxO). The interface is divided into three main sections: HTML, CSS (SCSS), and JS.

```

HTML:


<svg id="hello-svg" data-name="hello" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 582 197">
    <title>Hello</title>
    <path class="path path-1" d="M208,338c38-16.67,73.74-45.72,97.33-66.21,33-18.33,32.67-35.67,37.33-52.67c47.12,203.12,344,192,332,192c-11,0-21,10.33-24.94,27.68-4.52,19.89-"/>


```

```

CSS (SCSS):
body {
  background: #ff837b;
}
#container {
  width: 600px;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}
.fin {
  .path-1, .path-2,
  .path-3 {

```

```

JS:
$(function() {
  function animationStart() {
    $('#container').addClass('fin');
  }
  setTimeout(animationStart, 250);
});

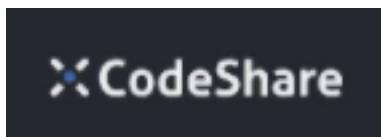
```

#### 2.0.14 Codepen.io (online code sharing)

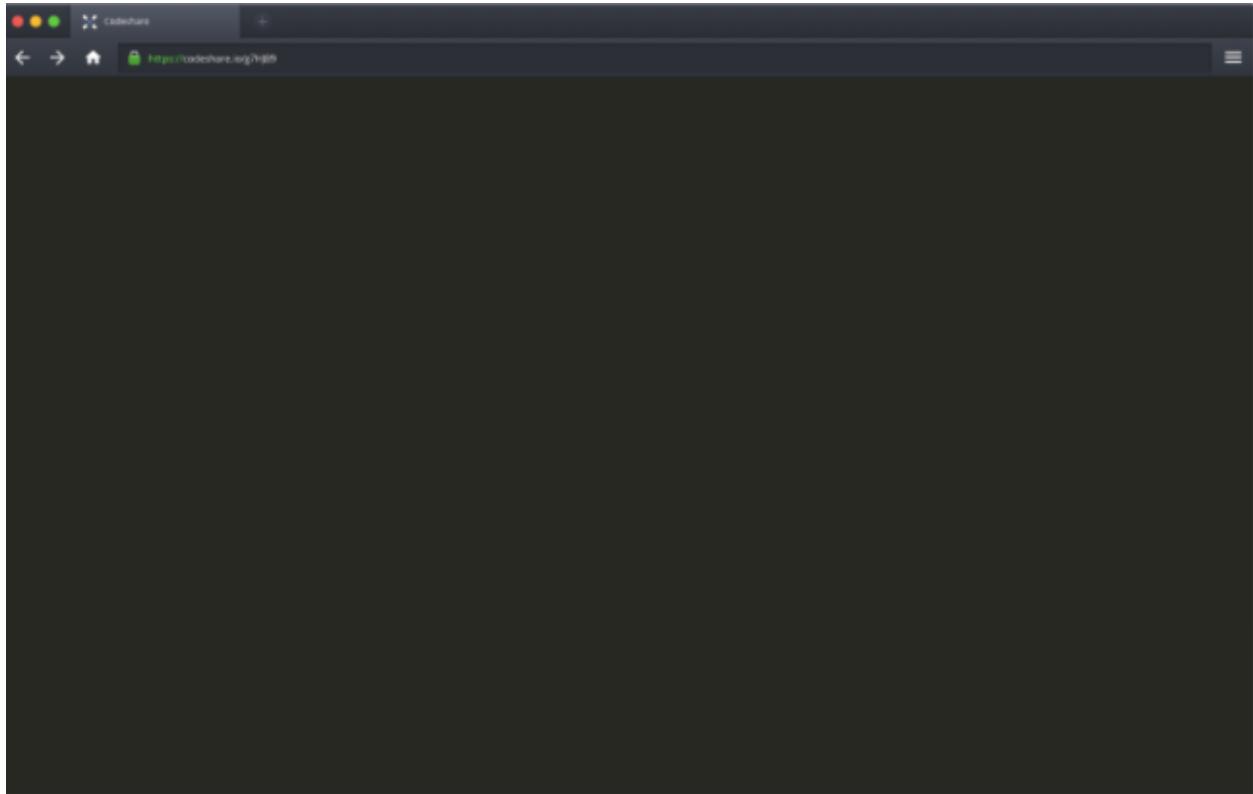
- Credit Card Sample on Codepen
  - <https://codepen.io/quinlo/pen/YONMEA>
- Checkout trends <https://codepen.io/trending>

---

#### 2.0.15 Codeshare.io (real-time code sharing)

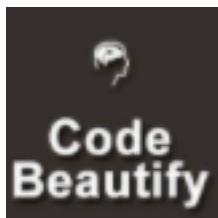


- <https://codeshare.io/>
- Share Code in Real-time with Developers, An online code editor for interviews, troubleshooting, teaching & more...



---

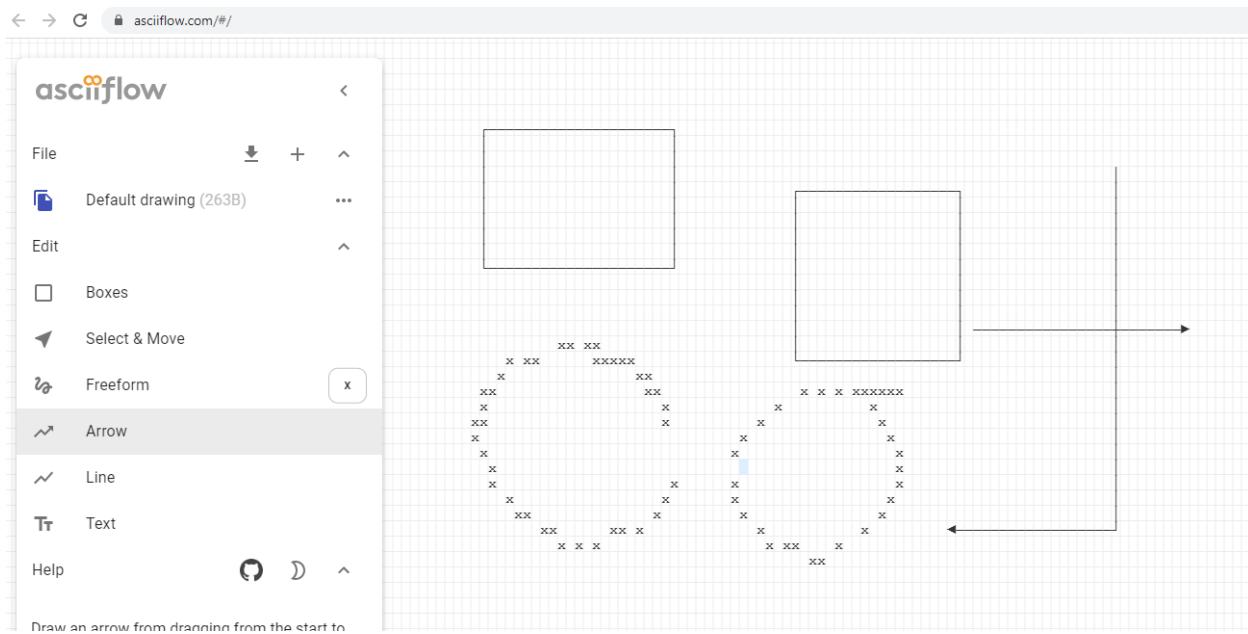
#### 2.0.16 Codebeautify.org (online data conversion tools)



- Has several tools for developers (Code Formatter, JSON Beautifier, XML Viewer, Hex Converters and more...)
  - <https://codebeautify.org/>
- 

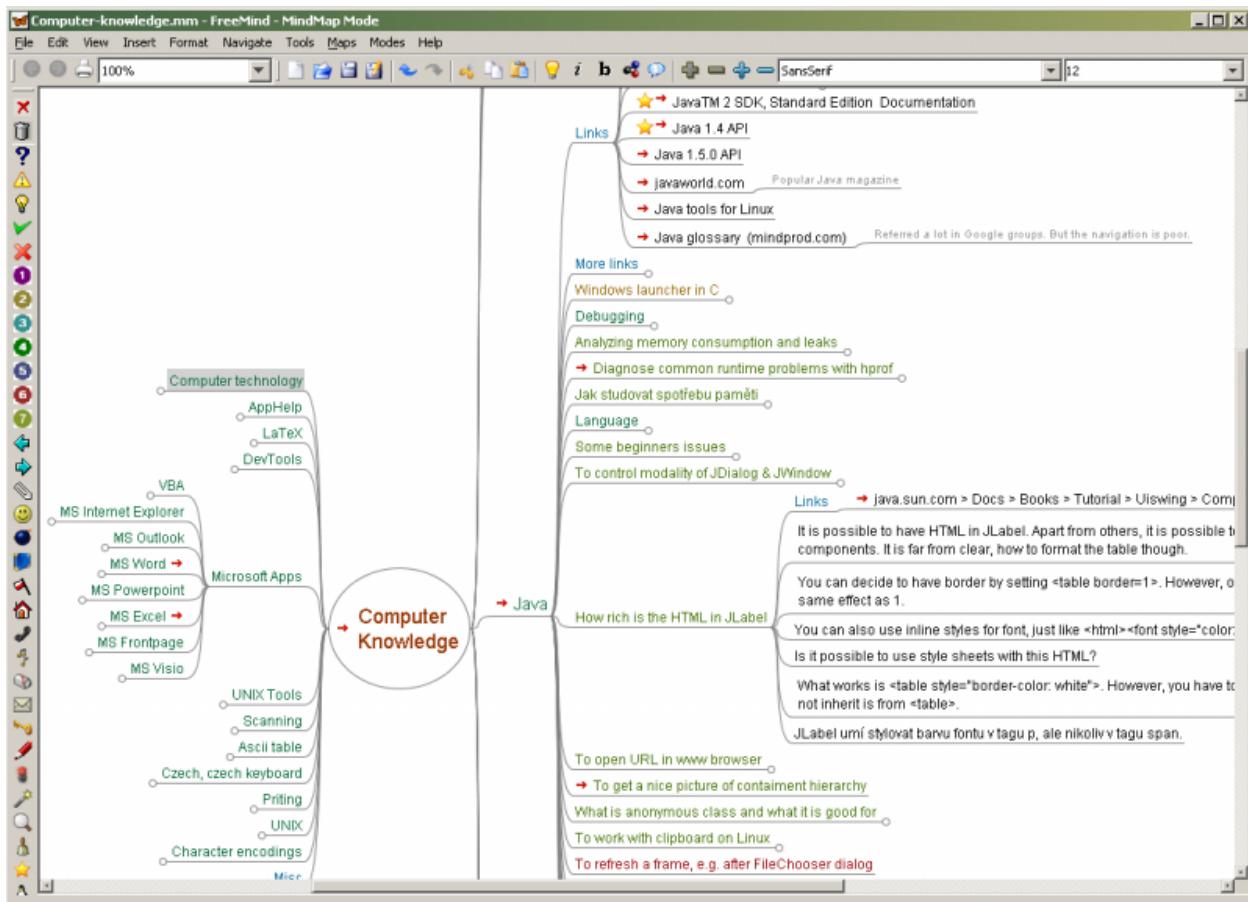
#### 2.0.17 AsciiFlow.com (ASCII drawing tool)

- Asciflow provides ascii based drawings that you can copy directly to textfiles and source codes. Visit the following link
  - <https://asciiflow.com/>



## 2.0.18 Freemind (opensource mindmap application)

- Freemind is open source java based desktop mindmap application. Can export files to several formats
  - Main Page - FreeMind<sup>43</sup>



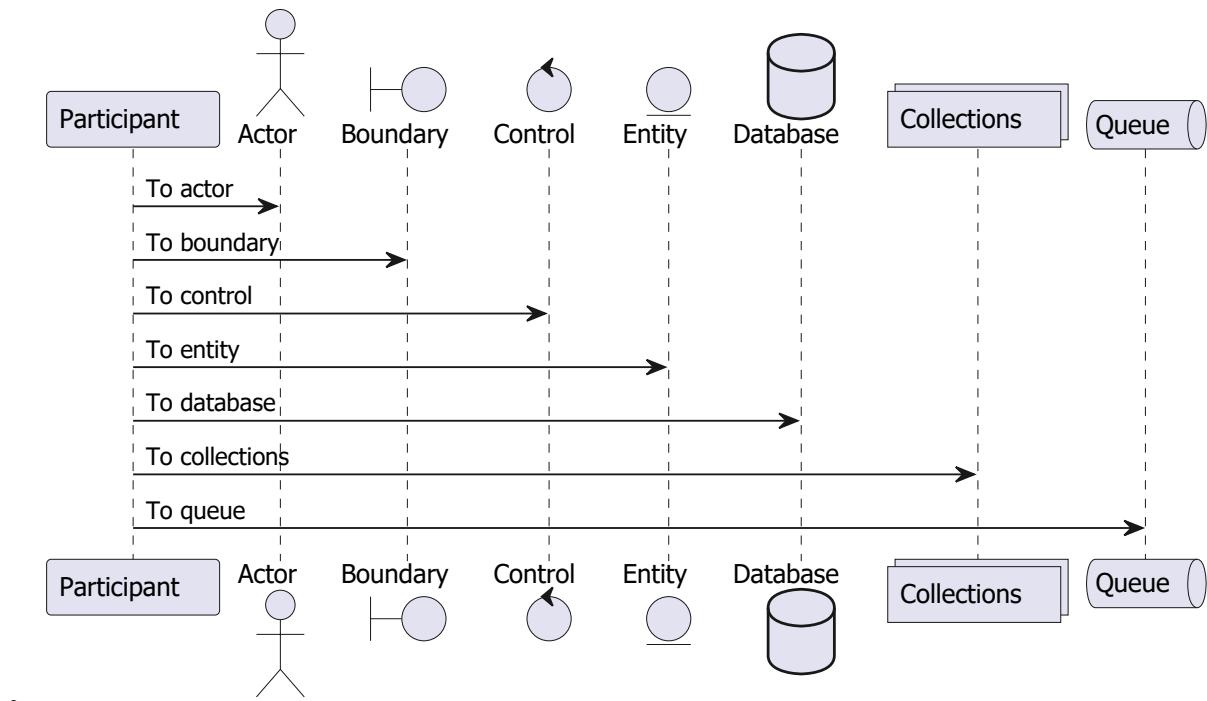
<sup>43</sup>[https://freemind.sourceforge.net/wiki/index.php/Main\\_Page](https://freemind.sourceforge.net/wiki/index.php/Main_Page)

