

CE103 Algorithms and Programming I

Introduction to Code Reusability and Automate Testing

Author: Asst. Prof. Dr. Uğur CORUH

Contents

0.1	CE103 Algorithms and Programming I	1
0.2	Week-4	1
0.3	Introduction to Code Reusability and Automated Testing	2
0.4	Shared Library Development	2
0.4.1	C Programming (Static Library)	2
0.4.2	C++ Programming (Static Library)	32
0.4.3	C/C++ WSL Option	32
0.4.4	C/C++ Remote Linux Option over SSH	36
0.4.5	C# Programming (Dinamik Library)	37
0.4.6	Java Programming	49
0.5	Program Testing	118
0.6	Unit Test Development	118
0.6.1	C Unit Tests	118
0.6.2	C++ Unit Tests	119
0.6.3	C# Unit Tests	123
0.6.4	Visual Studio Community Edition (MSTestV2+.Net)	123
1	TL;DR	141
1.0.1	Download and Setup OpenCover, NUnit Console, Report Generator without Package Manager	145
1.0.2	OpenCover	145
1.0.3	ReportGenerator	152
1.0.4	NUnit Console	153
1.0.5	NUnit + MSTest Batch Report Generation (Not Tested)	157
1.0.6	Java Unit Tests	157
1.1	TDD (Test Driven Development)	186
1.2	Test and Deployment Automation Management	186
2	References	187

List of Figures

List of Tables

0.1 CE103 Algorithms and Programming I

0.2 Week-4

0.2.0.1 Fall Semester, 2021-2022 Download DOC¹, SLIDE², PPTX³

¹ce103-week-4-test.md_doc.pdf

²ce103-week-4-test.md_slide.pdf

³ce103-week-4-test.md_slide.pptx

0.3 Introduction to Code Reusability and Automated Testing

During this course we will use entry level of shared library development and their tests and test automations. Also we will see TDD(Test Driven Development) approach.

During this course we will use **Windows OS, Eclipse and Visual Studio Community Edition** environments for examples.

Each example will include two function

“Hello ” printing function with name sayHelloTo(name) and
sum of two variable function for basic, sum = sum(a,b).

This sum function will add a to b and return result to sum variable.

We will locate them in library and use them from a console application, also we will create unit tests for testing their functionalities and return variables

0.4 Shared Library Development

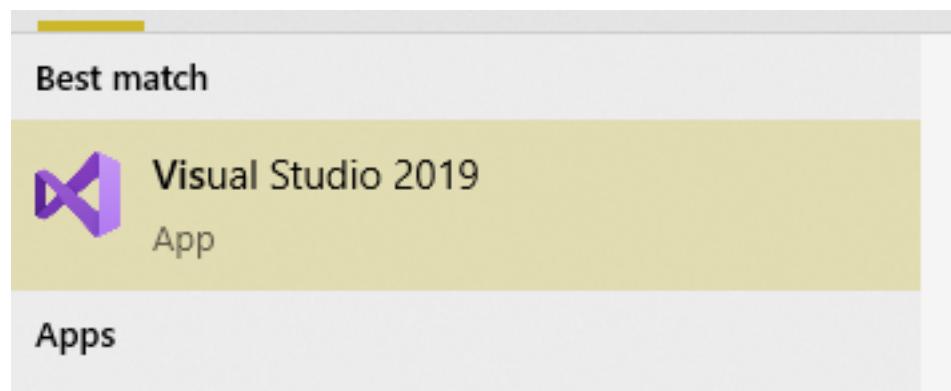
0.4.1 C Programming (Static Library)

0.4.1.1 Visual Studio Community Edition In this sample we will create **c-lib-sample** project that contains library, executable, unit tests and unit test runners.

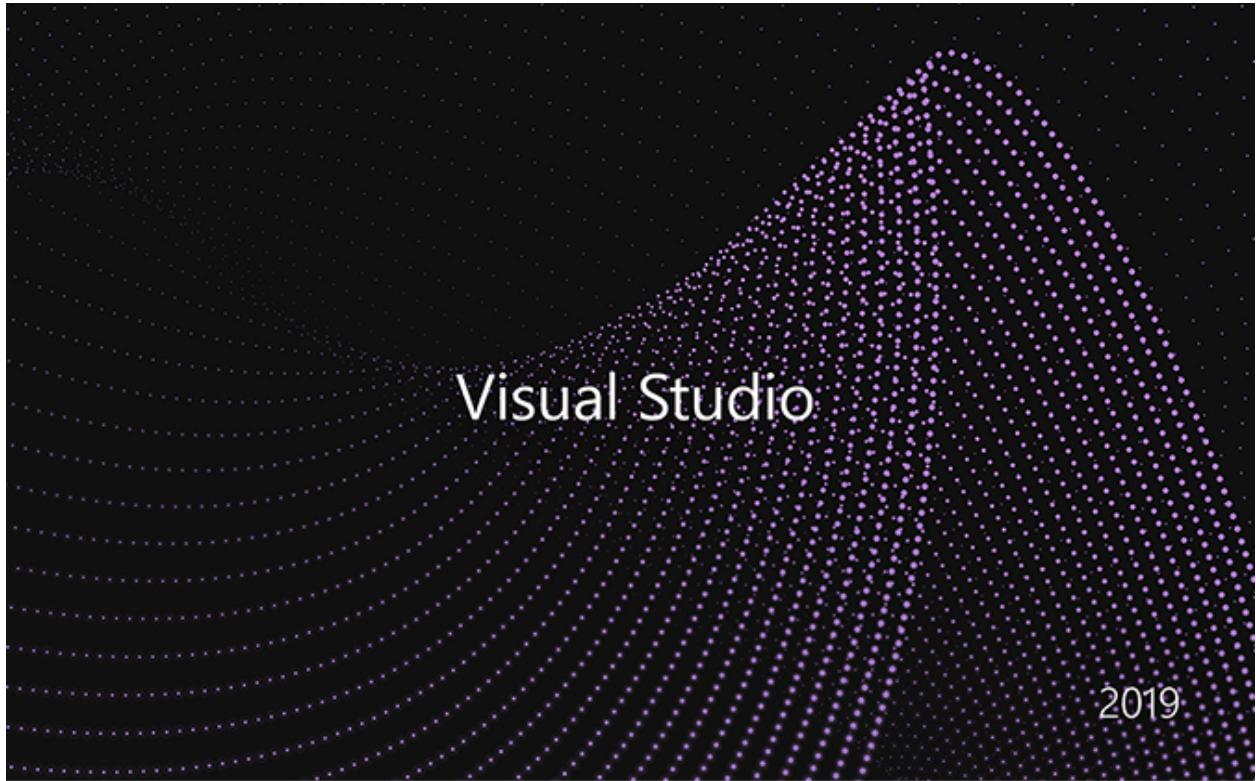
First of all you install Visual Studio Community Edition from website

Visual Studio 2019 Community Edition - Son Ücretsiz Sürümü İndir⁴

Open visual studio community edition and select create a new project



⁴<https://visualstudio.microsoft.com/tr/vs/community/>



Select create a new project

Get started



Clone a repository

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

Navigate and edit code within any folder



Create a new project

Choose a project template with code scaffolding to get started

[Continue without code →](#)

Select C++ static library from project list

Search for templates (Alt+S)  Clear all

C++ Windows All project types

 Epic Games Launcher (Unreal Engine install client)
Unreal Engine 4 is a complete suite of game development tools.

C++ Windows Games

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

C++ Windows Library

 Static Library
Build a .lib that can be packaged inside other Windows executables.

C++ Windows Library

 Shared Items Project
A Shared Items project is used for sharing files between multiple projects.

C++ Windows Android iOS Linux Desktop Console Library UWP

Name static library project

Configure your new project

Static Library C++ Windows Library

Project name

c-sample-lib

Location

E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programming I\Lectures\ce1l

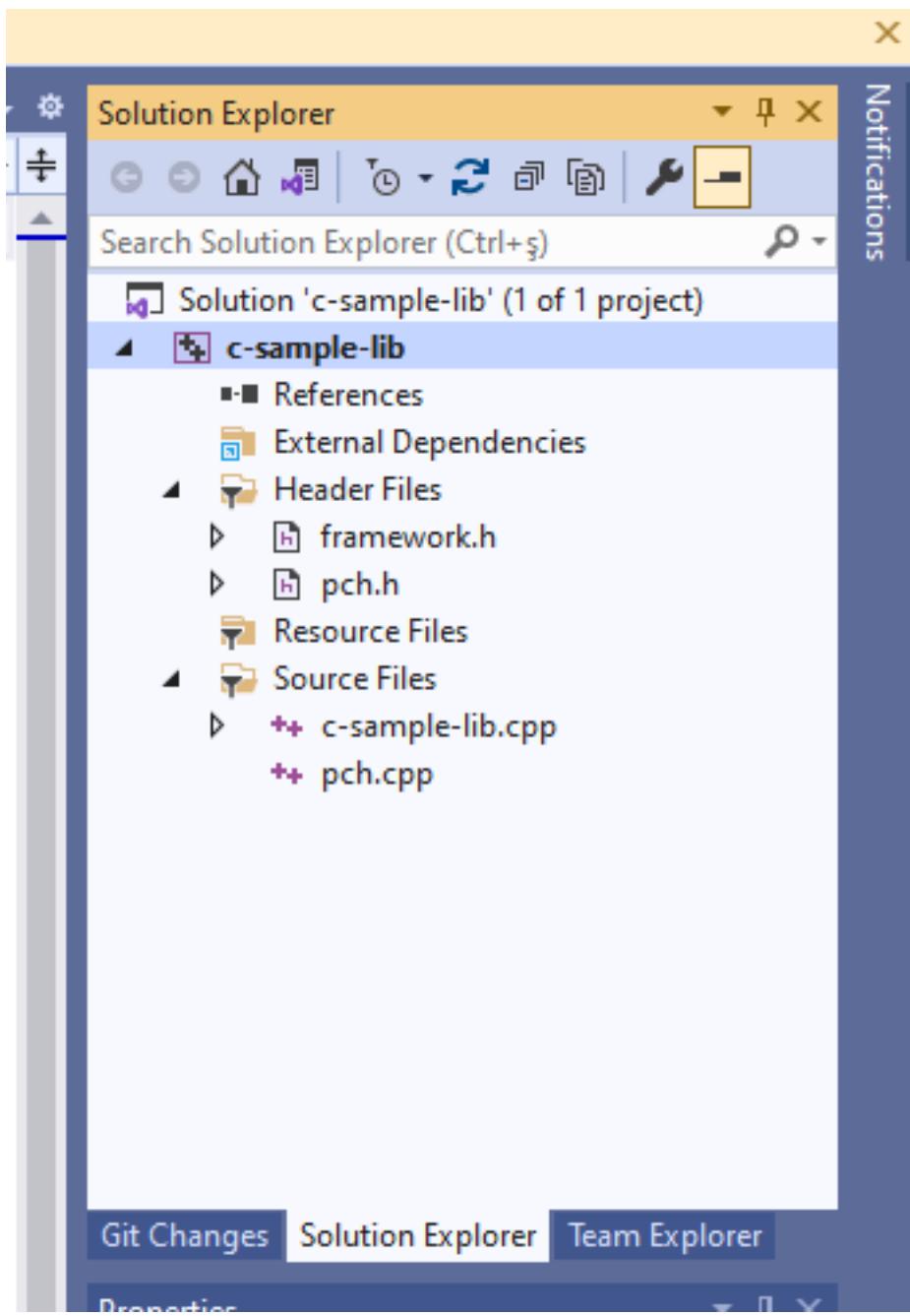
...

Solution name 

c-sample-lib

Place solution and project in the same directory

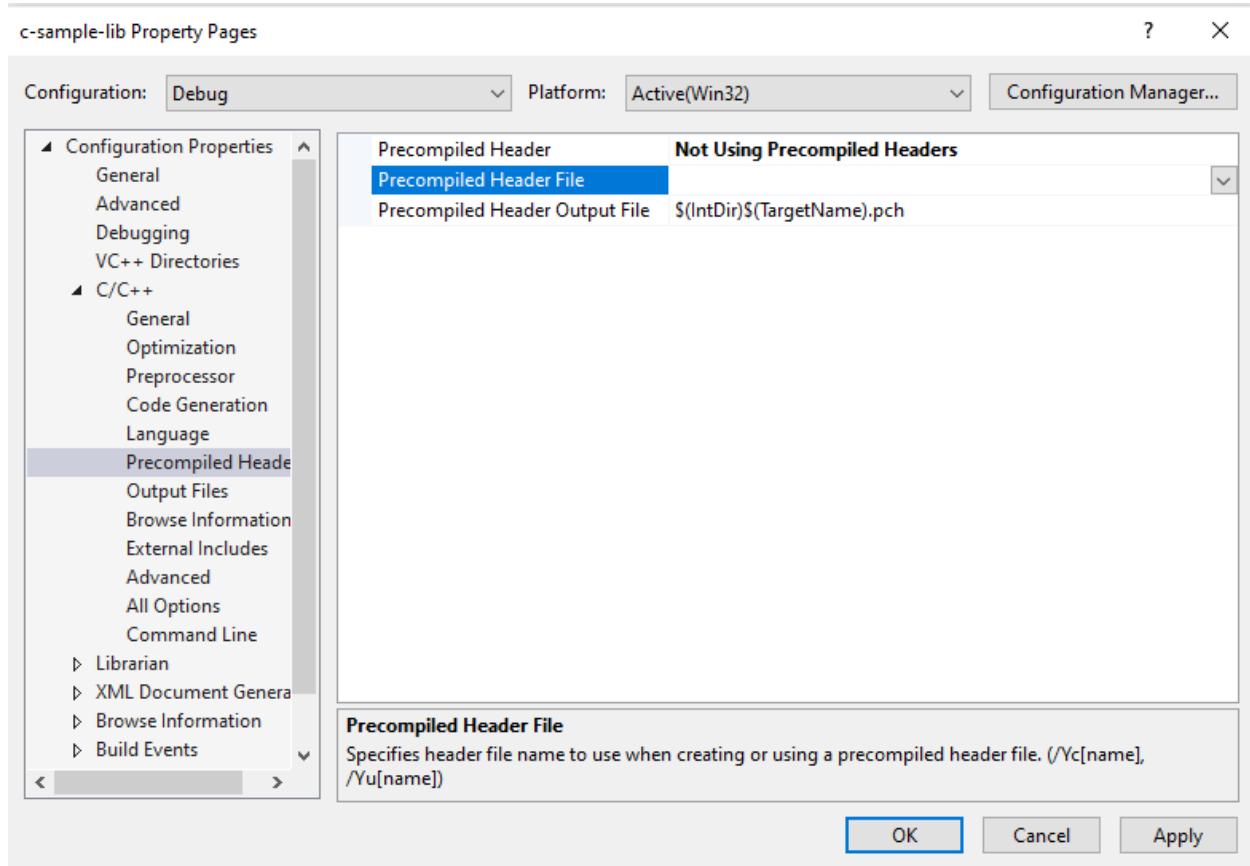
Default configuration come with C++ project types and setting



In the c-sample-lib.cpp you will sample function

```
void fnCSampleLib()
{
}
```

Delete pch.h and pch.c files. Also disable use precompiled header settings from configurations and change to “Not Using Precompiled Headers”, also you can delete precompiled Header File.



Customize library header name and update “framework.h” to “samplelib.h”

Insert your functions inside the c-sample-lib.c and update header files also.

```
// c-sample-lib.cpp : Defines the functions for the static library.
//

#include "samplelib.h"
#include "stdio.h"

/// <summary>
///
/// </summary>
/// <param name="name"></param>
void sayHelloTo(char* name){

    if (name != NULL){
        printf("Hello %s \n",name);
    }
    else {
        printf("Hello There\n");
    }
}

/// <summary>
///
/// </summary>
/// <param name="a"></param>
```

```

/// <param name="b"></param>
/// <returns></returns>
int sum(int a, int b){

    int c = 0;
    c = a + b;
    return c;
}

```

also update samplelib.h

```

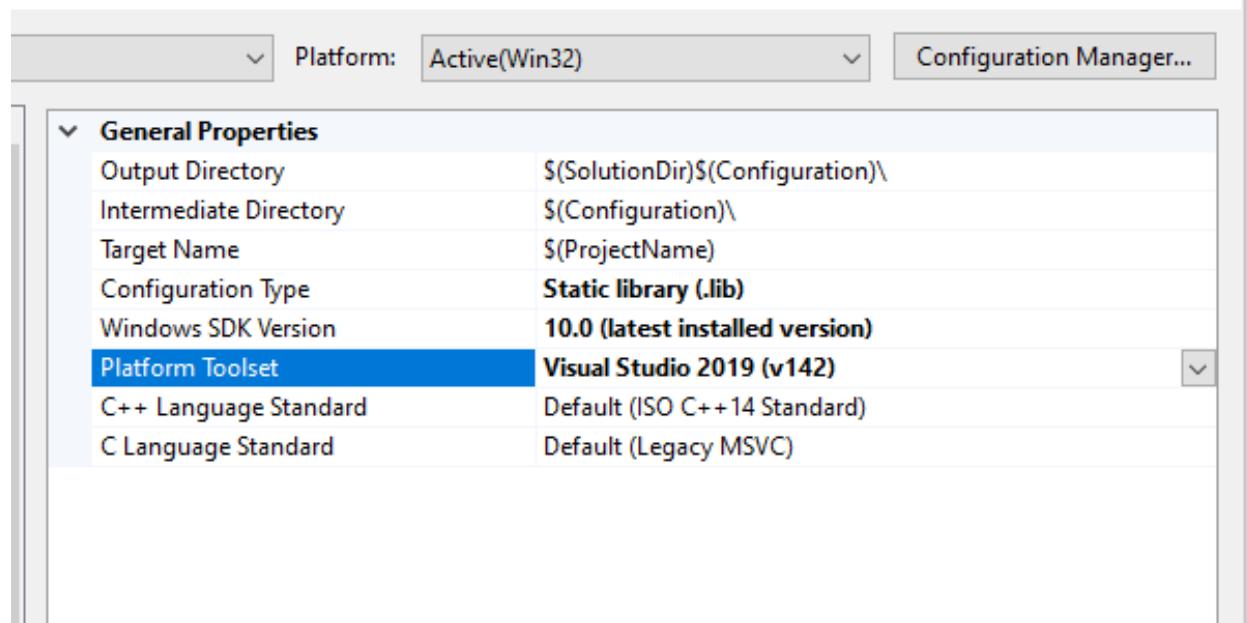
#pragma once

#define WIN32_LEAN_AND_MEAN           // Exclude rarely-used stuff from Windows headers

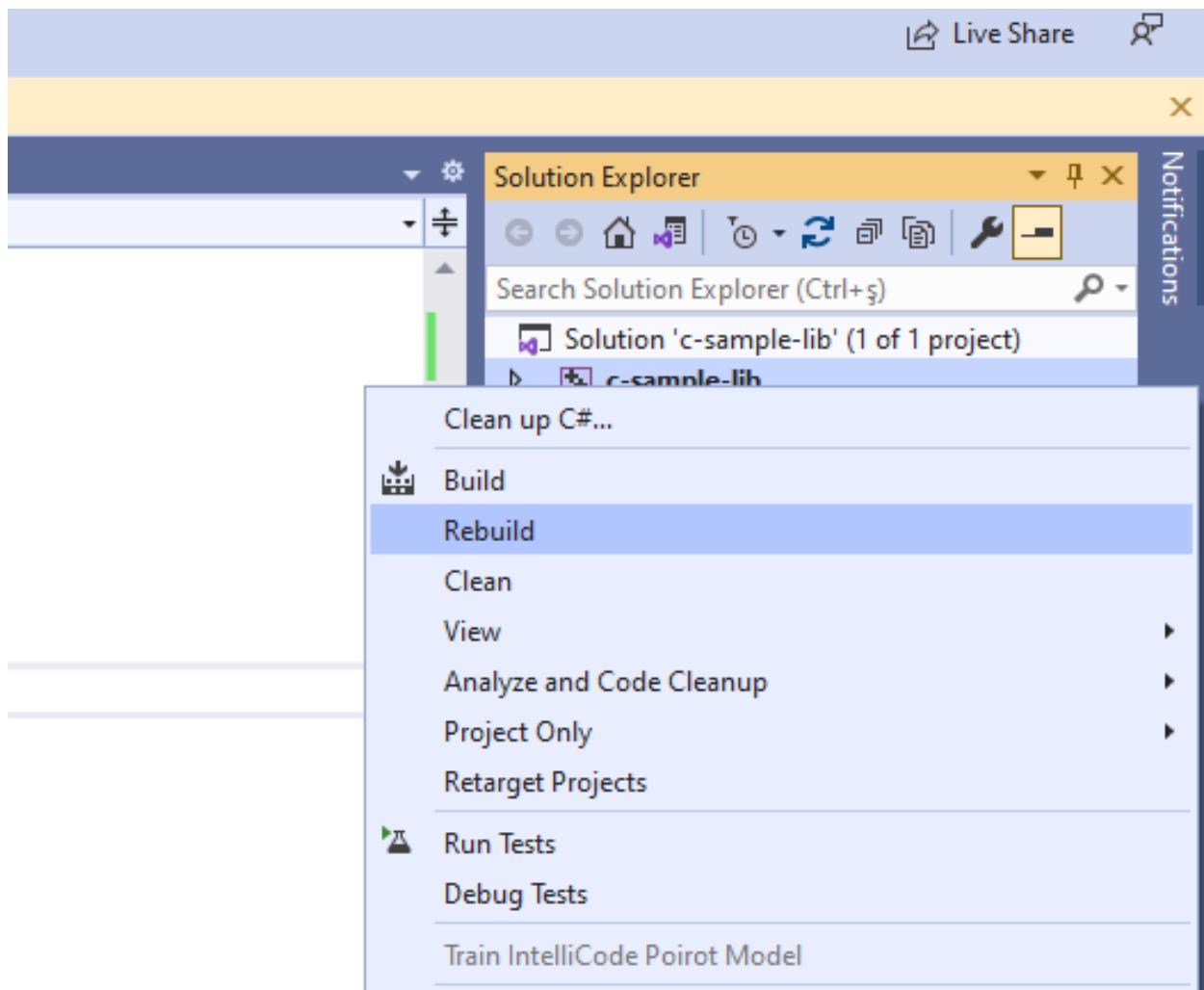
void sayHelloTo(char* name);
int sum(int a, int b);

```

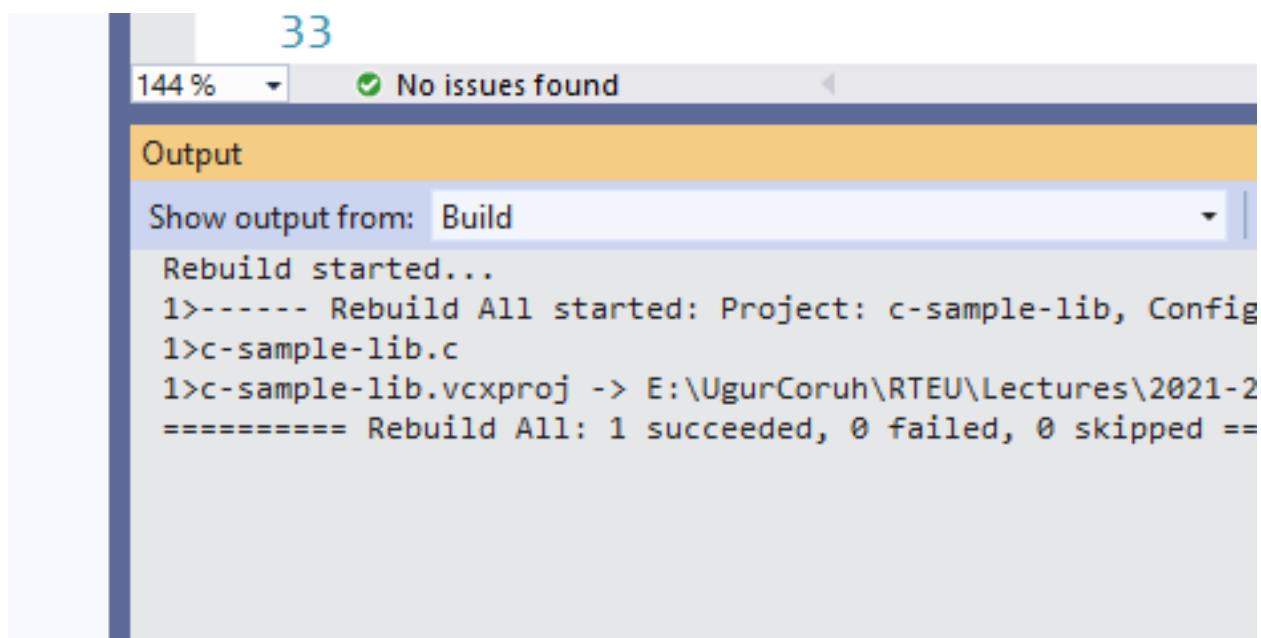
If you check configuration you will see that for C complier we are using Microsoft Environment and Toolkits



Now we can compile our library



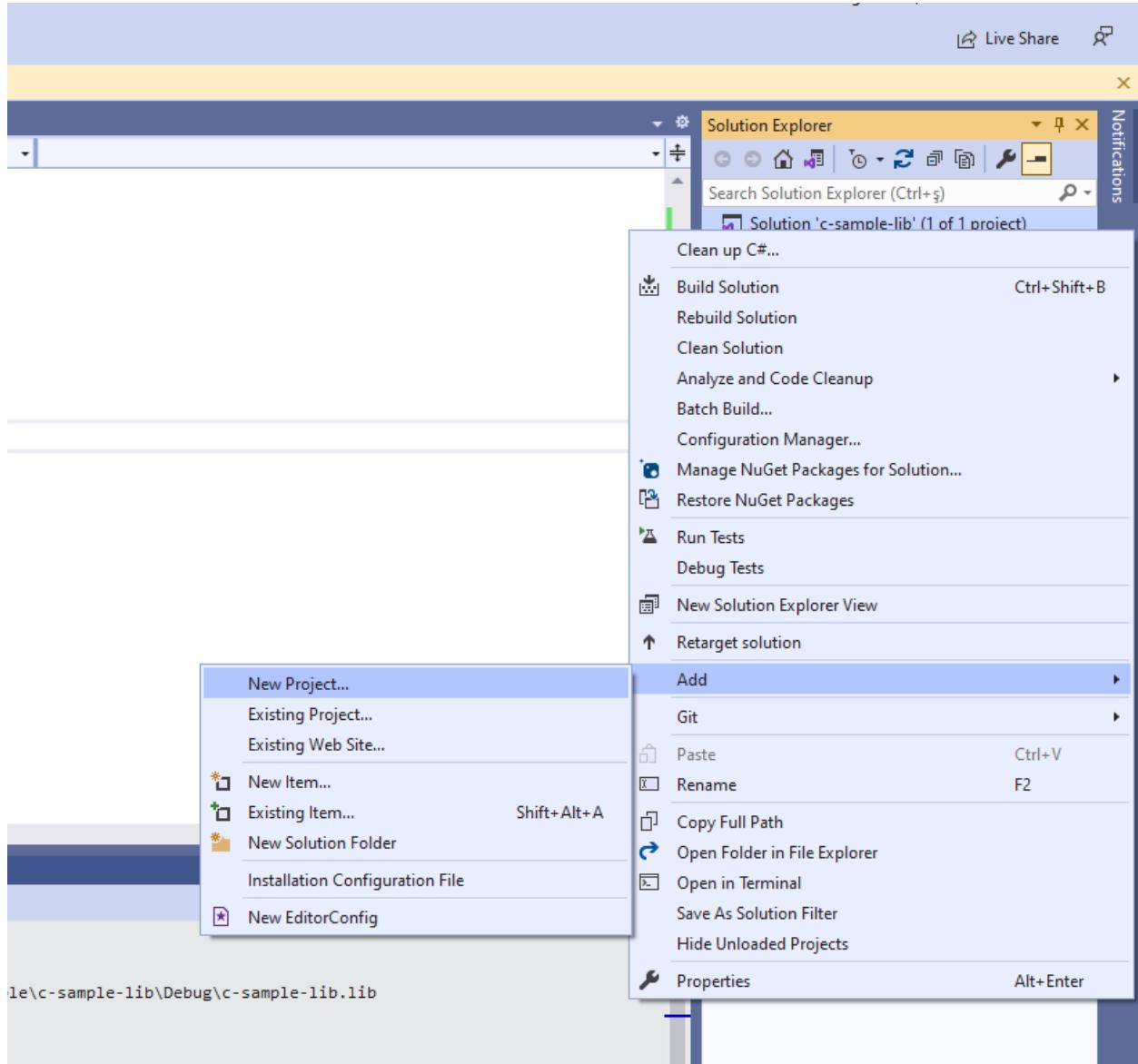
You can follow operation from output window



in debug folder we will see our output

Name
 c-sample-lib.idb
 c-sample-lib.lib
 c-sample-lib.pdb

now we will add a console application c-sample-app and use our library



select C++ Windows Console Application from list

Search for templates (Alt+S)

C++ Windows All project types

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

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

CMake Project
Build modern, cross-platform C++ apps that don't depend on .sln or .vcxproj files.

Windows Desktop Wizard

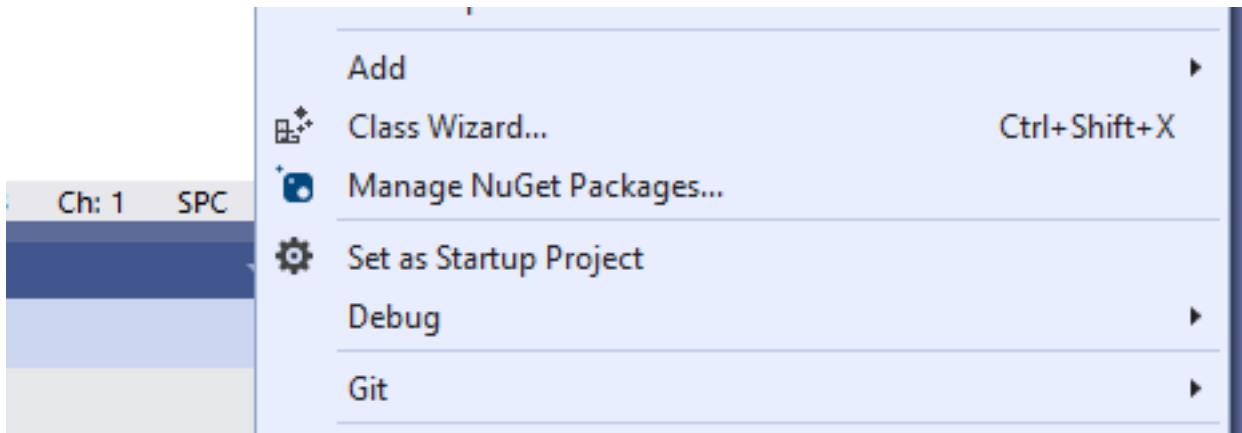
C++ Console Application Selection will generate a C++ console project we can change extension to C to compile our application as C application.

we will convert c-sample-app.c to following code

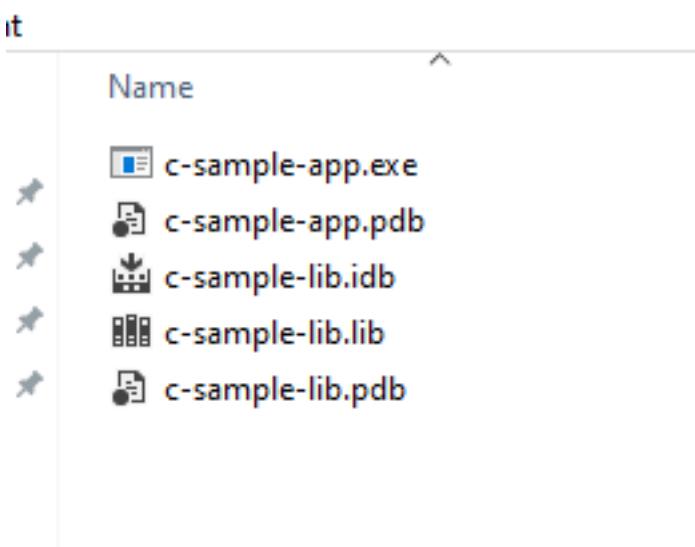
```
#include <stdio.h>

/// <summary>
///
/// </summary>
/// <returns></returns>
int main()
{
    printf("Hello World!\n");
}
```

after conversion set c-sample-app as startup project and build it



this will create c-sample-app.exe in the same folder with c-sample-lib.lib library

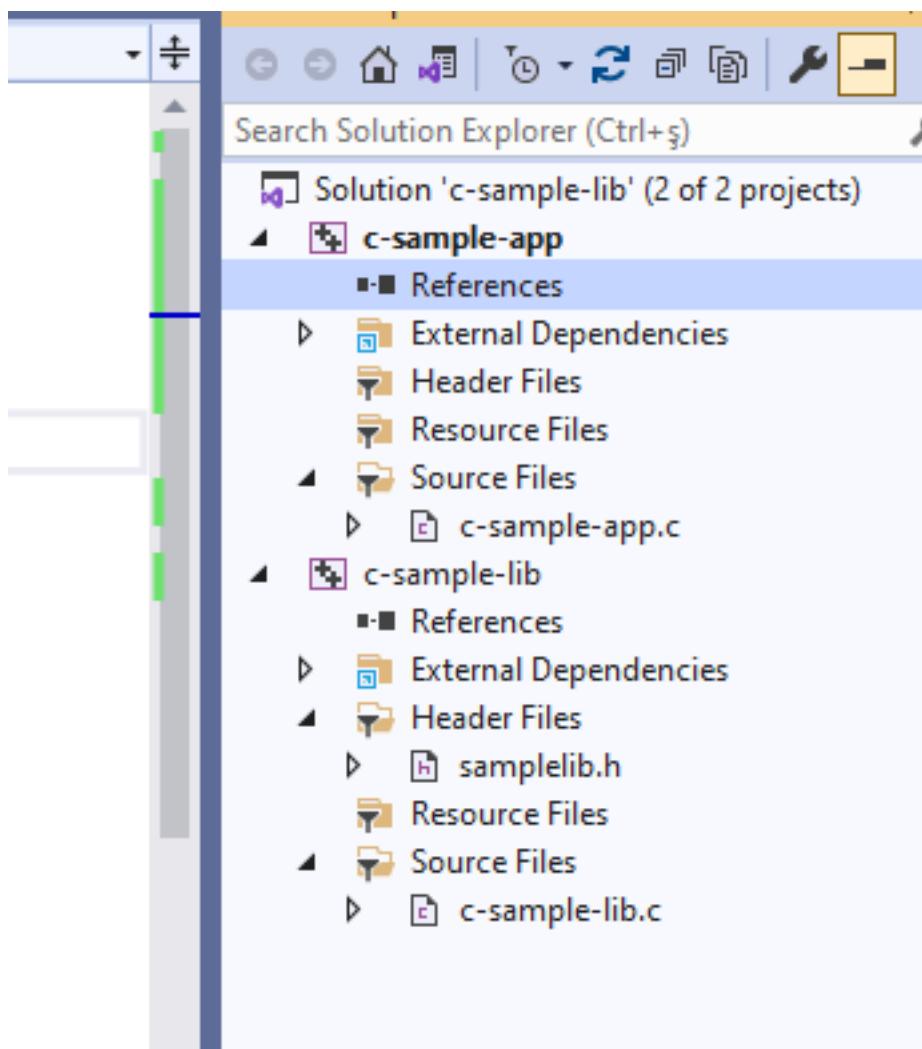


if we run application we will see only "Hello World"

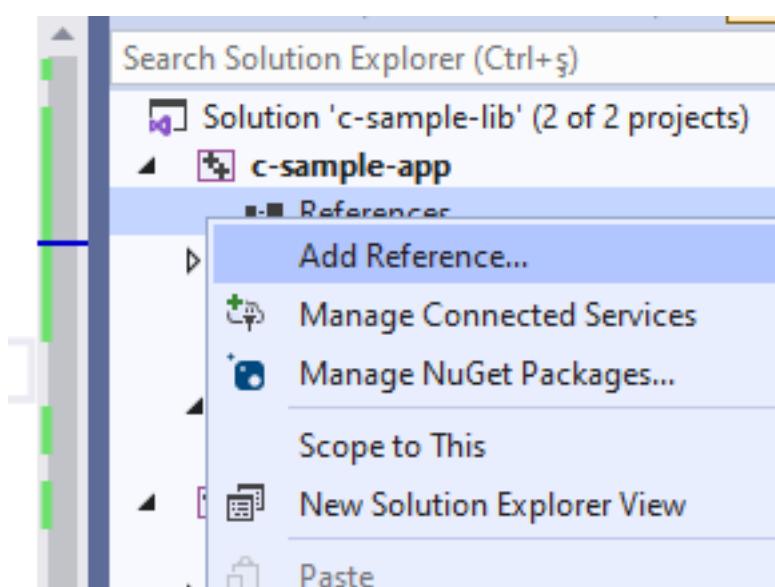
now we will see two options to add library as references in our application and use its functions.