### 2.0.18.1 Mockup Designers

- Mockflow
  - Signup - MockFlow<sup>44</sup>
- Wireflow
  - <https://wireflow.co/>

### 2.0.19 PlantUML (software designer)

- Text based designer for software engineers
  - <https://plantuml.com/>



- Also visit course notes that related to plantuml CE204 Object-Oriented Programming - RTEU C204 Object Oriented Programming Course Notes<sup>45</sup>

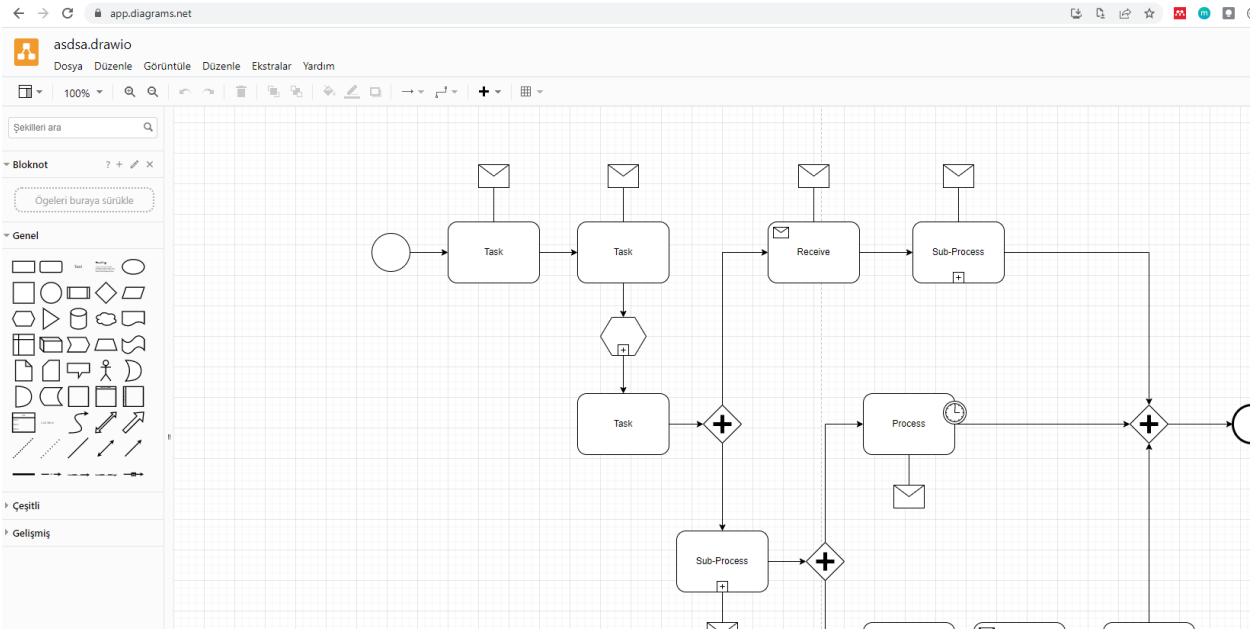
### 2.0.20 Drawio (drawing tool)

- Online and Offline Drawing Tool
  - <https://app.diagrams.net/>
- Offline Installer
  - Releases · jgraph/drawio-desktop · GitHub<sup>46</sup>

<sup>44</sup><https://mockflow.com/signup/>

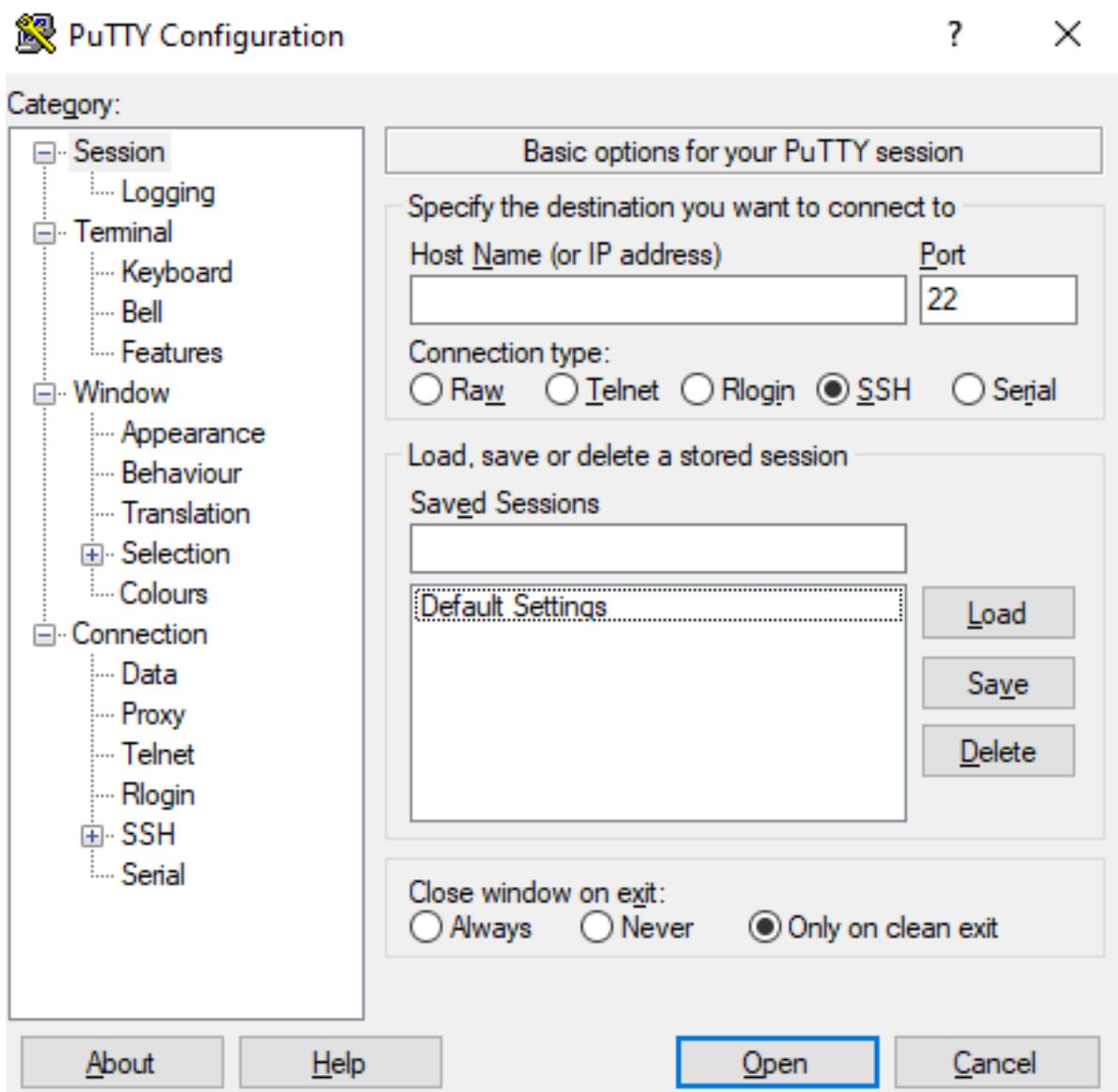
<sup>45</sup><https://ucoruh.github.io/ce204-object-oriented-programming/week-5/ce204-week-5/>

<sup>46</sup><https://github.com/jgraph/drawio-desktop/releases/>

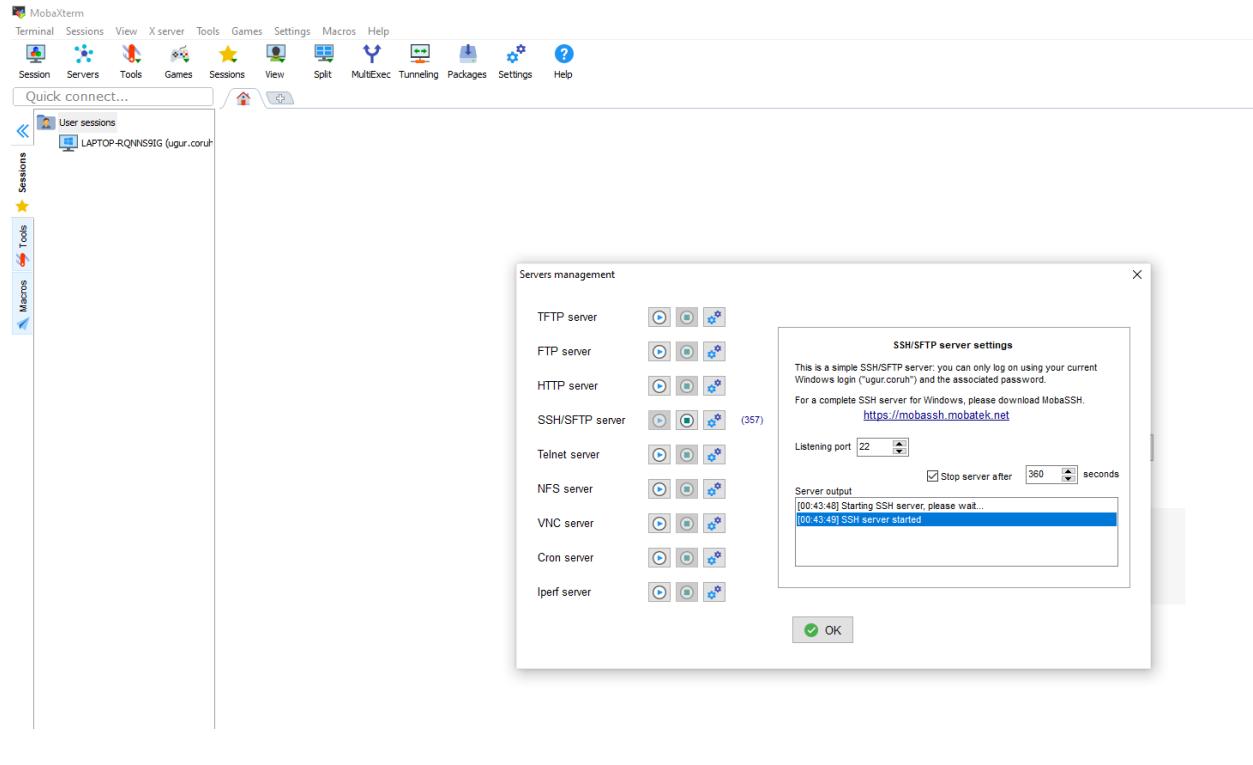


### 2.0.21 Putty (Remote Connection)

- Commonly use for SSH connection



- We can run a SSH server with MobaXterm and can connect to same computer with Putty.





## PuTTY Configuration

?

X

## Category:

- Session
  - ... Logging
- Terminal
  - ... Keyboard
  - ... Bell
  - ... Features
- Window
  - ... Appearance
  - ... Behaviour
  - ... Translation
  - + Selection
  - ... Colours
- Connection
  - ... Data
  - ... Proxy
  - ... Telnet
  - ... Rlogin
  - + SSH
  - ... Serial

## Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address)

Port

127.0.0.1

22

Connection type:

 Raw    Telnet    Rlogin    SSH    Serial

Load, save or delete a stored session

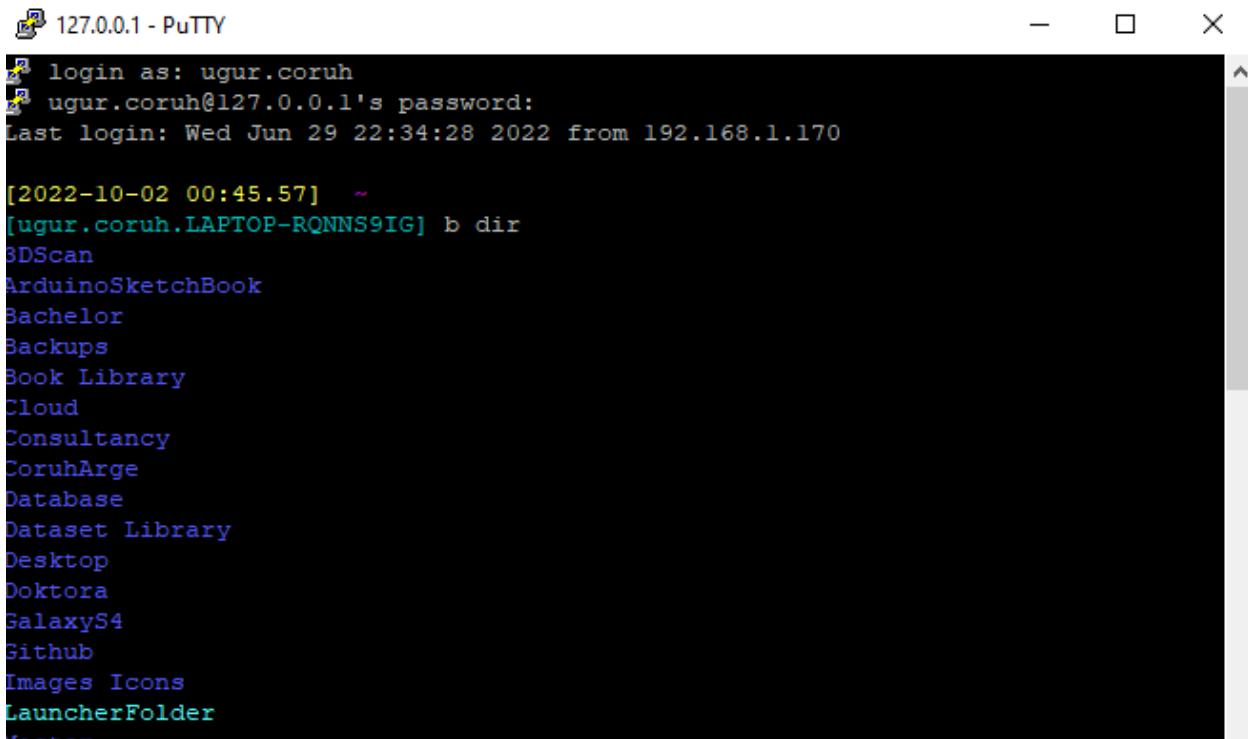
## Saved Sessions

## Default Settings

LoadSaveDelete

Close window on exit:

 Always    Never    Only on clean exitAboutHelpOpenCancel



A screenshot of a PuTTY terminal window titled "127.0.0.1 - PuTTY". The session log shows the user "ugur.coruh" logging in with their password and last login details. The user then runs the command "b dir" which lists the contents of the current directory. The directory contains many subfolders, including "3DScan", "ArduinoSketchBook", "Bachelor", "Backups", "Book Library", "Cloud", "Consultancy", "CoruhArge", "Database", "Dataset Library", "Desktop", "Doktora", "GalaxyS4", "Sithub", "Images Icons", and "LauncherFolder".

```
[ugur.coruh@127.0.0.1's password:
Last login: Wed Jun 29 22:34:28 2022 from 192.168.1.170

[2022-10-02 00:45:57] ~ [ugur.coruh.LAPTOP-RQNNNS9IG] b dir
3DScan
ArduinoSketchBook
Bachelor
Backups
Book Library
Cloud
Consultancy
CoruhArge
Database
Dataset Library
Desktop
Doktora
GalaxyS4
Sithub
Images Icons
LauncherFolder
```

---

## 2.1 Download file over SSH Protocol

- How to Download and Upload Files over SSH – TecAdmin<sup>47</sup>

Here are some useful examples for downloading files from the remote system over SSH protocol.

- This will connect to example.com server with user “username” and copy the /backup/file.zip file to local system directory /local/dir. To use theis command replace the values as per your environment.

```
scp username@example.com:/backup/file.zip /local/dir
```

- If the SSH is running on a non-standard port, You can specify the port using -P option with SCP command.

```
scp -P 2222 username@example.com:/backup/file.zip /local/dir
```

- If your remote server required a private key to connect server, You can use -i followed by a private key file path to connect your server using the SCP command. This can be helpful for AWS servers.

```
scp -i private_key.pem username@example.com:/backup/file.zip /local/dir
```

---

## 2.2 Upload file using SSH

You can also upload files to the remote server using SSH protocol using the SCP command. Use the following example command for uploading files to the SSH server.

```
scp file.zip username@example.com:/remote/dir
```

Similarity you can use -P switch to define port of the SSH server and -i to define private key for the user authentication.

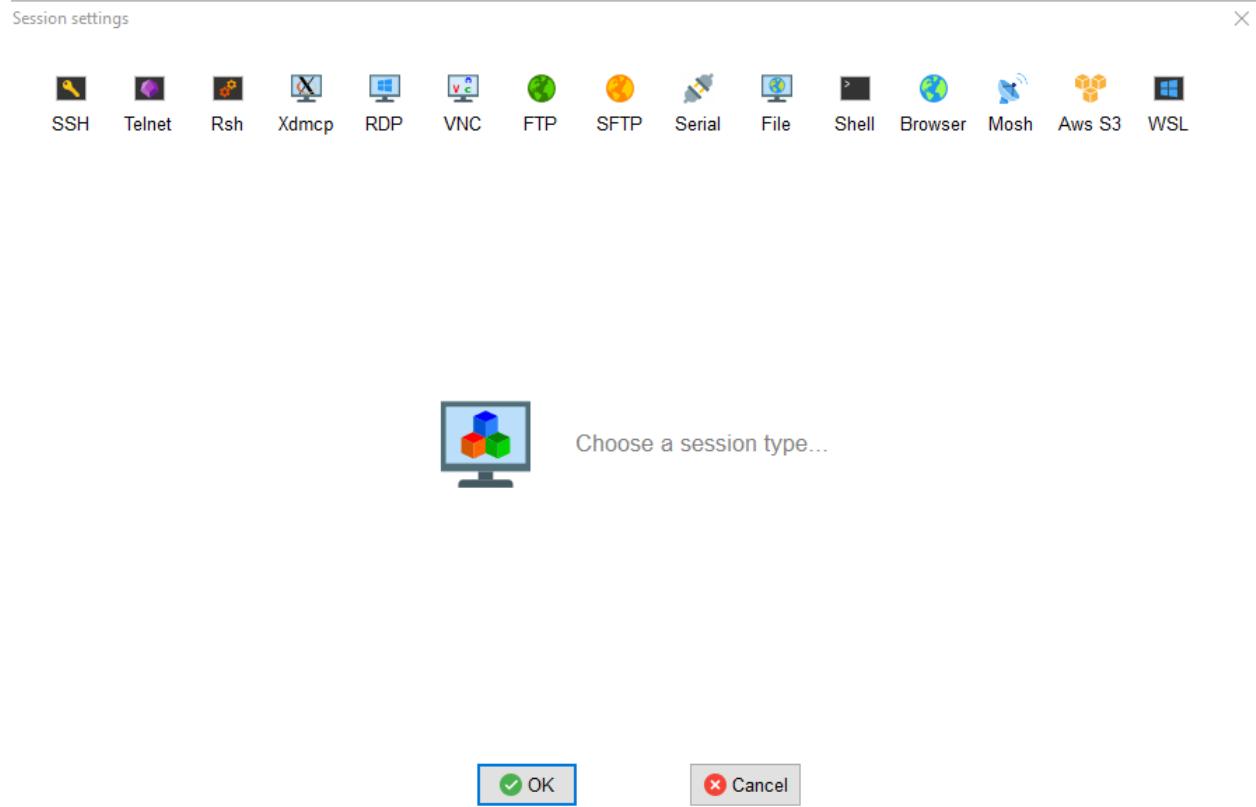
- 
- Also you can use SSH tunnels for remote code development

<sup>47</sup><https://tecatmin.net/download-file-using-ssh/>

- Developing on Remote Machines using SSH and Visual Studio Code<sup>48</sup>
  - Visual Studio Code Server<sup>49</sup>
- 
- 

### 2.2.1 MobaXterm (Remote Connection)

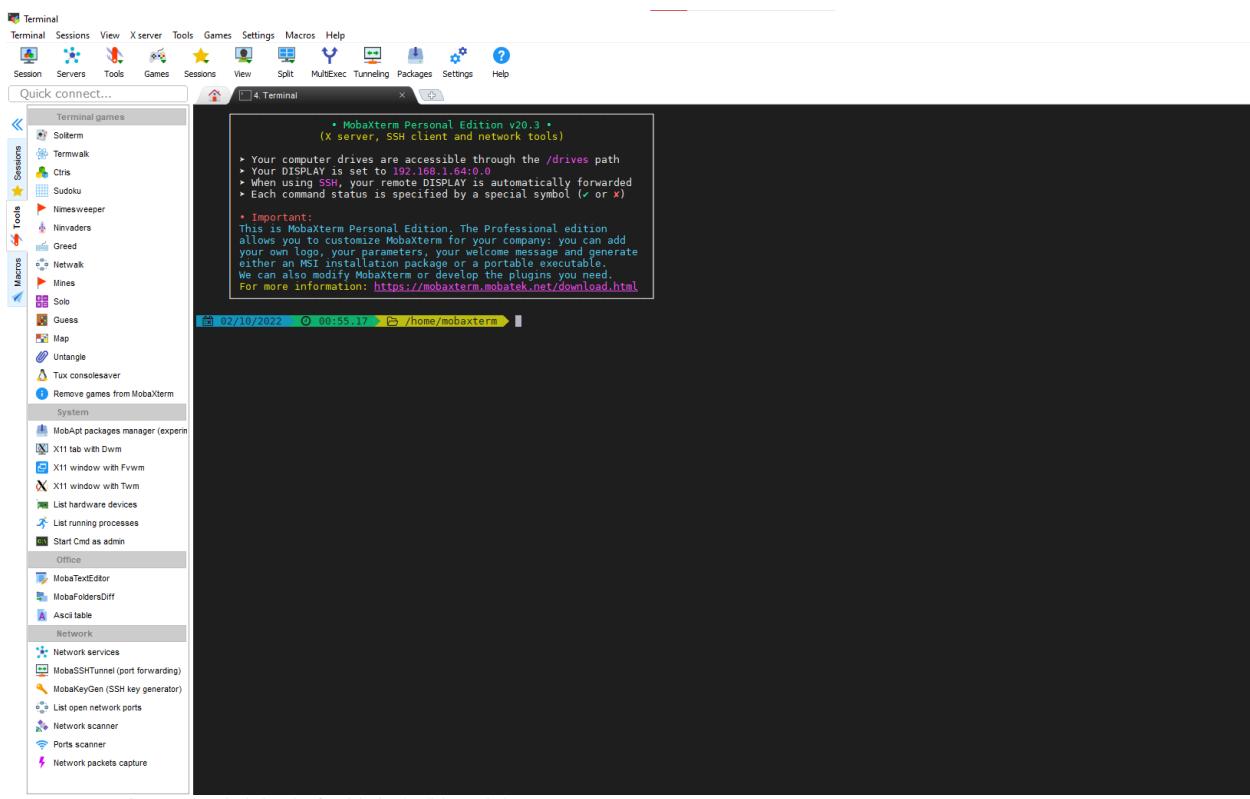
- Multip Purpose Remote Connection Toolkit



---

<sup>48</sup><https://code.visualstudio.com/docs/remote/ssh>

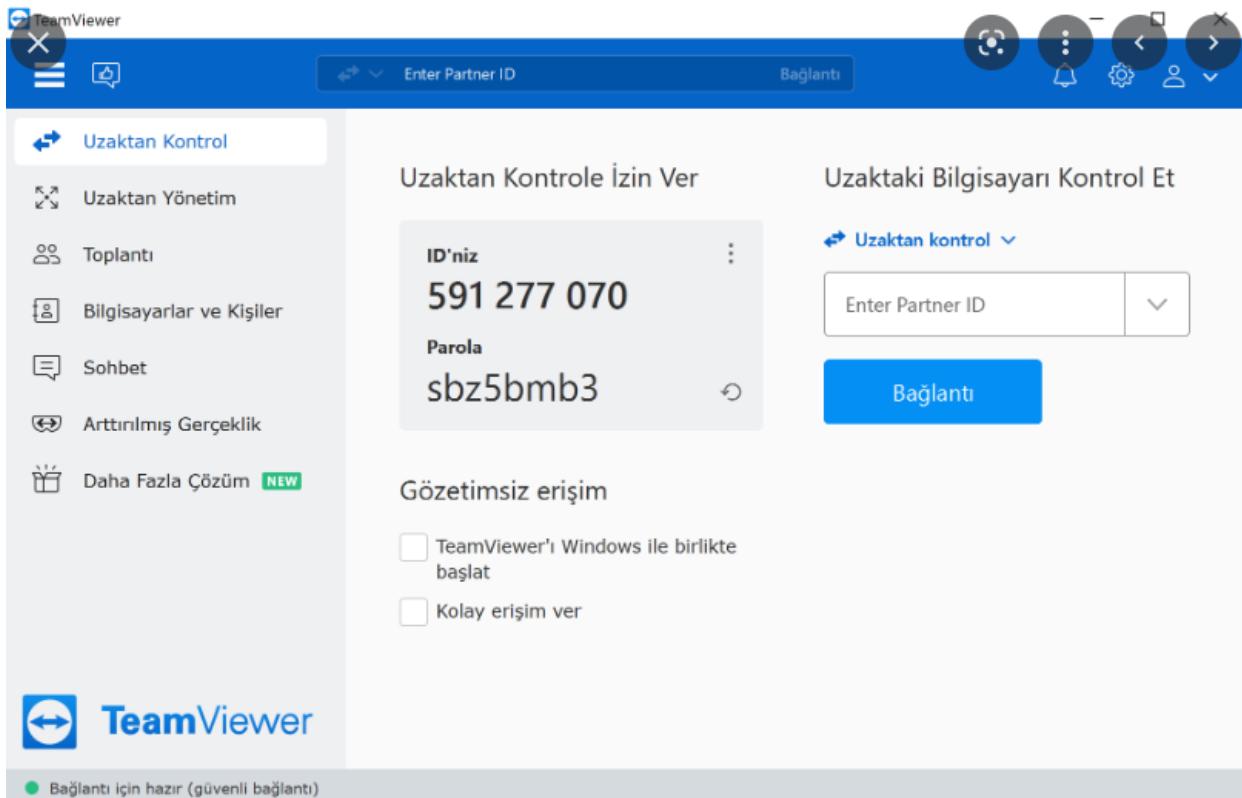
<sup>49</sup><https://code.visualstudio.com/docs/remote/vscode-server>



## 2.2.2 Teamviewer (Remote Connection)

- Remote connection tool

- TeamViewer – Uzaktan Destek, Uzaktan Erişim, Hizmet Masası, Çevrimiçi İşbirliği ve Toplantılar<sup>50</sup>