First option

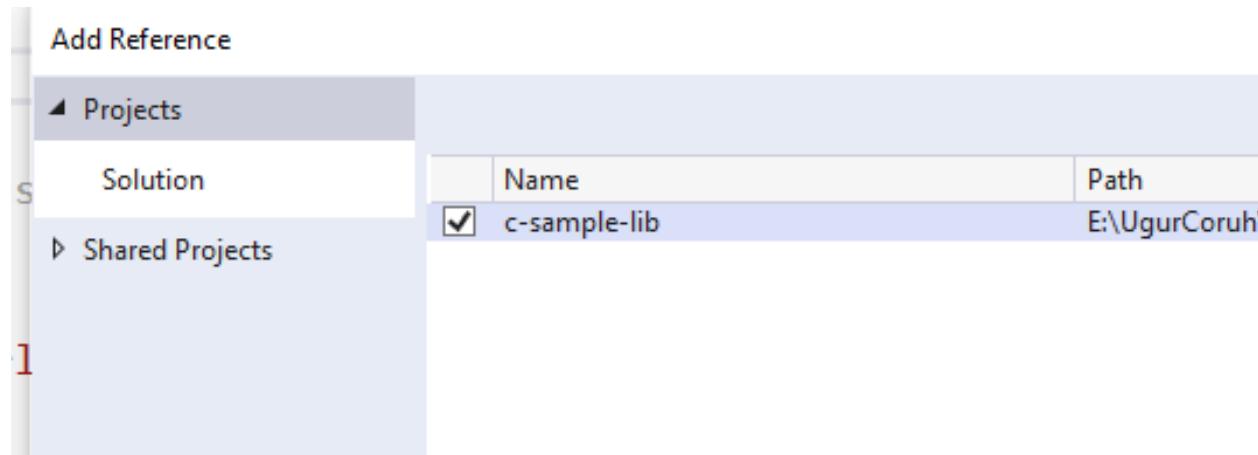
right click references for c-sample-app and add current library as reference



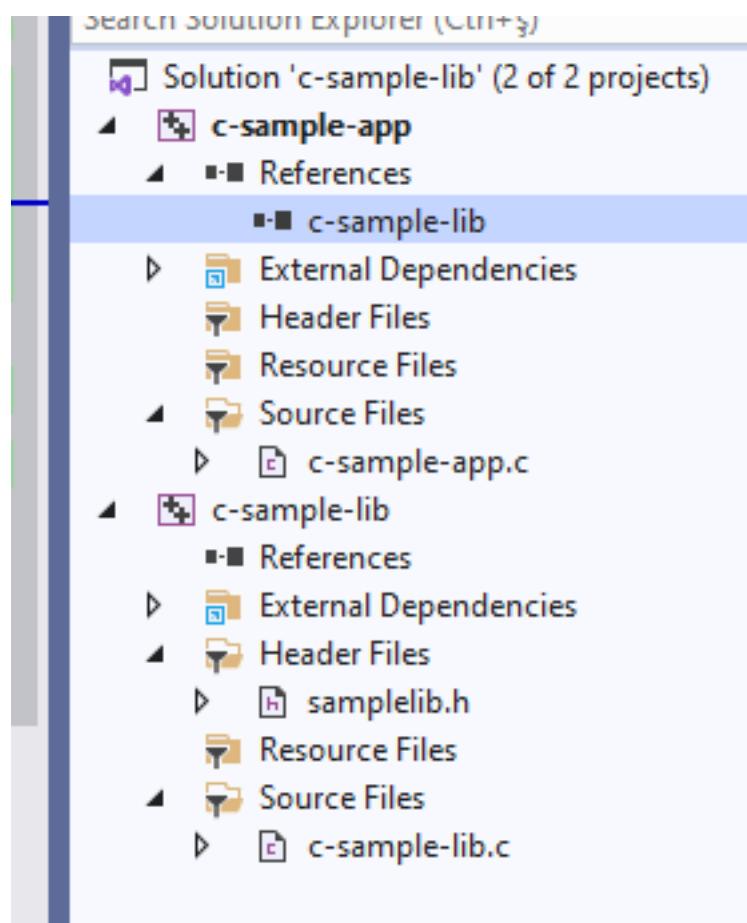
Select Add Reference



Browse for solution and select c-sample-lib



You can check added reference from references section



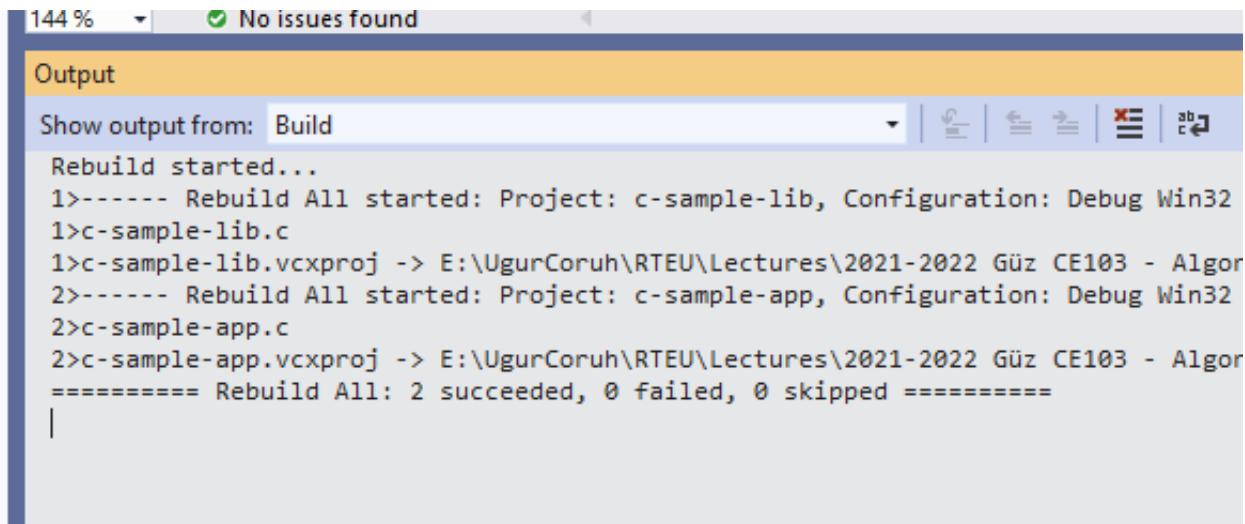
now we can include required headers from c-sample-lib folder and use it.

we can include required header with relative path as follow or with configuration

```
#include <stdio.h>
#include "../c-sample-lib/samplelib.h"
/// <summary>
///
```

```
/// </summary>
/// <returns></returns>
int main()
{
    printf("Hello World!\n");
}
```

we can build our c-sample-app



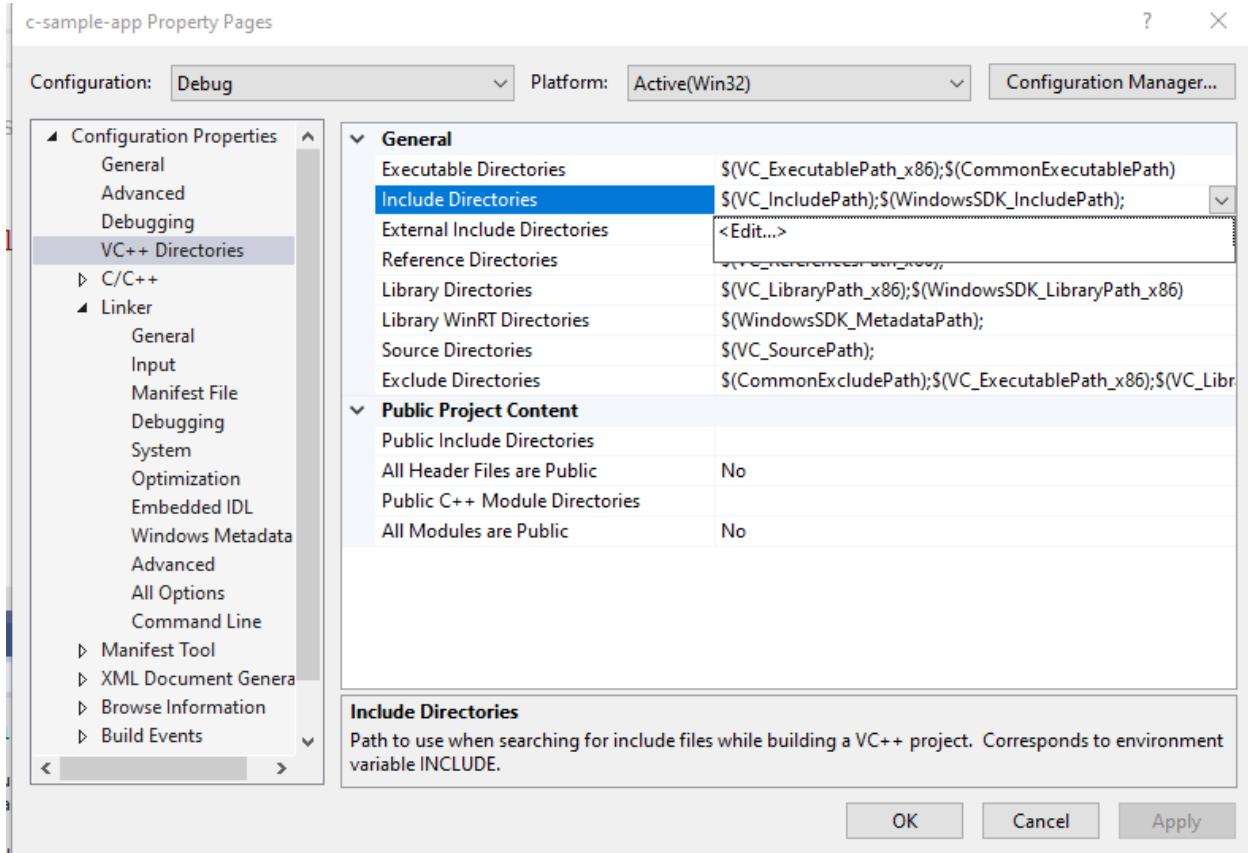
A screenshot of a terminal window titled "Output". The window shows the command "Rebuild started..." followed by a series of log entries from a rebuild process. The entries include paths like "E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algor" and file names like "c-sample-lib.c" and "c-sample-app.c". The log concludes with "===== Rebuild All: 2 succeeded, 0 failed, 0 skipped =====". The terminal has a dark theme with light-colored text.

```
144 %  No issues found
Output
Show output from: Build
Rebuild started...
1>----- Rebuild All started: Project: c-sample-lib, Configuration: Debug Win32
1>c-sample-lib.c
1>c-sample-lib.vcxproj -> E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algor
2>----- Rebuild All started: Project: c-sample-app, Configuration: Debug Win32
2>c-sample-app.c
2>c-sample-app.vcxproj -> E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algor
===== Rebuild All: 2 succeeded, 0 failed, 0 skipped =====
```

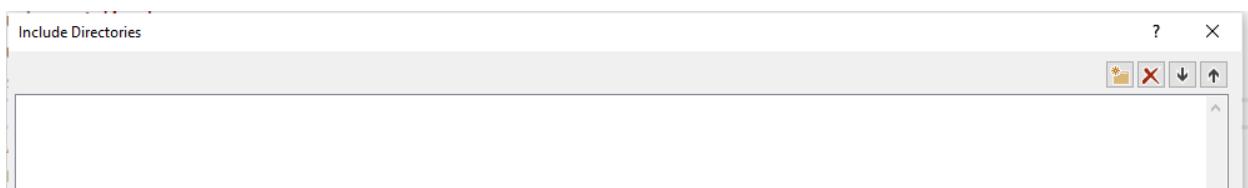
also we can only write header name

```
#include <samplelib.h>
```

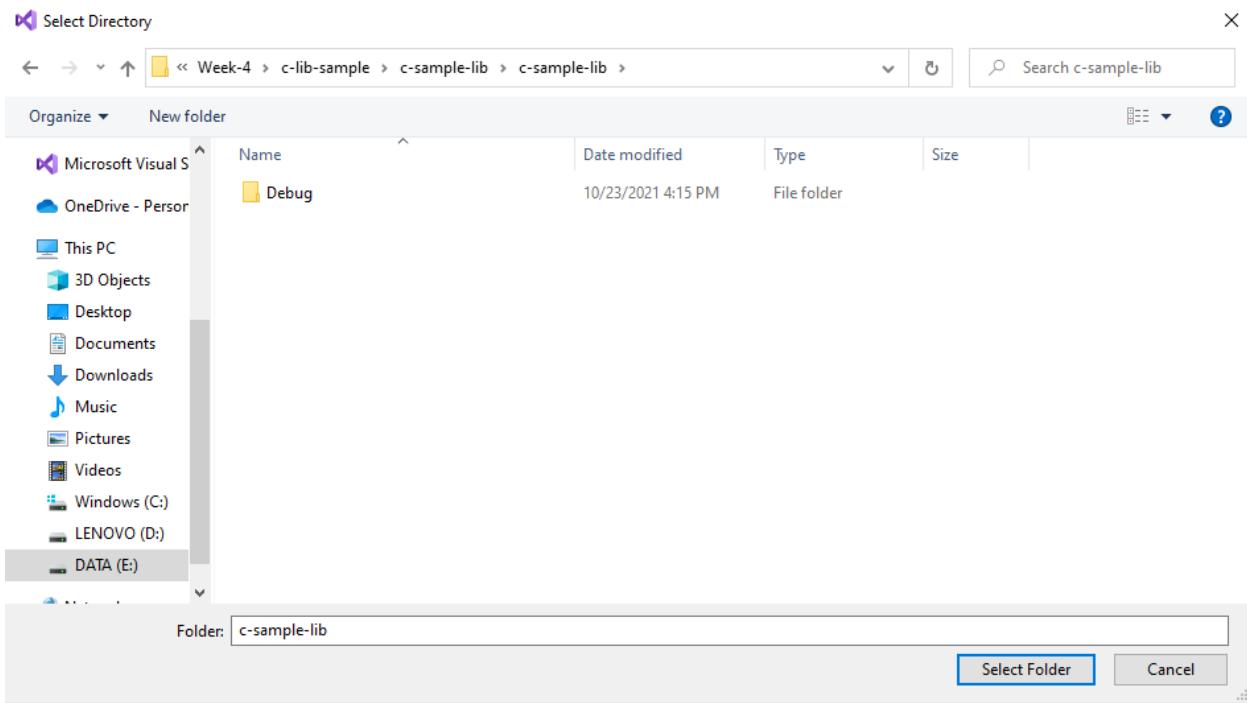
for this we need to configure include directories



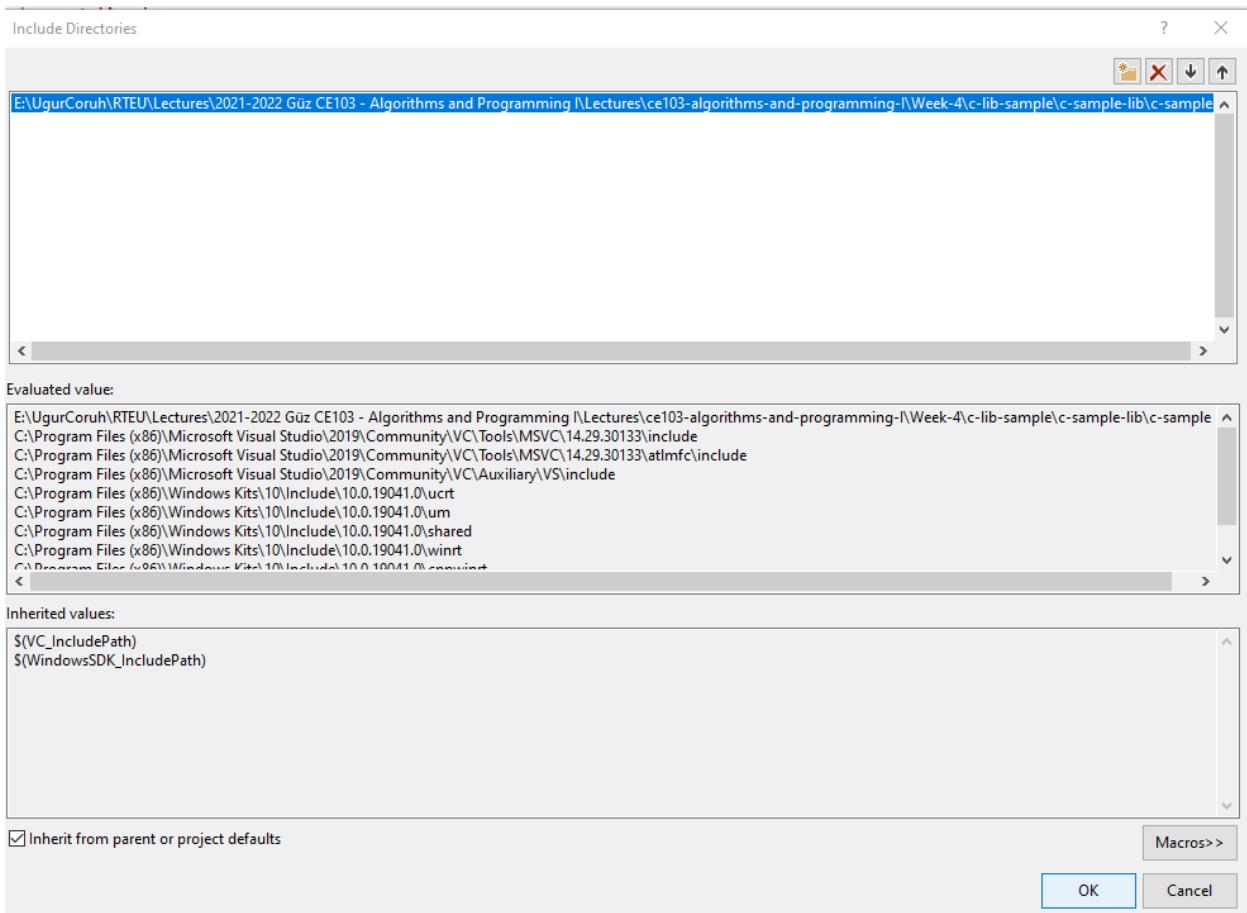
select c-sample-lib header file location



browse for folder



your full path will be added to your configuration

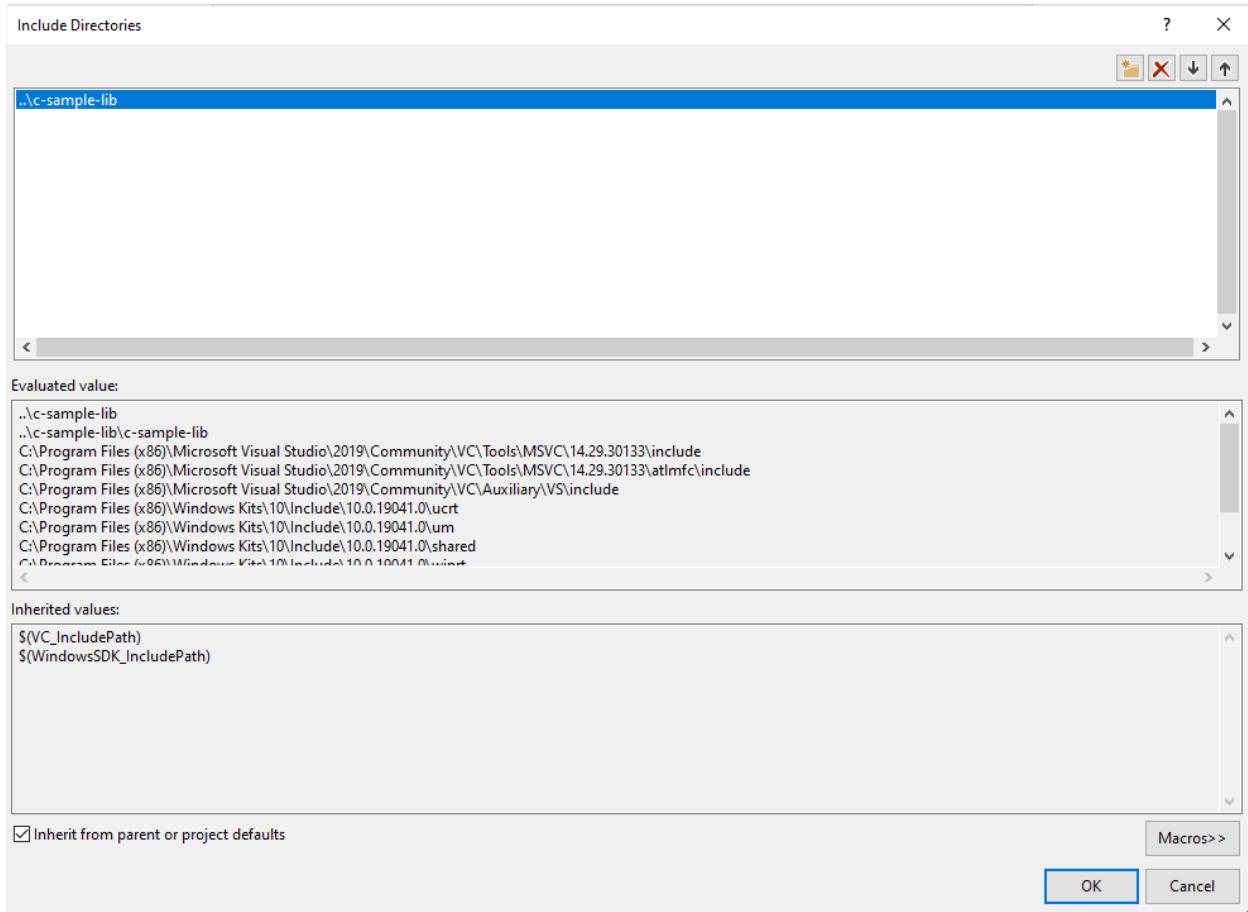


if you add header file paths to your configuration you can use header files by name in your source code

```
#include <stdio.h>
#include <samplelib.h>
/// <summary>
///
/// </summary>
/// <returns></returns>
int main()
{
    printf("Hello World!\n");
}
```

we can compile the following we don't have problems but here we need to configure relative paths for configuration open include library settings and update with relative path

..\\c-sample-lib



now we have portable source code configuration. we can call our functions and then we can update header and library folder configurations.

```
#include <stdio.h>
#include <samplelib.h>
/// <summary>
///
/// </summary>
/// <returns></returns>
```

```

int main()
{
    int result = 0;
    //printf("Hello World!\n");
    result = sum(5, 4);
    sayHelloTo("Computer");
    printf("Result is %d \n",result);
    printf("Press any key to continue...\n");
    getchar();
    return 0;
}

```

when you run you will see the following outputs, that mean we called library functions.



static library is a code sharing approach if you want to share your source code with your customers then you can share static libraries and header files together. Another case you can use a precompiled static library with you or this library can be part of any installation then if there is a installed app and static libraries are placed on system folder or any different location then you can use configuration files to set library path and included header paths

Now we can remove project from c-sample-app references but we will set library file in configuration

Before this copy static library and header files to a folder like that

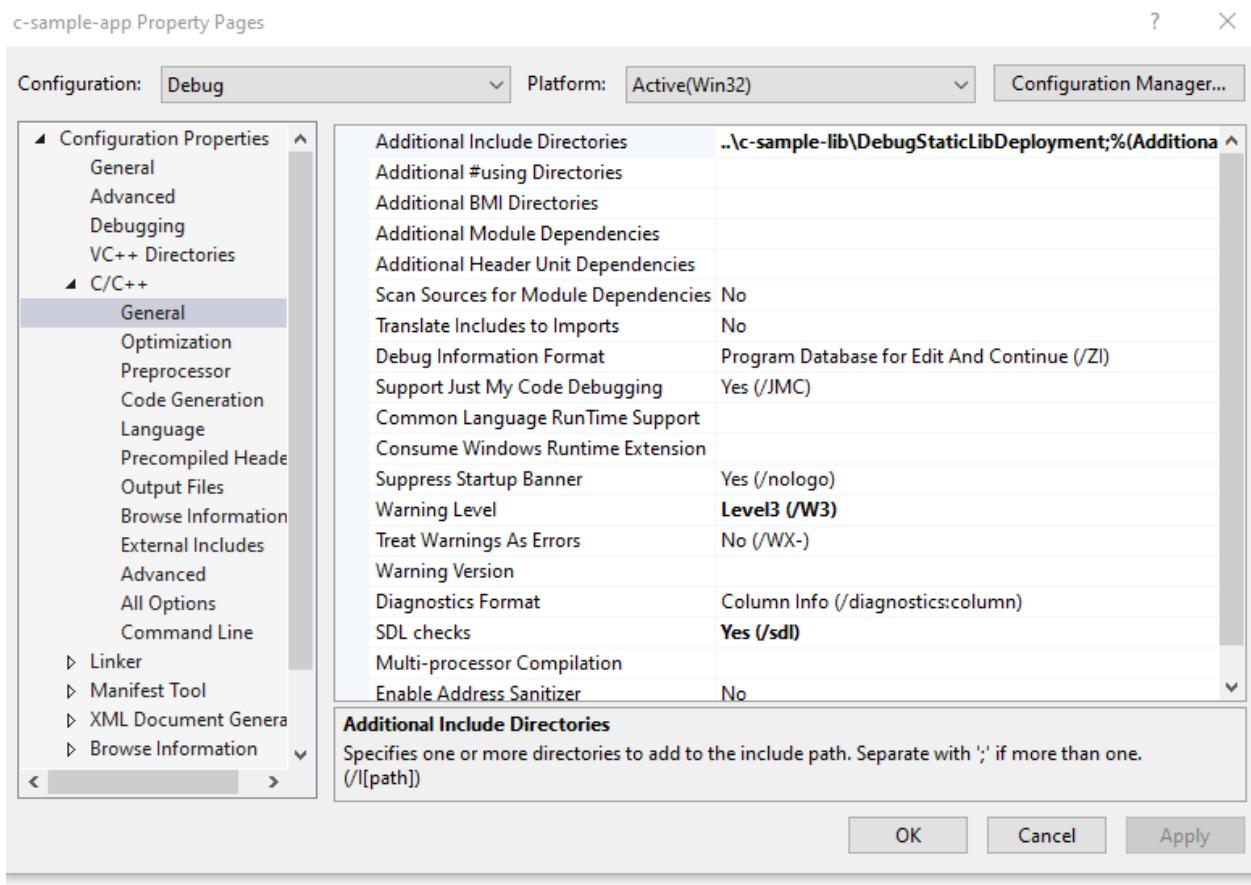
DebugStaticLibDeployment

- Set C/C++ -> General -> Additional Include Directories

There is a bug in configurations and relative path not finding headers so for this reason we will set full path but this is not a good practice for team working

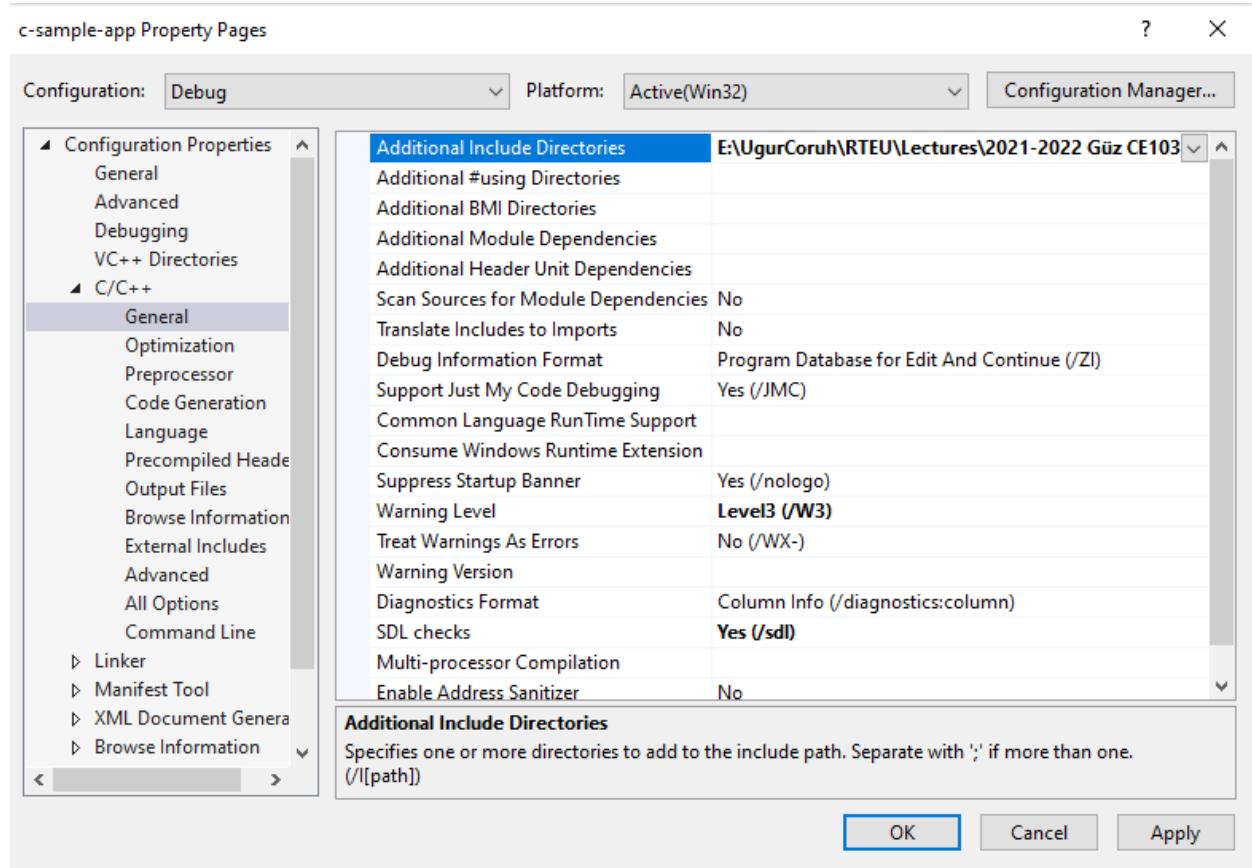
Not Working

`..\\c-sample-lib\\DebugStaticLibDeployment`



Working

E:\...\c-lib-sample\c-sample-lib\DebugStaticLibDeployment



Now we will set library folder that our static library placed

we will set VC++ Directories -> Library Directories

Here is the same issue if we use relative path it doesn't work we need to set full path for library folder

Working

E:\...\\c-lib-sample\\c-sample-lib\\DebugStaticLibDeployment

Configuration: Debug Platform: Active(Win32) Configuration Manager...

▲ Configuration Properties ▼

- General
- Advanced
- Debugging
- VC++ Directories
- ▶ C/C++
 - General
 - Optimization
 - Preprocessor
 - Code Generation
 - Language
 - Precompiled Headers
 - Output Files
 - Browse Information
 - External Includes
 - Advanced
 - All Options
 - Command Line
- ▷ Linker
- ▷ Manifest Tool
- ▷ XML Document Generation
- ▷ Browse Information

▼ General

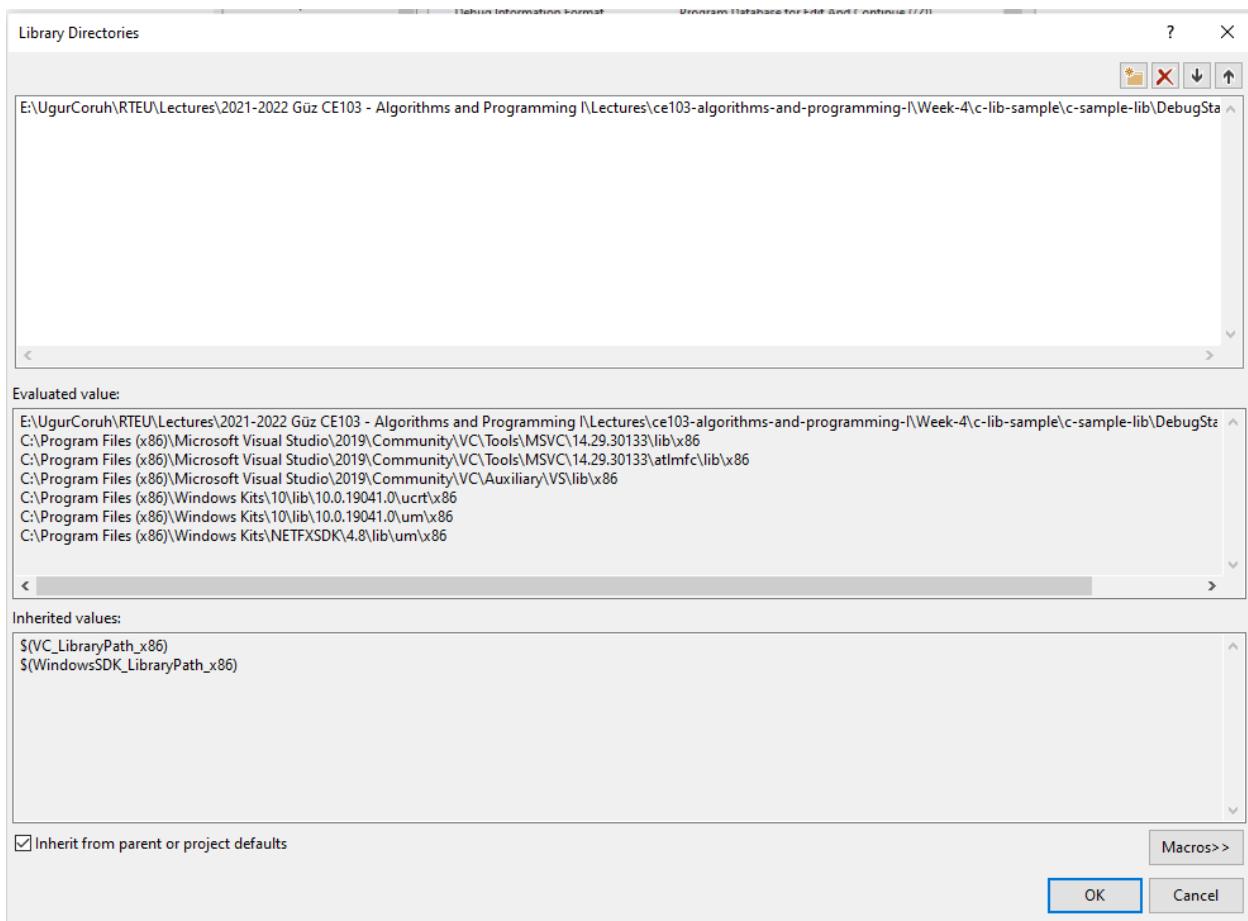
Executable Directories	<code>\$(VC_ExecutePath_x86);\$(CommonExecutablePath)</code>
Include Directories	<code>\$(IncludePath)</code>
External Include Directories	<code>\$(ExternalAllIncludePath)</code>
Reference Directories	<code>\$(VC_ReferencesPath_x86);</code>
Library Directories	<code>E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - A\</code>
Library WinRT Directories	<code>\$(WindowsSDK_MetadataPath);</code>
Source Directories	<code>\$(VC_SourcePath);</code>
Exclude Directories	<code>\$(CommonExcludePath);\$(VC_ExecutePath_x86);\$(VC_Libr</code>

▼ Public Project Content

Public Include Directories	No
All Header Files are Public	No
Public C++ Module Directories	
All Modules are Public	

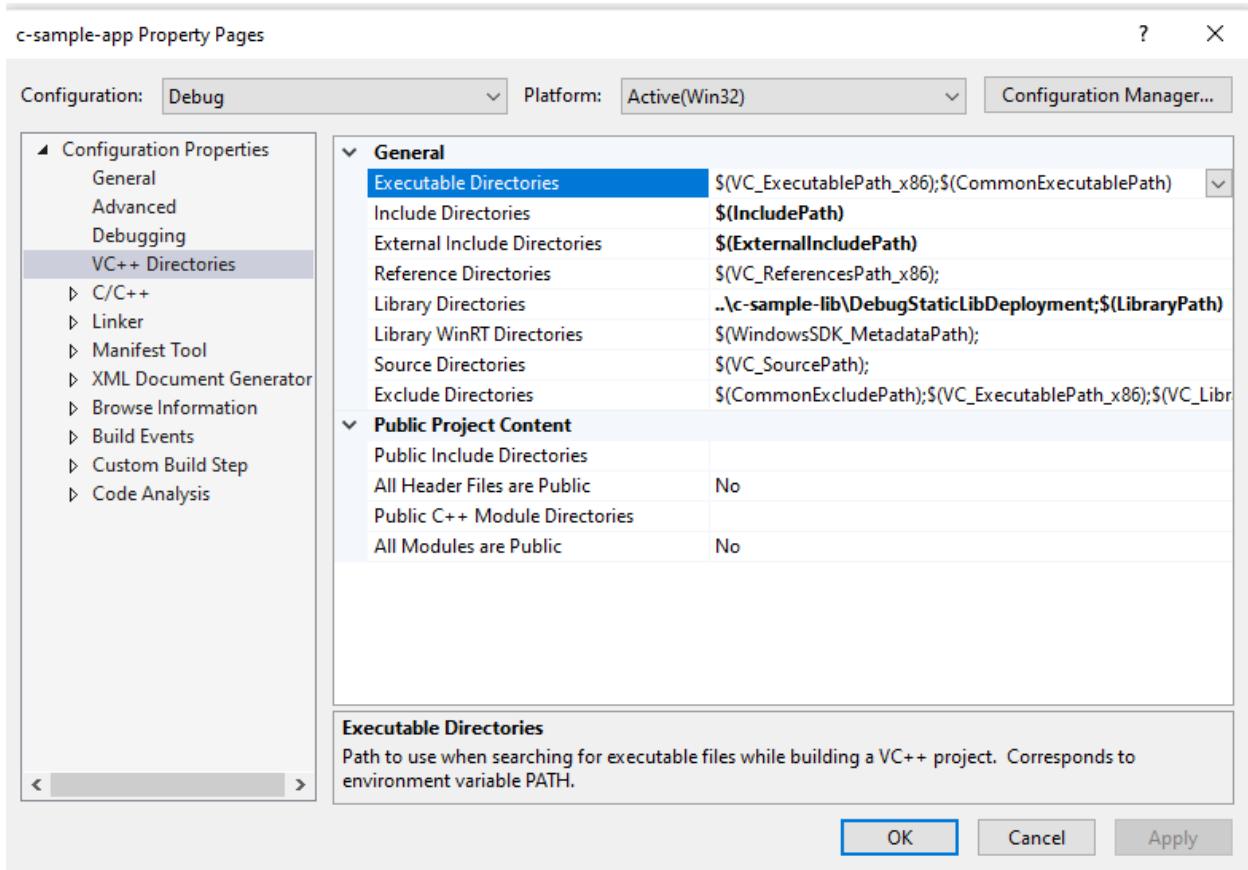
Library Directories
Path to use when searching for library files while building a VC++ project. Corresponds to environment variable LIB.