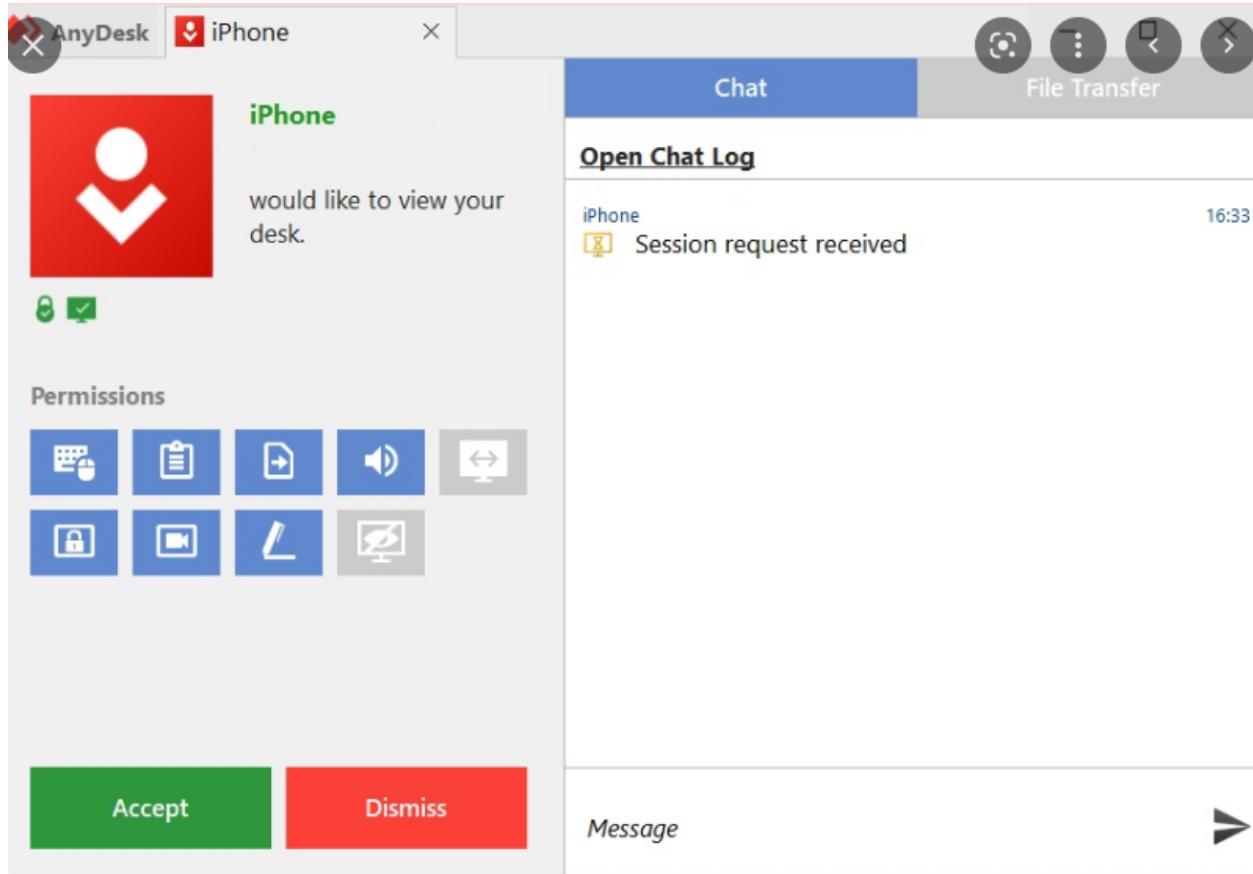


### 2.2.3 AnyDesk

- Remote connection tool
  - The Fast Remote Desktop Application – AnyDesk<sup>51</sup>

<sup>50</sup><https://www.teamviewer.com/tr/>

<sup>51</sup><https://anydesk.com/en>

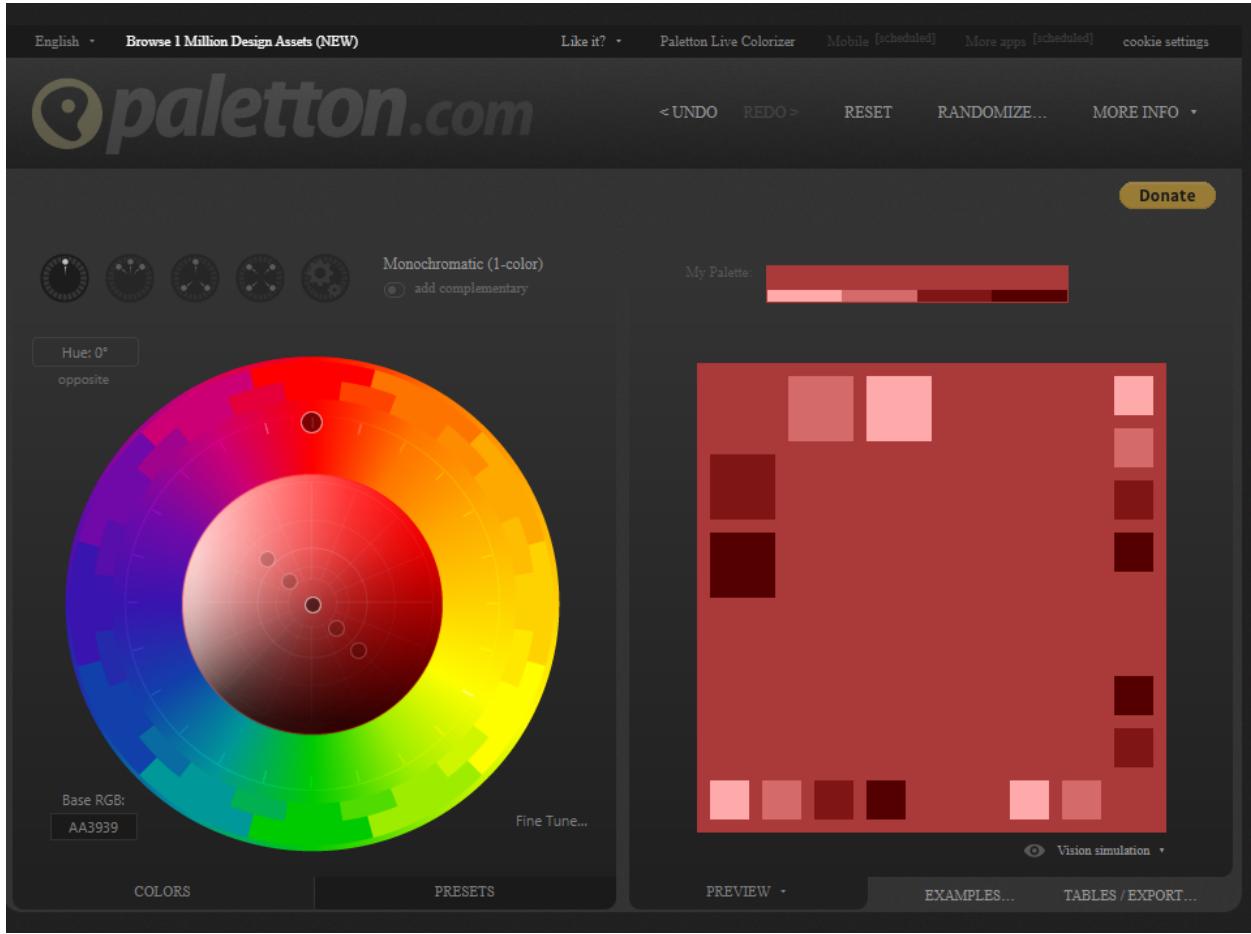


#### 2.2.4 Paletton.com and Colorhunt.co (Color Chooser)

- Generates color palettes and sample usages
  - Paletton - The Color Scheme Designer<sup>52</sup>
  - <https://colorhunt.co/>
  - Also check Colors Tutorial<sup>53</sup>

<sup>52</sup><https://paletton.com/#uid=1000u0kllllaFw0g0qFqFg0w0aF>

<sup>53</sup><https://www.w3schools.com/colors/>



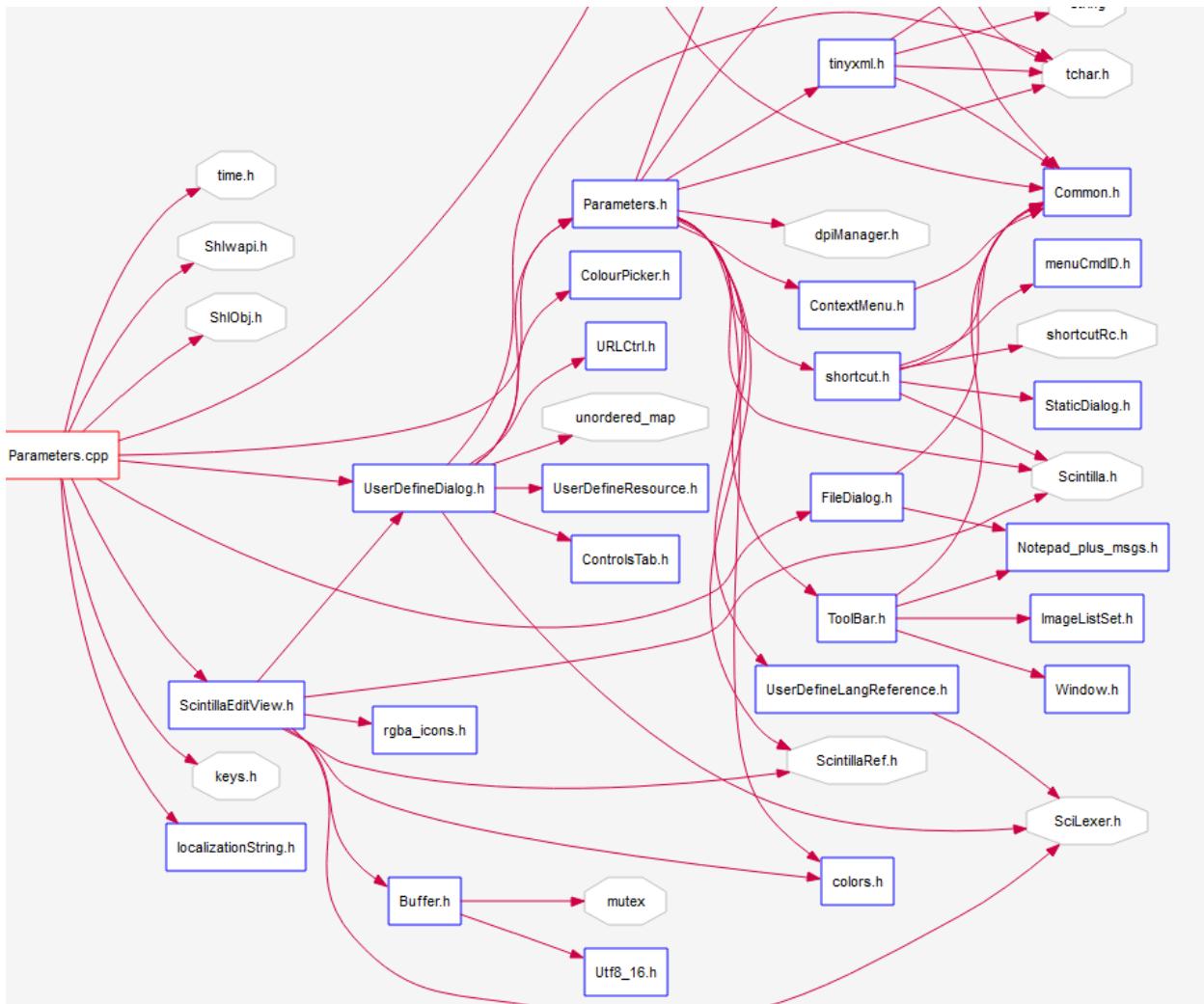


## 2.2.5 Understand (Static Code Analysis)

- <https://emenda.com/scitoools-understand/>

The screenshot displays a software interface for code analysis and visualization. The main window is divided into several panes:

- Left pane:** Shows a tree view of the project structure under "File System".
- Top-left pane:** Displays a large block of C++ code with syntax highlighting and annotations.
- Top-right pane:** Shows a detailed dependency graph where nodes represent functions and edges represent calls between them. A specific function, `checkForReentrantCall()`, is highlighted in yellow.
- Bottom-right pane:** A 3D bar chart titled "Code Volume Distribution" showing the relative size of different code metrics. The legend indicates: **Volume Metrics** (red), **Control Flow** (blue), **Control Data** (green), and **Control Function** (cyan).
- Bottom-left pane:** A table titled "File Details" listing various files with their last modified dates.
- Right side:** A vertical stack of three windows: "Dependency Graph - Summary", "Code Volume - Directory Structure", and "Code Volume - Directory Structure".



### 2.2.6 JD Project (Java Decompiler)

- Java Decomplier for Jar and Class Files, If code is not obfuscated it recover source code from compiled files. Just drag and drop files to GUI
    - <http://java-decompiler.github.io/>
    - You can use it standalone app or with eclipse

The screenshot shows the JD-GUI interface. On the left, there's a tree view of the jar file contents, including the META-INF directory with versions/9, MANIFEST.MF, and org.jd.core.test. Inside org.jd.core.test, there's a Basic class with several fields: e (double), index (int), int78 (int), int\_max (int), long12 (long), long34 (long), and long\_min (long). On the right, the decompiled code for the Basic class is shown:

```

public void operator(int paramInt) {
    System.out.println("start");

    int i = 50 / (25 + (paramInt = 789));

    i = paramInt += 100;

    this.int78 = paramInt = this.int78 += 456
    this.int78 = 50 / (25 + (this.int78 = 789))

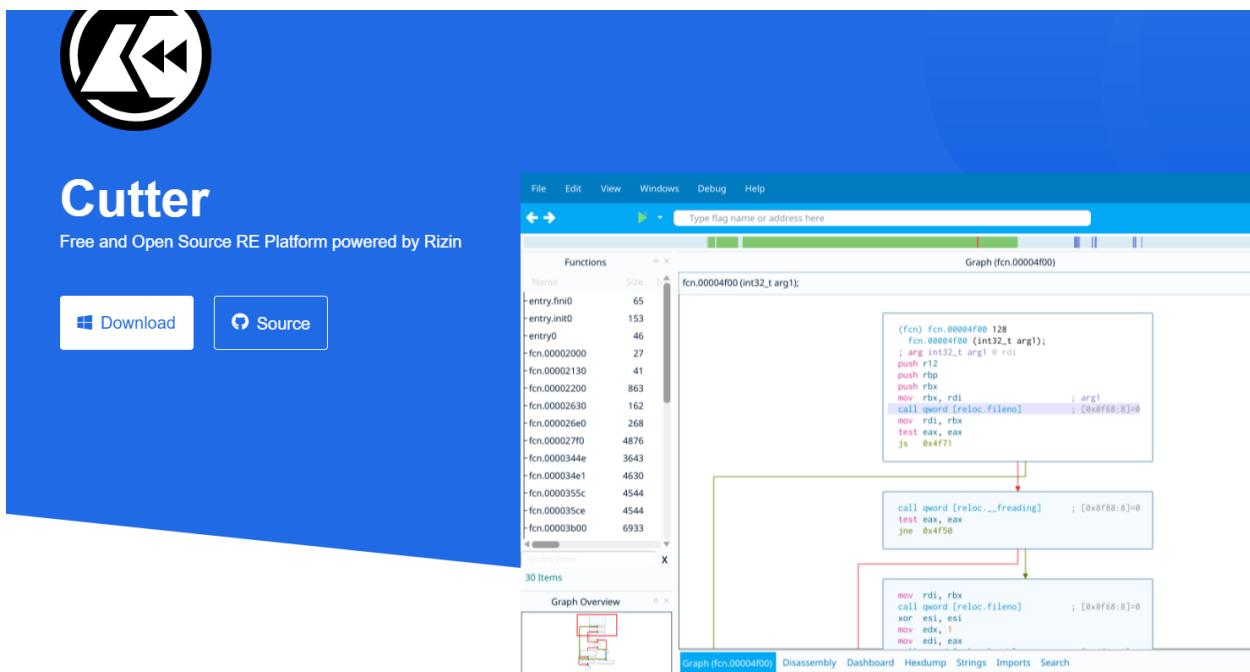
    paramInt = ++this.int78;
    paramInt = this.int78++;
    paramInt *= 10;
}

```

A search bar at the bottom left says "Find: int78". Below the search bar are buttons for "Next" and "Previous".