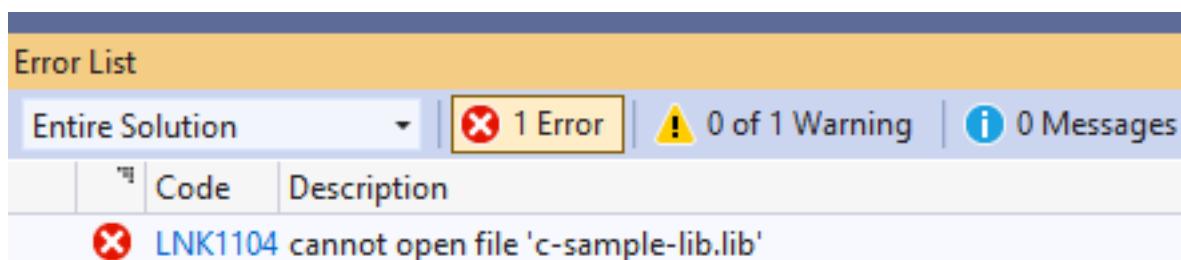
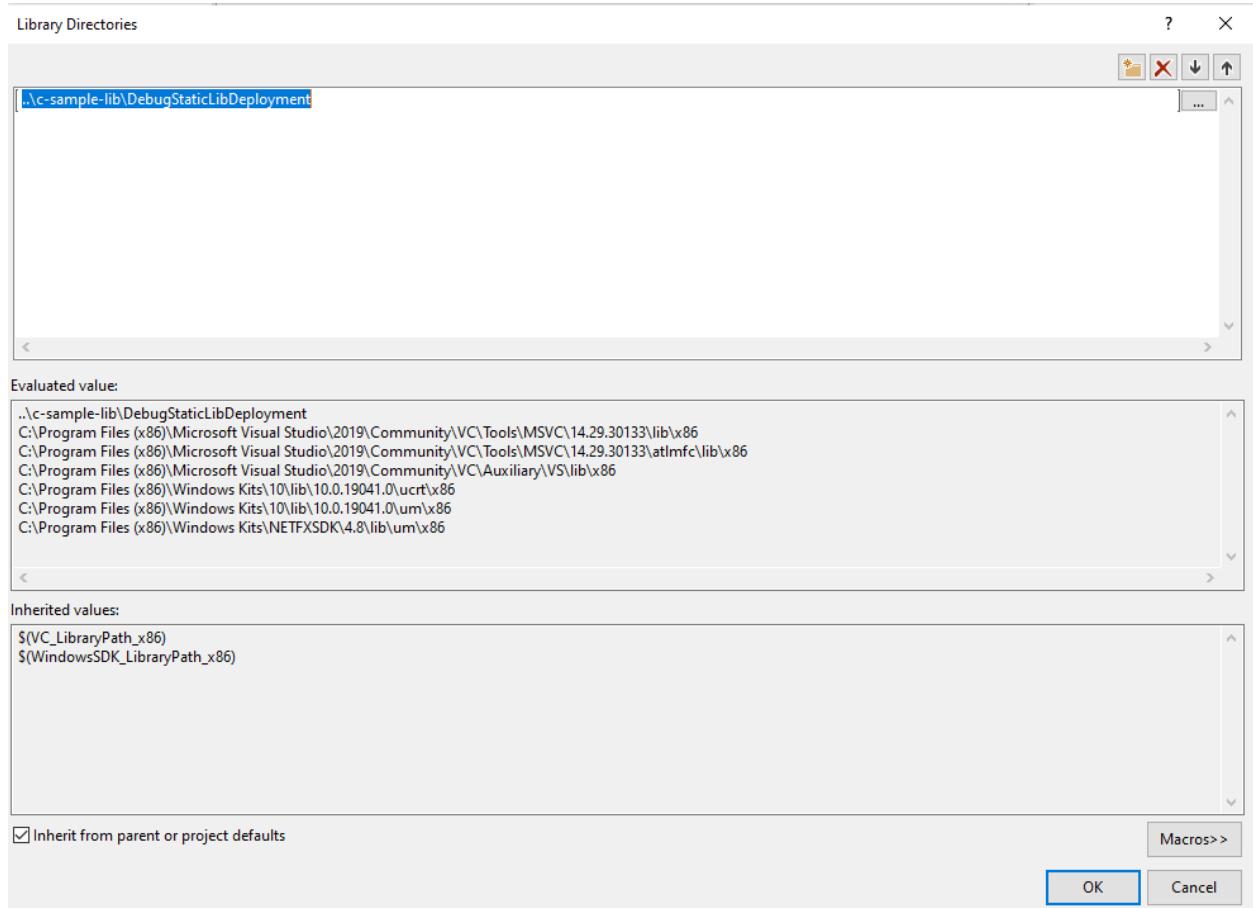
OK Cancel Apply



Not Working

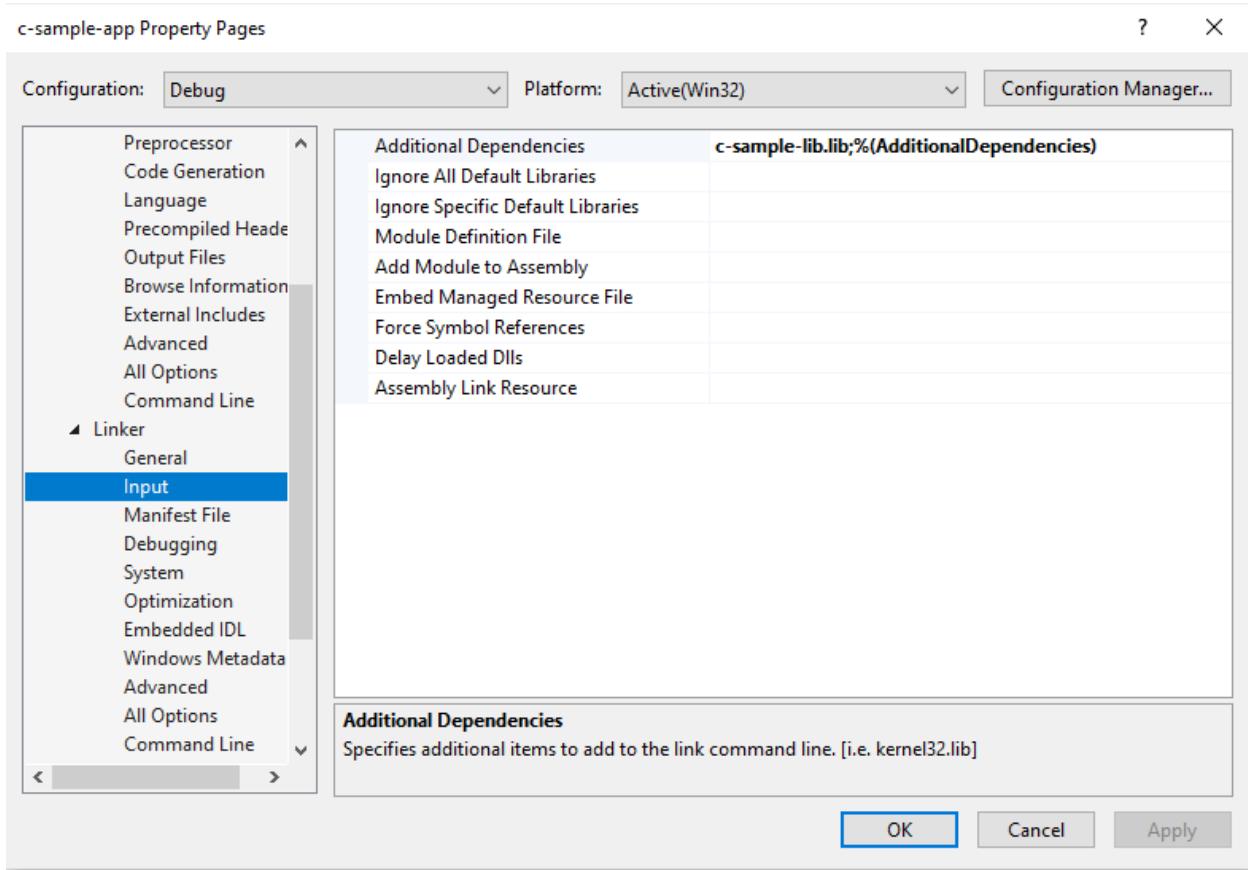
.. \c-sample-lib\DebugStaticLibDeployment





If we set full path for both libraries and headers then we need to set library name for project

Linker->Input->Additional Dependencies



In this case we will compile c-sample-app and we do not need to compile c-sample-lib because we copied output files to different location and they are ready to use.

current source code will be like that nothing changed

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

/// <summary>
///
/// </summary>
/// <returns></returns>
int main()
{
    int result = 0;
    //printf("Hello World!\n");
    result = sum(5, 4);
    sayHelloTo("Computer");
    printf("Result is %d \n",result);
    printf("Press any key to continue...\n");
    getchar();
    return 0;
}
```

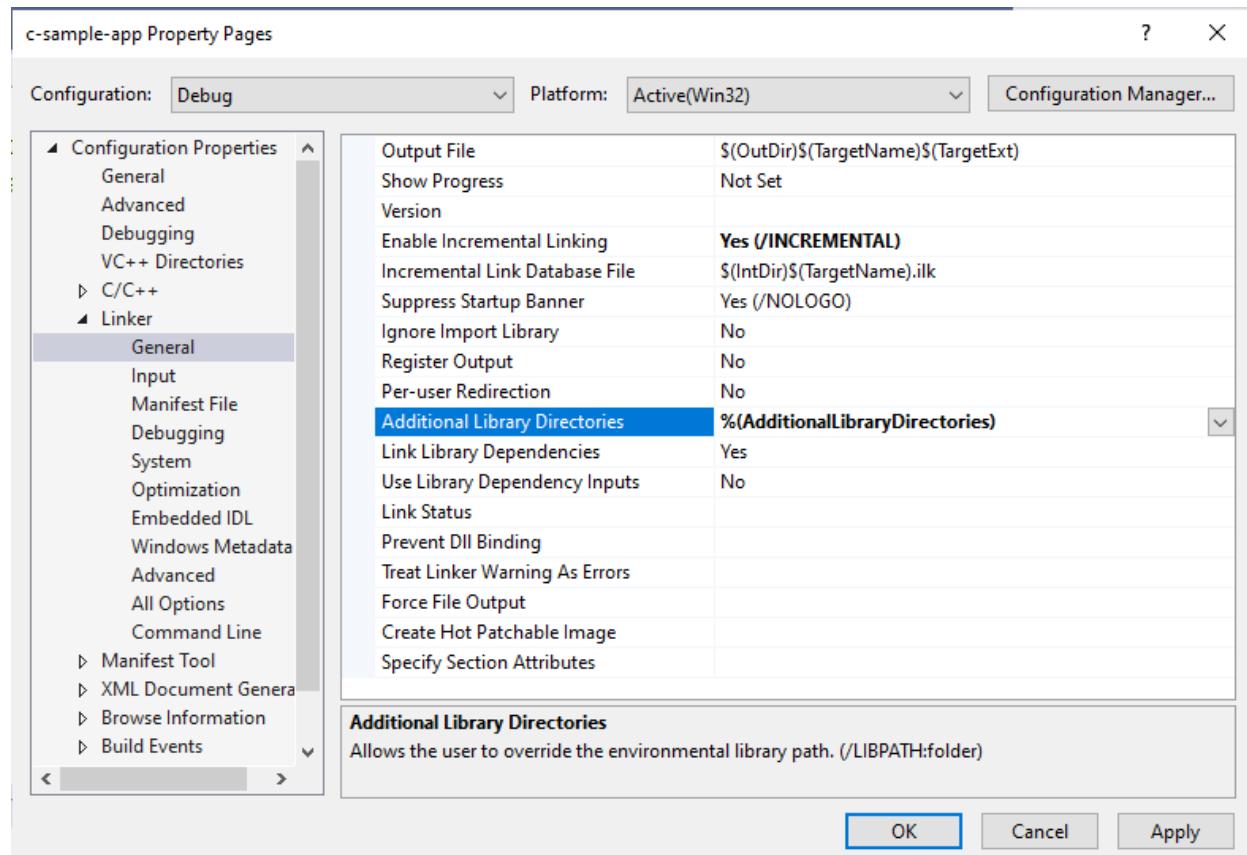
and output

```
E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programmin...  
Hello Computer  
Result is 9  
Press any key to continue...
```

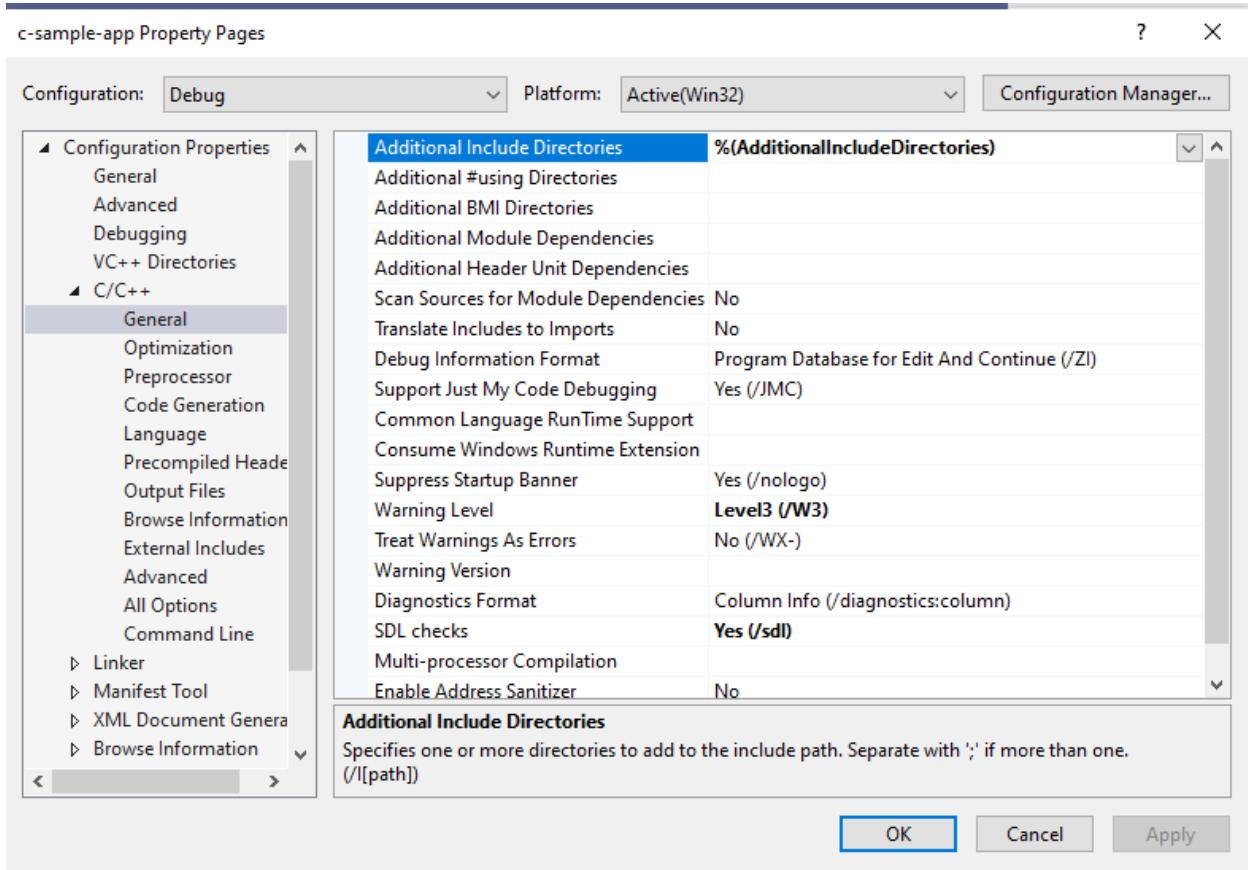
There is a option about portability that we can set for team works

We will remove all library related settings from configurations and we will write them in source code

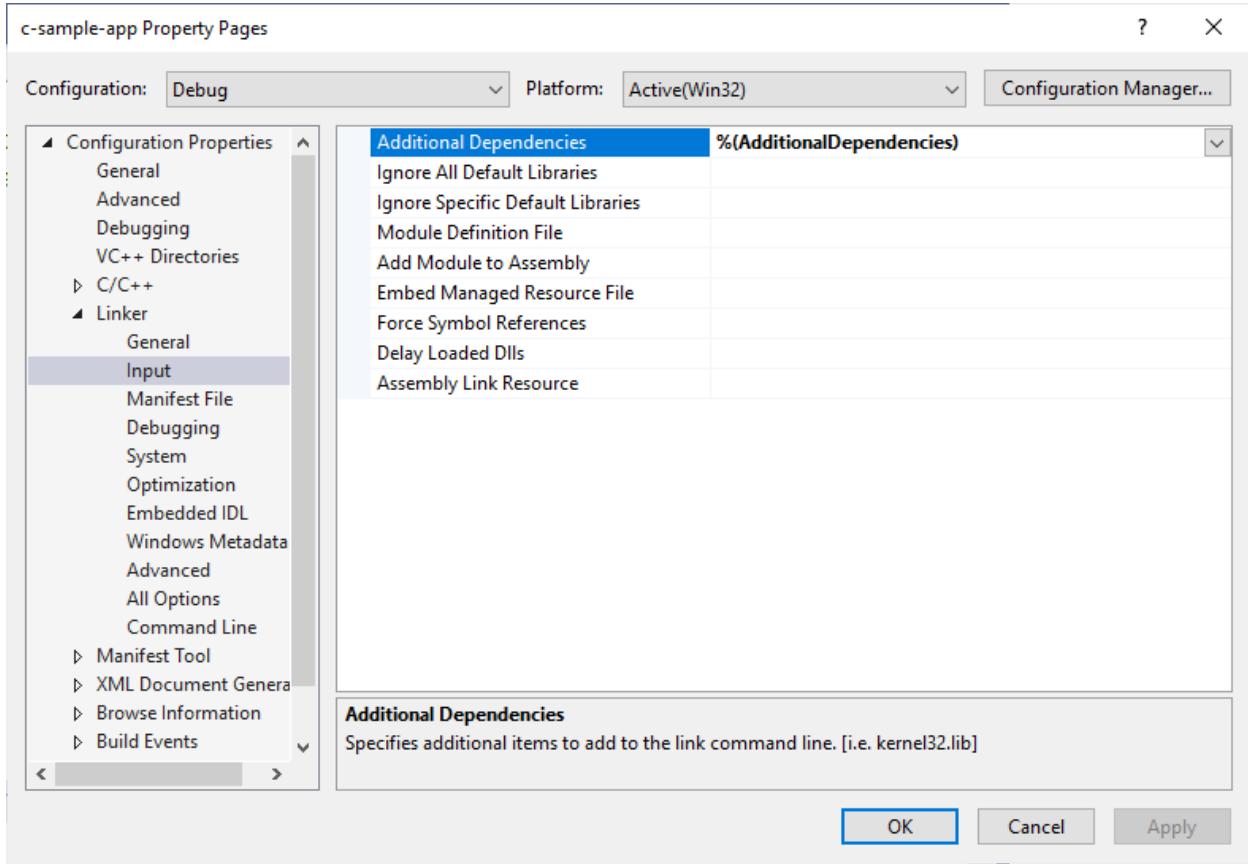
Clear linker->general->additional library directories



Clear C/C++ -> General -> Additional Include Directories



Clear Linker->Input->Additional Dependencies



Now we can set this configurations in source code as follow

```
#pragma comment(lib, "..\\DebugStaticLibDeployment\\c-sample-lib.lib")
#include "..\\DebugStaticLibDeployment\\samplelib.h"

#include <stdio.h>

/// <summary>
///
/// </summary>
/// <returns></returns>
int main()
{
    int result = 0;
    //printf("Hello World!\n");
    result = sum(5, 4);
    sayHelloTo("Computer");
    printf("Result is %d \n",result);
    printf("Press any key to continue...\n");
    getchar();
    return 0;
}
```

with this configuration if your friends download this code then they can run them with their environment without setting a path.

0.4.2 C++ Programming (Static Library)

0.4.2.1 Visual Studio Community Edition All steps are similar with C programming above, but you do not need to delete pch.h

You should take care about compiled source codes

for example if your code is compiled for x86 then your application also should use the x86 configuration else x64 then library should be x64 complied version.

Source will look like the following

```
// cpp-sample-app.cpp : This file contains the 'main' function. Program execution begins and ends there
//  
  
#pragma comment(lib, "..\\DebugStaticLibDeployment\\cpp-sample-lib.lib")  
  
#include "..\DebugStaticLibDeployment\samplelib.h"  
  
#include <iostream>  
  
int main()  
{  
    std::cout << "Hello World!\n";  
  
    int result = 0;  
    //printf("Hello World!\n");  
    result = sum(5, 4);  
    sayHelloTo("Computer");  
    printf("Result is %d \n", result);  
    printf("Press any key to continue...\n");  
    getchar();  
    return 0;  
}
```

0.4.3 C/C++ WSL Option

Install WSL

GitHub - ucoruh/ns3-wsl-win10-setup: ns3 windows 10 WSL2 setup and usage⁵

Create a Linux project

⁵<https://github.com/ucoruh/ns3-wsl-win10-setup>

C++

Linux

All project types



CMake Project

Build modern, cross-platform C++ apps that don't depend on .sln or .vcxproj files.

C++

Windows

Linux

Console



Console Application

Run code in a Linux terminal. Prints "hello" by default.

C++

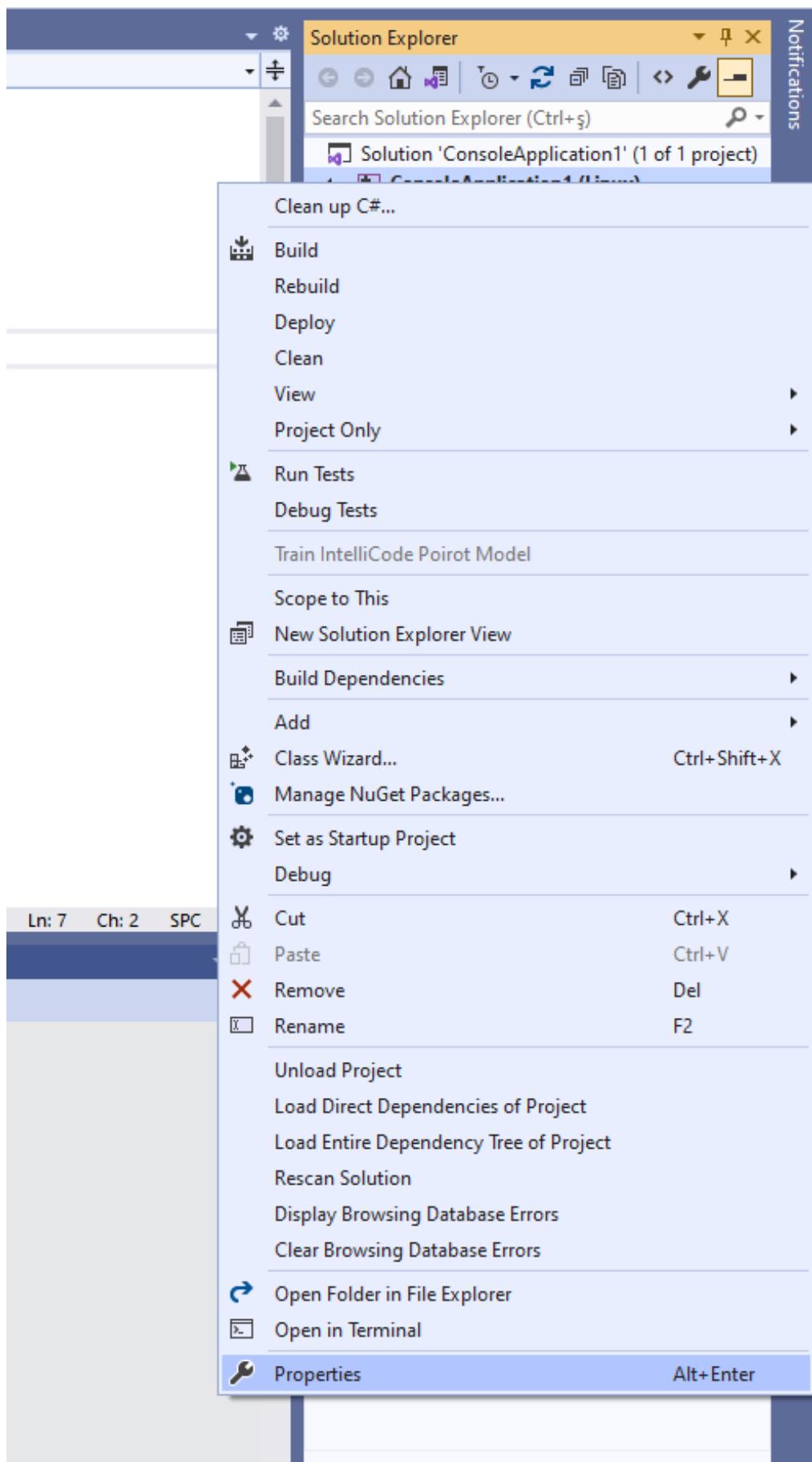
Linux

Console

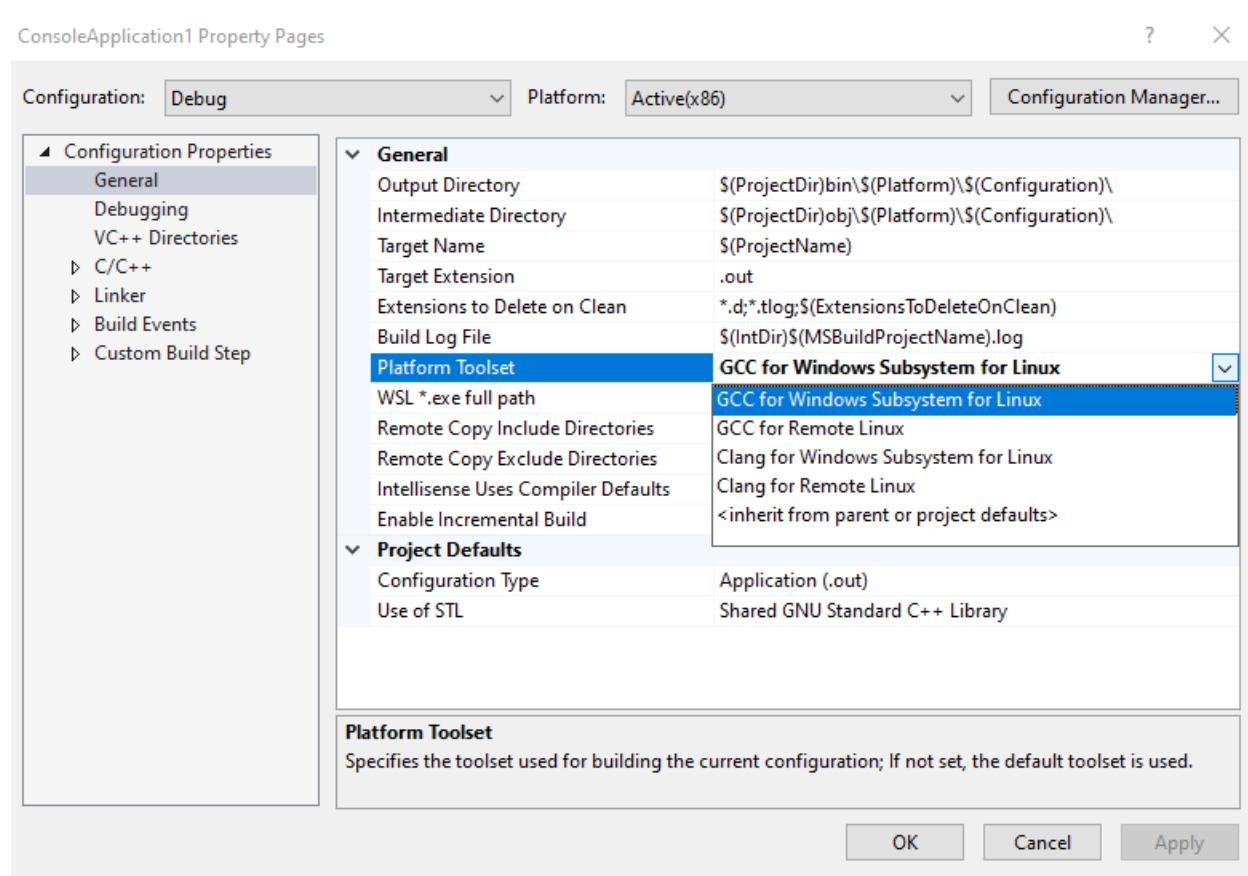


Empty Project

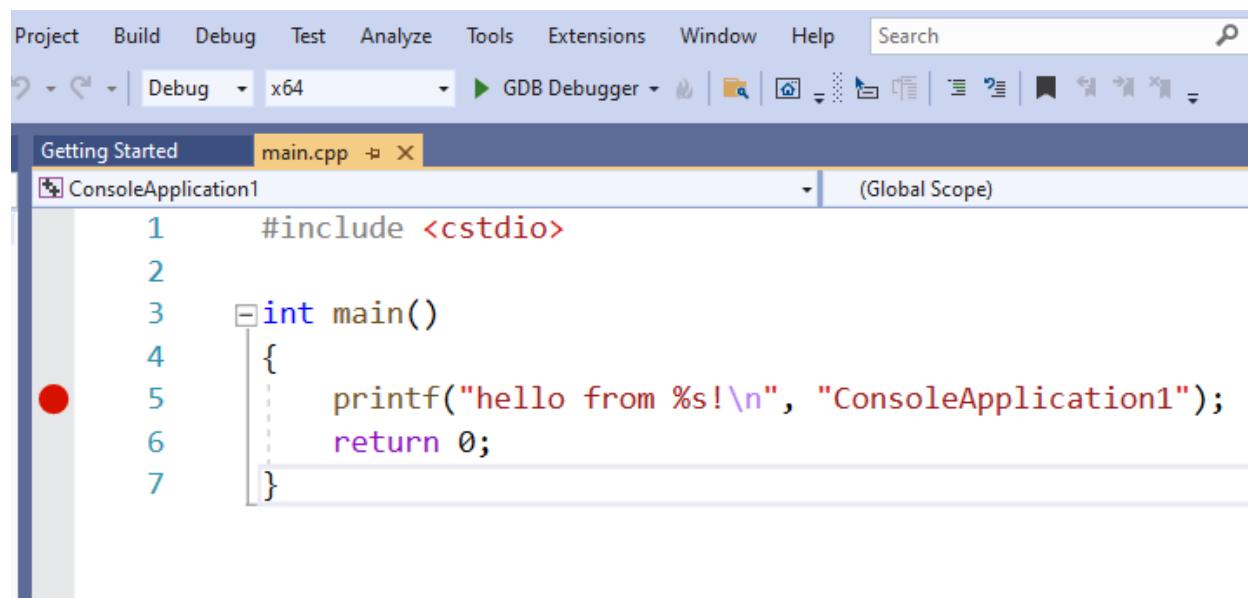
Configure Platform Toolset to WSL



Select GCC for Windows Subsystem for Linux



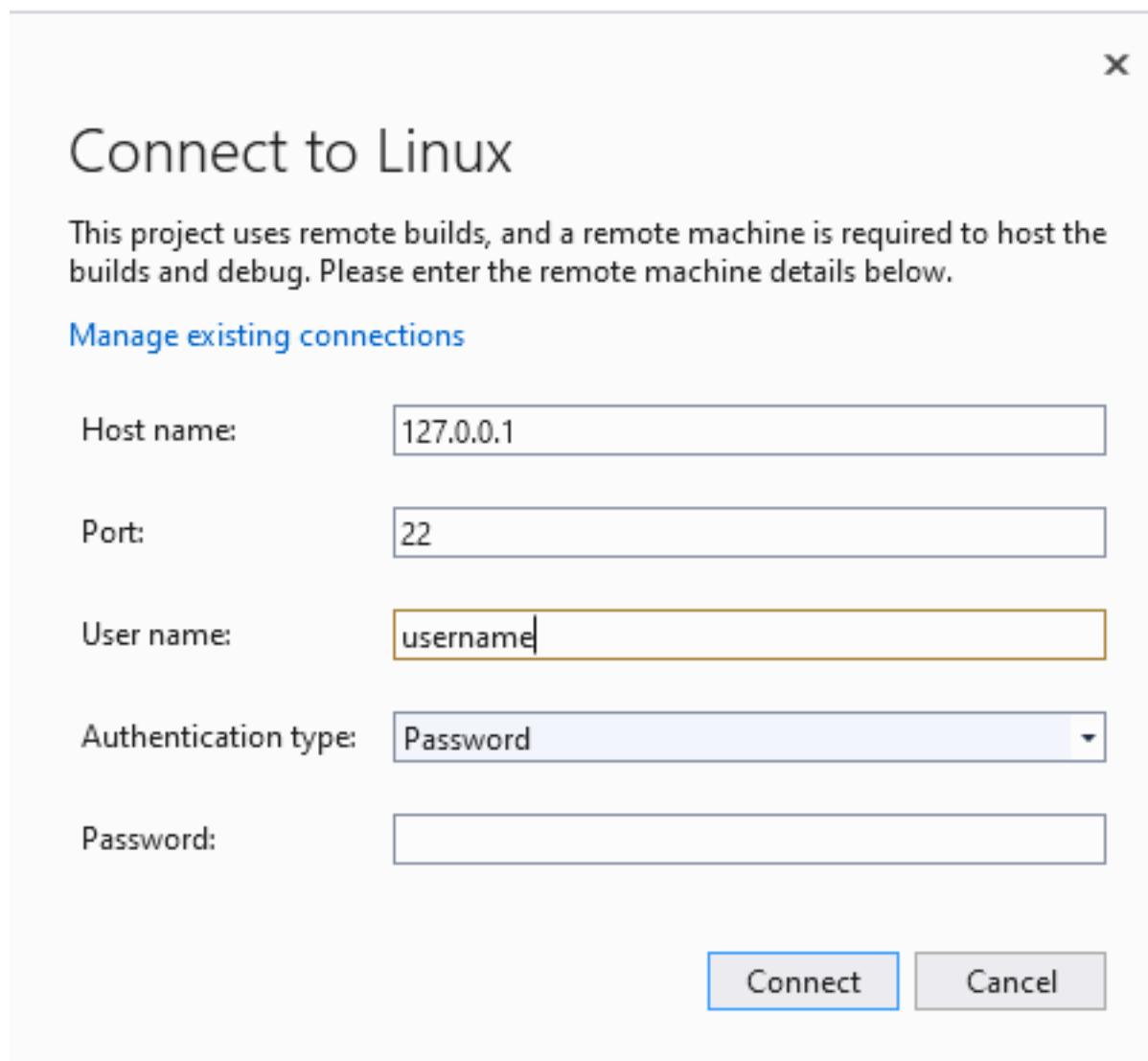
Put a breakpoint and run debugger



In the debugger for WSL you can use local WSL installation but if you want to run it on Release setting it require a SSH connection.