## 2.2.7 Cutter (Multi-Platform Reverse Engineering Tool)

- Cutter's goal is to be an advanced FREE and open-source reverse-engineering platform while keeping the user experience at mind. Cutter is created by reverse engineers for reverse engineers.
- <https://cutter.re/>



## 2.2.8 IDA Pro / Freeware (Native Reverse Engineering Tool)

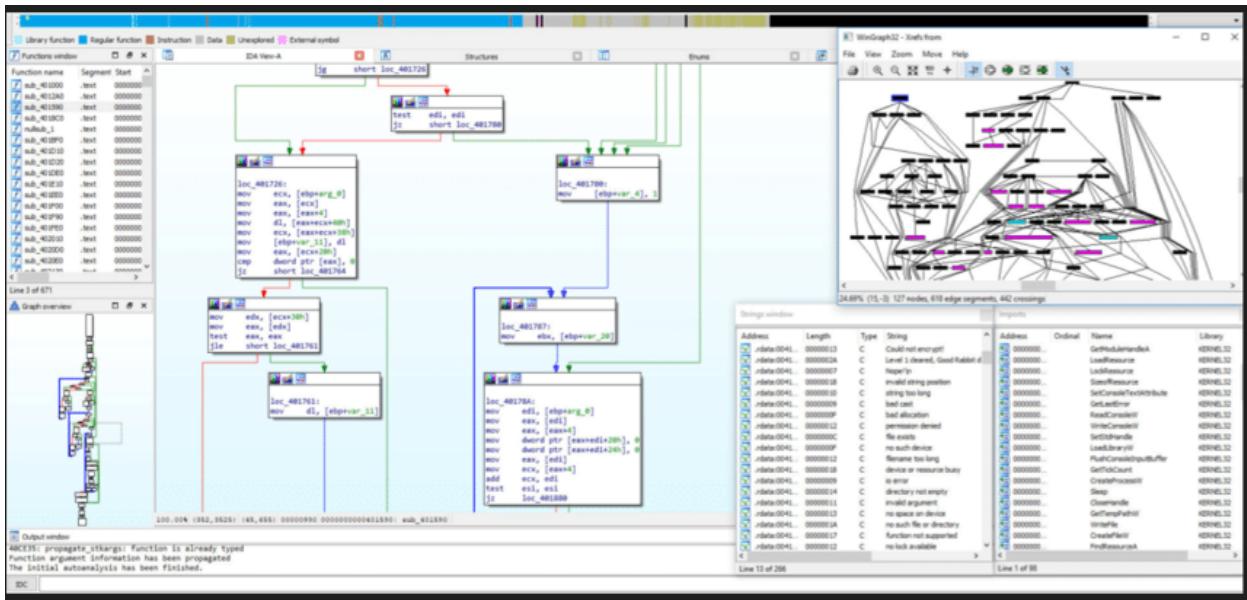
- IDA Pro as a disassembler is capable of creating maps of their execution to show the binary instructions that are actually executed by the processor in a symbolic representation (assembly language). Advanced techniques have been implemented into IDA Pro so that it can generate assembly language source code from machine-executable code and make this complex code more human-readable.

## 2.2.9 IDA Pro / Freeware (Native Reverse Engineering Tool)

- Hex Rays - State-of-the-art binary code analysis solutions<sup>54</sup>

<sup>54</sup> <https://hex-rays.com/ida-pro/>

## 2.2.10 IDA Pro / Freeware (Native Reverse Engineering Tool)



## 2.2.11 Code Visualization (Python, C, C++, Java)

- This coding tutor tool helps you learn Python, JavaScript, C, C++, and Java by visualizing code execution.
  - <https://pythontutor.com/>

Python 3.6

```

1 def listSum(numbers):
2     if not numbers:
3         return 0
4     else:
5         (f, rest) = numbers
6         return f + listSum(rest)
7
8 myList = (1, (2, (3, None)))
9 total = listSum(myList)

```

[Edit this code](#)

line that just executed  
next line to execute

< Prev Next >  
Step 11 of 22

[Visualized using Python Tutor](#)  
[Customize visualization](#)

Frames	Objects
Global frame	function listSum(numbers)
listSum	tuple 0 1
myList	tuple 0 1
listSum	tuple 0 2
numbers	tuple 0 3 None
f	1
rest	None
listSum	tuple 2
numbers	tuple 2
f	2
rest	None

## 2.2.12 Assembly of C Code

- Multilanguage supported. Convert source code to assembly codes
  - <https://godbolt.org/>

The screenshot shows the Compiler Explorer interface. On the left, the C++ source code is displayed:

```

1 #include <stdio.h>
2 int addNumbers(int a, int b); // function prototype
3
4 int main()
5 {
6     int n1,n2,sum;
7
8     printf("Enters two numbers: ");
9     scanf("%d",&n1,&n2);
10
11     sum = addNumbers(n1, n2); // function call
12     printf("sum = %d",sum);
13
14     return 0;
15 }
16
17 int addNumbers(int a, int b) // function definition ...
18 {
19     int result;
20     result = a+b;
21     return result; // return statement
22 }

```

On the right, the assembly output for x86-64 gcc 12.2 is shown:

```

1 .LC0: .string "Enters two numbers: "
2 .LC1: .string "%d %d"
3 .LC2: .string "sum = %d"
4
5 main:
6     push rbp
7     mov rbp, rsp
8     sub rsp, 16
9     mov edi, OFFSET FLAT:_LC0
10    mov eax, 0
11    call printf
12    lea rdx, [rbp-12]
13    lea rax, [rbp-8]
14    mov rsi, rax
15    mov edi, OFFSET FLAT:_LC1
16    mov eax, 0
17    call _isoc99_scanf
18    mov edx, DWORD PTR [rbp-12]
19    mov eax, DWORD PTR [rbp-8]
20    mov esi, edx
21    mov edi, eax
22    mov eax, 0
23    call addNumbers(int, int)
24    mov eax, DWORD PTR [rbp-4], eax
25    mov esi, eax
26    mov edi, OFFSET FLAT:_LC2
27    mov eax, 0
28    call printf
29    leave
30    mov eax, 0
31    ret
32
33 addNumbers(int, int):
34

```

### 2.2.13 Mobile Device Screen Sharing for Demo

- Show USB or Wifi connected Mobile Device Screen on PC and Provide Controls
  - GitHub - Genymobile/scrcpy: Display and control your Android device<sup>55</sup>
  - Run scrcpy for single mobile phone.
  - Open Source Project - Scrcpy now works wirelessly<sup>56</sup>

---

### 2.2.14 Travis-CI

- Travis-CI is a continues integration platform
  - Travis-CI free option removed for this reason, its not in our scope.
  - It uses Travis.yml files for actions.
- 

### 2.2.15 AppVeyor

- Another CI platform it has free option for public builds.
  - <https://www.appveyor.com>
  - GitHub - Kimserey/hello-world-nuget<sup>57</sup>
  - hello-world-nuget/appveyor.yml at master · Kimserey/hello-world-nuget · GitHub<sup>58</sup>

---

<sup>55</sup><https://github.com/Genymobile/scrcpy>

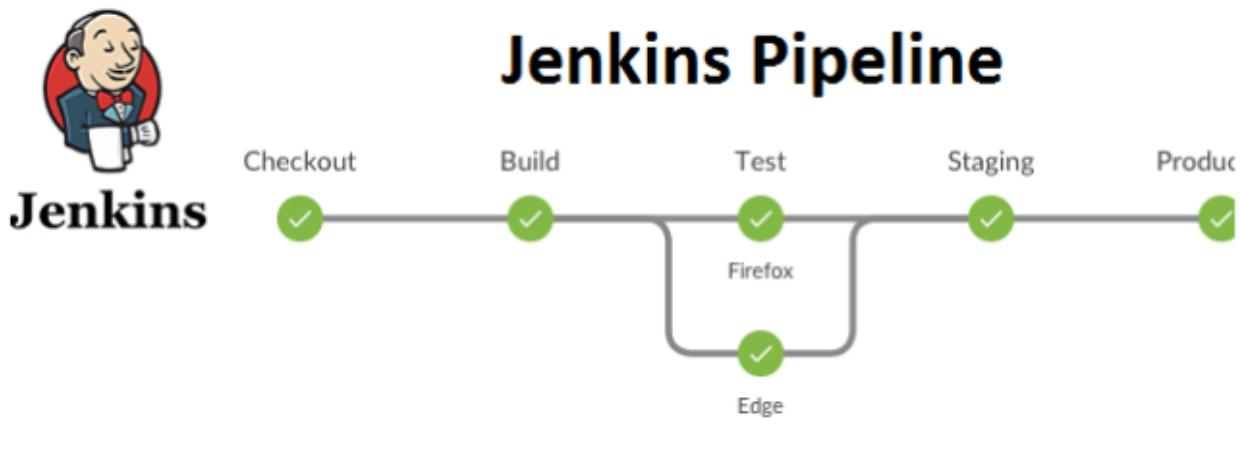
<sup>56</sup><https://www.genymotion.com/blog/open-source-project-scrcpy-now-works-wirelessly/>

<sup>57</sup><https://github.com/Kimserey/hello-world-nuget>

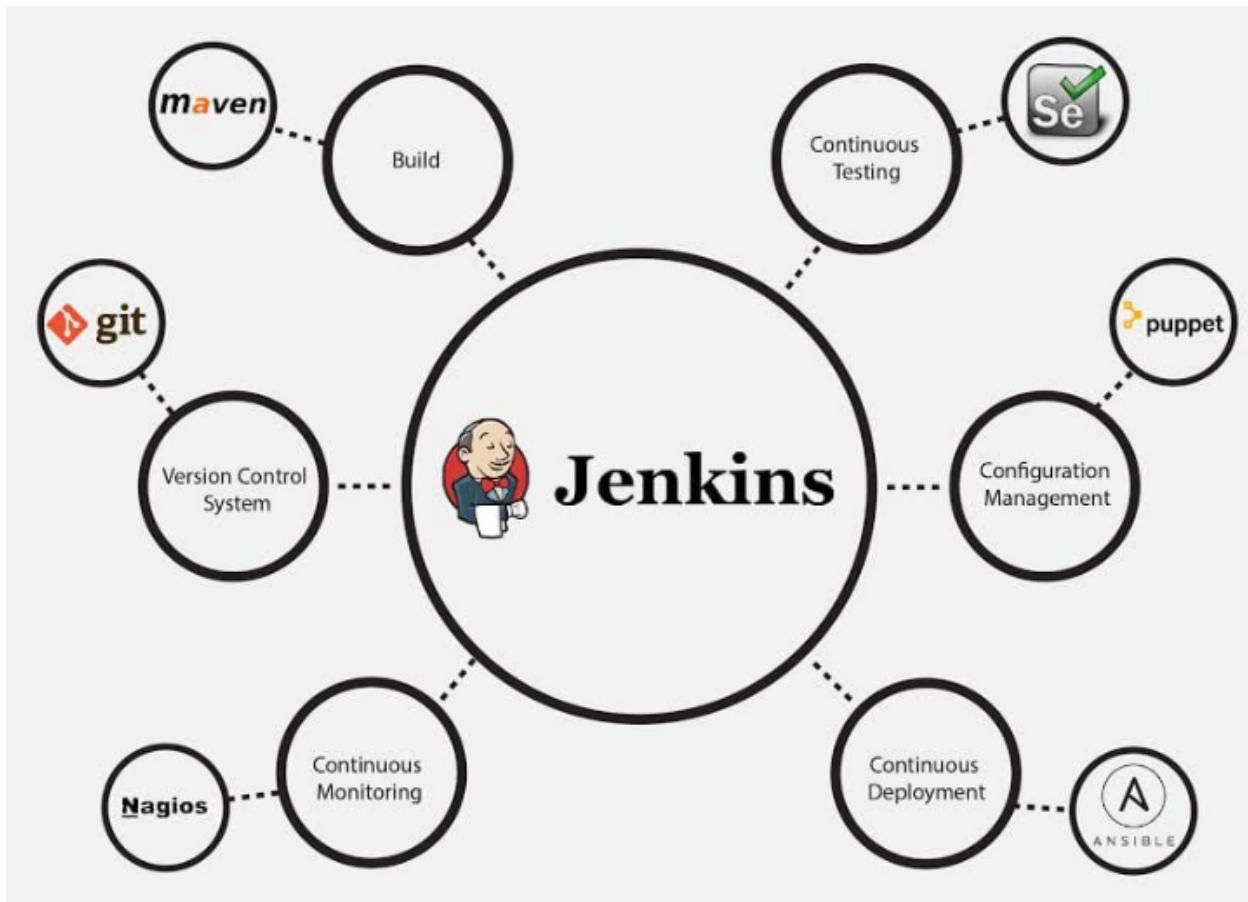
<sup>58</sup><https://github.com/Kimserey/hello-world-nuget/blob/master/appveyor.yml>

### 2.2.16 Jenkins

- Self-hosted solution for CI operations, Has integration with Github and several platforms.
  - <https://www.jenkins.io/>
  - <https://www.jenkins.io/doc/pipeline/tour/hello-world/>



### 2.2.17 Jenkins



## 2.2.18 Jenkins

- <https://www.jenkins.io/solutions/github/>

 **Configure Global Security**

Enable security

TCP port for JNLP slave agents  Fixed :   Random  Disable

Disable remember me

Access Control

**Security Realm**

Delegate to servlet container  Github Authentication Plugin

**Global GitHub OAuth Settings**

GitHub Web URI	<a href="https://github.com">https://github.com</a>
GitHub API URI	<a href="https://api.github.com">https://api.github.com</a>
Client ID	<input type="text"/>
Client Secret	<input type="text"/>
OAuth Scope(s)	read:org,user:email

Jenkins' own user database  LDAP  Unix user/group database

**Authorization**

Anyone can do anything  Github Committer Authorization Strategy

**Github Authorization Settings**

Admin User Names	tyler, kohsuke
Participant in Organization	<input type="text"/>
Use Github repository permissions	<input type="checkbox"/>
Grant READ permissions to all Authenticated Users	<input type="checkbox"/>
Grant CREATE Job permissions to all Authenticated Users	<input type="checkbox"/>

## 2.2.19 Vagrant

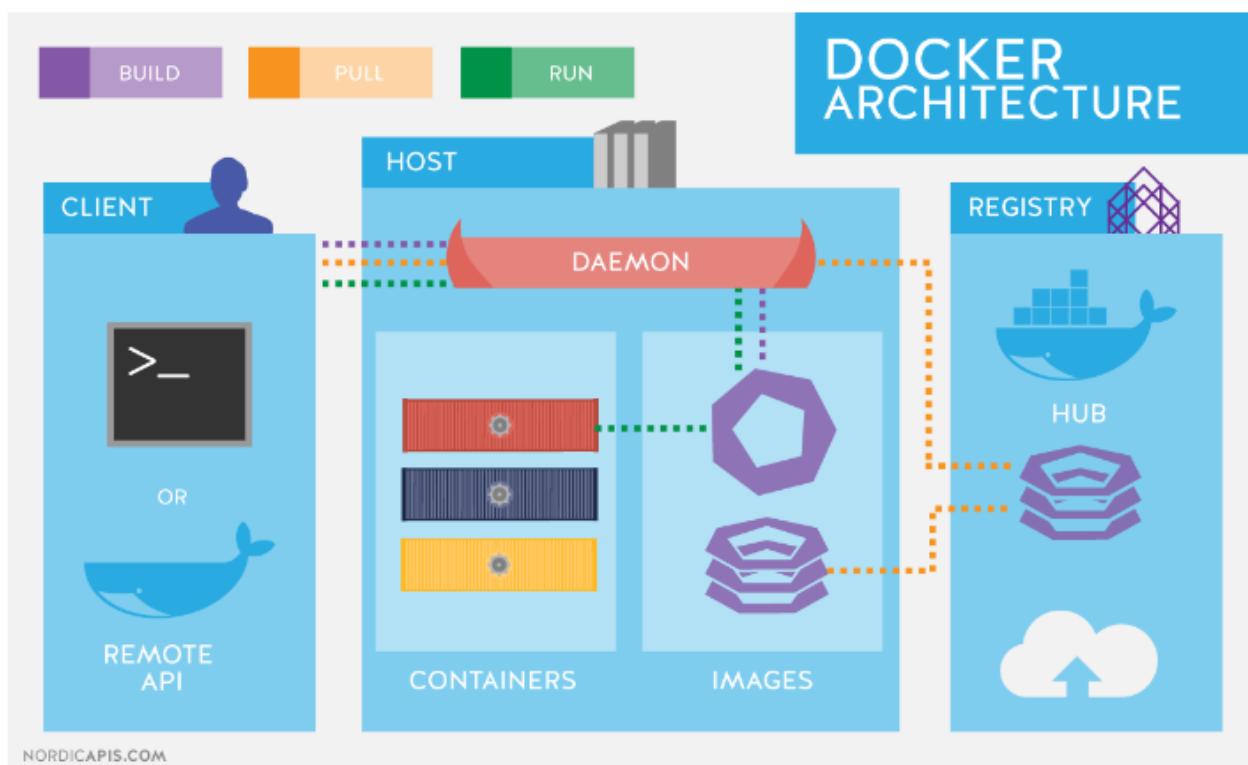
- Vagrant is a tool for building and managing virtual machine environments in a single workflow. With an easy-to-use workflow and focus on automation, Vagrant lowers development environment setup time, increases production parity, and makes the “works on my machine” excuse a relic of the past.

- <https://www.vagrantup.com/>
  - Setup Development Environment with Vagrant
    - Setting Up Development Environment Using Vagrant - Edureka<sup>59</sup>
    - [development-environment/Vagrantfile at master · gantsign/development-environment](https://github.com/gantsign/development-environment) GitHub<sup>60</sup>
- 

### 2.2.20 Docker / Docker Compose / Kubernetes (1)

- Docker takes away repetitive, mundane configuration tasks and is used throughout the development lifecycle for fast, easy and portable application development – desktop and cloud.
    - [https://www.youtube.com/watch?v=nBwJm0onzeo&ab\\_channel=GaryExplains](https://www.youtube.com/watch?v=nBwJm0onzeo&ab_channel=GaryExplains) Dockerfile
    - <https://devopedia.org/docker>
  - DockerHub
  - Docker Compose Yaml
  - Dockerrun.aws.json (AWS)
- 

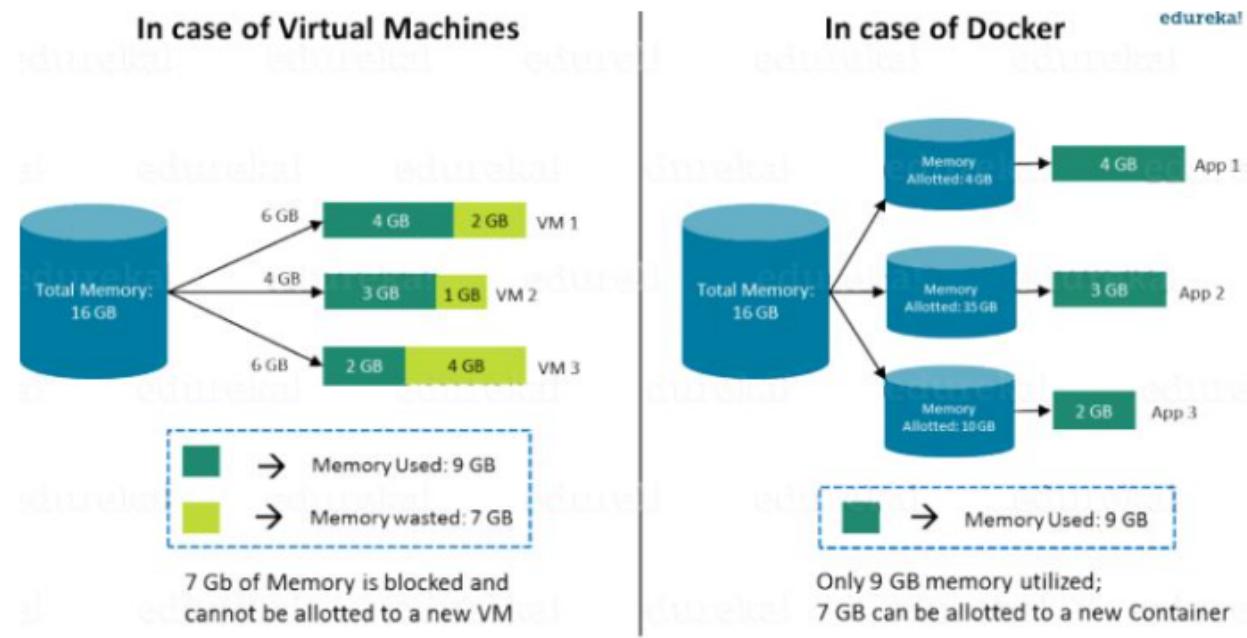
### 2.2.21 Docker / Docker Compose / Kubernetes (2)



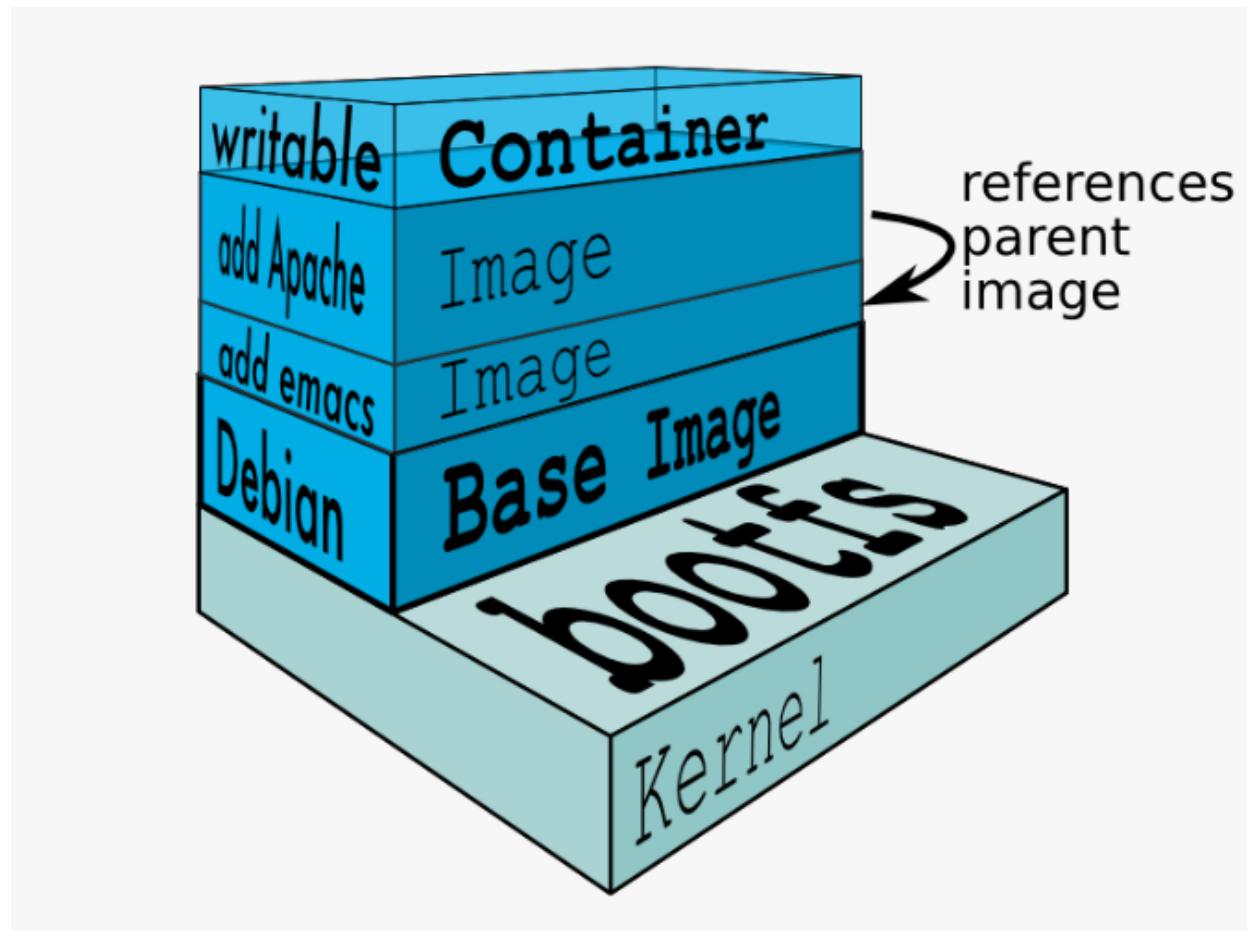
<sup>59</sup><https://www.edureka.co/blog/development-environment-using-vagrant/>

<sup>60</sup><https://github.com/gantsign/development-environment/blob/master/Vagrantfile>

## 2.2.22 Docker / Docker Compose / Kubernetes (3)



## 2.2.23 Docker / Docker Compose / Kubernetes (4)



---

2.2.24 Docker / Docker Compose / Kubernetes (5)

```
FROM node:9.3.0-alpine

RUN npm install -g @angular/cli@1.5.5 \
    && mkdir -p /usr/src/pintail-whoami

WORKDIR /usr/src/pintail-whoami

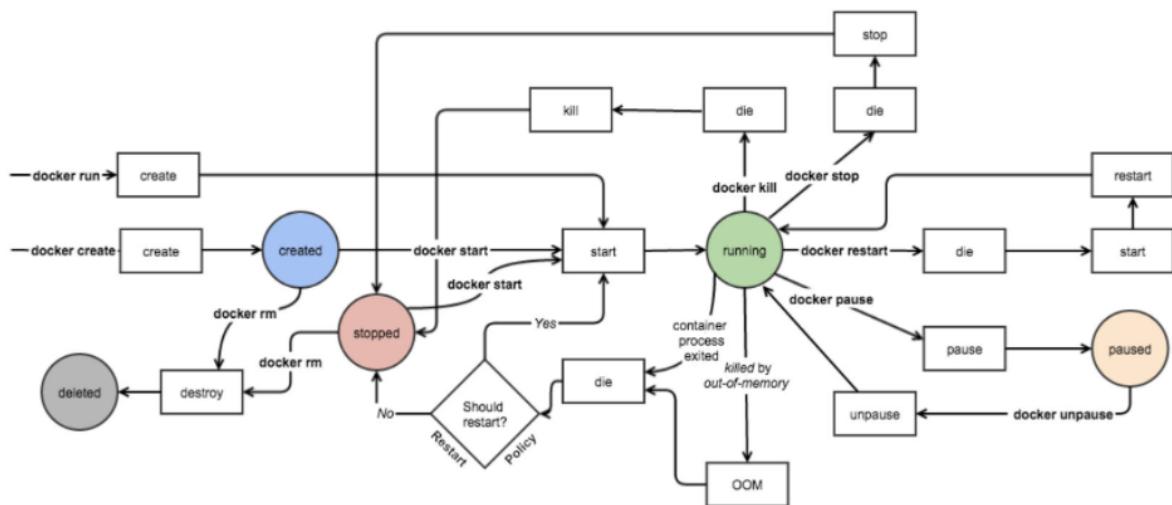
ADD . /usr/src/pintail-whoami

RUN npm install && ng build

EXPOSE 80

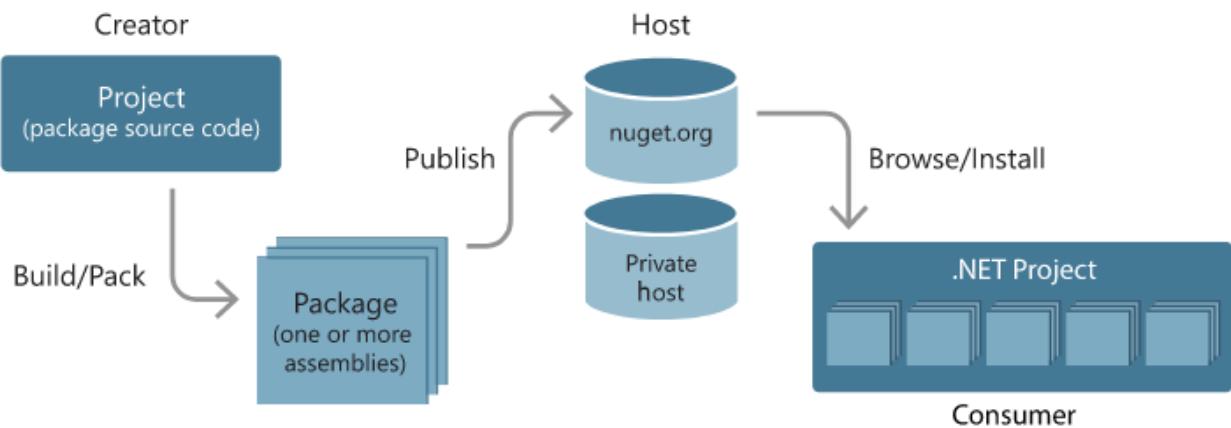
CMD node server.js $HOSTNAME
```