```
#include <cstdio>
```

Configure SSH parameters



so you have to complete the following steps.

0.4.4 C/C++ Remote Linux Option over SSH

Enable SSH

SSH on Windows Subsystem for Linux (WSL) | Illuminia Studios⁶

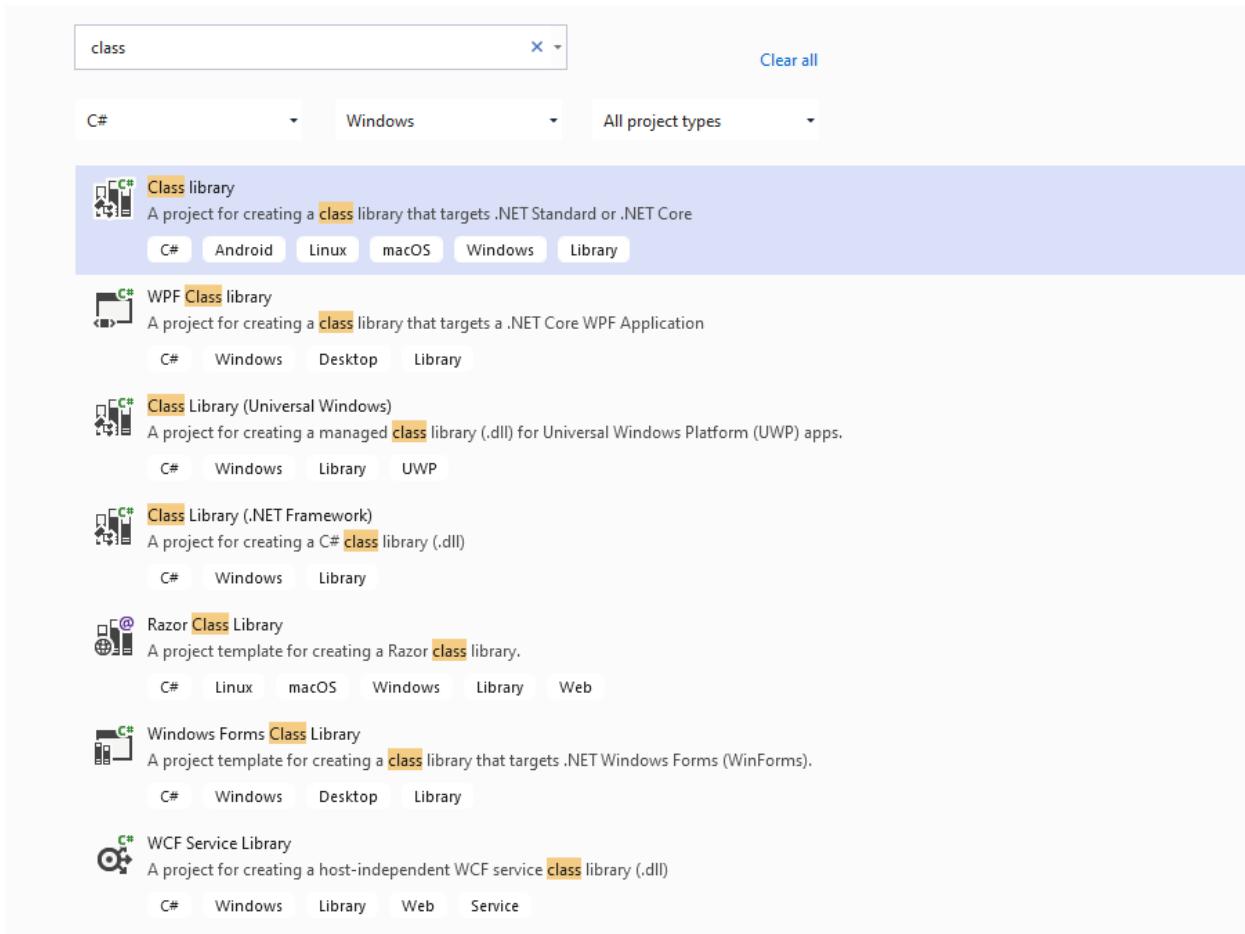
Connect to Remote WSL Environment

⁶<https://www.illuminastudios.com/dev-diaries/ssh-on-windows-subsystem-for-linux/>

0.4.5 C# Programming (Dinamik Library)

0.4.5.1 Visual Studio Community Edition In C# project we will create class library we have several options

for this sample we will select .NET core that we can build cross platform library



There is no static library option

⁷<https://docs.microsoft.com/tr-tr/cpp/linux/connect-to-your-remote-linux-computer?view=msvc-160>

Configure your new project

Class library C# Android Linux macOS Windows Library

Project name

csharp-sample-lib

Location

E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programming \Lectures\ce11

...

Solution name 

csharp-sample-lib

Place solution and project in the same directory

We will select .Net Core 3.1

Additional information

Class library C# Android Linux macOS Windows Library

Target Framework 

.NET Core 3.1 (Long-term support)

.NET Standard 2.0

.NET Standard 2.1

.NET Core 2.1 (Long-term support)

.NET Core 3.1 (Long-term support)

.NET 5.0 (Current)

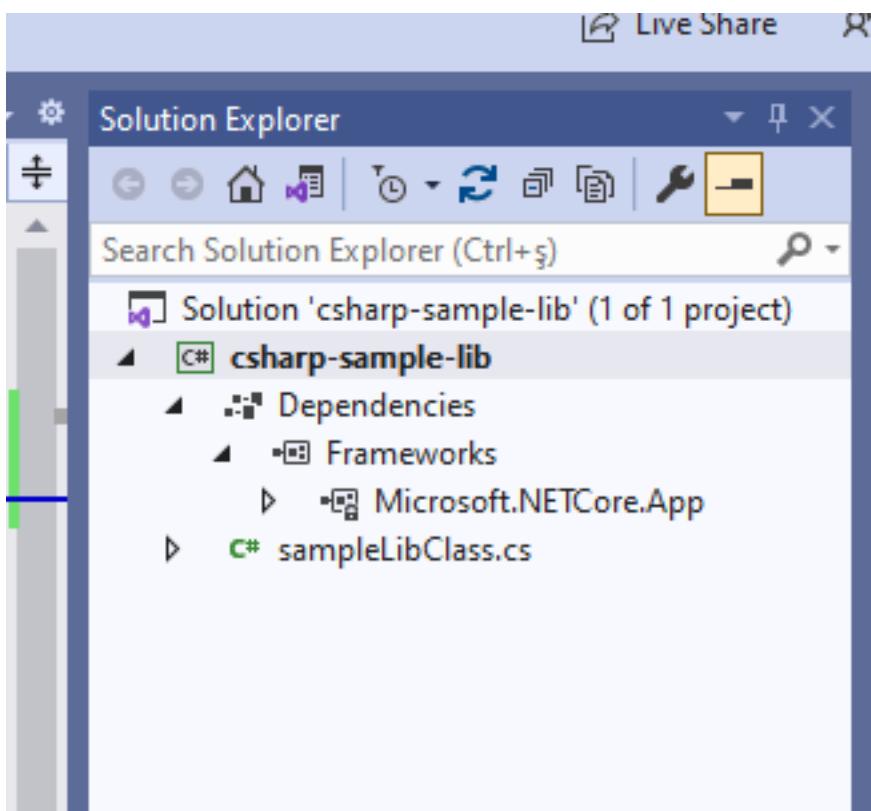
You will have default empty class library file

The screenshot shows a code editor window with the file `sampleLibClass.cs` open. The code contains a single class definition:

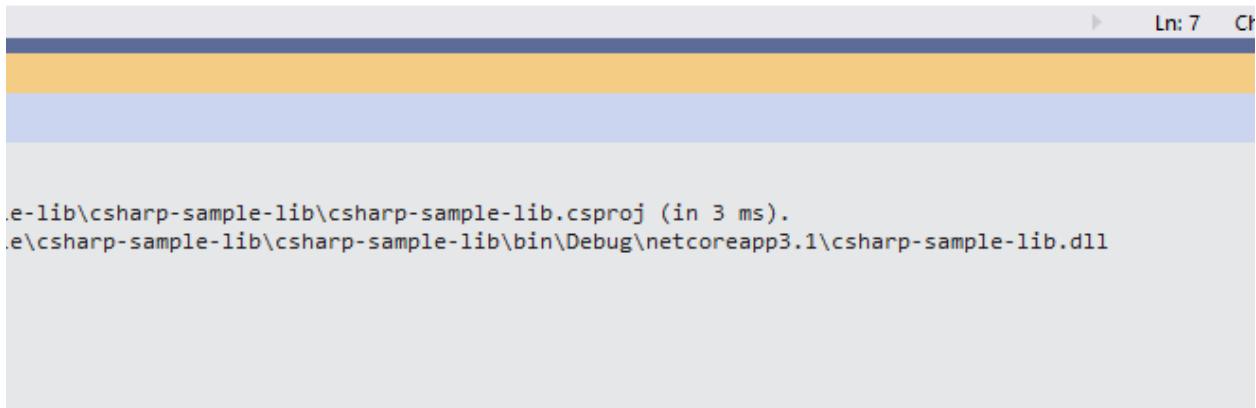
```
1  using System;
2
3  namespace csharp_sample_lib
4  {
5      public class sampleLibClass
6      {
7          ...
8      }
9  }
```

A green vertical bar highlights the opening brace of the `sampleLibClass` constructor, indicating a syntax error or warning.

In the project you can see .NETcore reference



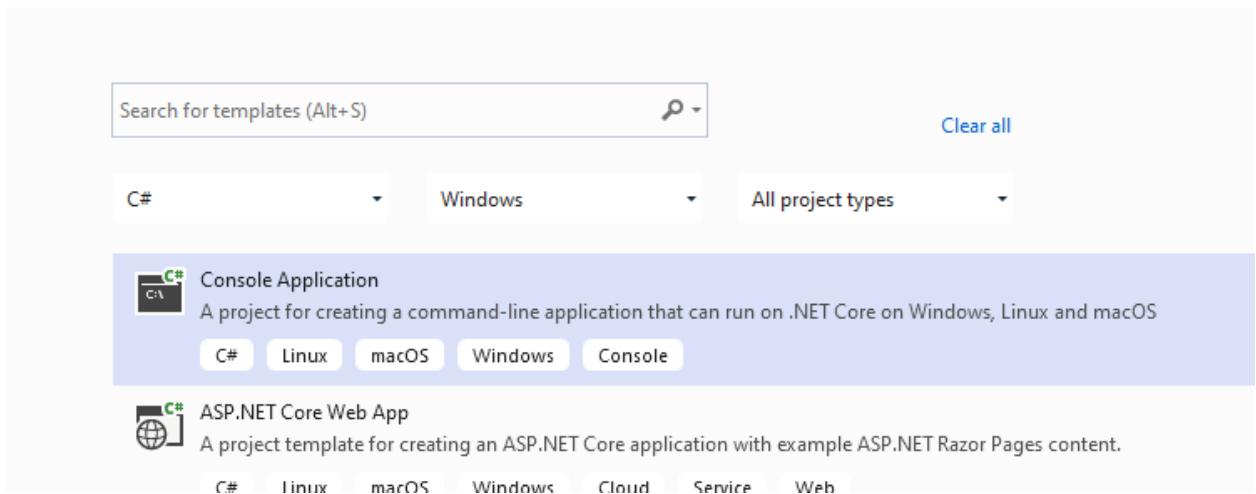
We can build empty class library that generate dll for our application



```
.\e-lib\csharp-sample-lib\csharp-sample-lib.csproj (in 3 ms).
.\e\csharp-sample-lib\csharp-sample-lib\bin\Debug\netcoreapp3.1\csharp-sample-lib.dll
```

Now we will add Console Application but this will also use .NETCore

Select New Project



Name the project

Configure your new project

Console Application C# Linux macOS Windows Console

Project name

csharp-sample-app

Location

C:\Users\ugur.coruh\Desktop\csharp-lib-sample\csharp-sample-lib\

...

Select .NETCore framework

Additional information

Console Application C# Linux macOS Windows Console

Target Framework ⓘ

.NET Core 3.1 (Long-term support)

.NET Core 2.1 (Long-term support)

.NET Core 3.1 (Long-term support)

.NET 5.0 (Current)

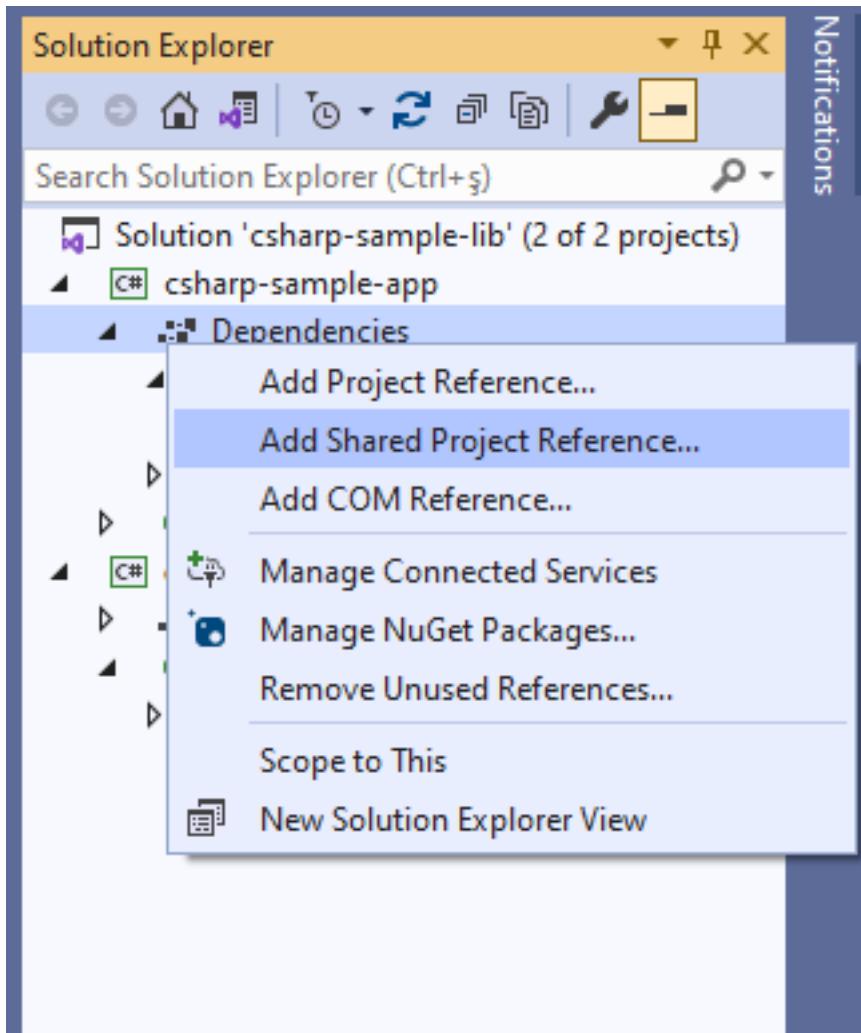
You will have the following sample main.cs file

```
using System;

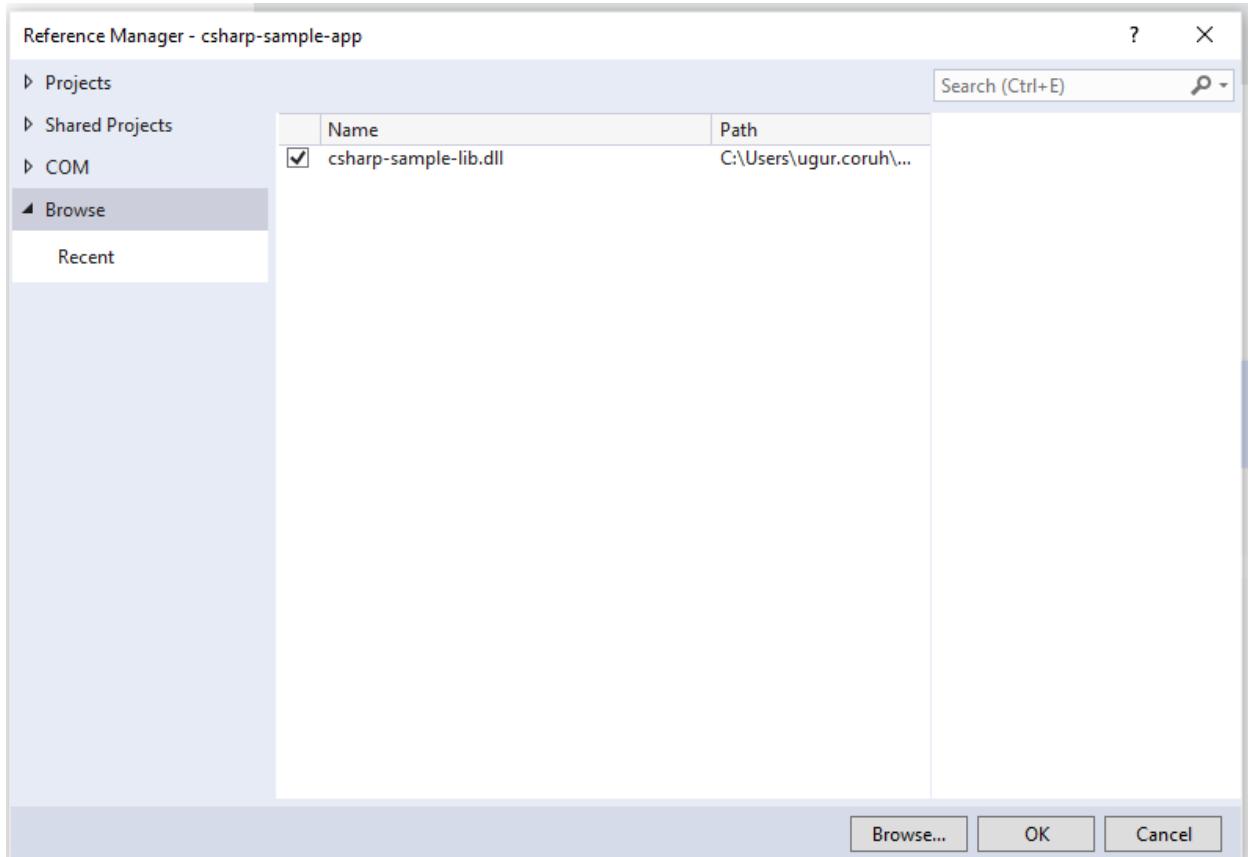
namespace csharp_sample_app
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
}
```

```
    }  
}
```

Now we can link projects with adding references open reference section



browse for class library project output folder and select output dll file for console application



now we can update our library code and use it in console application

copy following sample to sampleLibClass file in the library

```
using System;

namespace csharp_sample_lib
{
    public class sampleLibClass
    {
        public static void sayHelloTo(string name)
        {
            if (!String.IsNullOrEmpty(name))
            {
                Console.WriteLine("Hello " + name);
            }
            else
            {
                Console.WriteLine("Hello There");
            }
        }

        public static int sum(int a, int b)
        {
            int c = 0;
            c = a + b;
            return c;
        }
    }
}
```

```
    }
}
```

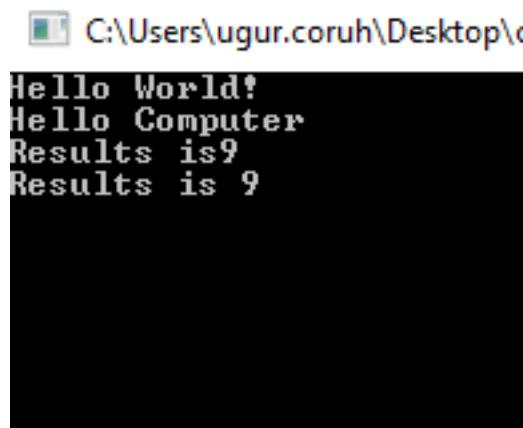
after this operation copy following sample to console application and build app then you can run

```
using csharp_sample_lib;
using System;

namespace csharp_sample_app
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");

            sampleLibClass.sayHelloTo("Computer");
            int result = sampleLibClass.sum(5, 4);
            Console.WriteLine("Results is" + result);
            Console.WriteLine("Results is {0}", result);
            Console.Read();
        }
    }
}
```

You will see following output that mean we called DLL functions



```
C:\Users\ugur.coruh\Desktop\c
Hello World!
Hello Computer
Results is9
Results is 9
```

Also we can publish this console application with dll for linux environment or others
for linux environment we should install .NETCore

follow the link below or commands that I shared with you as below for deployment

How to Install Dotnet Core on Ubuntu 20.04 – TecAdmin⁸

Step 1 – Enable Microsoft PPA

```
wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb
```

⁸<https://tecatmin.net/how-to-install-net-core-on-ubuntu-20-04/>

Step 2 – Installing Dotnet Core SDK

```
sudo apt update  
sudo apt install apt-transport-https  
sudo apt install dotnet-sdk-3.1
```

Step 3 – Install Dotnet Core Runtime Only

To install .NET Core Runtime on Ubuntu 20.04 LTS system, execute the commands:

```
sudo apt update
```

To install the previous version of .Net core runtime 2.1, type:

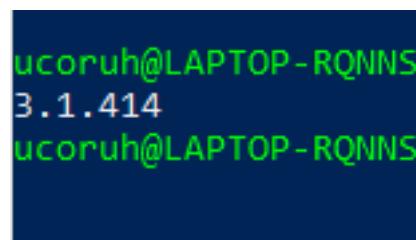
```
sudo apt install dotnet-runtime-2.1
```

Press “y” for any input prompted by the installer.

Step 4 – (Optional) Check .NET Core Version

You can use dotnet command line utility to check installed version of .NET Core on your system. To check dotnet version, type:

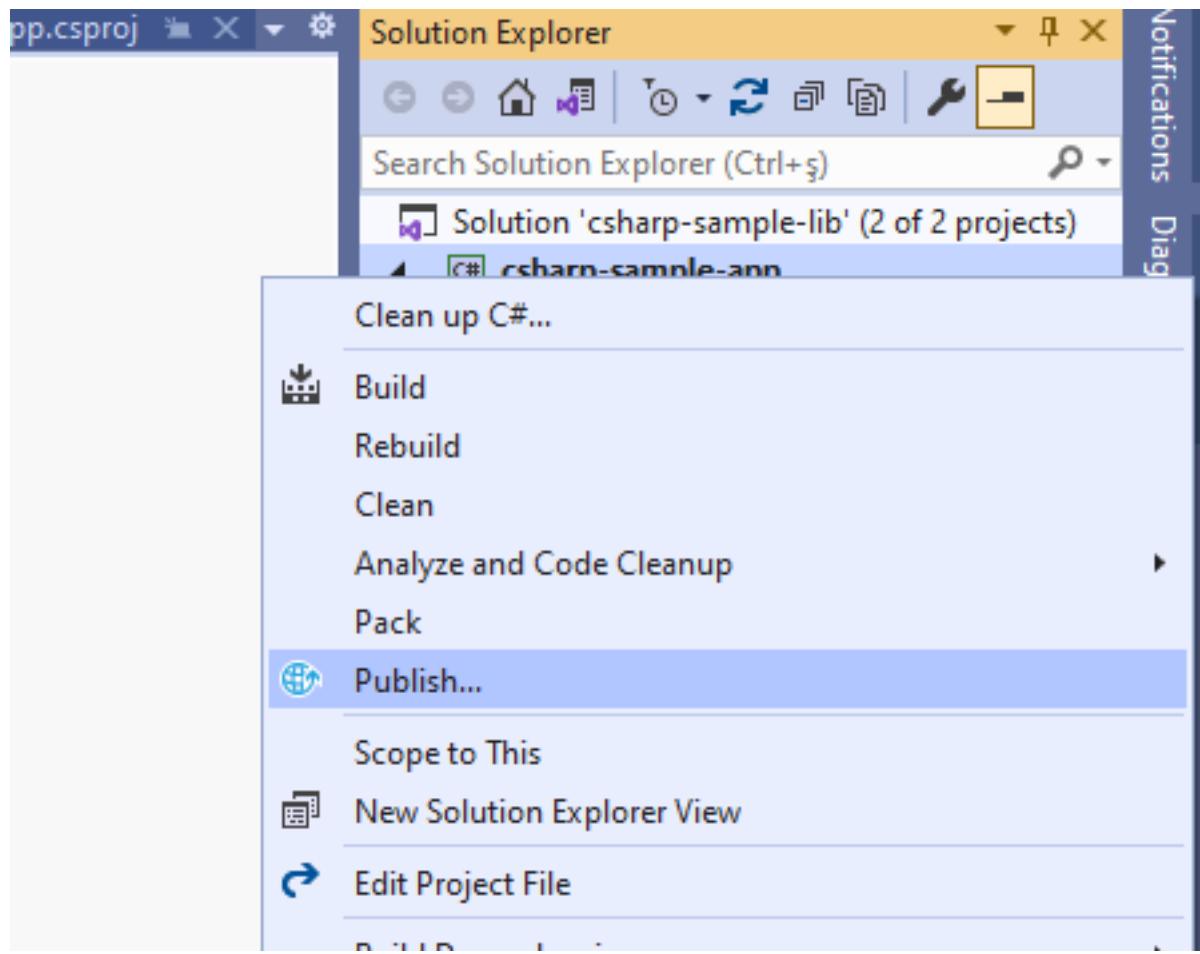
```
dotnet --version
```

A terminal window with a dark blue background and white text. It displays the command "dotnet --version" and its output: "ucoruh@LAPTOP-RQNNNS 3.1.414 ucoruh@LAPTOP-RQNNNS".

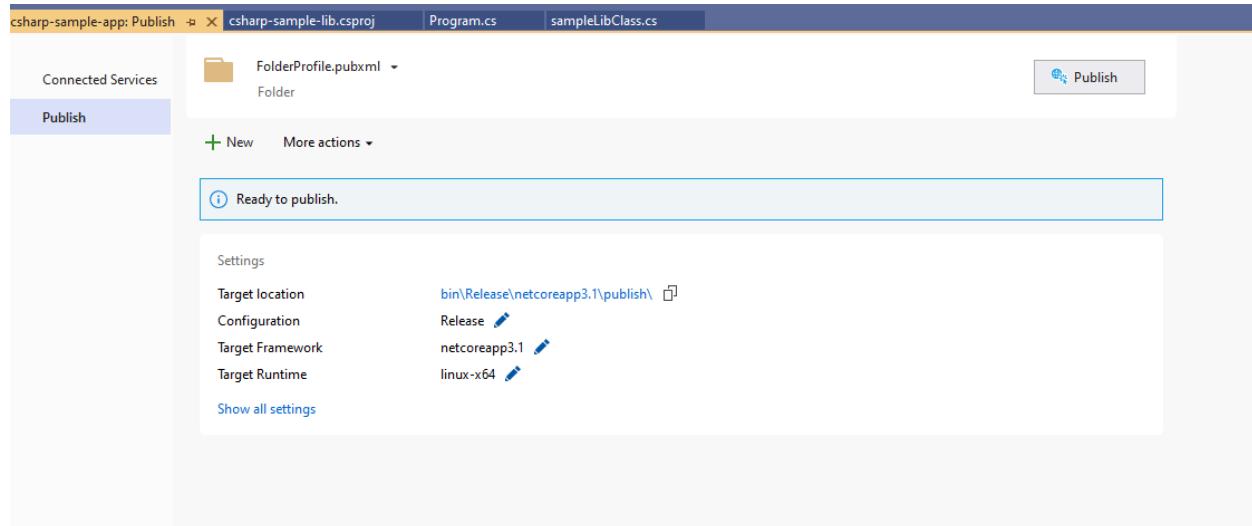
```
ucoruh@LAPTOP-RQNNNS  
3.1.414  
ucoruh@LAPTOP-RQNNNS
```

Now we will publish our application as single executable

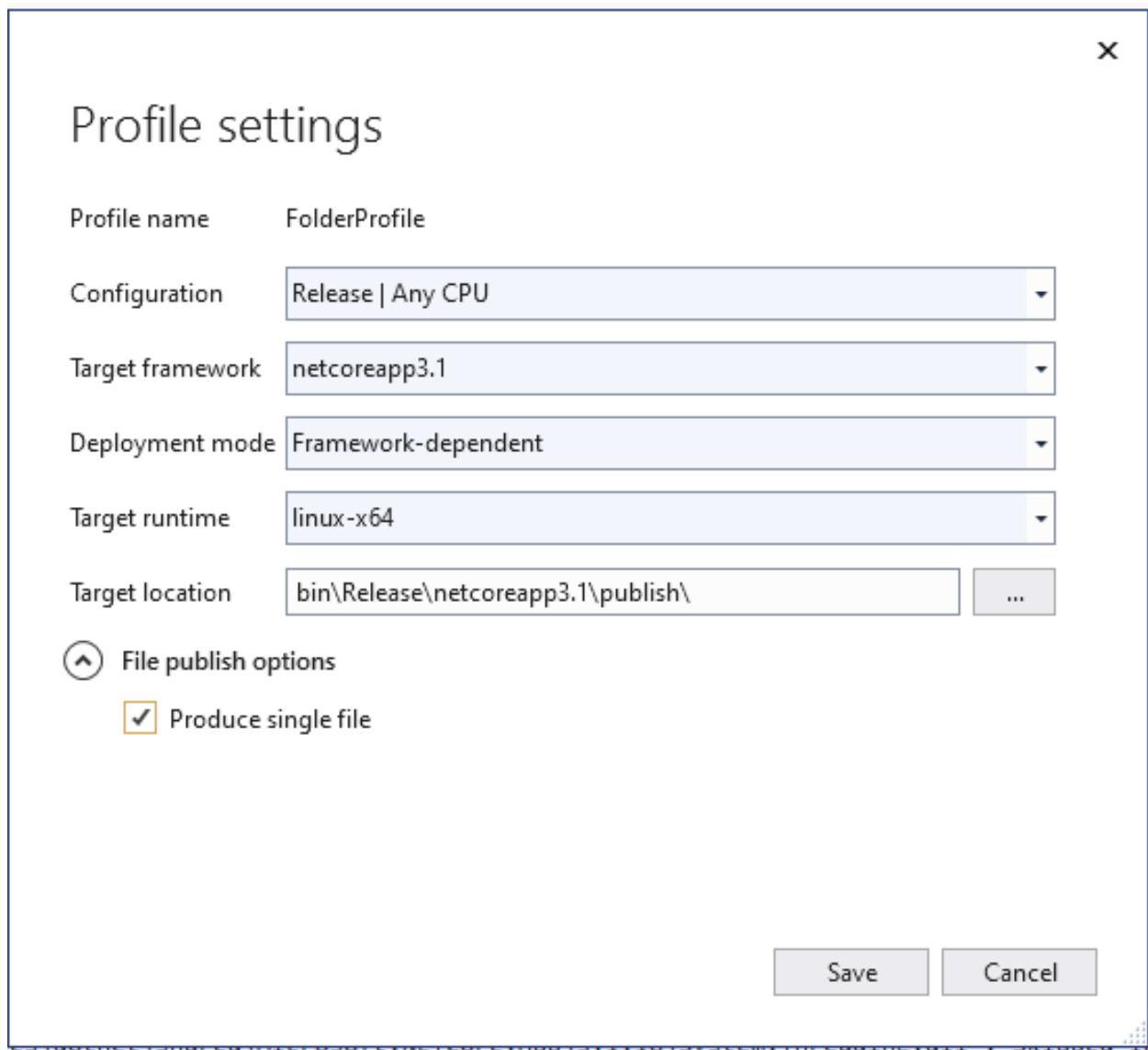
Open publish menu



Select netcoreapp3.1 and Release for linux-x64



Select produce single file



After successful publish you will have linux binary that you can run with WSL

Desktop > csharp-lib-sample > csharp-sample-lib > csharp-sample-app > bin > Release > netcoreapp3.1 > publish

Name	Date modified	Type	Size
csharp-sample-app	10/24/2021 1:36 AM	File	97 KB
csharp-sample-app.pdb	10/24/2021 1:36 AM	Program Debug D...	10 KB
csharp-sample-lib.pdb	10/24/2021 1:30 AM	Program Debug D...	10 KB
packages-microsoft-prod.deb	4/23/2020 10:02 PM	DEB File	4 KB

Open WSL and enter the path where this folder located
and run application as follow

```
Processing triggers for man-db (2.9.1-1) ...
ucoruh@LAPTOP-RQNN9IG:/mnt/c/Users/ugur/Desktop/csharp-lib-sample/csharp-sample-lib/csharp-sample-app/bin/Release/netcoreapp3.1/publish$ dotnet --version
3.1.414
ucoruh@LAPTOP-RQNN9IG:/mnt/c/Users/ugur/Desktop/csharp-lib-sample/csharp-sample-app/bin/Release/netcoreapp3.1/publish$ ./csharp-sample-app
csharp-sample-app      csharp-sample-app.pdb      csharp-sample-lib.lib      packages-microsoft-prod.deb
ucoruh@LAPTOP-RQNN9IG:/mnt/c/Users/ugur/Desktop/csharp-lib-sample/csharp-sample-app/bin/Release/netcoreapp3.1/publish$ ./csharp-sample-app
Hello World!
Hello Computer
Results is9
Results is 9
```

check dotnet –version and then run application

```
publish$ dotnet --version
publish$ ./
publish$ ./csharp-sample-app
```

you will see similar output

```
ucoruh@LAPTOP-RQNN9IG:/mnt/c/
csharp-sample-app
ucoruh@LAPTOP-RQNN9IG:/mnt/c/
Hello World!
Hello Computer
Results is9
Results is 9
```

In this sample we created single application from settings lets try with shared library located option uncheck the “produce single file” option and publish again.

Then you will have the following outputs

Name	Date modified	Type	Size
csharp-sample-app	10/24/2021 1:36 AM	File	88 KB
csharp-sample-app.deps.json	10/24/2021 1:36 AM	JSON File	1 KB
csharp-sample-app.dll	10/24/2021 1:36 AM	Application exten...	4 KB
csharp-sample-app.pdb	10/24/2021 1:36 AM	Program Debug D...	10 KB
csharp-sample-app.runtimeconfig.json	10/24/2021 1:36 AM	JSON File	1 KB
csharp-sample-lib.dll	10/24/2021 1:30 AM	Application exten...	4 KB
csharp-sample-lib.pdb	10/24/2021 1:30 AM	Program Debug D...	10 KB

If you run csharp-sample-app
you will have the same output

```
ucoruh@LAPTOP-RQNN5:~$ Hello World!
Hello Computer
Results is 9
Results is 9
```

0.4.6 Java Programming

0.4.6.1 Eclipse IDE You should download and install eclipse installer and then you should select Eclipse IDE for Java Developers

Eclipse Installer 2021-09 R | Eclipse Packages⁹

⁹<https://www.eclipse.org/downloads/packages/installer>



eclipse**installer**
by Oomph

eclipseinstaller

by Oomph



type filter text



Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration



Eclipse IDE for Enterprise Java and Web Developers

Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web...



Eclipse IDE for C/C++ Developers

An IDE for C/C++ developers.



Eclipse IDE for Embedded C/C++ Developers

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),...



Eclipse IDE for PHP Developers

The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, TypeScript, HTML, CSS and XML.

[Click here to...](#)

[★ DONATE](#)

eclipseinstaller

by Oomph



Eclipse IDE for Java Developers

[details](#)

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration.

Java 11+ VM

C:\Program Files\Java\jdk-16.0.1

**Installation Folder**

C:\Users\ugur.coruh\eclipse\java-2021-09

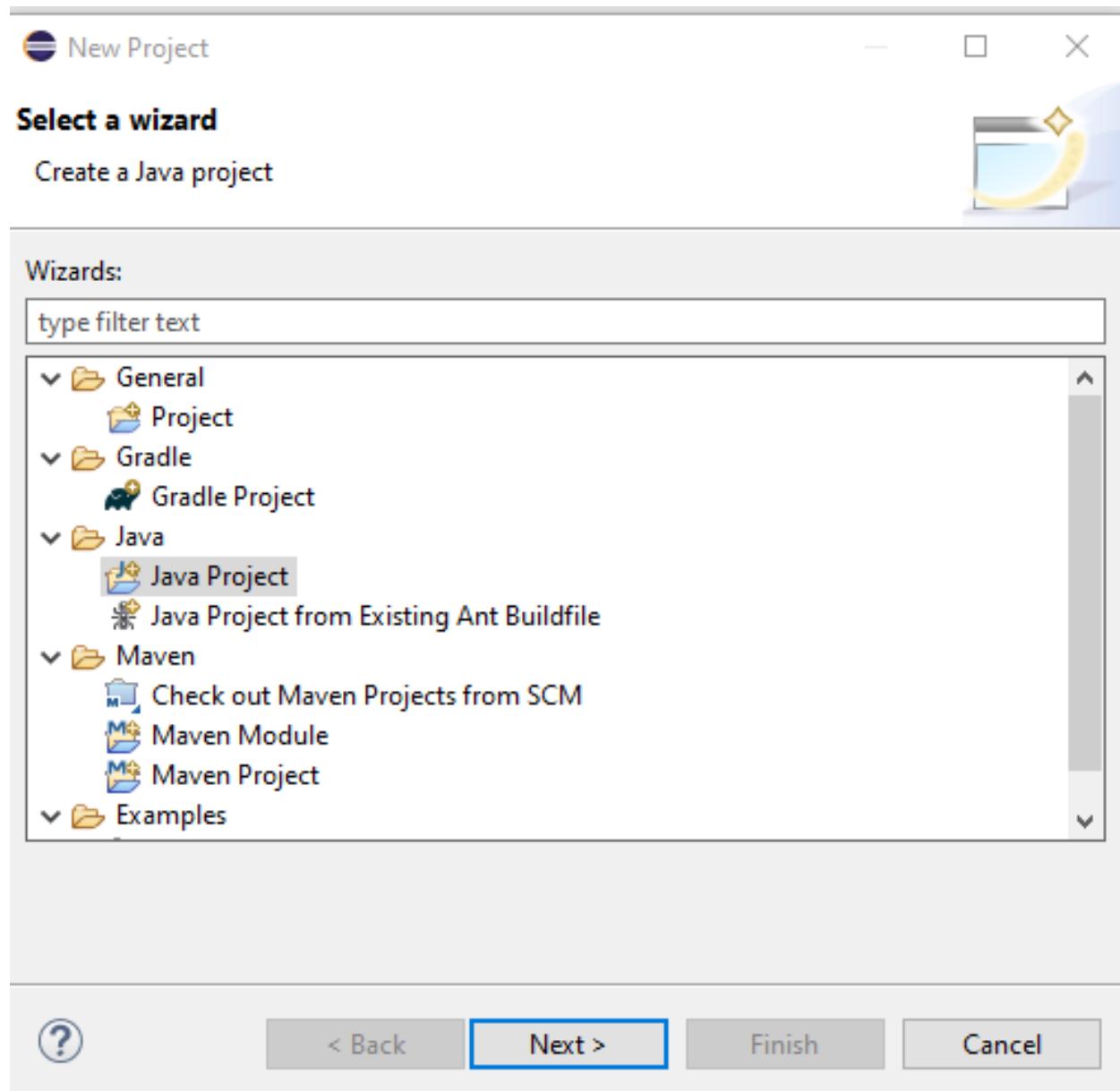
 **create start menu entry** **create desktop shortcut** **INSTALLING** **Cancel Installation** **BACK**



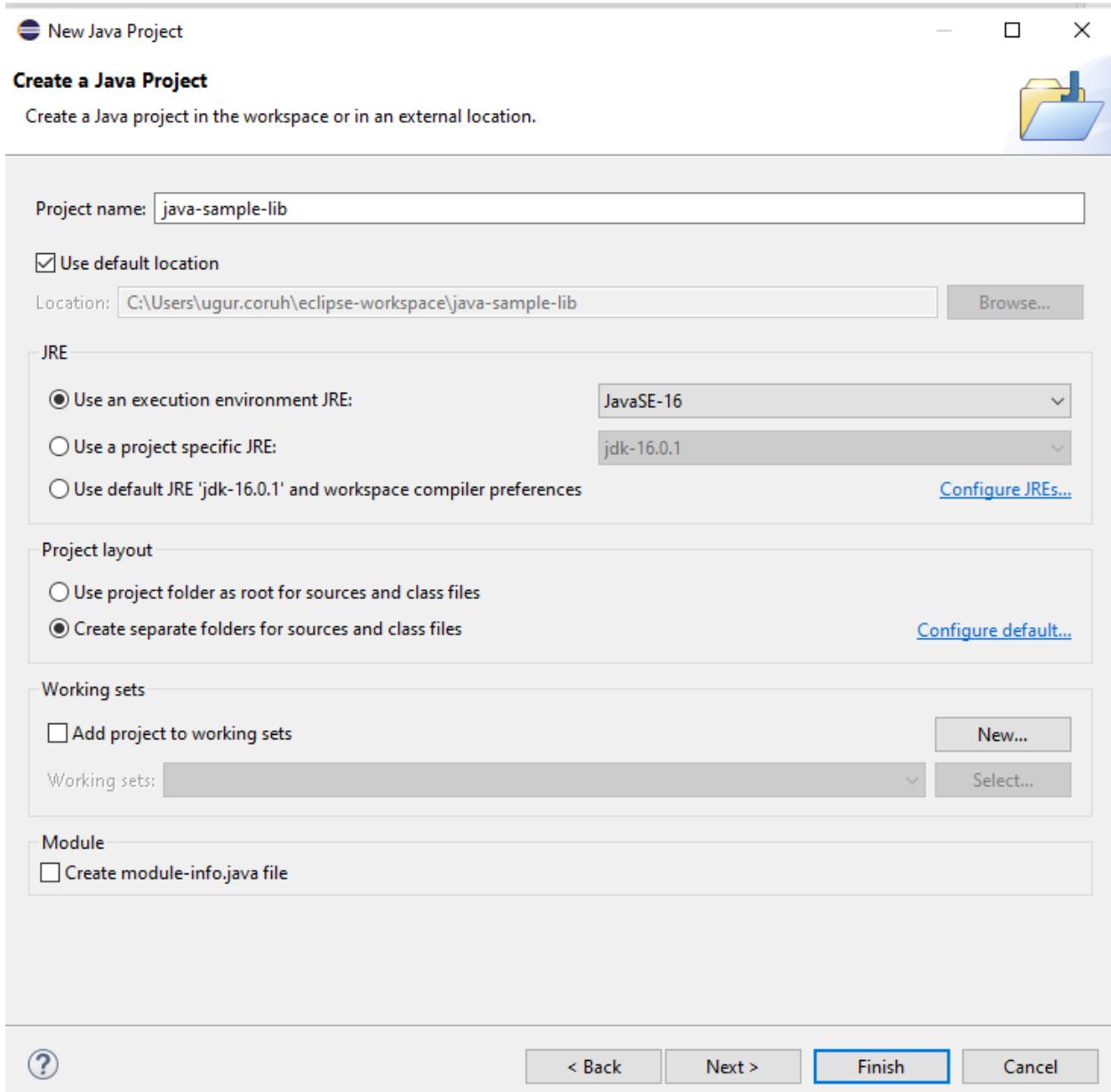
select create a project



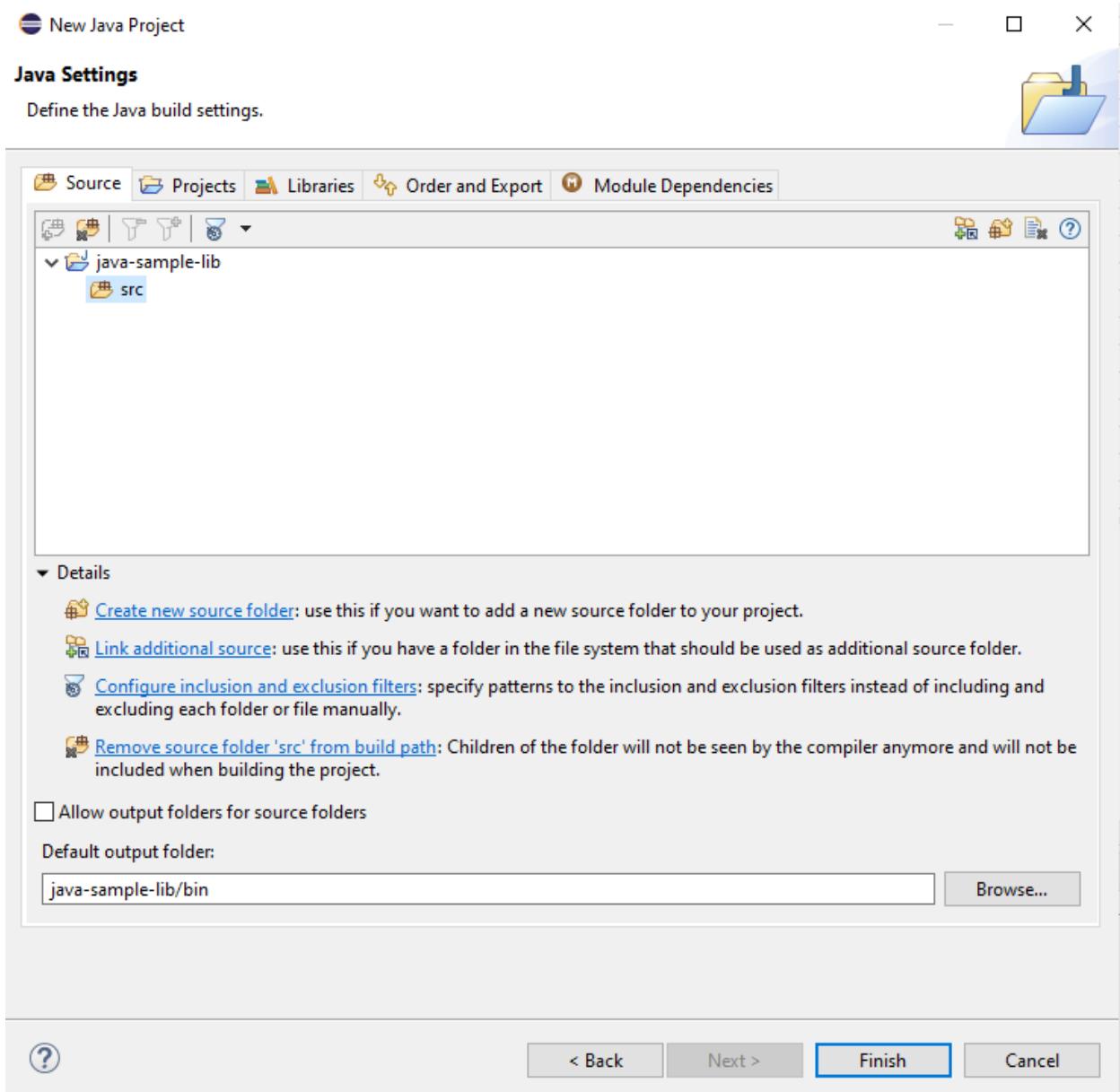
select java project



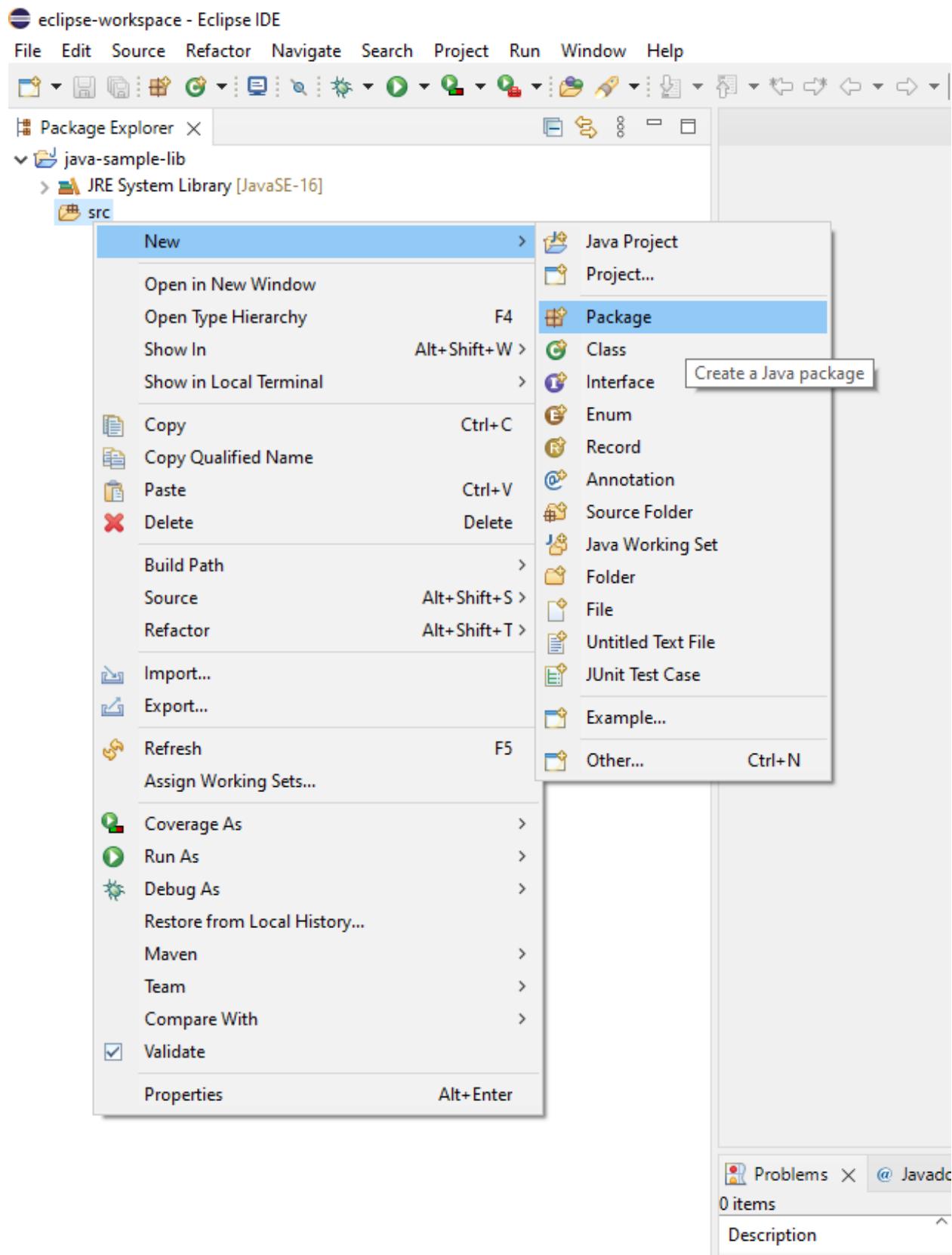
give project name



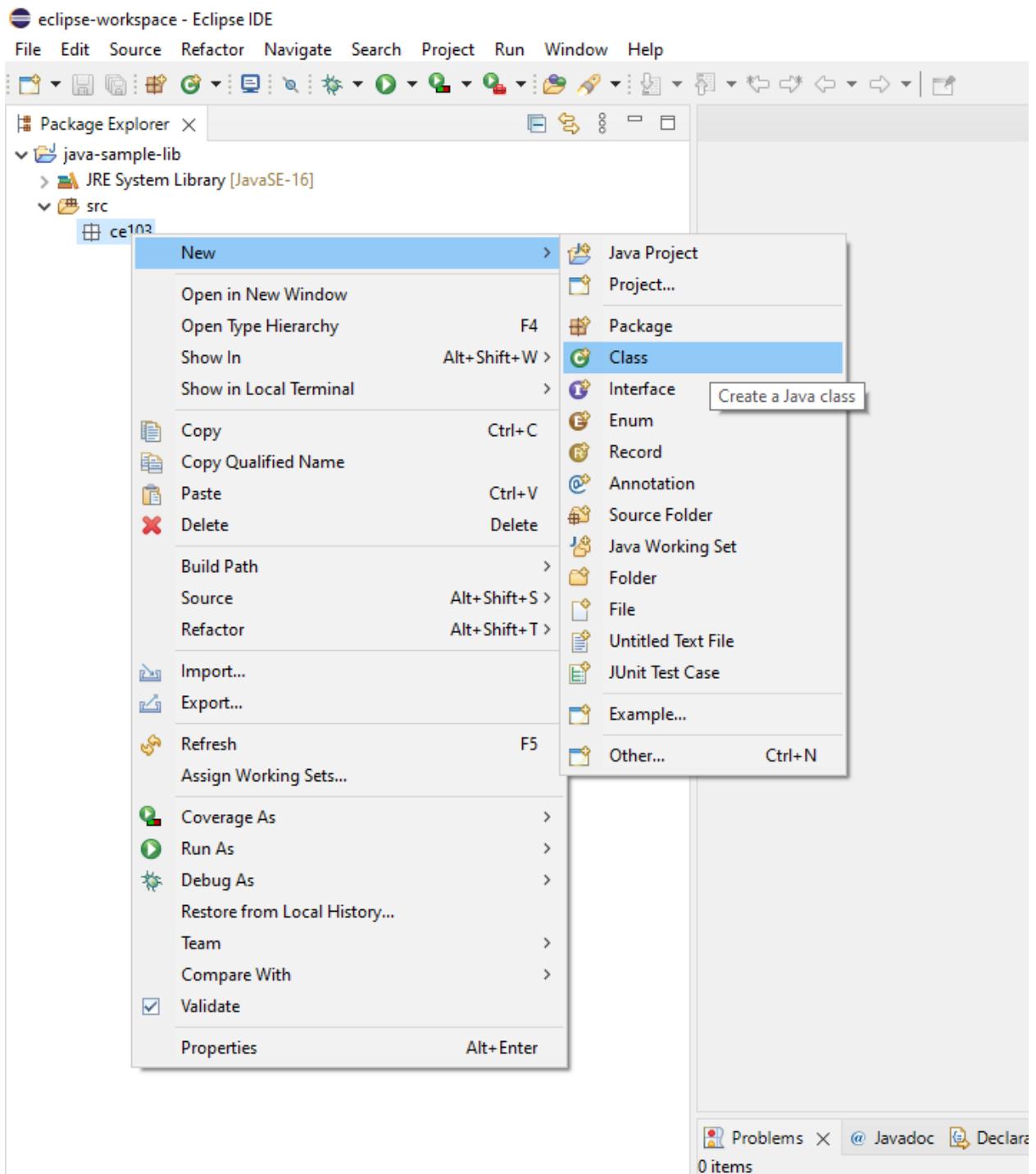
select finish



first we need to add a default package to keep everything organized



then we can create our class that includes our functions



give class a name

 New Java Class

Java Class

Create a new Java class.



Source folder:

Package:

Enclosing type:

Name:

Modifiers: public package private protected
 abstract final static

Superclass:

Interfaces:

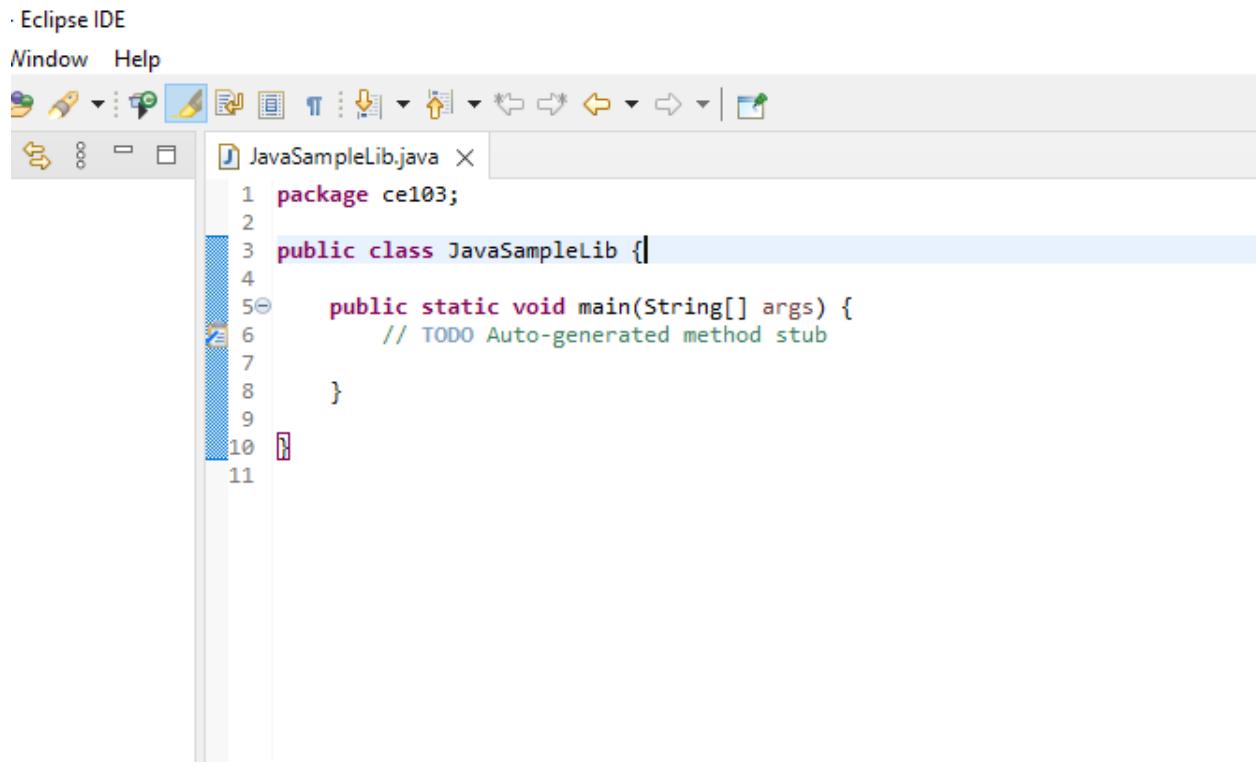
Which method stubs would you like to create?

public static void main(String[] args)
 Constructors from superclass
 Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

you will have following class with main



We will create sample java library with static functions as below.

```
package ce103;

import java.io.IOException;

public class JavaSampleLib {

    public static void sayHelloTo(String name) {
        if(name.isBlank() || name.isEmpty())
        {
            System.out.println("Hello "+name);
        }else {
            System.out.println("Hello There");
        }
    }

    public static int sum(int a,int b)
    {
        int c = 0;
        c = a+b;
        return c;
    }

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

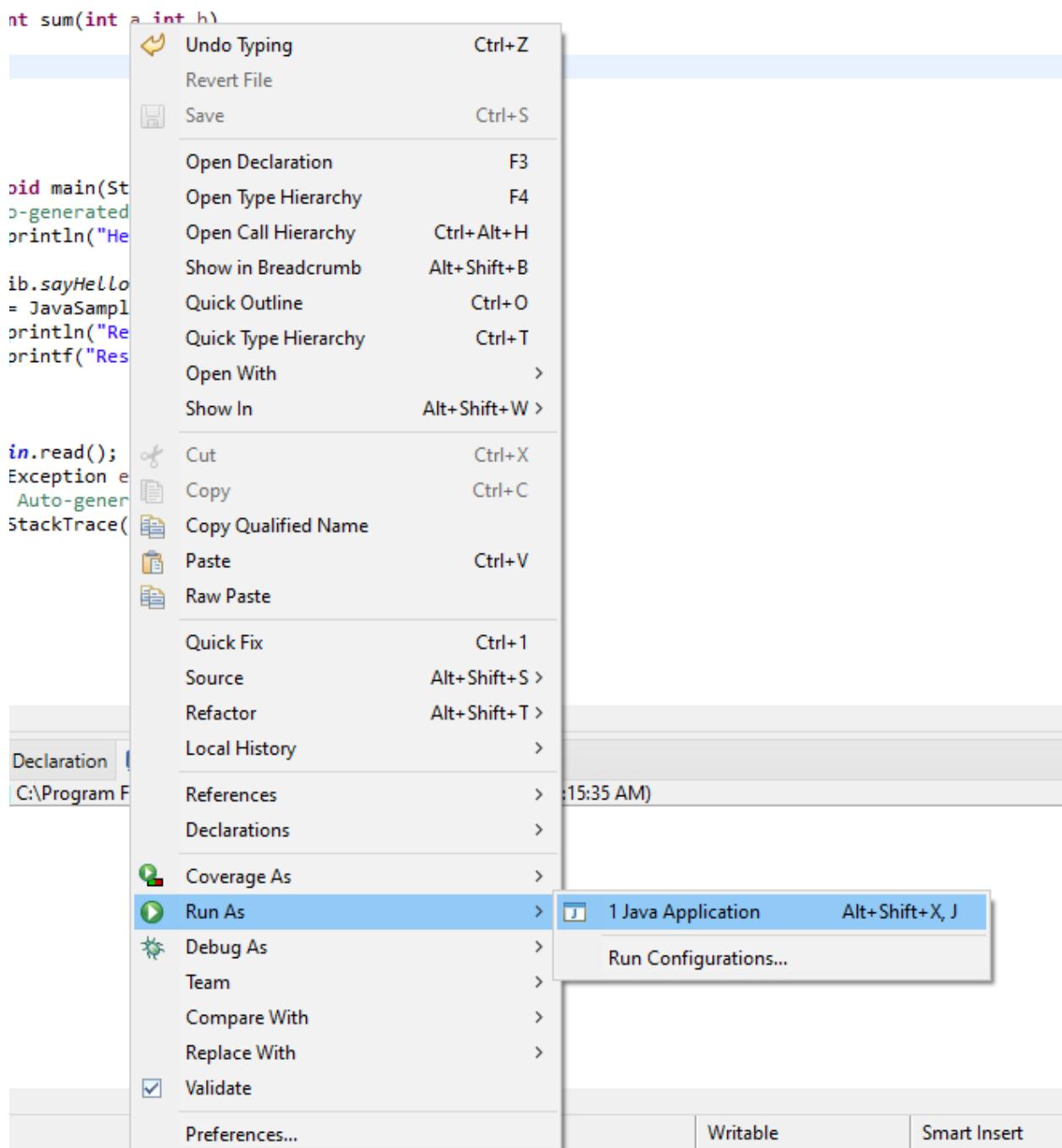
        JavaSampleLib.sayHelloTo("Computer");
        int result = JavaSampleLib.sum(5, 4);
        System.out.println("Results is" + result);
    }
}
```

```
System.out.printf("Results is %d \n", result);

try {
    System.in.read();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

}
```

also we can add main method to run our library functions. If we run this file its process main function



we can see output from console as below

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

- Project Explorer View:** Shows a project named "java-sample-lib" containing a "src" folder and a "ce103" package, which contains a file named "JavaSampleLib.java".
- Java Sample Code:** The code in "JavaSampleLib.java" is as follows:

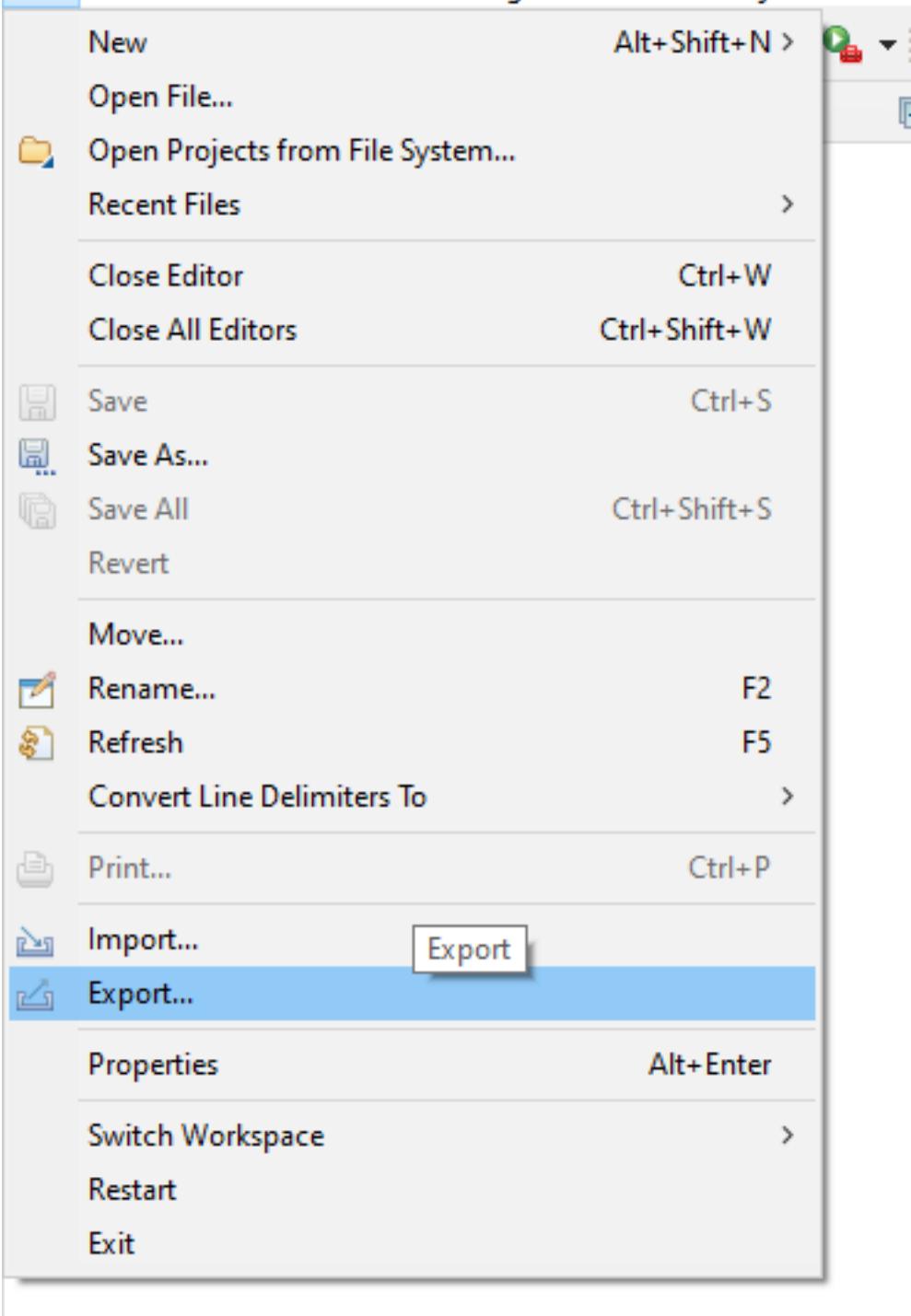
```
1 package ce103;
2
3 import java.io.IOException;
4
5 public class JavaSampleLib {
6
7     public static void sayHelloTo(String name) {
8         if(name.isBlank() || name.isEmpty())
9         {
10             System.out.println("Hello "+name);
11         }else{
12             System.out.println("Hello There");
13         }
14     }
15
16     public static int sum(int a,int b)
17     {
18         int c = 0;
19         c = a+b;
20         return c;
21     }
22
23     public static void main(String[] args) {
24         // TODO Auto-generated method stub
25         System.out.println("Hello World!");
26
27         JavaSampleLib.sayHelloTo("Computer");
28         int result = JavaSampleLib.sum(5, 4);
29         System.out.println("Results is" + result);
30         System.out.printf("Results is %d \n", result);
31
32         try {
33             System.in.read();
34         } catch (IOException e) {
35             // TODO Auto-generated catch block
36             e.printStackTrace();
37         }
38     }
39 }
40
41 }
42 }
```

- Console View:** Displays the output of the program execution:

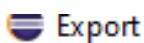
```
Hello World!
Hello There
Results is9
Results is 9
```

There is no exe files java runtime environment run class files but we can export this as an executable.

File Edit Source Refactor Navigate Search Project Run



Select Java->Runnable JAR File



Select

Export all resources required to run an application into a JAR file on the local file system.



Select an export wizard:

type filter text

- ▼ General
 - Ant Buildfiles
 - Archive File
 - File System
 - Preferences
- ▼ Install
 - Installed Software Items to File
- ▼ Java
 - JAR file
 - Javadoc
 - Runnable JAR file
- ▼ Run/Debug
 - Breakpoints
 - Coverage Session
 - Launch Configurations



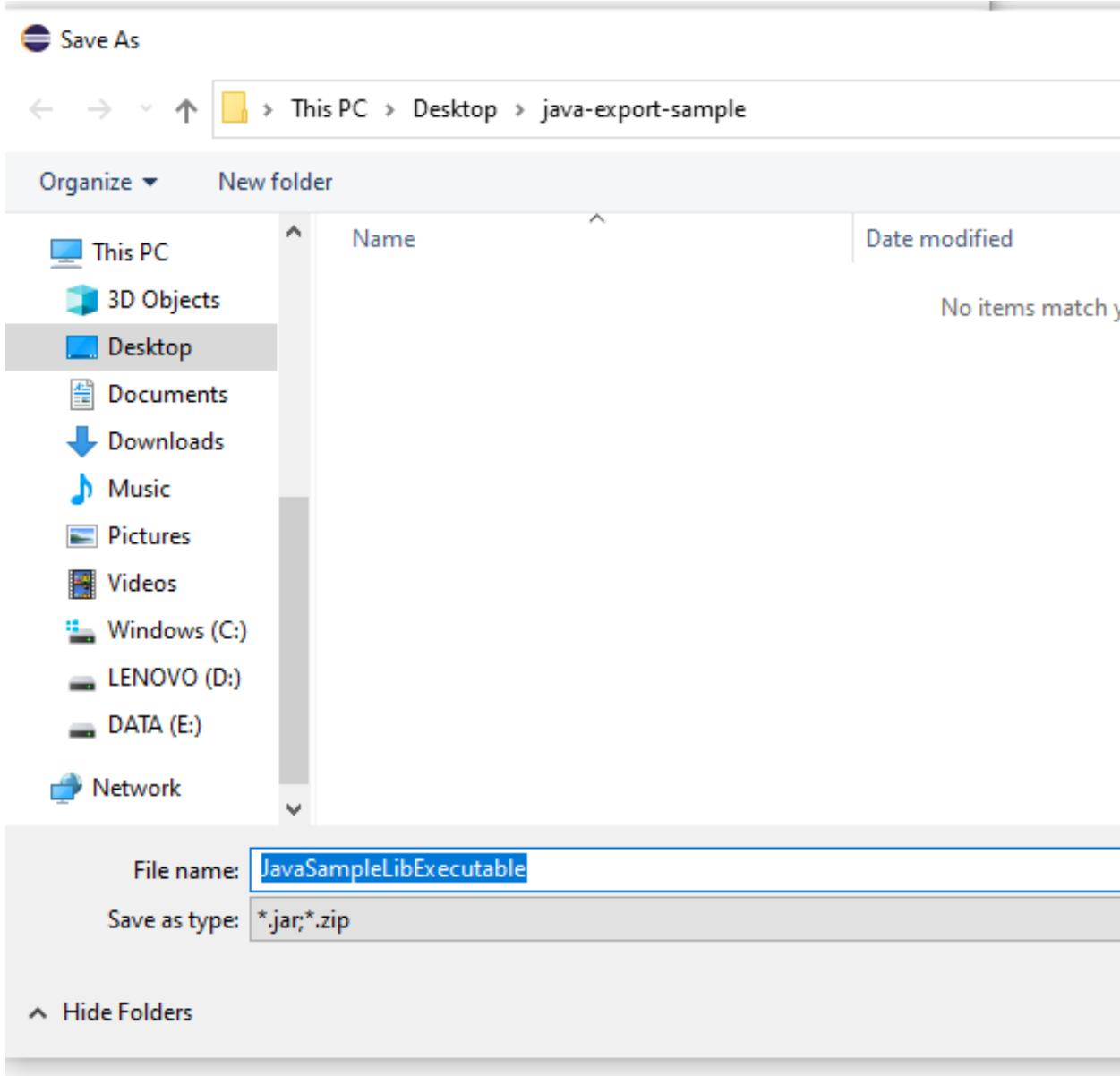
< Back

Next >

Finish

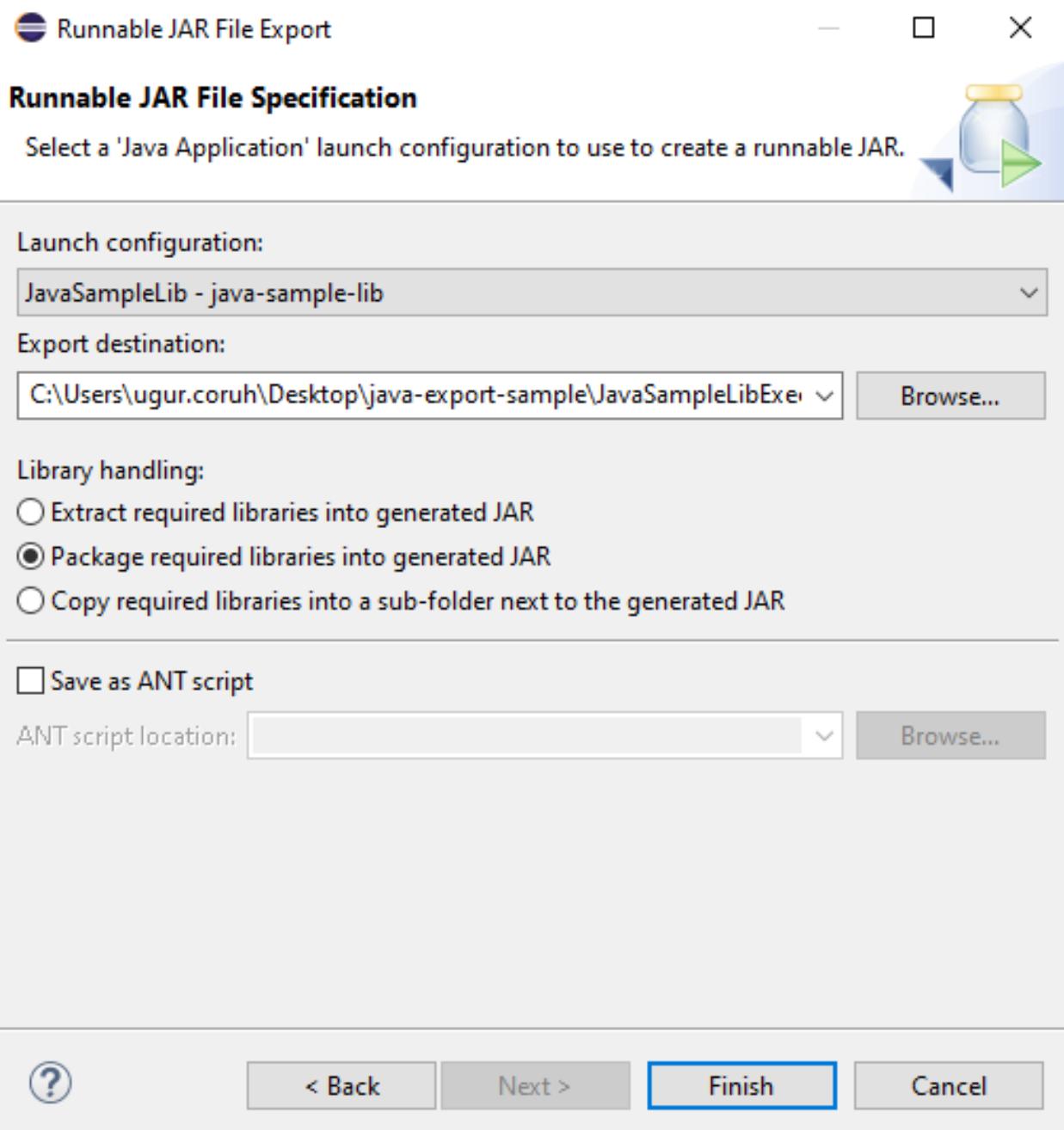
Cancel

click next and set output path for jar file

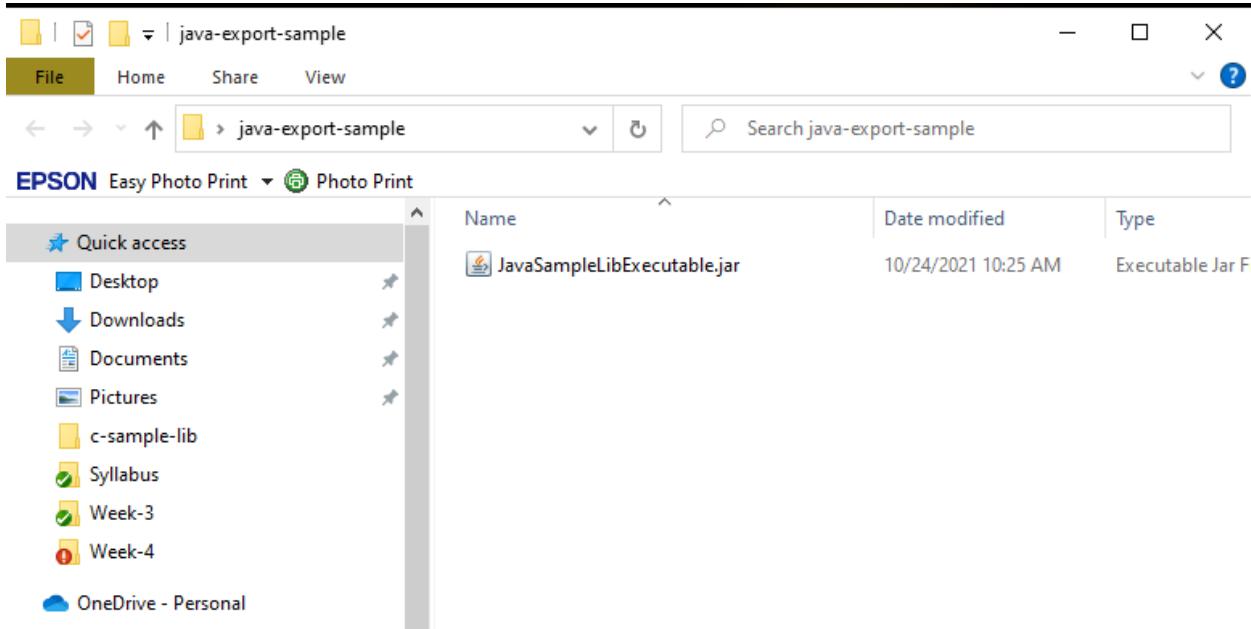


If our project has several external dependency then we can extract this required files (jar, so, dll) in seperated folder or we can combine them and generate a single executable jar

Lets pack everthing together, Select launch configuration that has main function



end of this operation we will have the following jar that we can by click



When you click application if cannot run then try command line to see problem
enter jar folder and run the following command

```
java -jar JavaSampleLibExecutable.jar
```

```
C:\Users\ugur.coruh\Desktop\java-export-sample>java -jar JavaSampleLibExecutable.jar
Exception in thread "main" java.lang.UnsupportedClassVersionError: ce103/JavaSampleLib has been compiled by a more recent
version of the Java Runtime (class file version 60.0), this version of the Java Runtime only recognizes class file ver
sions up to 52.0
        at java.lang.ClassLoader.defineClass1(Native Method)
        at java.lang.ClassLoader.defineClass(Unknown Source)
        at java.security.SecureClassLoader.defineClass(Unknown Source)
        at java.net.URLClassLoader.defineClass(Unknown Source)
        at java.net.URLClassLoader.access$100(Unknown Source)
        at java.net.URLClassLoader$1.run(Unknown Source)
        at java.net.URLClassLoader$1.run(Unknown Source)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.net.URLClassLoader.findClass(Unknown Source)
        at java.lang.ClassLoader.loadClass(Unknown Source)
        at java.lang.ClassLoader.loadClass(Unknown Source)
        at java.lang.Class.forName0(Native Method)
        at java.lang.Class.forName(Unknown Source)
        at org.eclipse.jdt.internal.jarinjarloader.JarRsrcLoader.main(JarRsrcLoader.java:59)
C:\Users\ugur.coruh\Desktop\java-export-sample>
```

In my case eclipse build JDK is newer than that I installed and set for my OS

If we check version we can see problem Java version 1.8.0_231

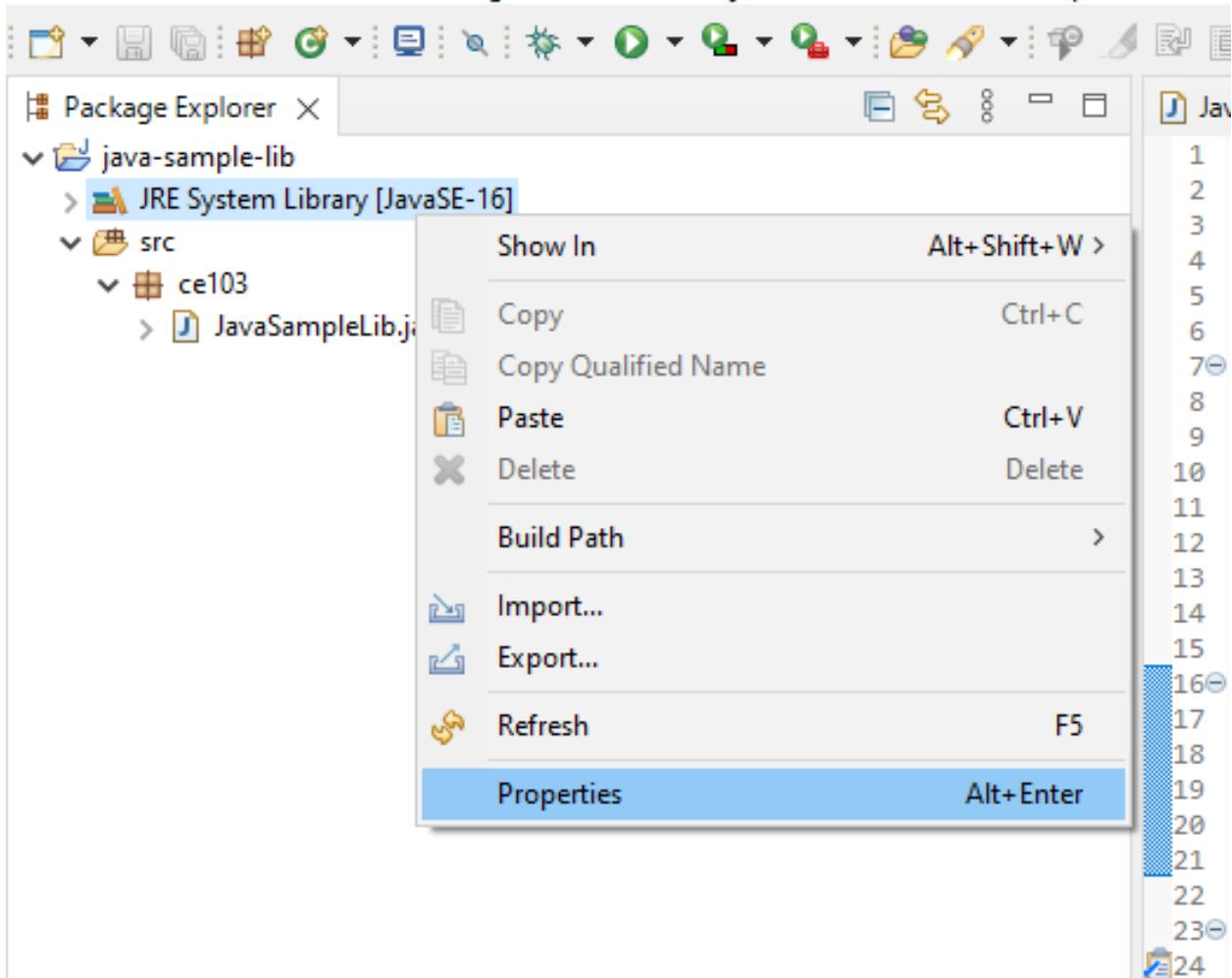
```
C:\Users\ugur.coruh\Desktop\java-export-sample>java -showversion
java version "1.8.0_231"
Java(TM) SE Runtime Environment (build 1.8.0_231-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.231-b11, mixed mode)