## 2.2.25 Docker / Docker Compose / Kubernetes (6)



## 2.2.26 Nuget Packages (1)

- <https://www.nuget.org/packages>
- What is NuGet and what does it do? | Microsoft Learn<sup>61</sup>

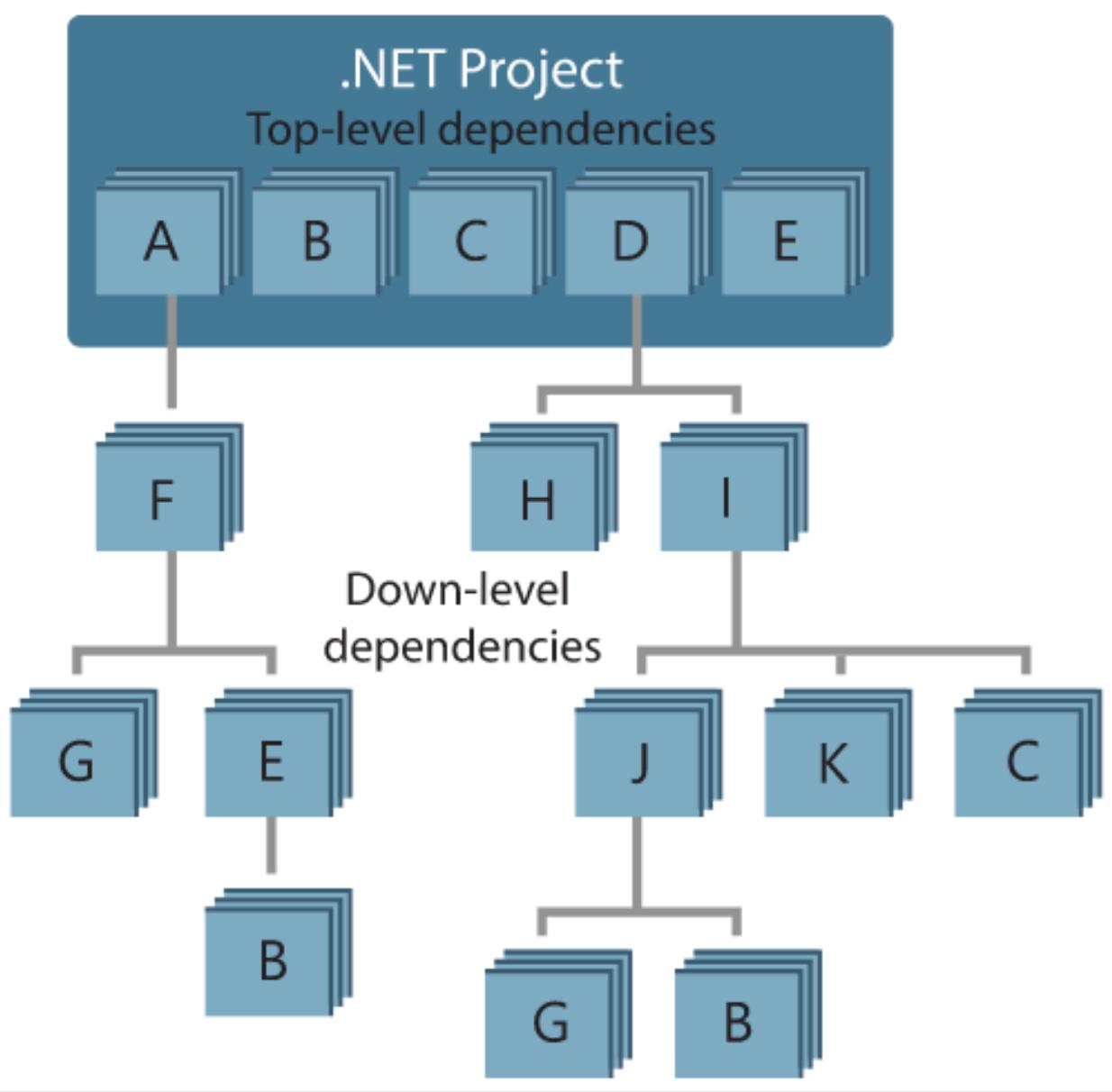


<sup>61</sup><https://learn.microsoft.com/en-us/nuget/what-is-nuget>

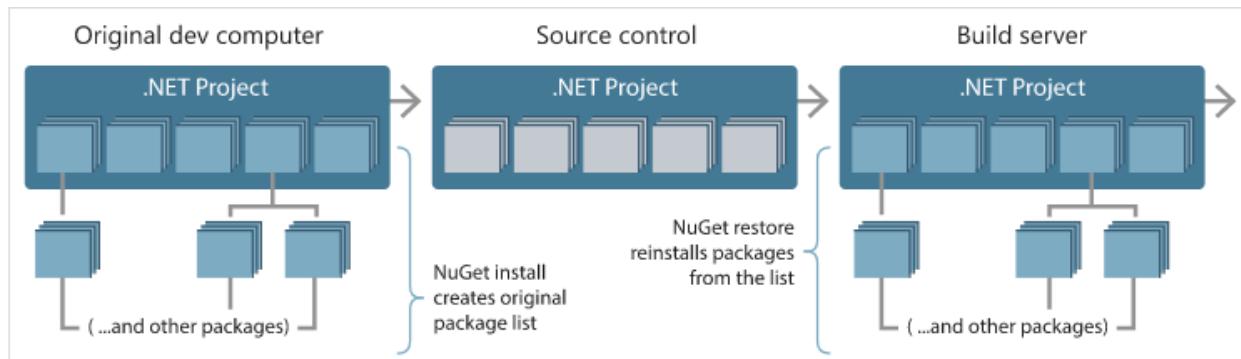
## 2.2.27 NuGet Tools (2)

Tool	Platforms	Applicable Scenarios	Description
<a href="#">dotnet CLI</a>	All	Creation, Consumption	CLI tool for .NET Core and .NET Standard libraries, and for SDK-style projects that target .NET Framework (see <a href="#">SDK attribute</a> ). Provides certain NuGet CLI capabilities directly within the .NET Core tool chain. As with the <code>nuget.exe</code> CLI, the dotnet CLI does not interact with Visual Studio projects.
<a href="#">nuget.exe CLI</a>	All	Creation, Consumption	CLI tool for .NET Framework libraries and non-SDK-style projects that target .NET Standard libraries. Provides all NuGet capabilities, with some commands applying specifically to package creators, some applying only to consumers, and others applying to both. For example, package creators use the <code>nuget pack</code> command to create a package from various assemblies and related files, package consumers use <code>nuget install</code> to include packages in a project folder, and everyone uses <code>nuget config</code> to set NuGet configuration variables. As a platform-agnostic tool, the NuGet CLI does not interact with Visual Studio projects.
<a href="#">Package Manager Console</a>	Visual Studio on Windows	Consumption	Provides <a href="#">PowerShell commands</a> for installing and managing packages in Visual Studio projects.
<a href="#">Package Manager UI</a>	Visual Studio on Windows	Consumption	Provides an easy-to-use UI for installing and managing packages in Visual Studio projects.
<a href="#">Manage NuGet UI</a>	Visual Studio for Mac	Consumption	Provides an easy-to-use UI for installing and managing packages in Visual Studio for Mac projects.
<a href="#">MSBuild</a>	Windows	Creation, Consumption	Provides the ability to create packages and restore packages used in a project directly through the MSBuild tool chain.