Usage: java [-options] class [args...]
```

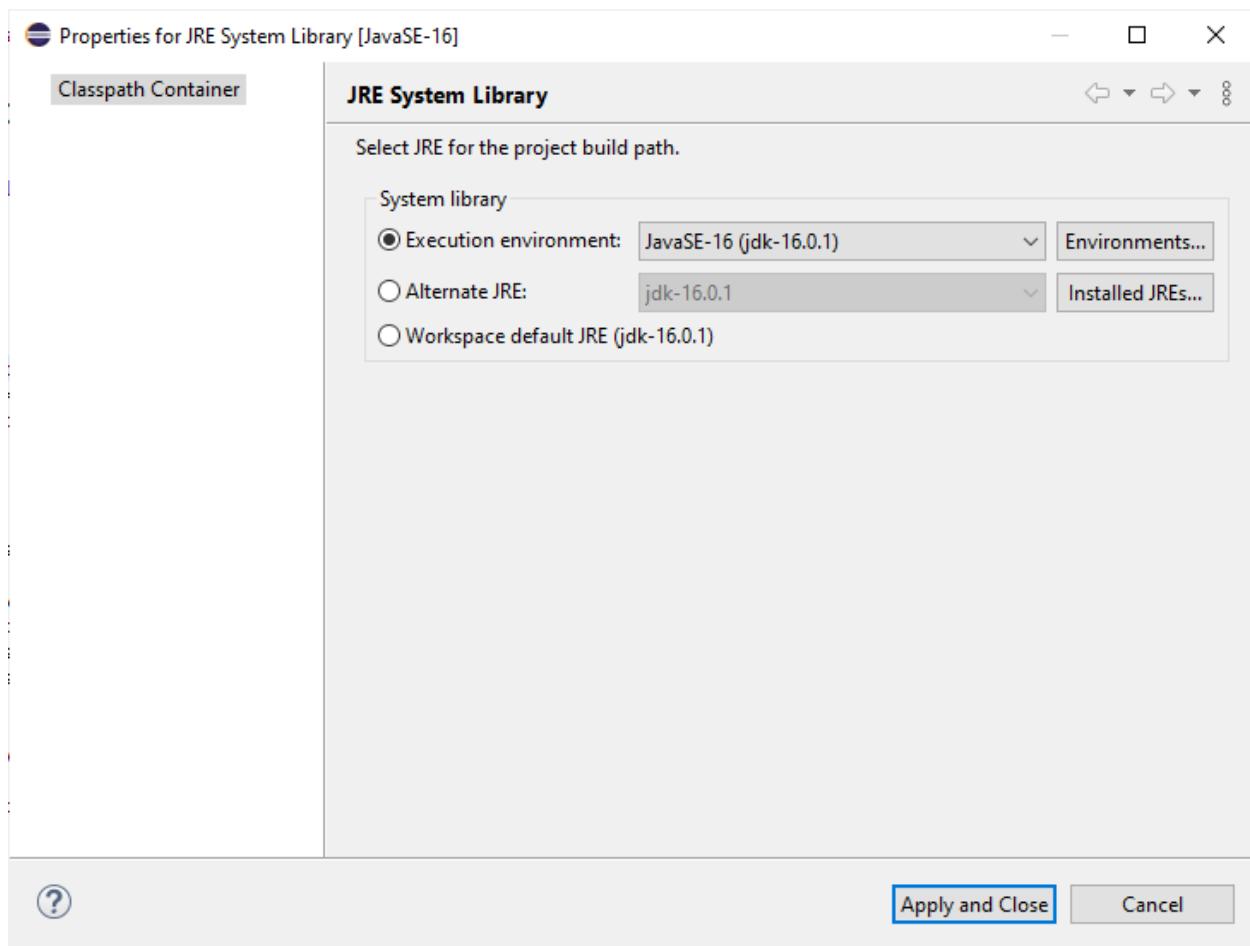
We can found installed and builded JDK for our application from Eclipse setting

eclipse-workspace - java-sample-lib/src/ce103/JavaSampleLib.java - Eclipse IDE

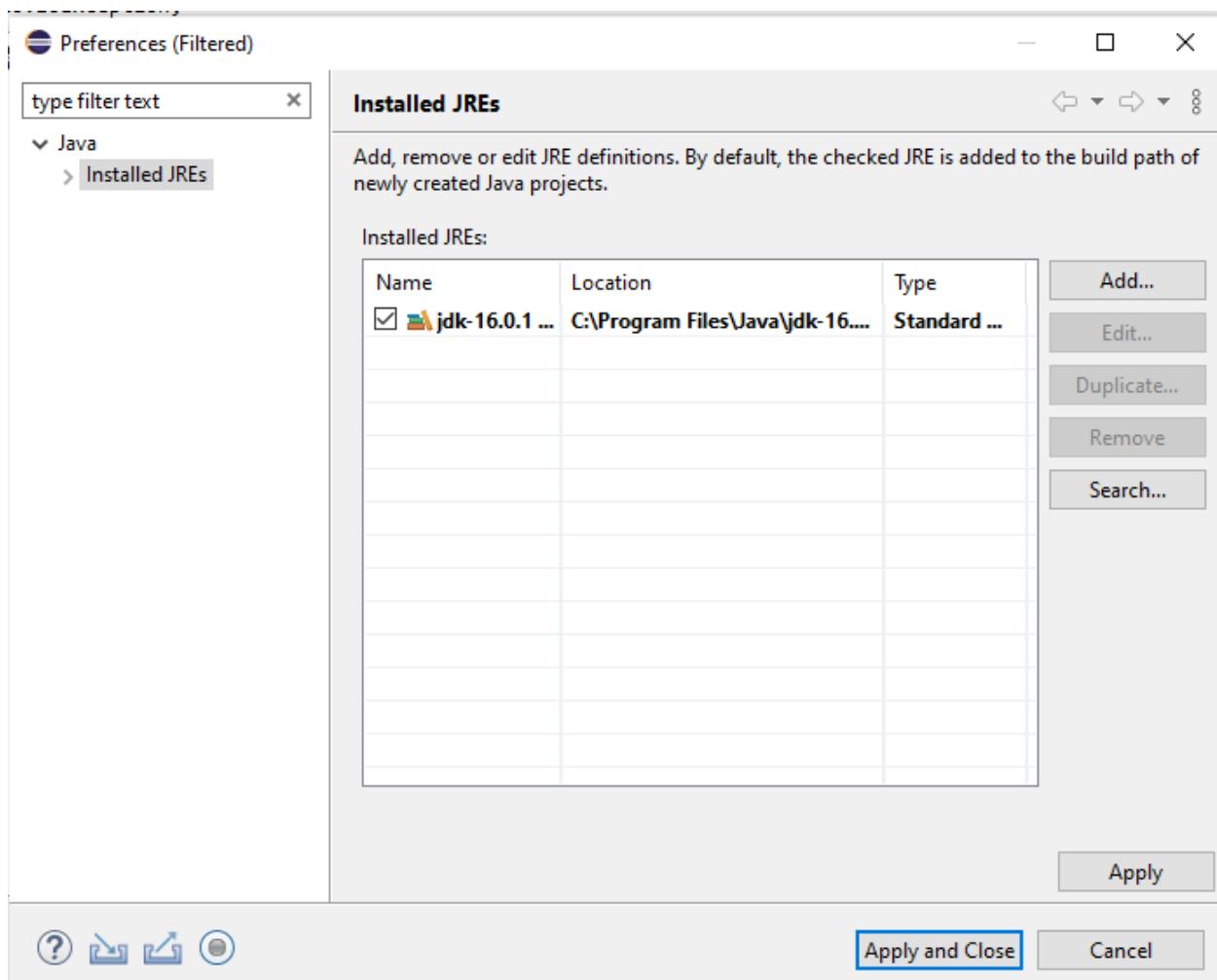
File Edit Source Refactor Navigate Search Project Run Window Help



select environments

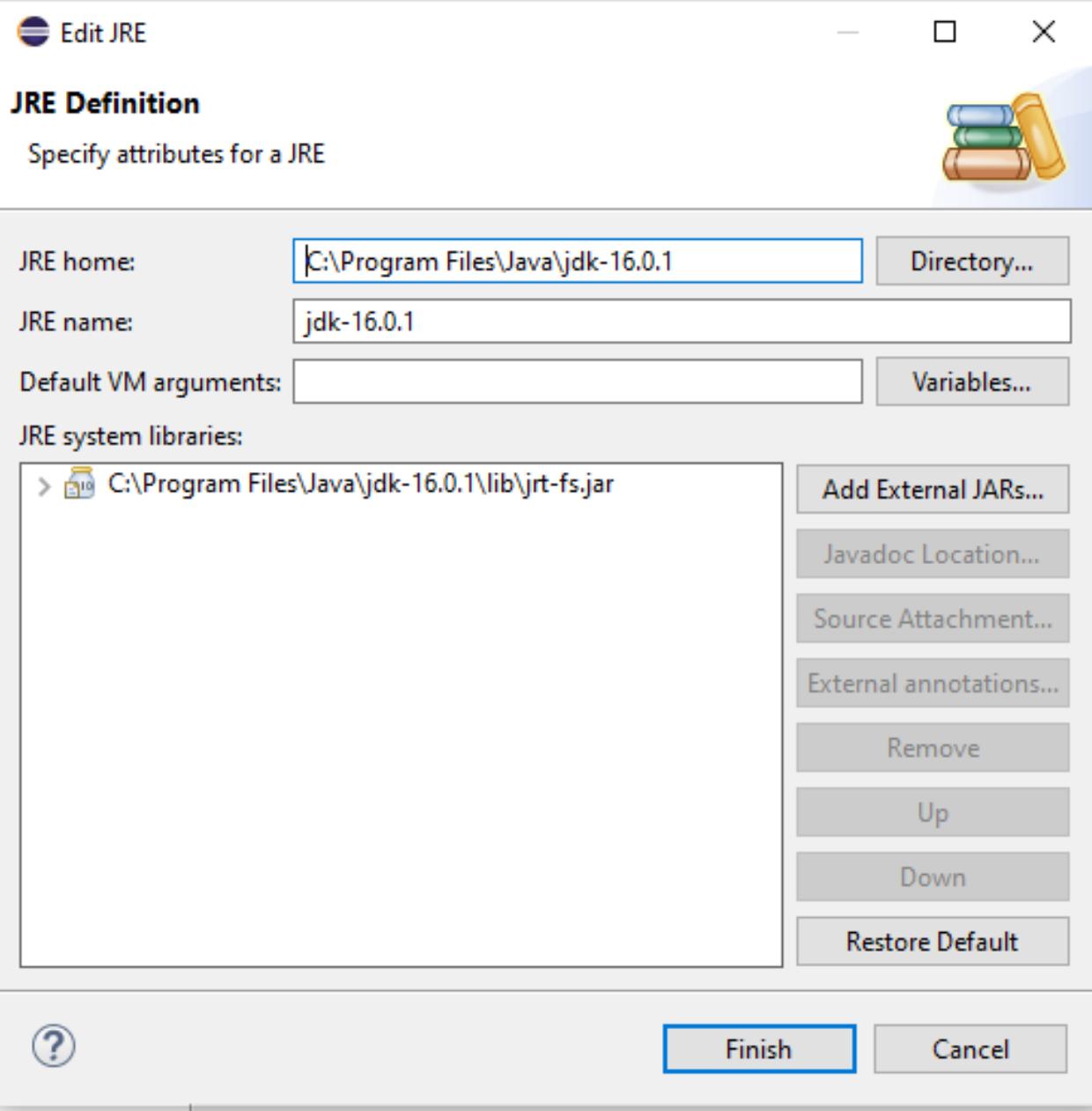


select installed JRE or JDK



you can see installed JRE or JDK home

C:\Program Files\Java\jdk-16.0.1



Open system environment to fix this problem

All Apps Documents Web More ▾

Best match



System Configuration

App



Settings



Edit the system environment variables



System



Reset this PC



Recovery



Recovery options



About your PC



Taskbar notification area



See if you have a 32-bit or 64-bit version of Windows



Search the web



syste - See web results



Apps (7+)

System Properties

X

Computer Name Hardware Advanced System Protection Remote

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

Settings...

User Profiles

Desktop settings related to your sign-in

Settings...

Startup and Recovery

System startup, system failure, and debugging information

Settings...

Environment Variables...

OK

Cancel

Apply

Check user settings first

Environment Variables

X

User variables for ugur.coruh

Variable	Value
ChocolateyLastPathUpdate	132416153103954791
GOPATH	C:\Users\ugur.coruh\go
IntelliJ IDEA Community Edit...	C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.1.1...
OneDrive	C:\Users\ugur.coruh\OneDrive
OneDriveConsumer	C:\Users\ugur.coruh\OneDrive
Path	C:\Program Files\Java\jdk-16.0.1\bin;C:\Python27;C:\Users\ugur.co...
TEMP	C:\Users\ugur.coruh\AppData\Local\Temp

New...

Edit...

Delete

System variables

Variable	Value
asl.log	Destination=file
ChocolateyInstall	C:\ProgramData\chocolatey
CHOKIDAR_USESPOLLING	true
ComSpec	C:\WINDOWS\system32\cmd.exe
configsetroot	C:\WINDOWS\ConfigSetRoot
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA HOME	C:\Program Files\Java\jdk-16.0.1\

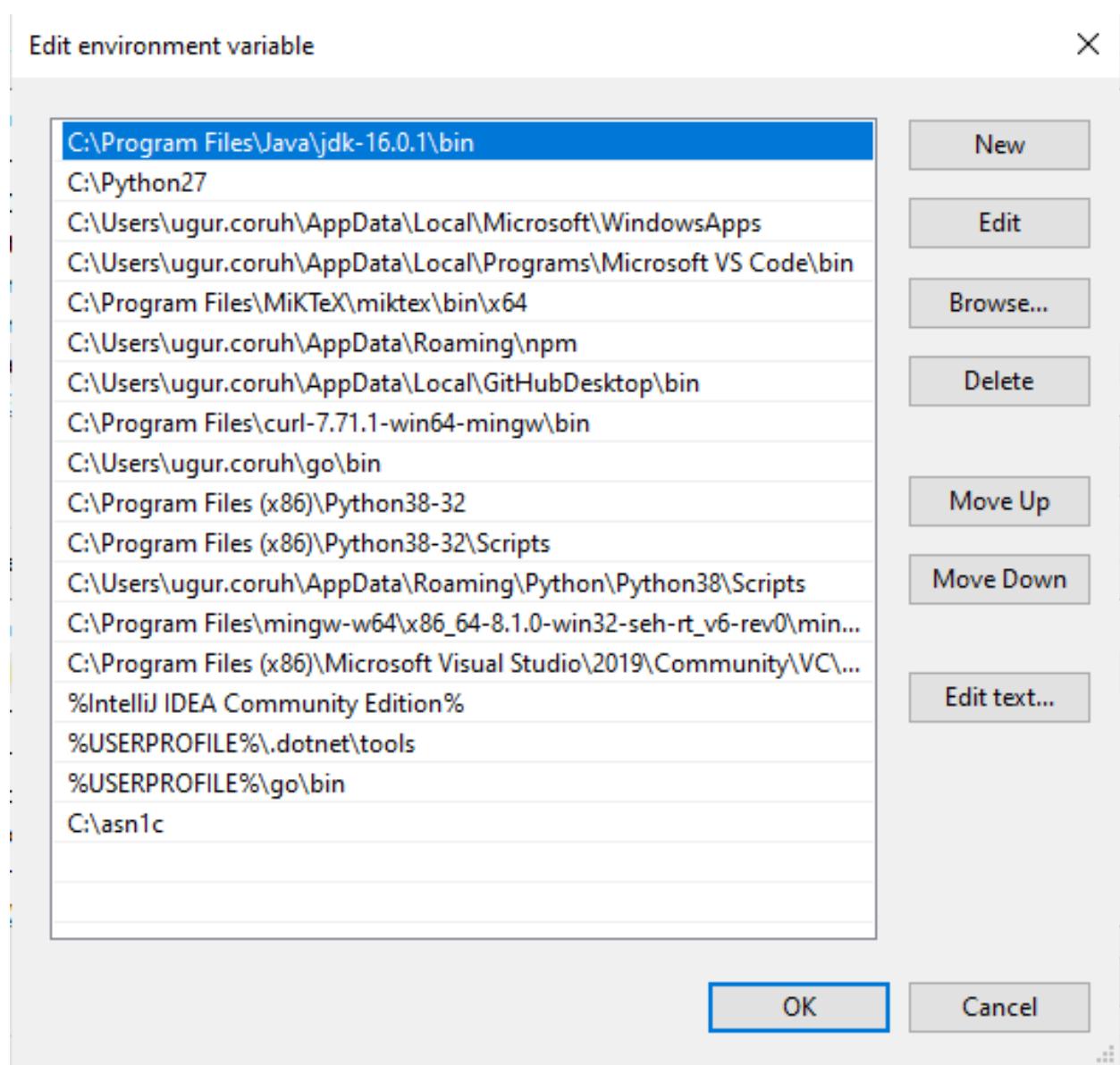
New...

Edit...

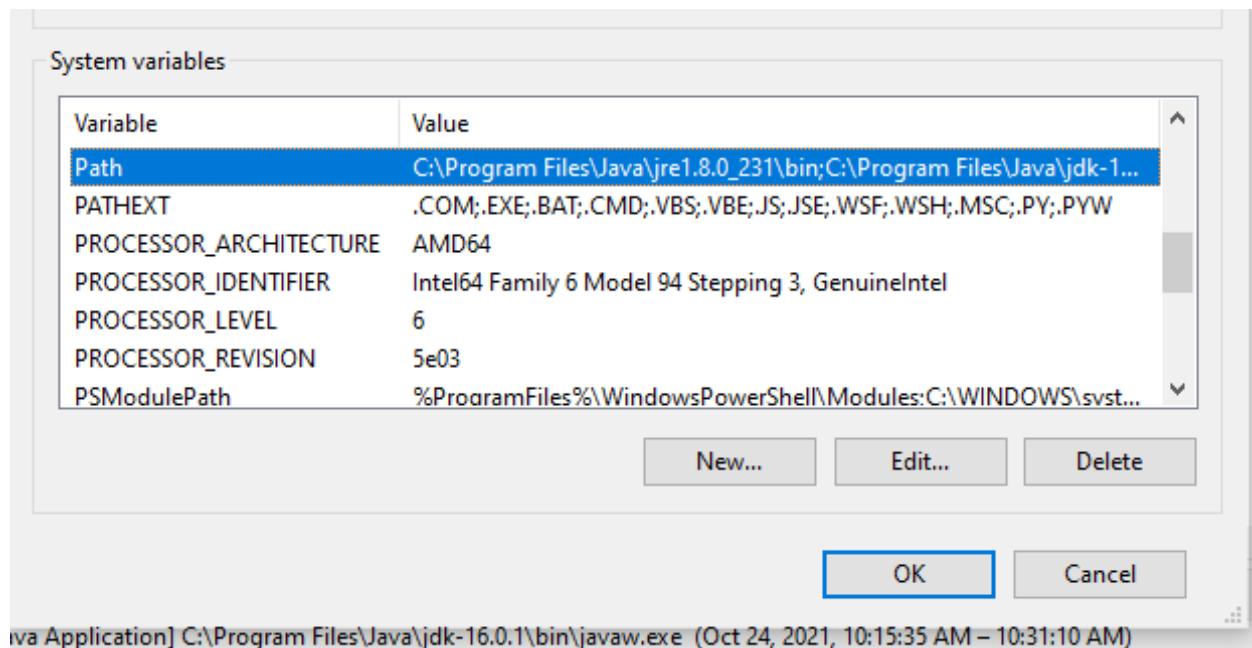
Delete

OK

Cancel



Check system settings



Edit environment variable

X

C:\Program Files\Java\jre1.8.0_231\bin
C:\Program Files\Java\jdk-16.0.1\bin
C:\Python27\
C:\Python27\Scripts
C:\Program Files\Amazon Corretto\jdk11.0.8_10\bin
C:\Program Files (x86)\Intel\Intel(R) Management Engine Compon...
C:\Program Files\Intel\Intel(R) Management Engine Components\i...
C:\ProgramData\Anaconda3
C:\ProgramData\Anaconda3\Library\mingw-w64\bin
C:\ProgramData\Anaconda3\Library\usr\bin
C:\ProgramData\Anaconda3\Library\bin
C:\ProgramData\Anaconda3\Scripts
C:\Program Files (x86)\Python38-32\Scripts\
C:\Program Files (x86)\Python38-32\
C:\Program Files\Python38\Scripts\
C:\Program Files\Python38\
C:\Windows\system32
C:\Windows
C:\Windows\System32\Wbem
C:\Windows\System32\WindowsPowerShell\v1.0\
...

New

Edit

Browse...

Delete

Move Up

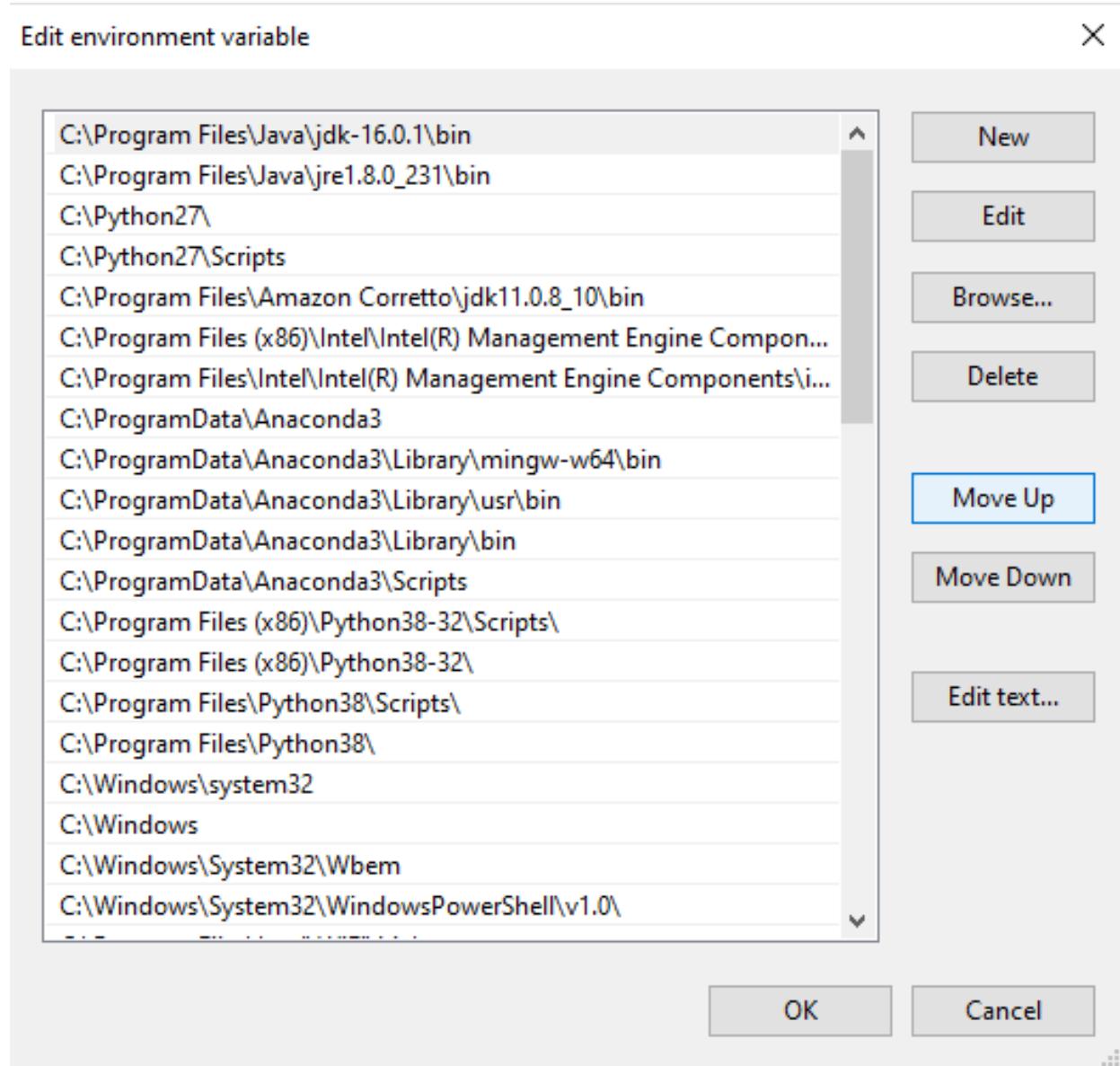
Move Down

Edit text...

OK

Cancel

we will move up the JDK 16 configuration then command line will run first java



Also in system setting check JAVA_HOME

System variables	
Variable	Value
JAVA_HOME	C:\Program Files\Java\jdk-16.0.1\
MOSQUITTO_DIR	C:\Program Files\mosquitto
NUMBER_OF_PROCESSORS	8
OS	Windows NT

After this settings close current command line and open new one
write

```
java --version
```

if you see java version updated and 16.0.1 then settings are correct

```
C:\ C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ugur.coruh>java --version
java 16.0.1 2021-04-20
Java(TM) SE Runtime Environment (build 16.0.1+9-24)
Java HotSpot(TM) 64-Bit Server VM (build 16.0.1+9-24, mixed mode, sharing)

C:\Users\ugur.coruh>
```

and now if we enter and run application as follow we will see output

```
C:\Users\ugur.coruh>cd Desktop
C:\Users\ugur.coruh\Desktop>cd java-export-sample
C:\Users\ugur.coruh\Desktop\java-export-sample>java -jar JavaSampleLibExecutable.jar
Hello World!
Hello There
Results is9
Results is 9
```

But when you click this jar its not running as you see so we have options to provide a clickable application there

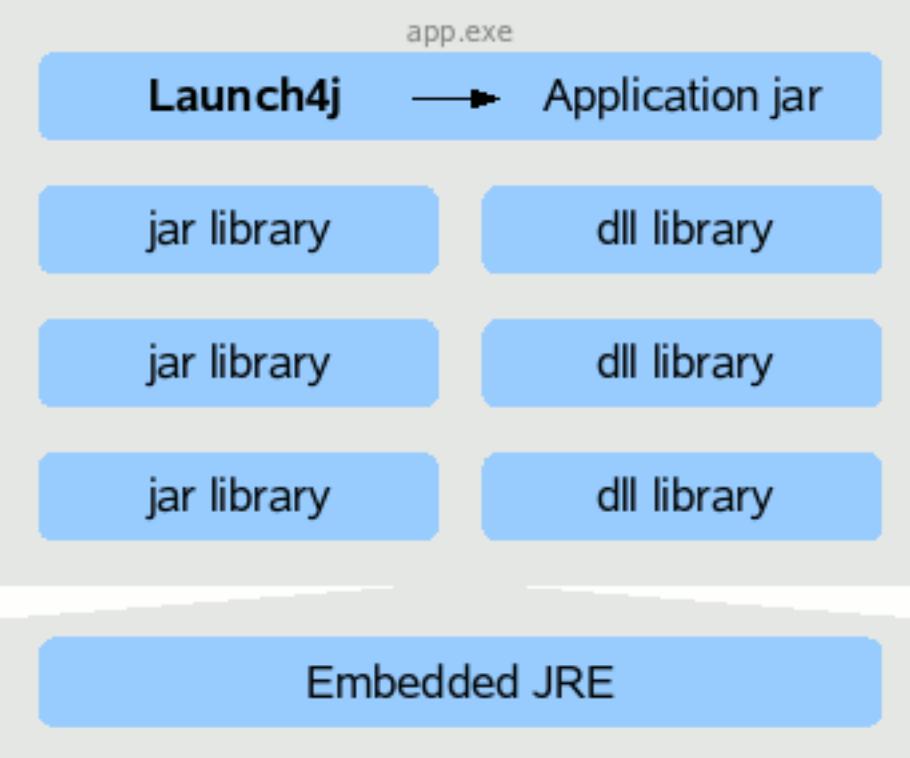
Launch4j is an option here

Launch4j - Cross-platform Java executable wrapper¹⁰

¹⁰<http://launch4j.sourceforge.net/index.html>



app-setup.exe
Installer created by an external installer program.

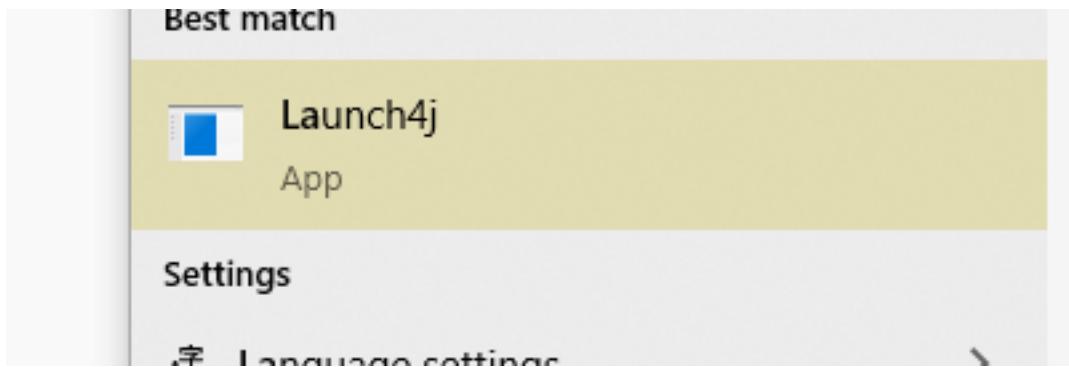


you can watch this tutorial also

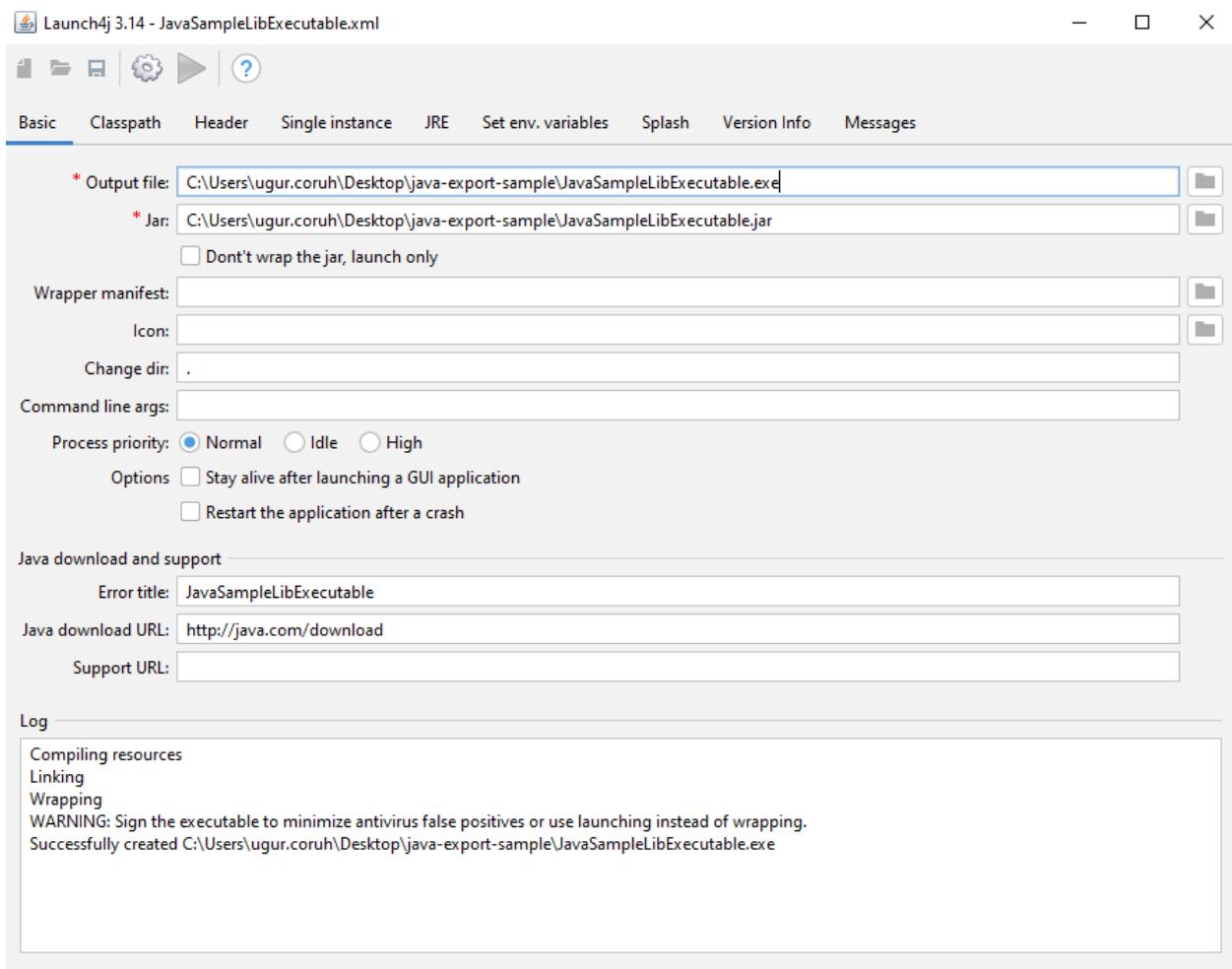
How to convert jar to exe using Launch4J Full explanation - YouTube¹¹

Download and install launch4j and open application

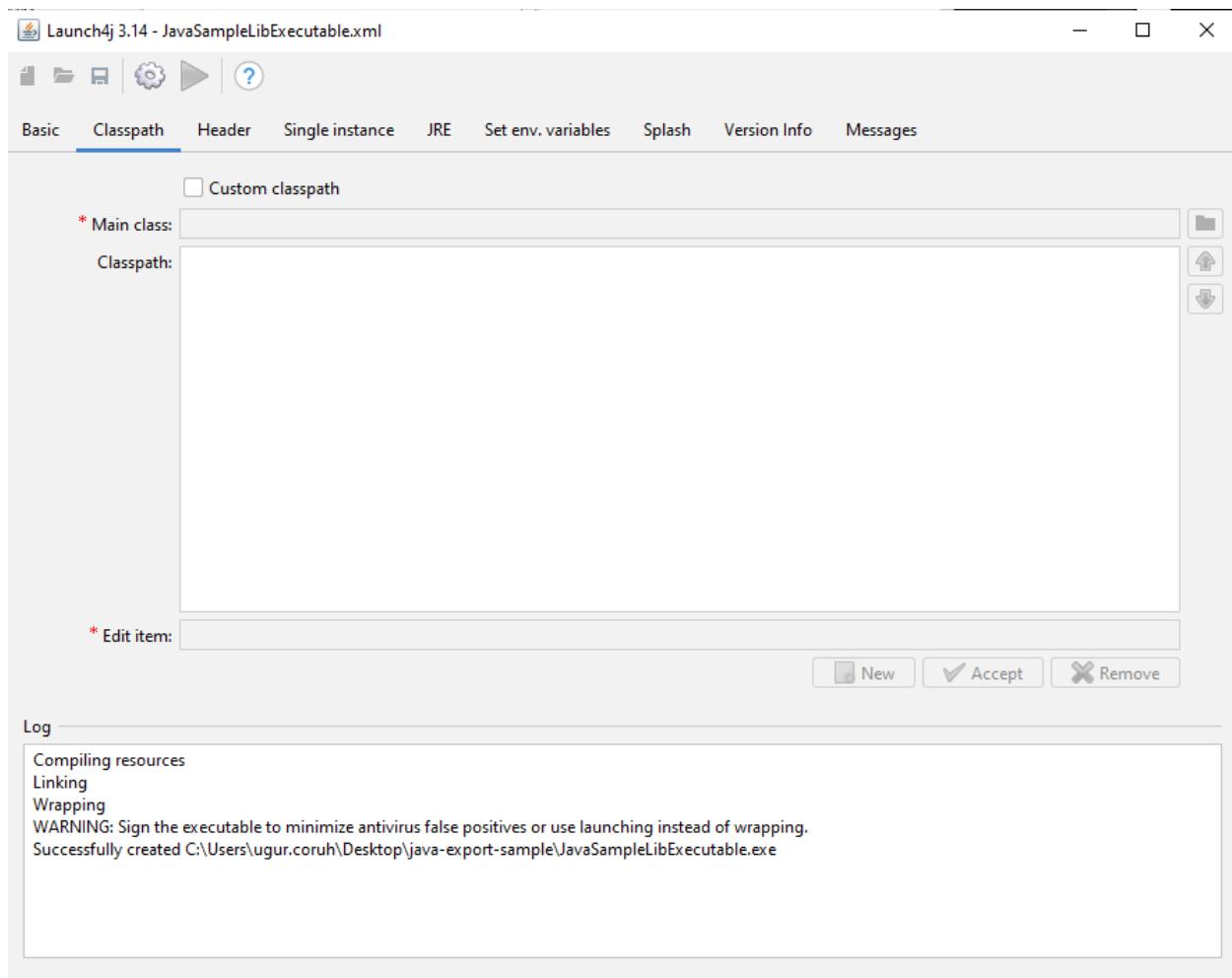
¹¹https://www.youtube.com/watch?v=MyMPPuYGN-U&ab_channel=GoXR3PlusStudio



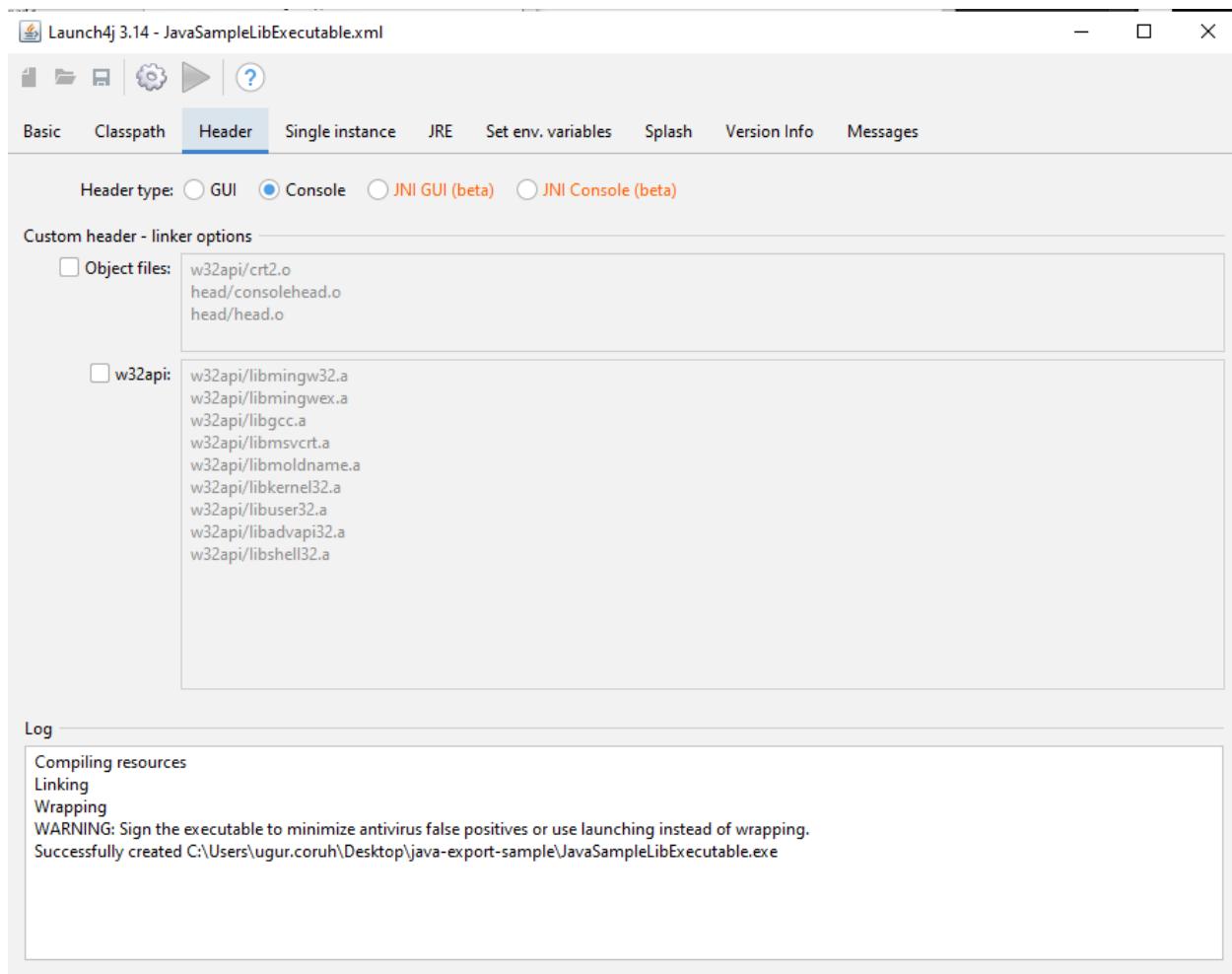
configure your application settings similar to below select jar file and exe output path



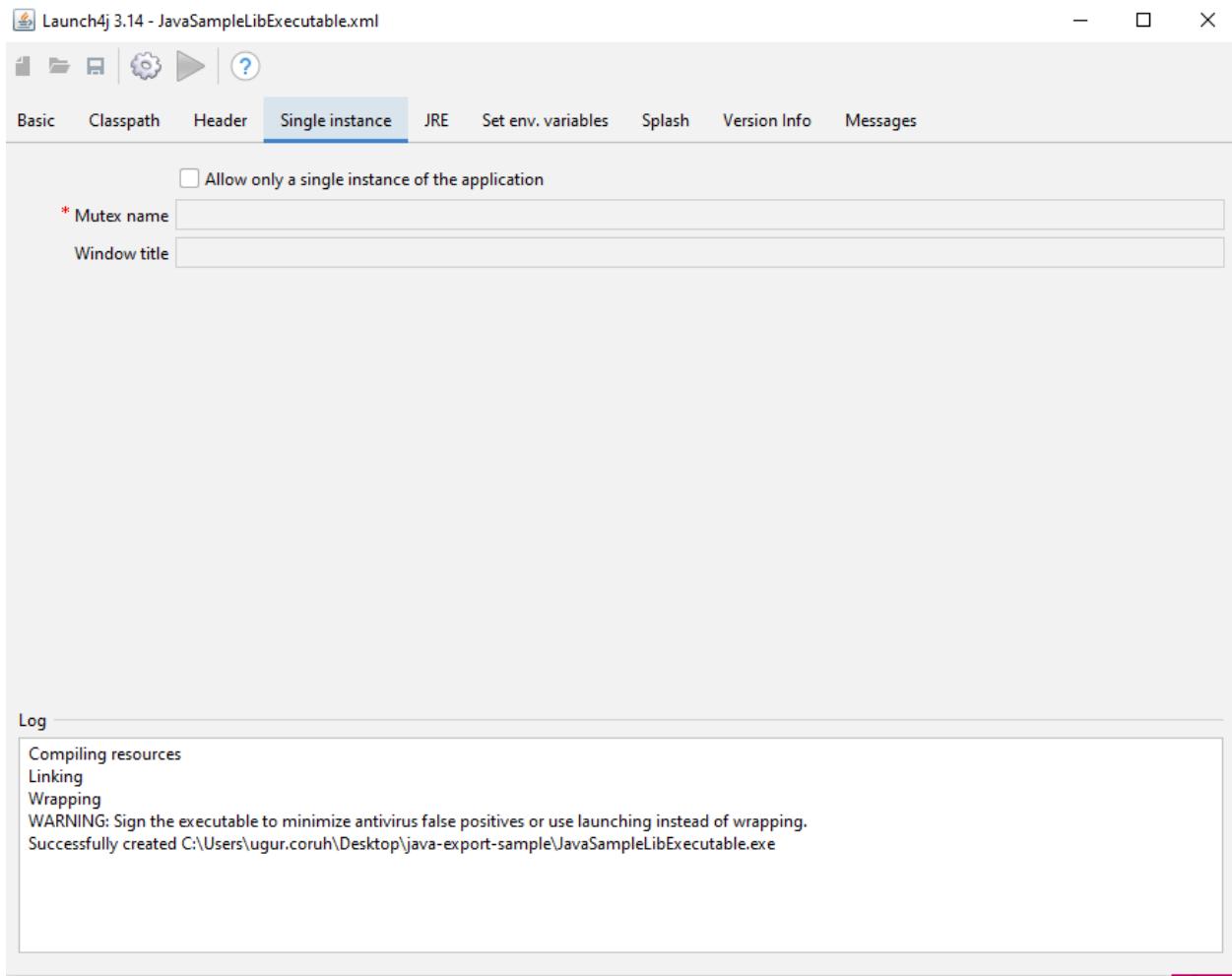
we can customize main class if have multiple main class



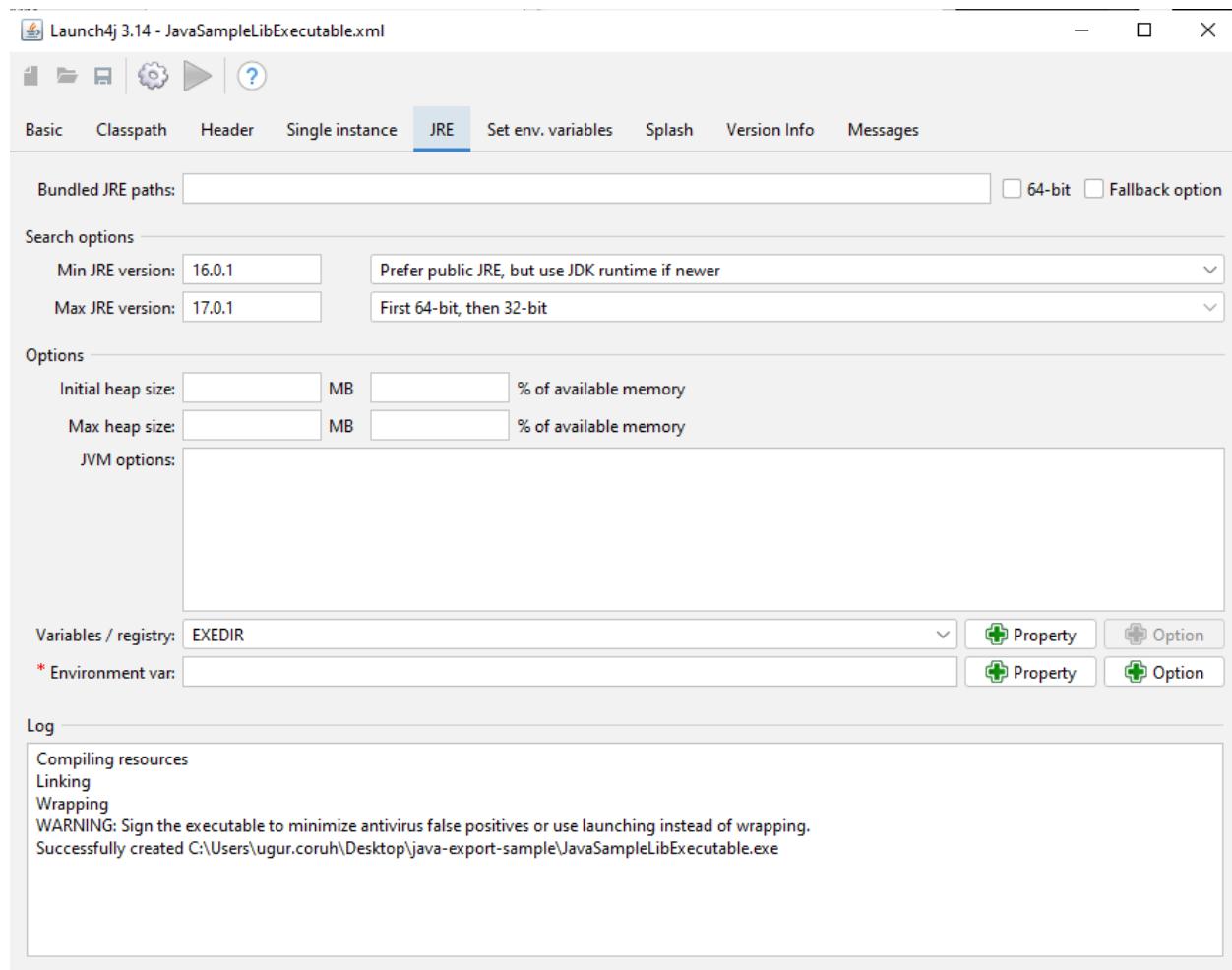
select console from setting for this application



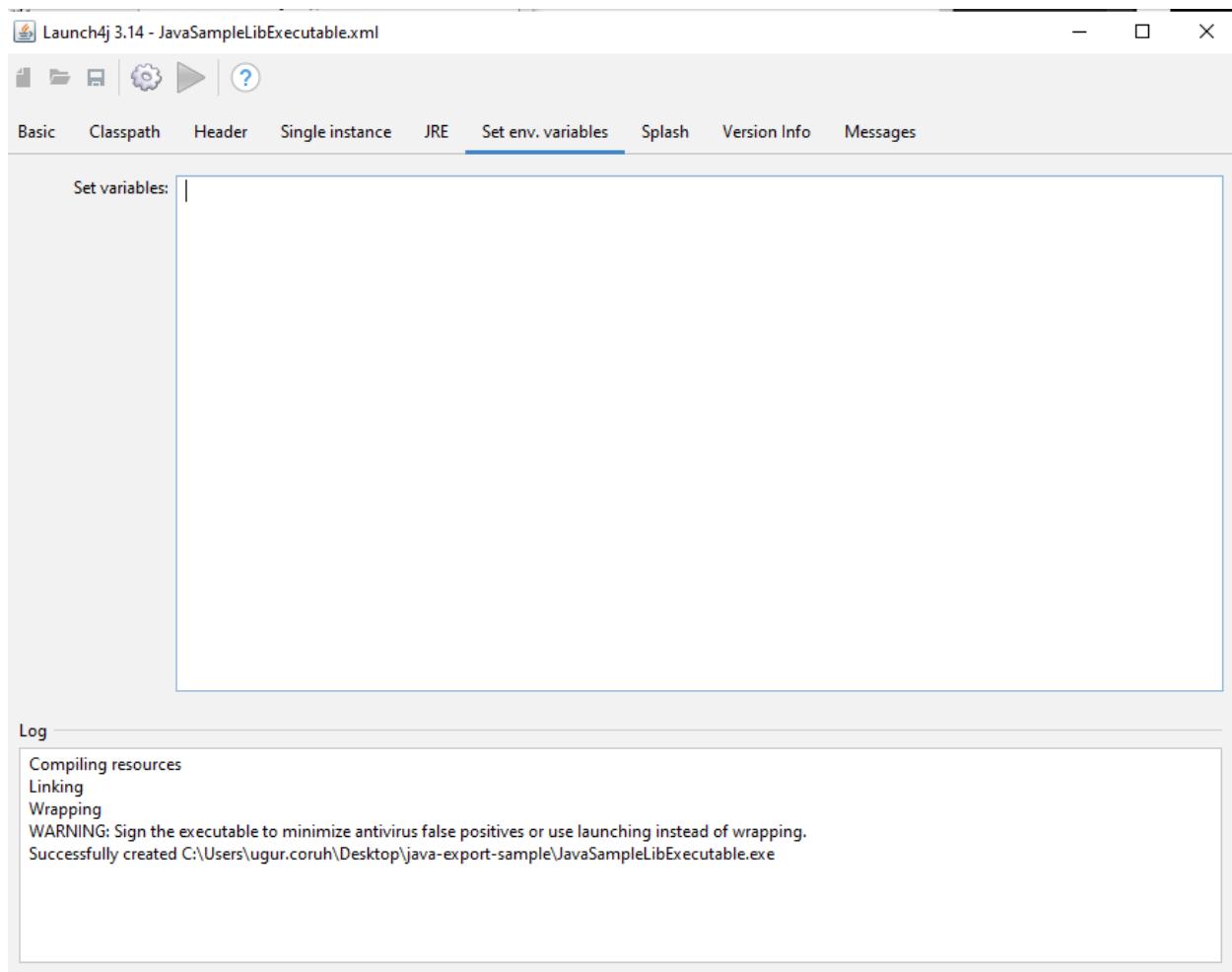
we can provide a single running application, this setting avoid to run multiple instances



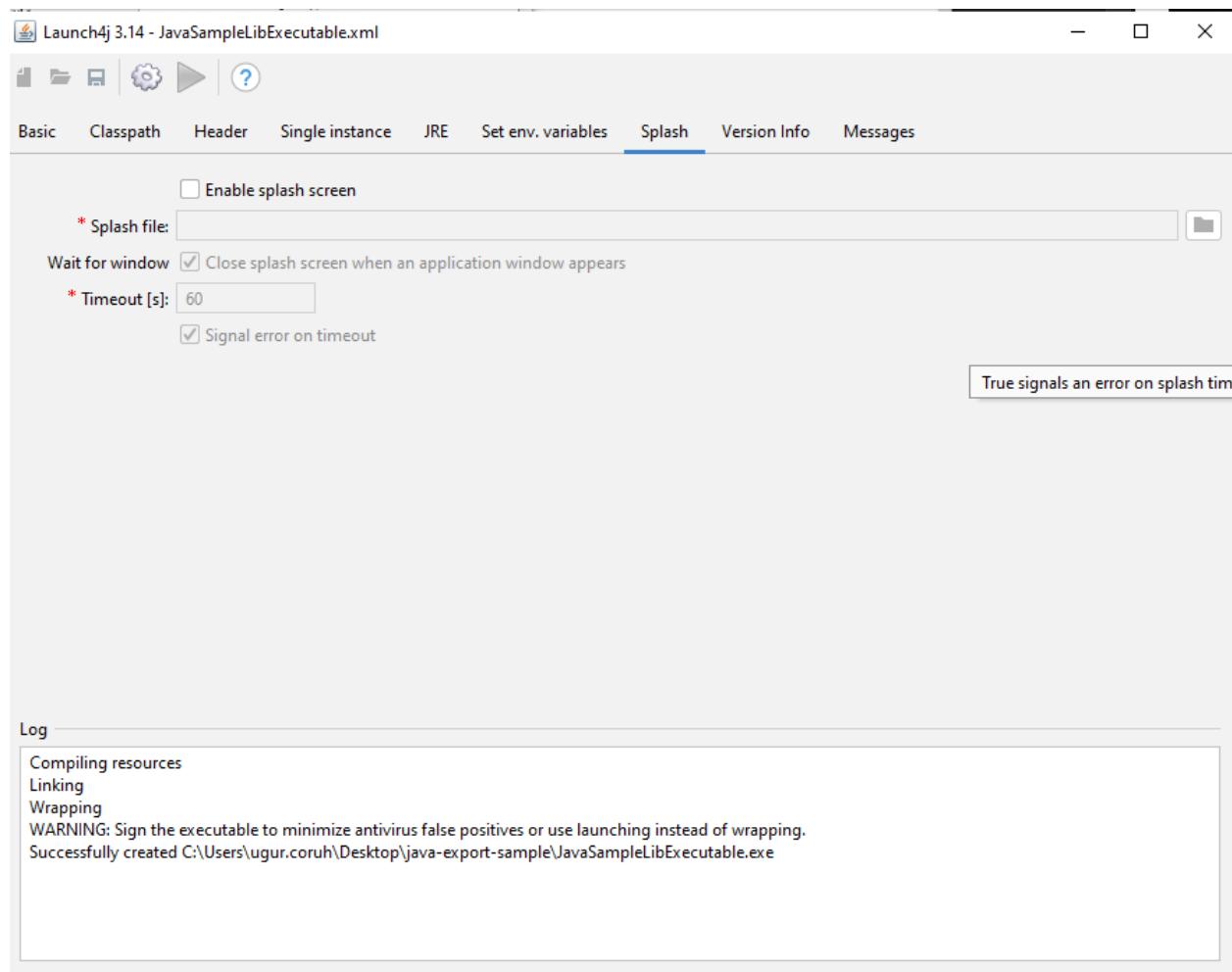
we need to set runtime environment versions



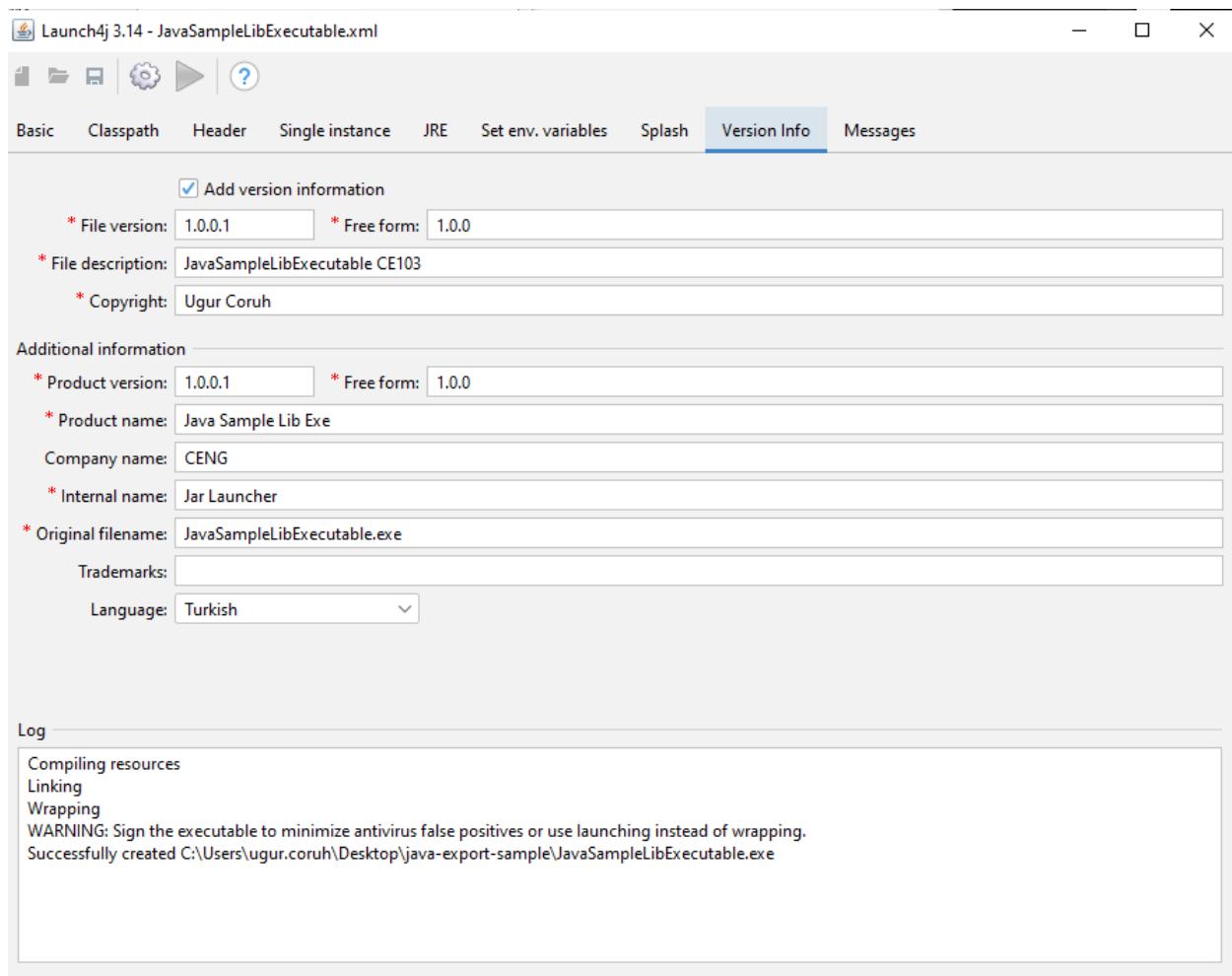
you can set system parameters before running application



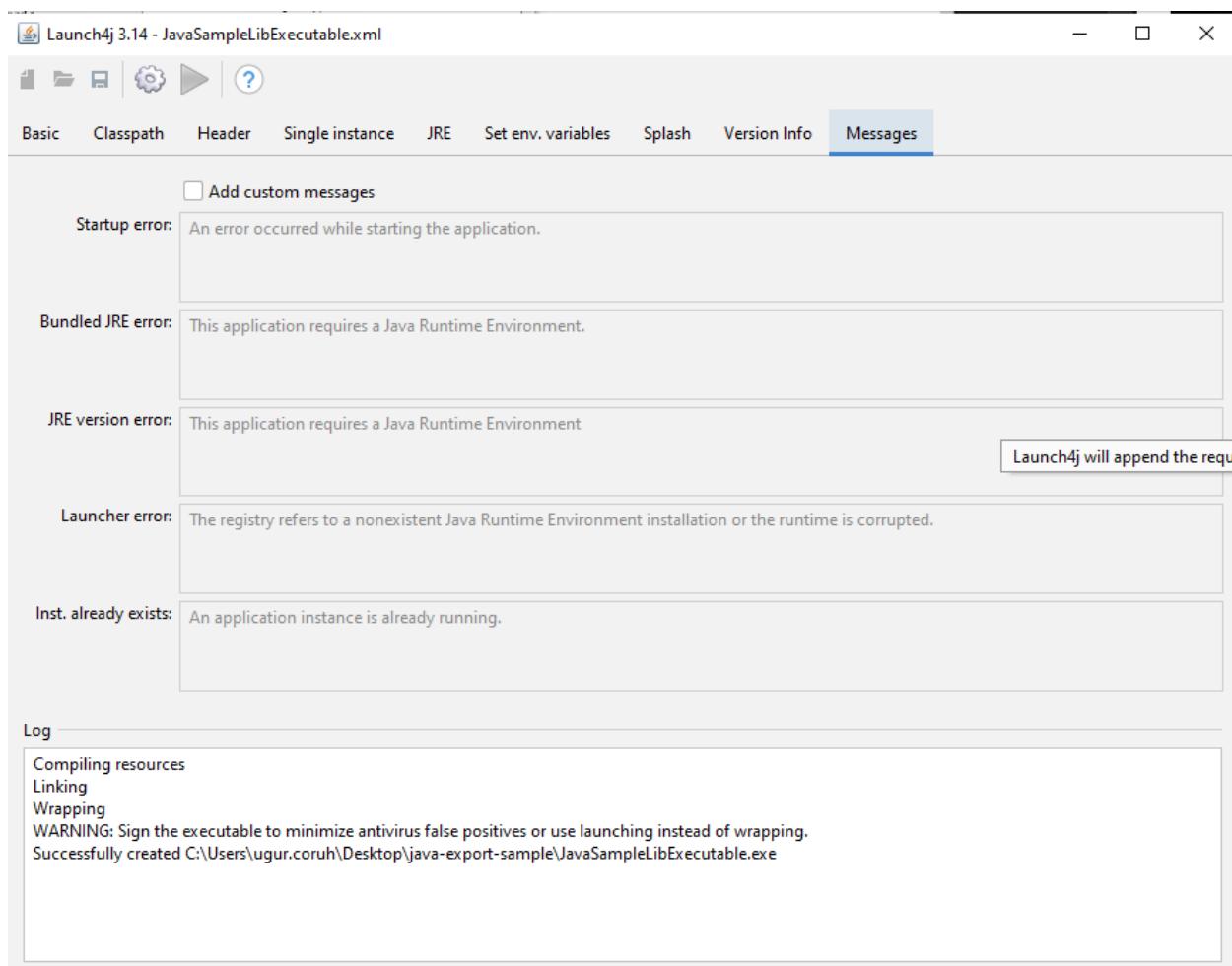
with splash screen you can show a splash screen image for your application



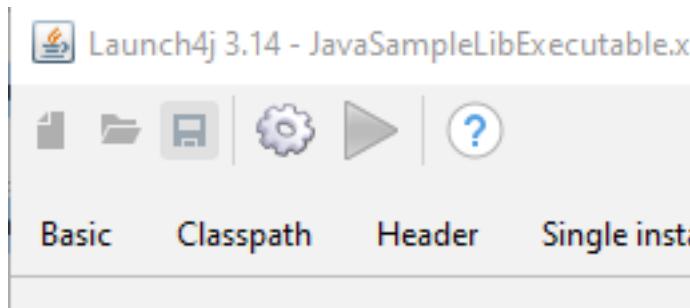
File attributes such as version product information is configured from version info tab



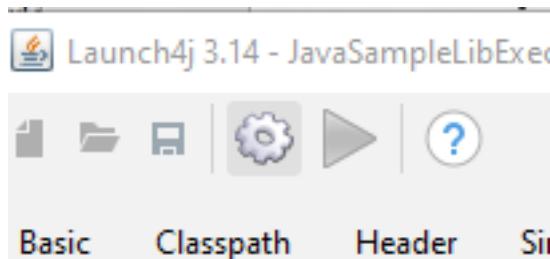
if your application runtime condition has an error then you can show this customized messages also



with this options save configuration file xml



and compile settings



you will see generated output file in log screen

Compiling resources

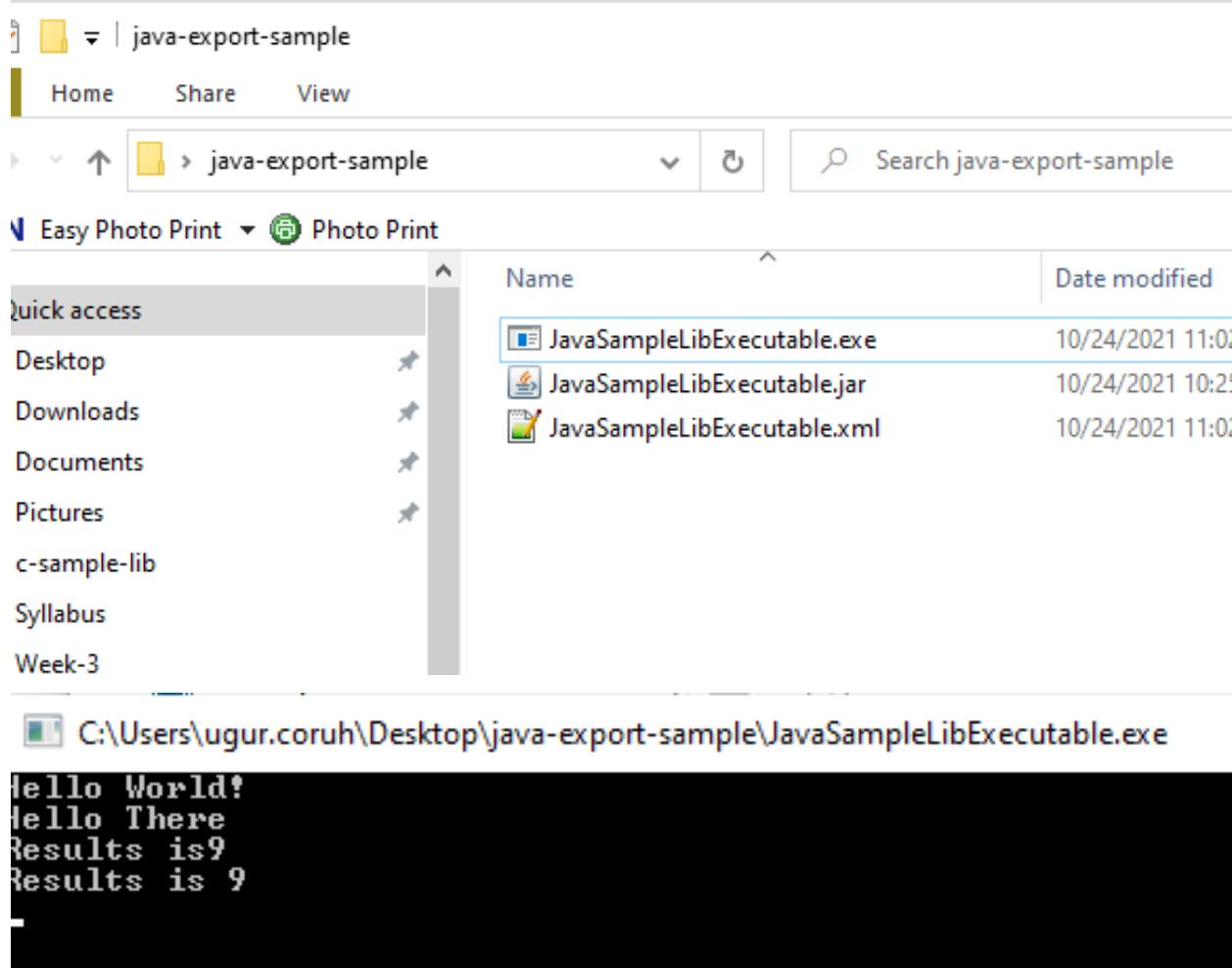
Linking

Wrapping

WARNING: Sign the executable to minimize antivirus false positives or use launching instead of wrapping

Successfully created C:\Users\ugur.coruh\Desktop\java-export-sample\JavaSampleLibExecutable.exe

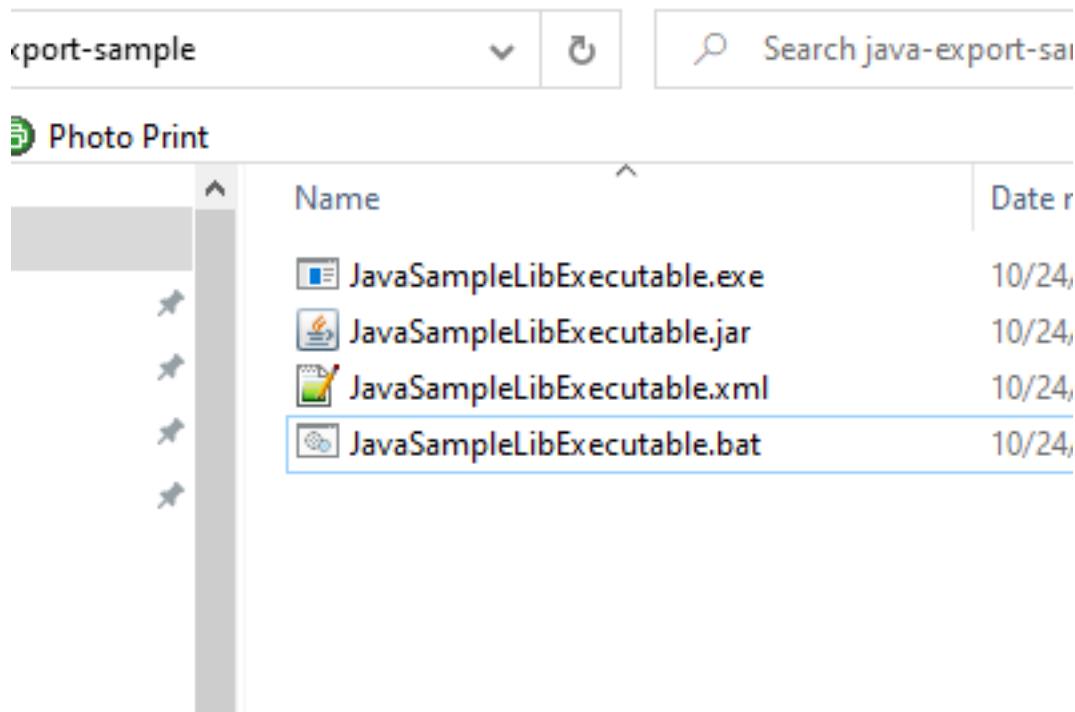
now we can run exe by click



another option here adding a bat file to run current jar file

JavaSampleLibExecutable.bat

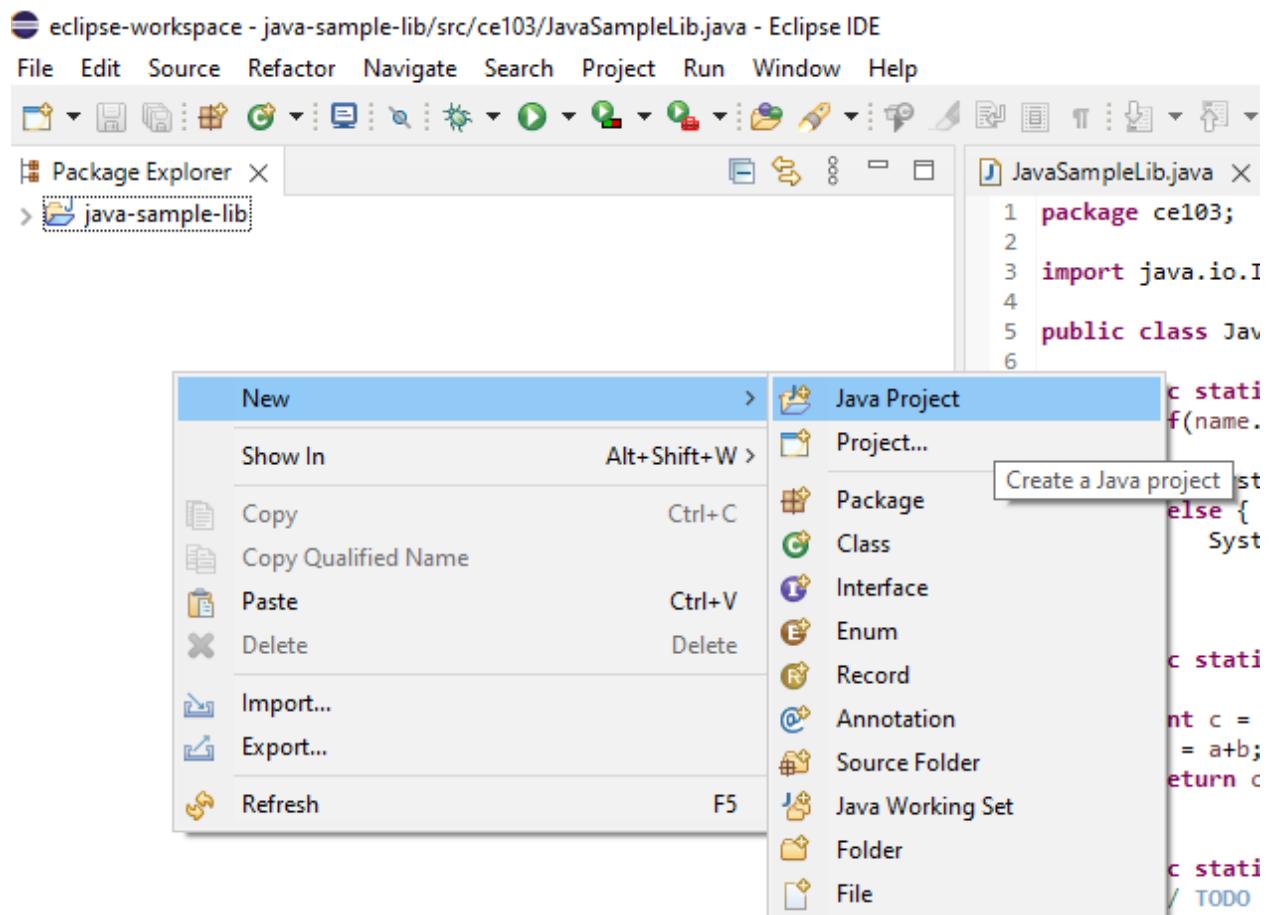
```
java -jar JavaSampleLibExecutable.jar
```

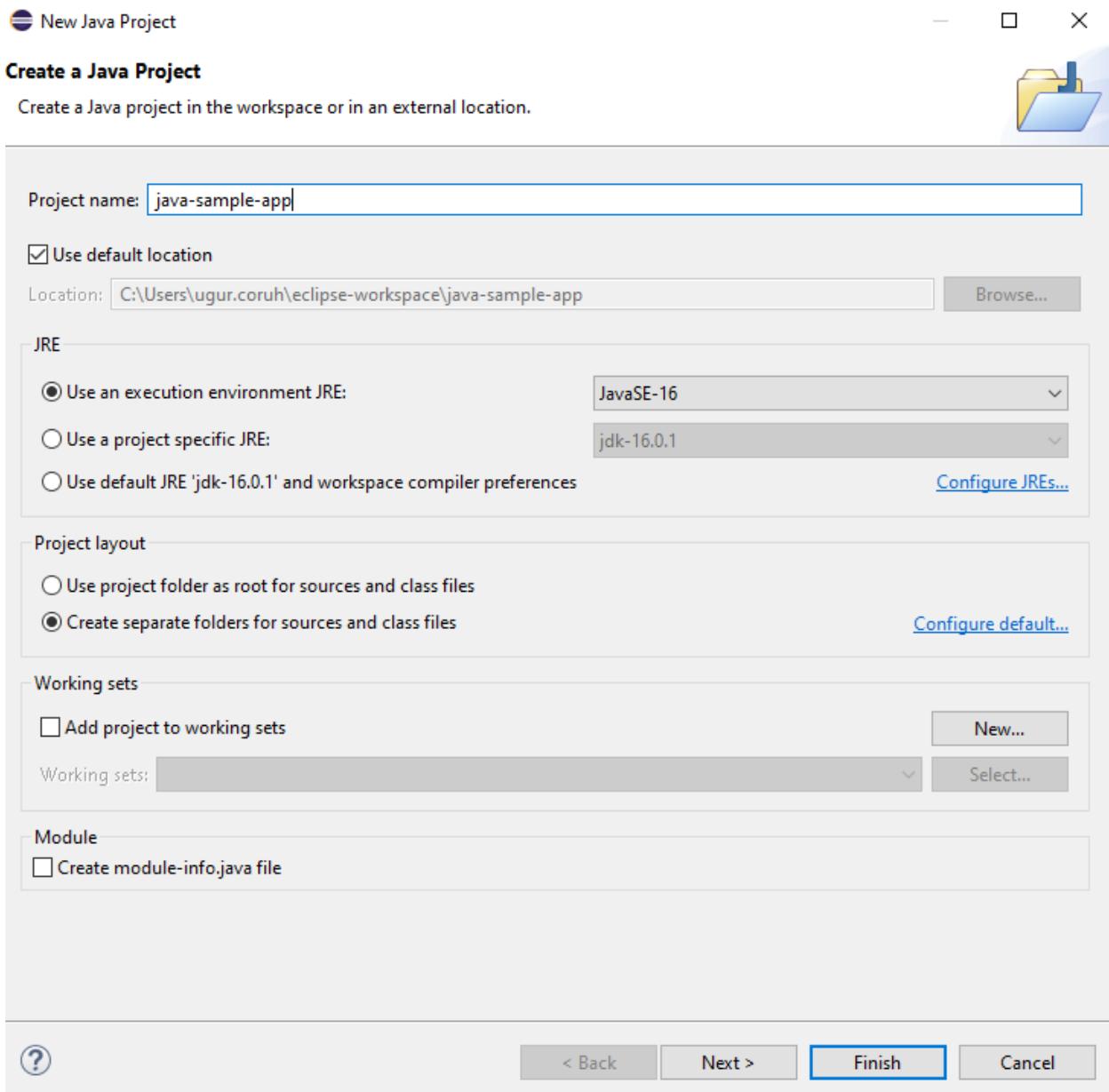


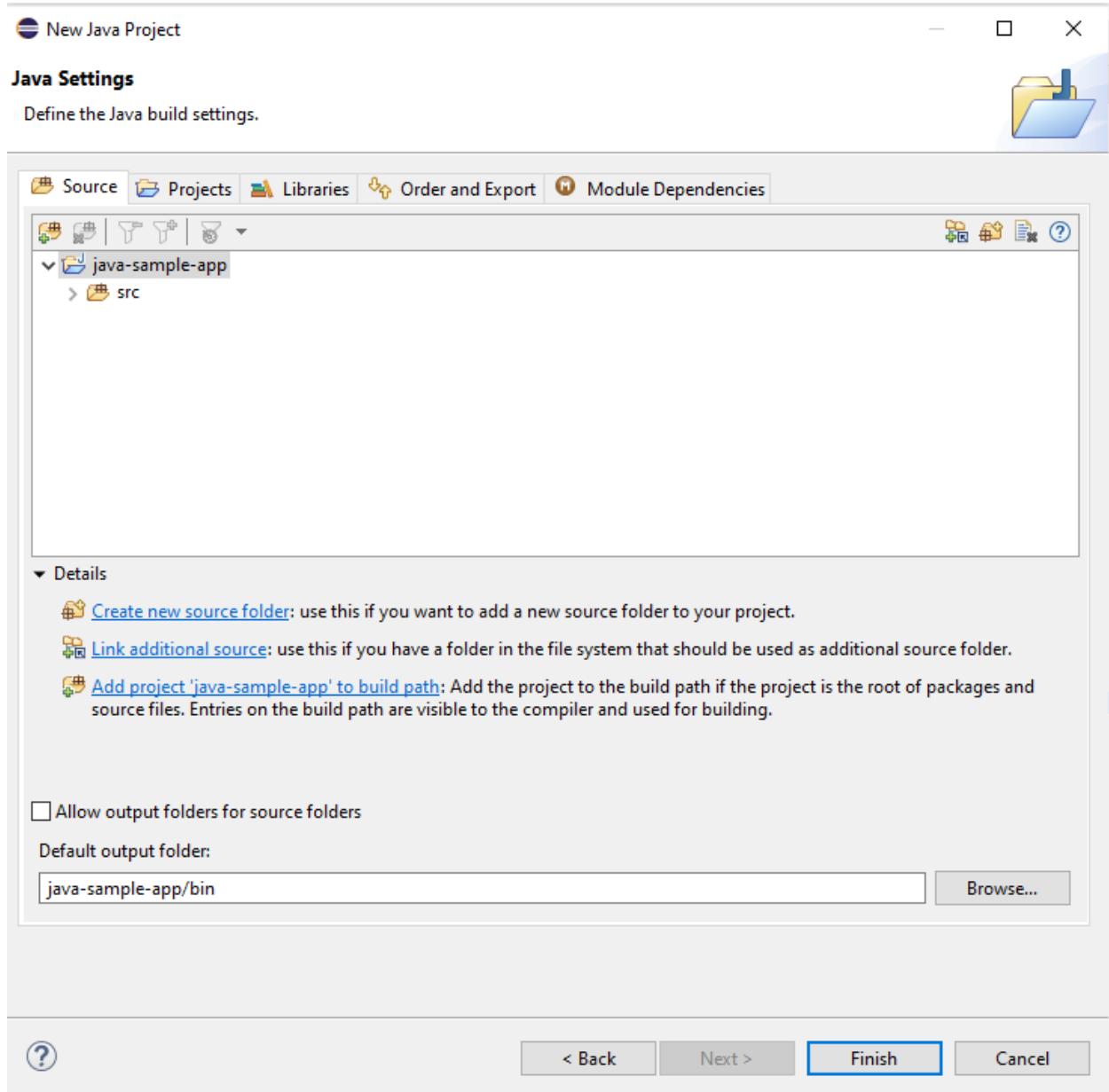
if we click bat file then we will automate command line task for current jar file