## 2.3 Managing dependencies (3)

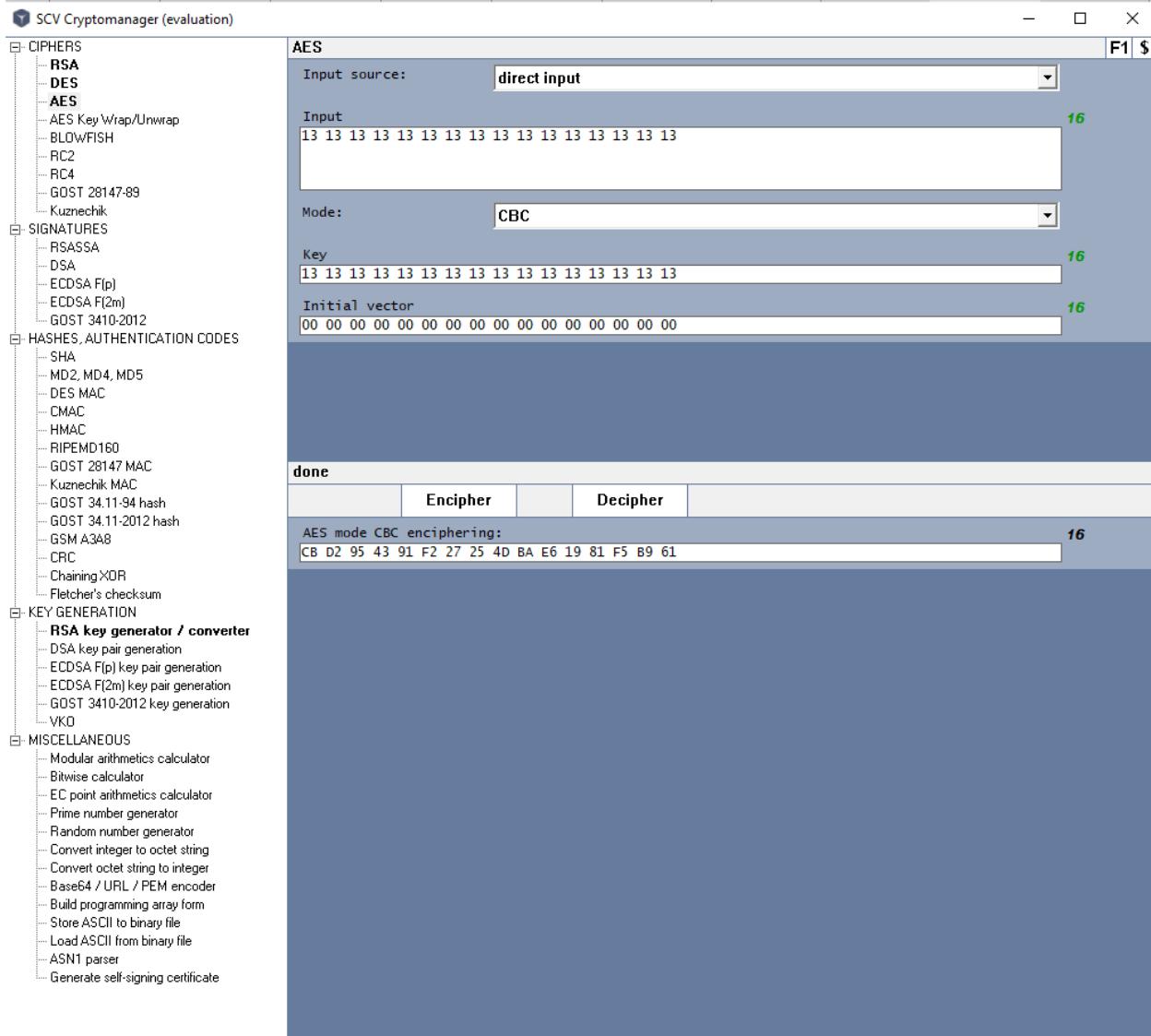


## 2.4 Tracking references and restoring packages (4)



#### 2.4.1 SCV Cryptomanager

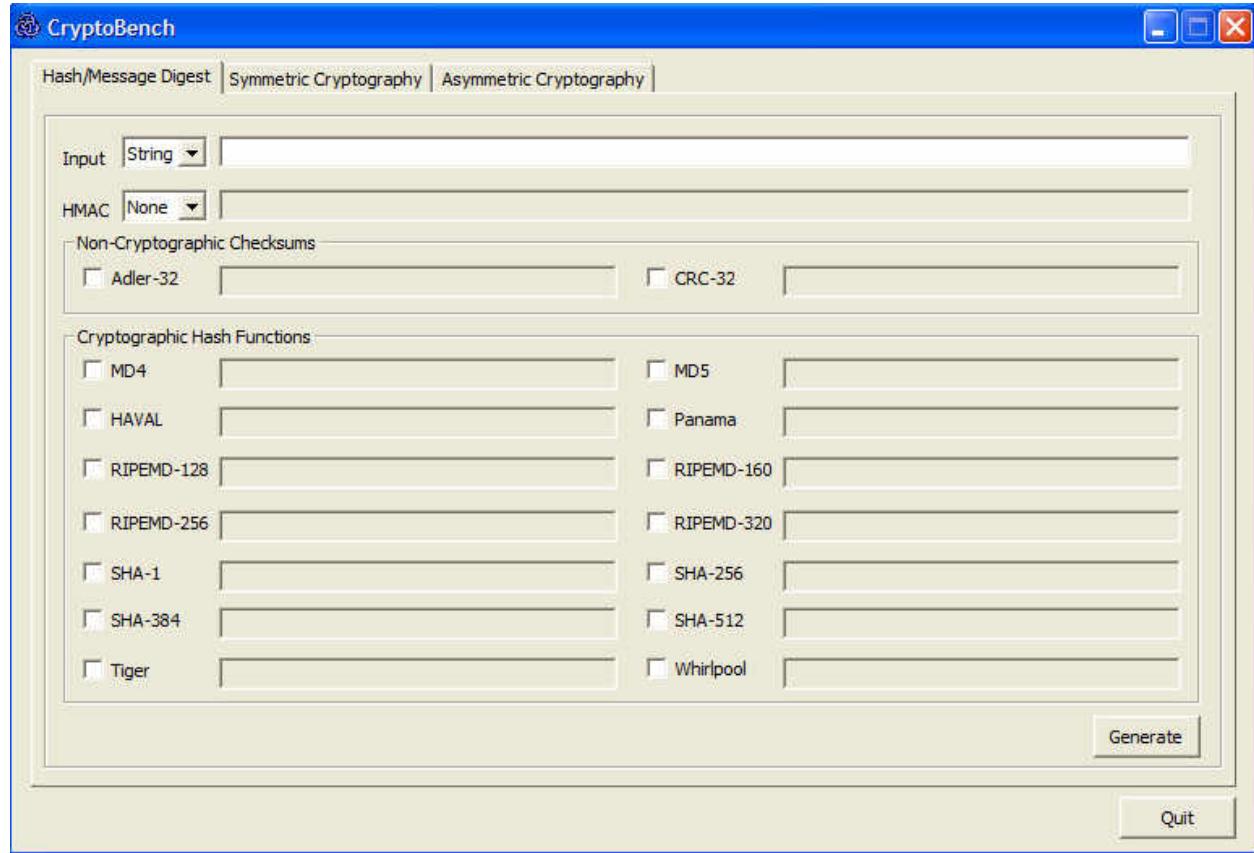
- SCV Crypto Manager has several tools for cryptographic operations.
  - <https://cryptomanager.com/download.php>



#### 2.4.2 Addario CryptoBench

- CryptoBench can be used for hash and symmetric asymmetric encryption-decryption operations.
  - CryptoBench Download Page<sup>62</sup>
  - <http://www.addario.org/files/CryptoBench%20v1.0.1.zip>

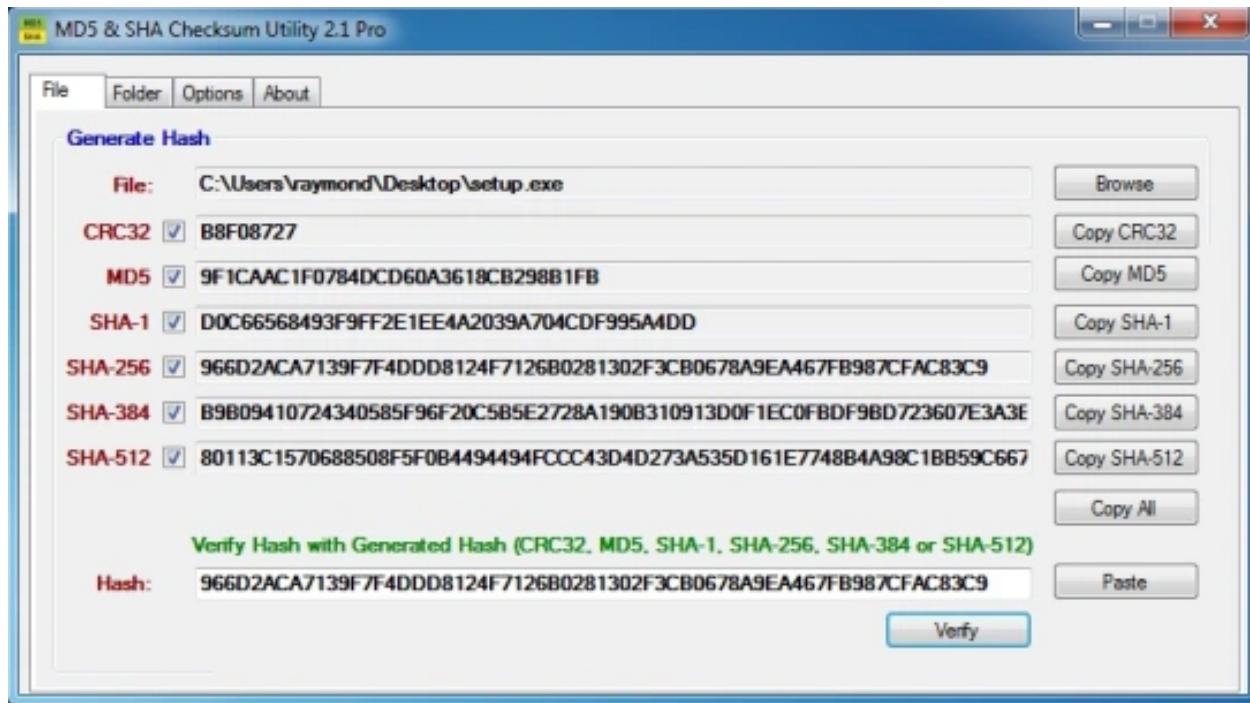
<sup>62</sup><http://www.addario.org/cryptobench/>



#### 2.4.3 Raymond's MD5 & SHA Checksum Utility

- Hash Calculation Utility
- MD5 & SHA Checksum Utility | Raymond's WordPress<sup>63</sup>

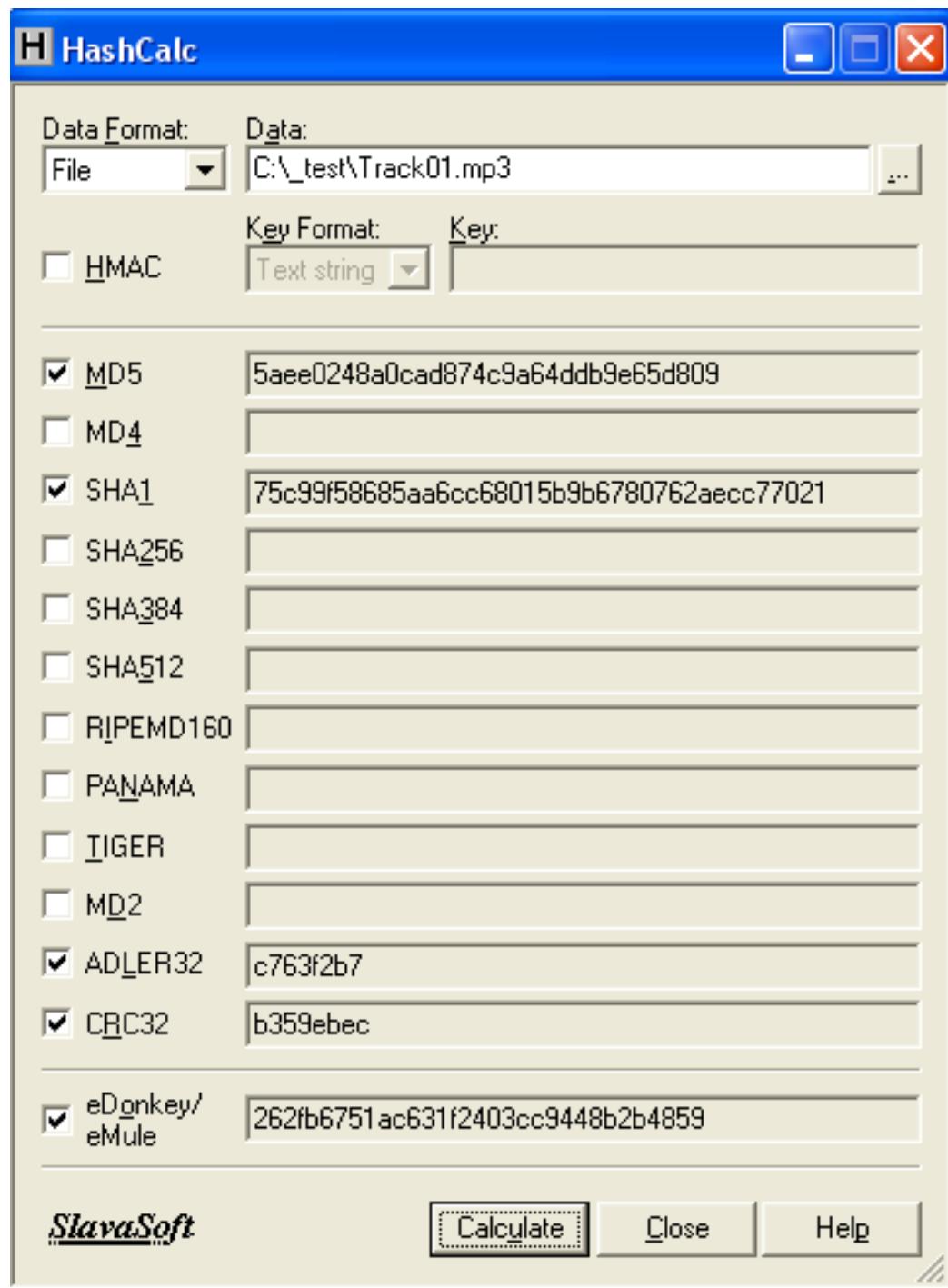
<sup>63</sup><https://raylin.wordpress.com/downloads/md5-sha-1-checksum-utility/>



#### 2.4.4 SlavaSoft HashCalc

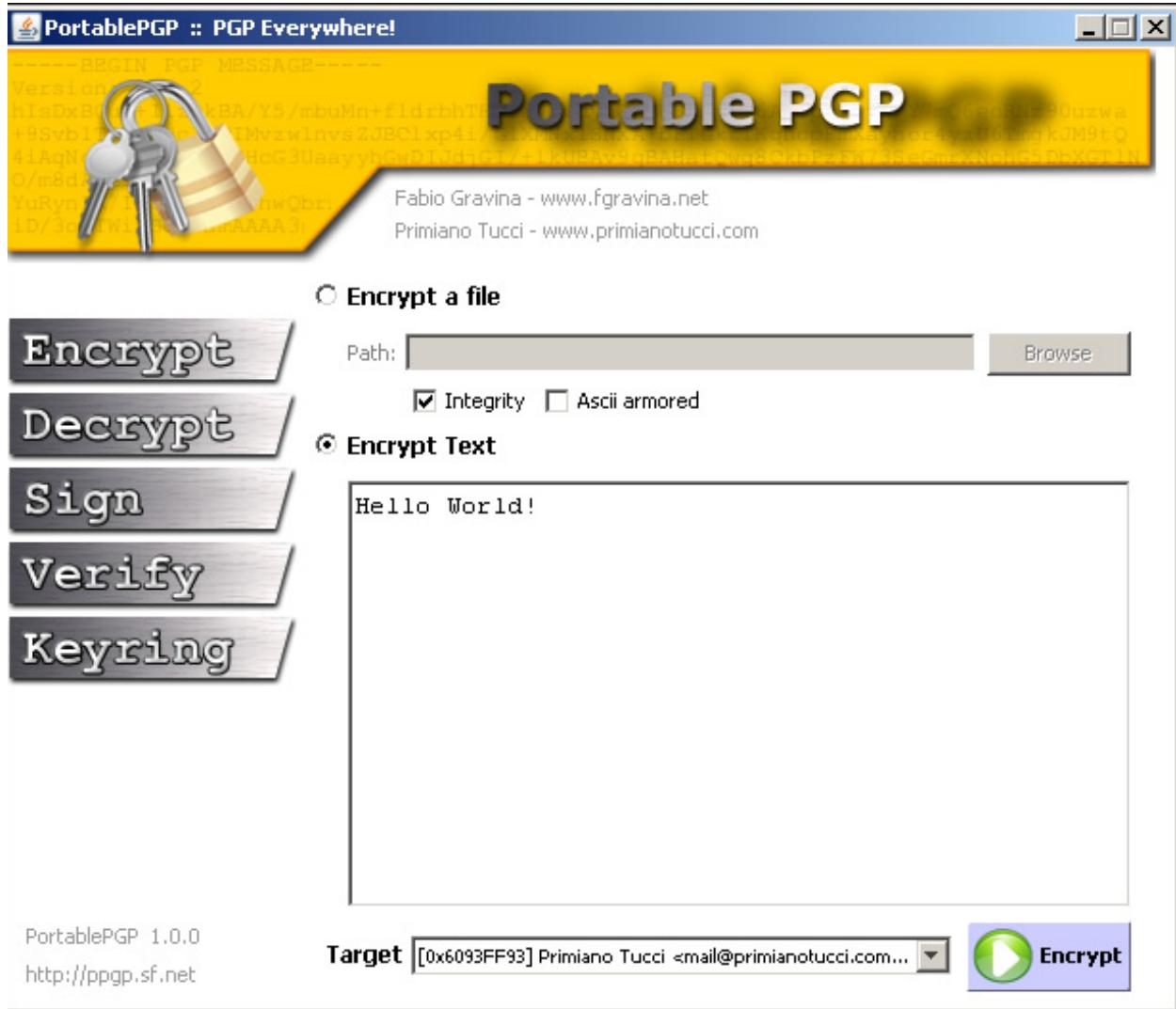
- SlavaSoft HashCalc - Hash, CRC, and HMAC Calculator<sup>64</sup>

<sup>64</sup><https://www.slavasoft.com/hashcalc/>



#### 2.4.5 Portable PGP

- Portable PGP uses for the generation of PGP keys to transfer files securely via e-mail or other channels.  
You can encrypt or sign your documents with this tool then the receiver can open or verify your e-mail.
- <https://ppgp.sourceforge.net/>



#### 2.4.6 Online Programming Environments

- Hackerrank
  - <https://www.hackerrank.com/>
- CS50 Sandbox
  - <https://sandbox.cs50.io/>
- Programiz C Online Compiler
  - Online C Compiler<sup>65</sup>

*End – Of – Week – 2*

---

<sup>65</sup><https://www.programiz.com/c-programming/online-compiler/>