```
C:\WINDOWS\system32\cmd.exe
C:\Users\sugur.coruh\Desktop\java-export-sample>java -jar JavaSampleLibExecutable.jar
Hello World!
Hello There
Results is 9
Results is 9
```

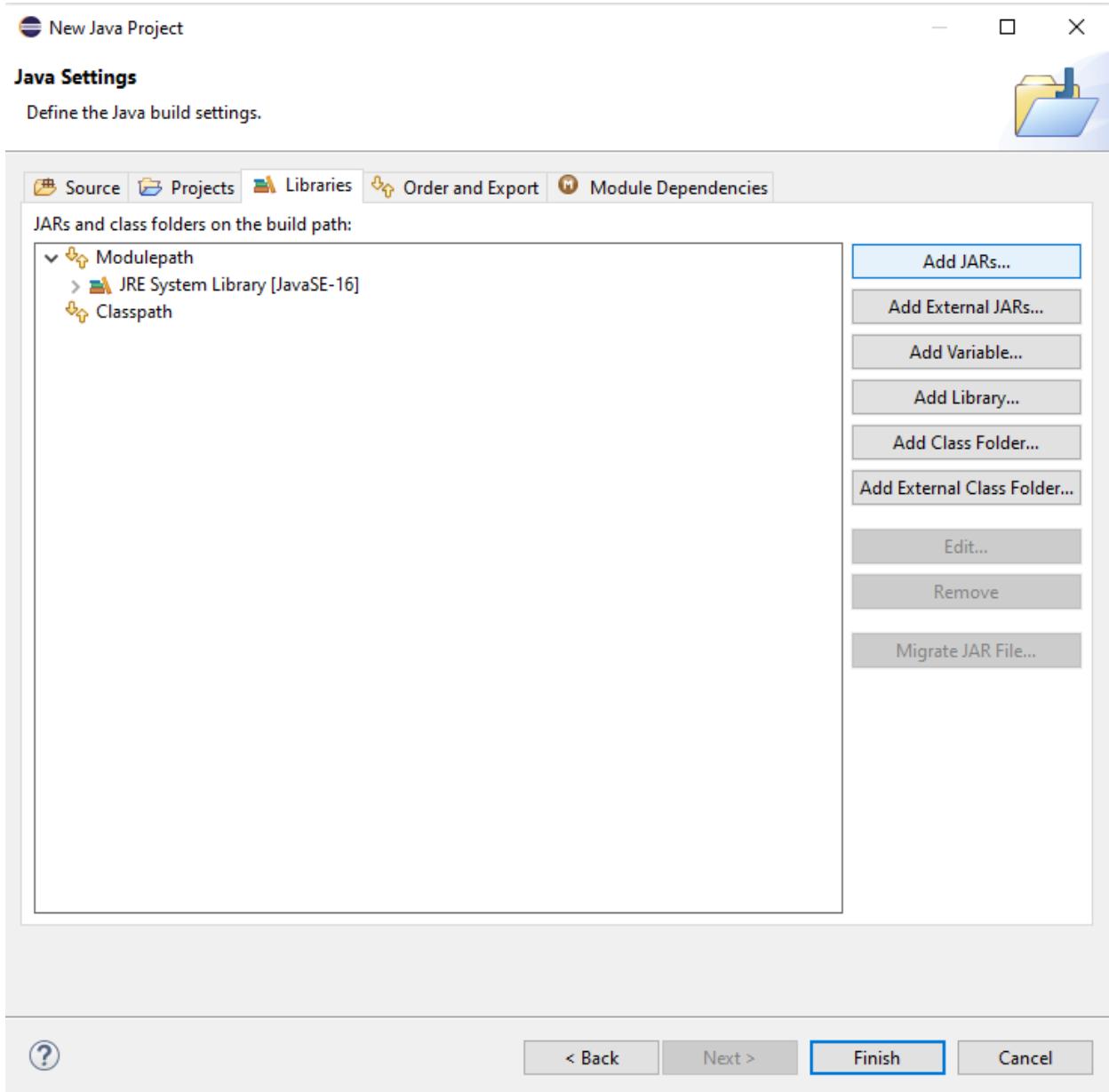
Now return back to our java library and create another console application that use library functions



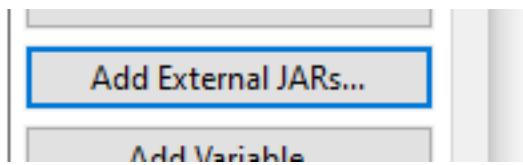




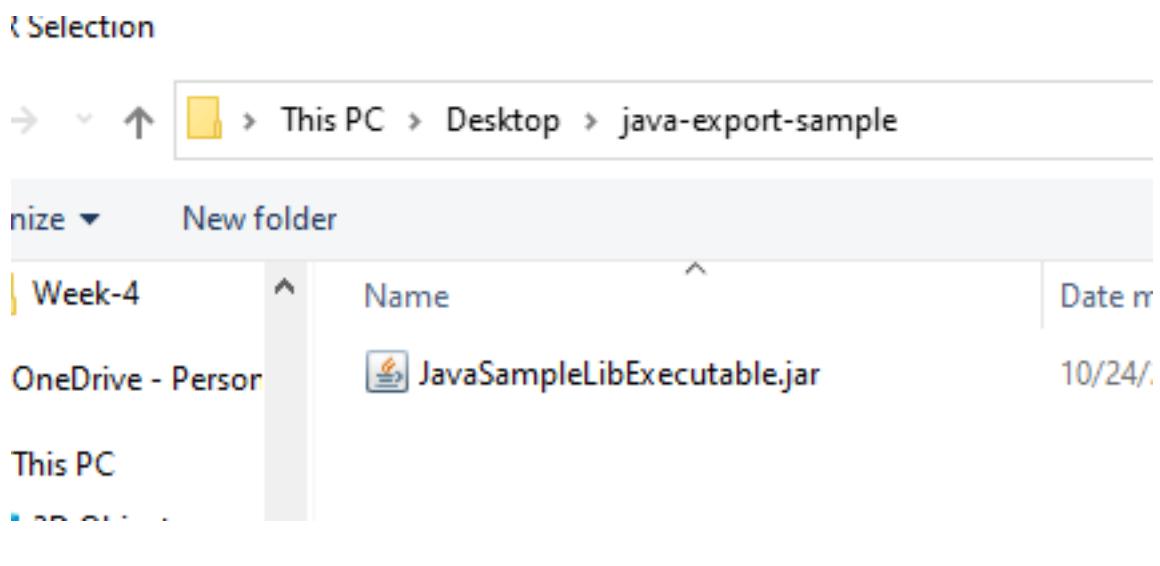
you can set libraries in this step from but our library should exported for our solution



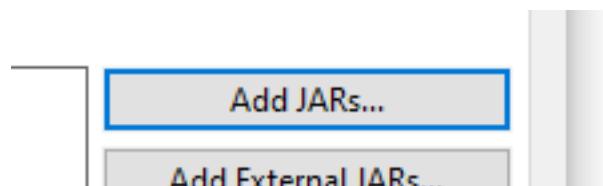
Select Add External JARs...

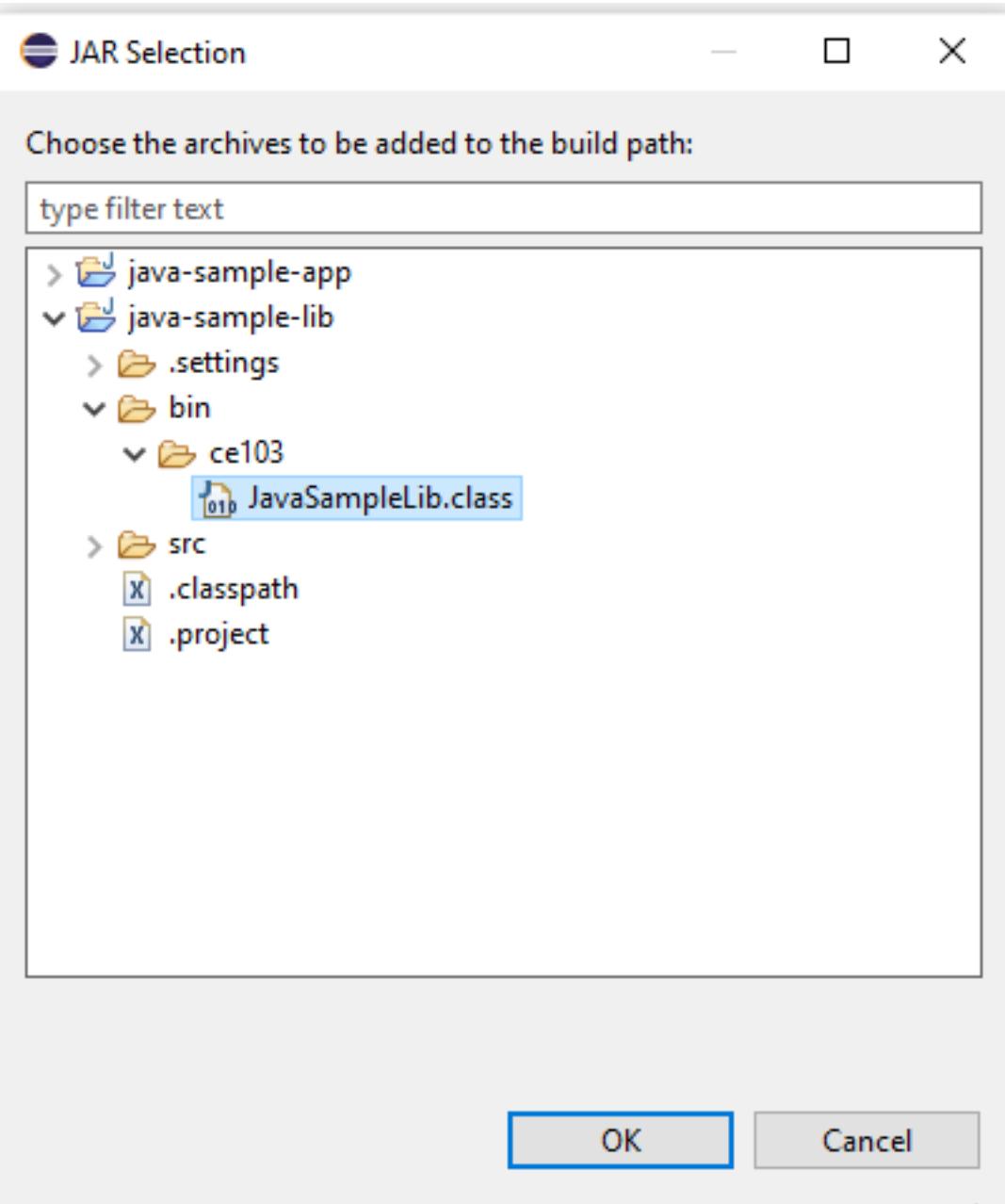


Open Exported jar folder and select

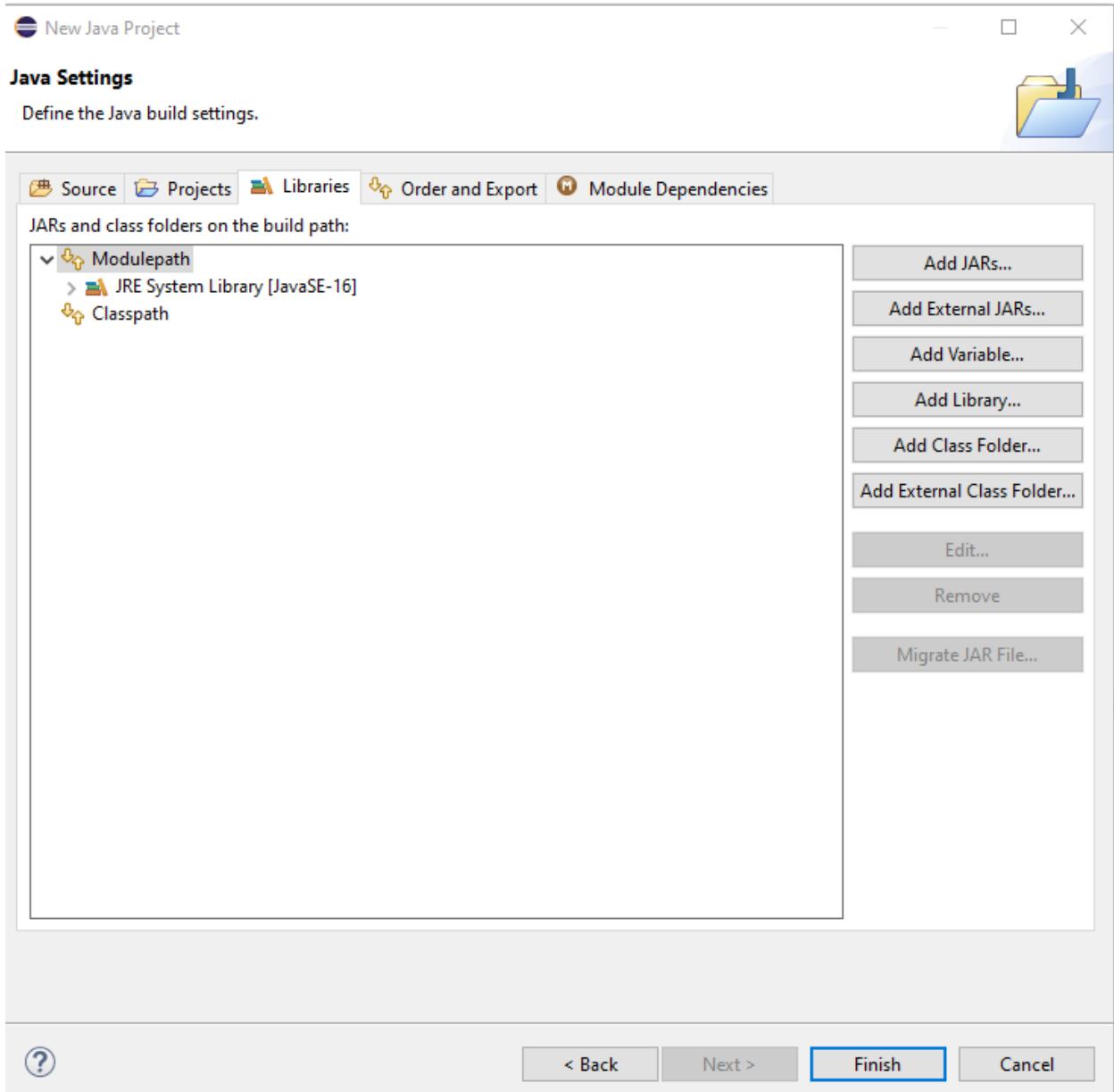


Or we can select by Add jar from current workspace

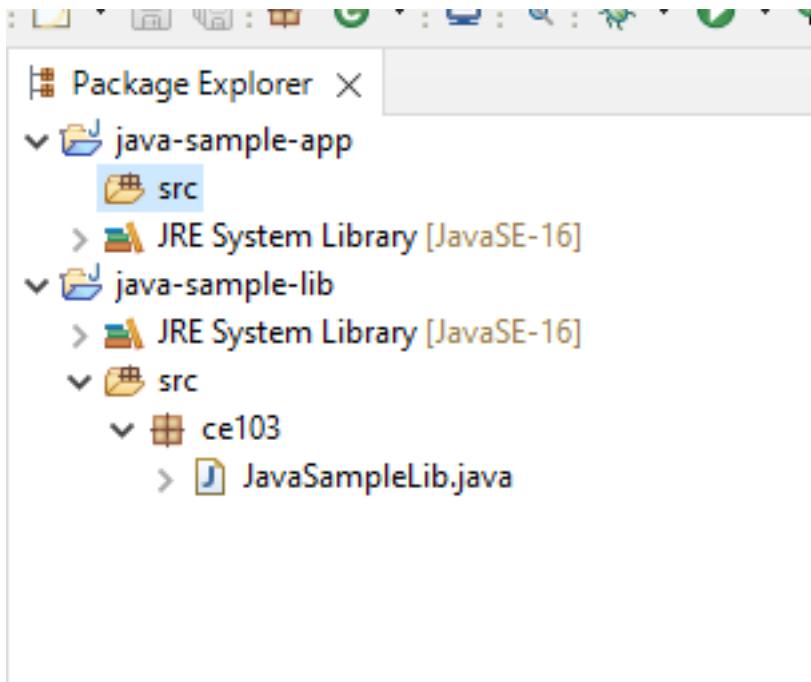




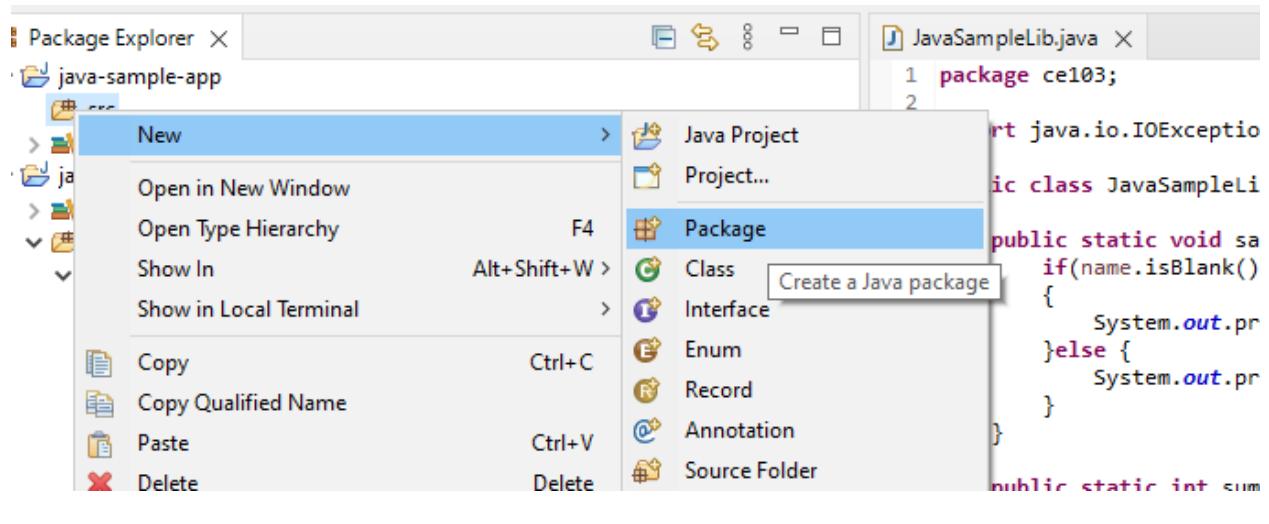
but in this step I won't add anything I'll add references later

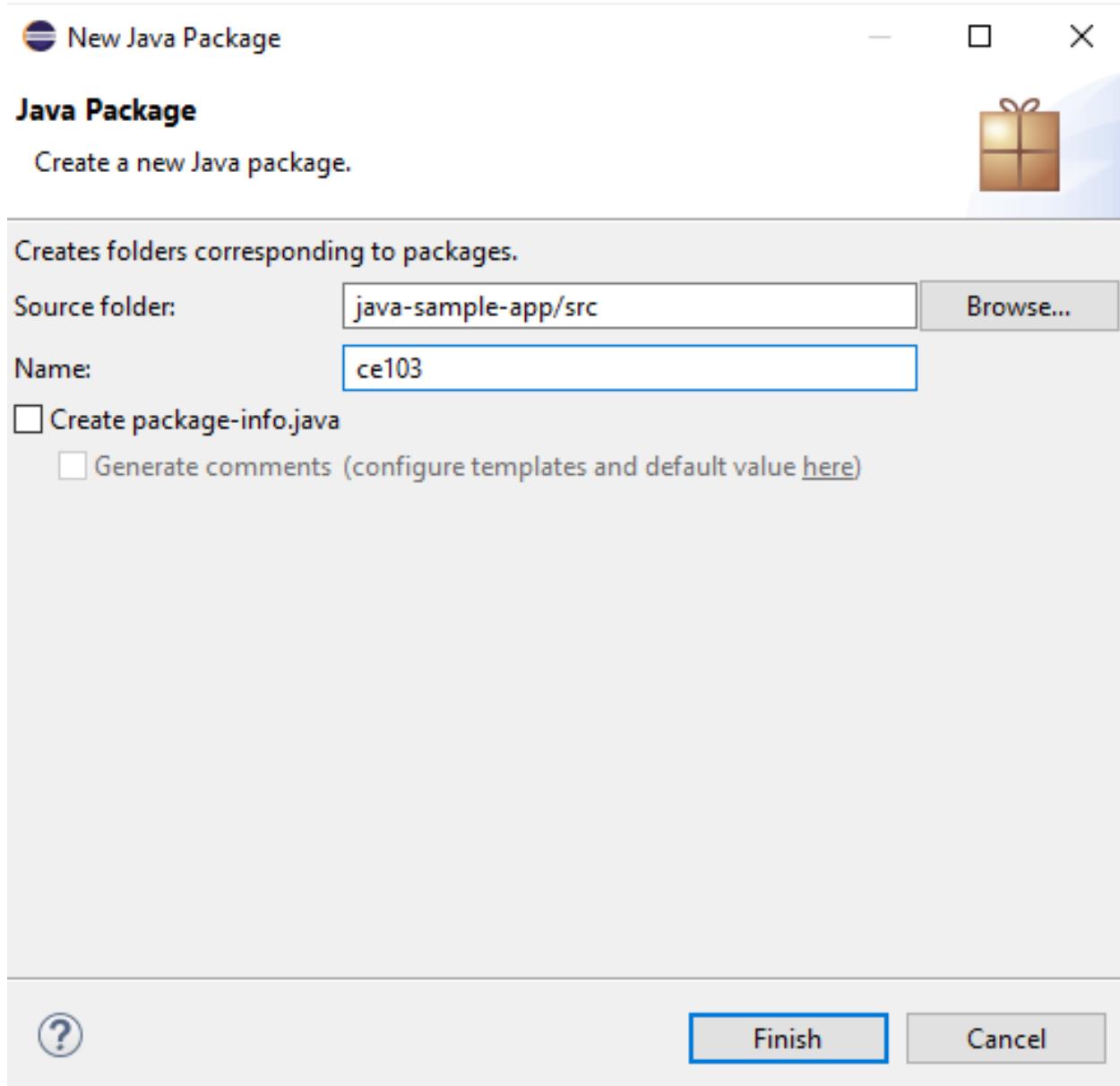


we will have the following project

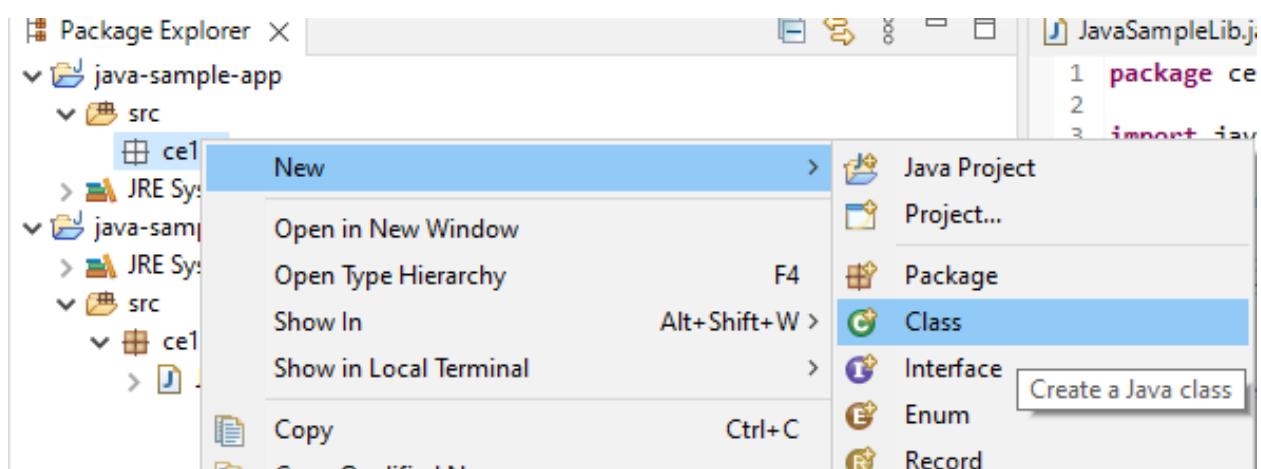


lets create a package





and lets create a main class for our application



check create main function

New Java Class

Java Class

Create a new Java class.

Source folder:

Package:

Enclosing type:

Name:

Modifiers: public package private protected
 abstract final static

Superclass:

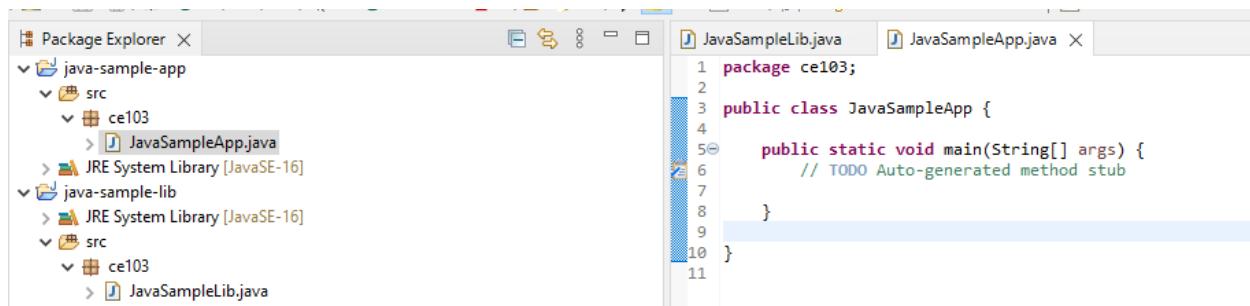
Interfaces:

Which method stubs would you like to create?

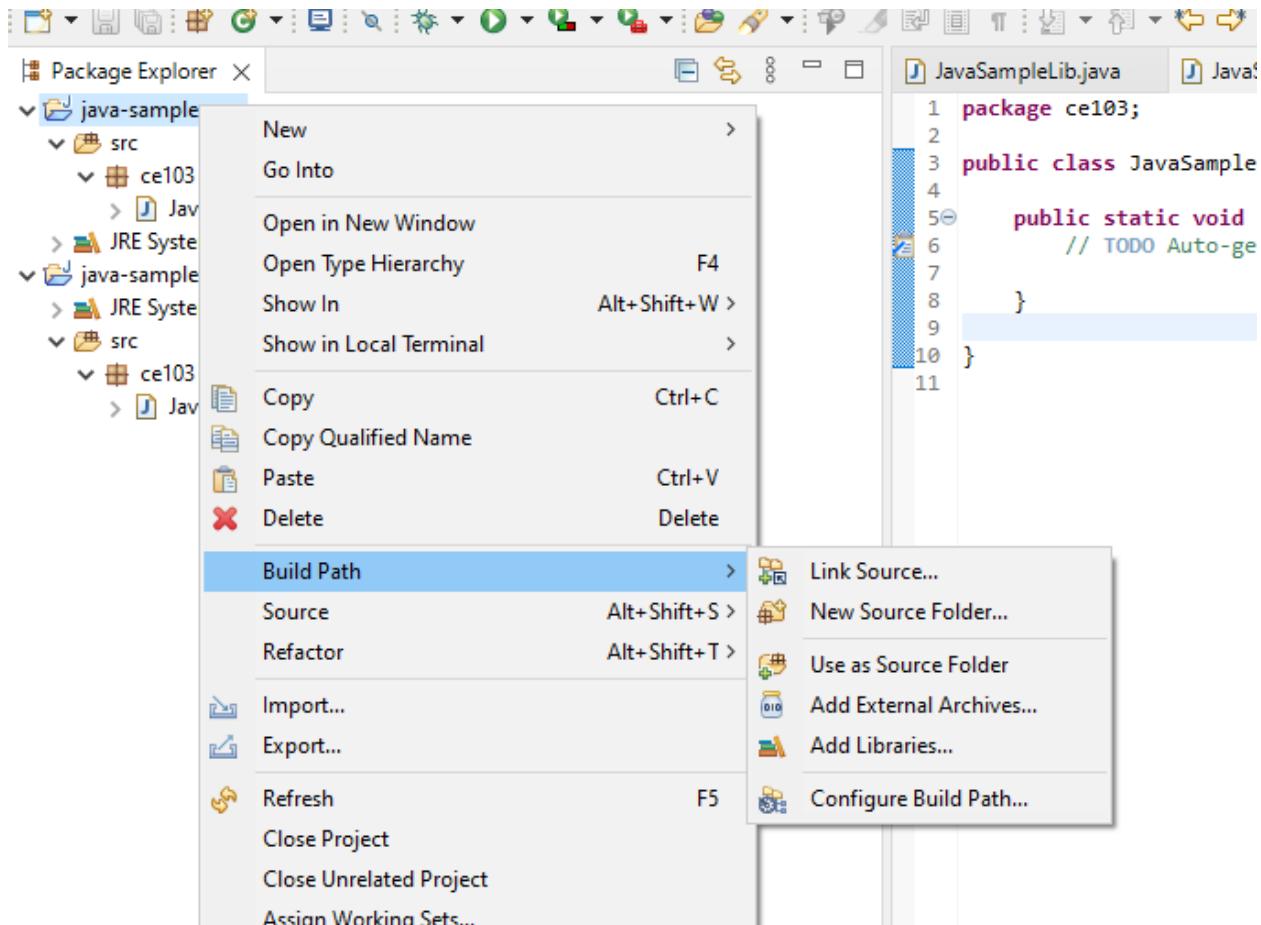
public static void main(String[] args)
 Constructors from superclass
 Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

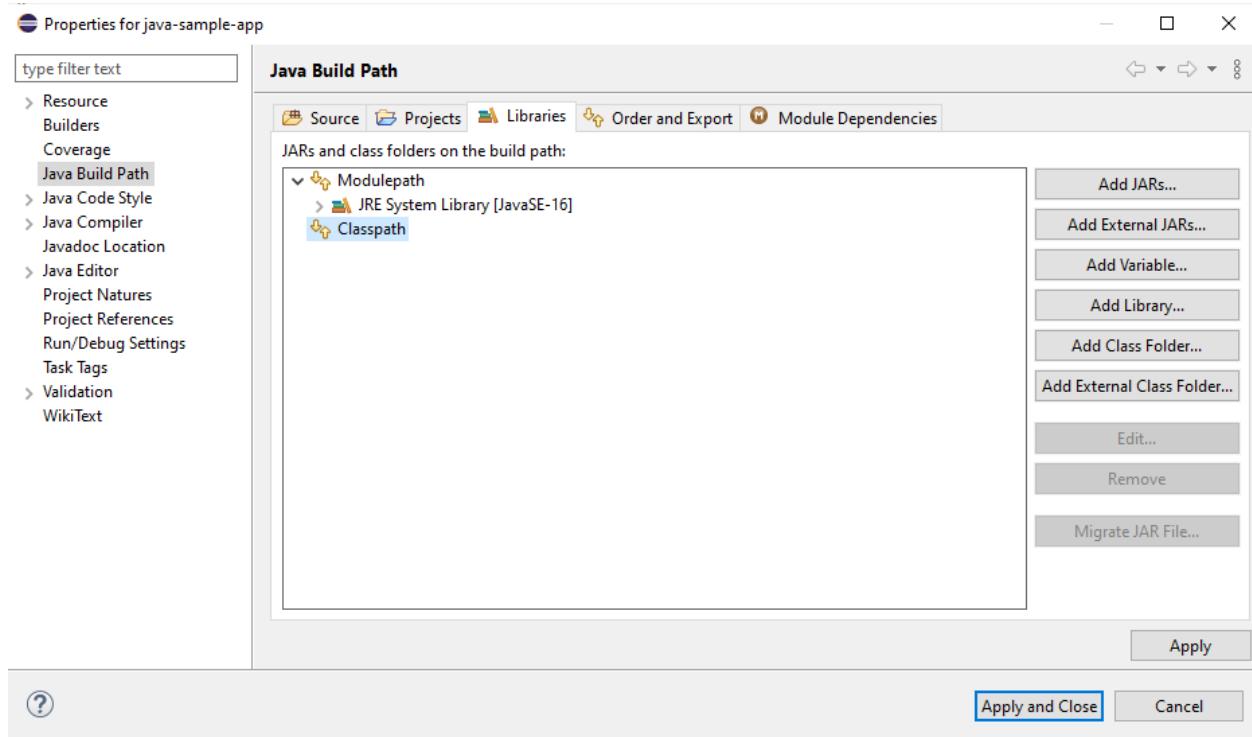
Generate comments



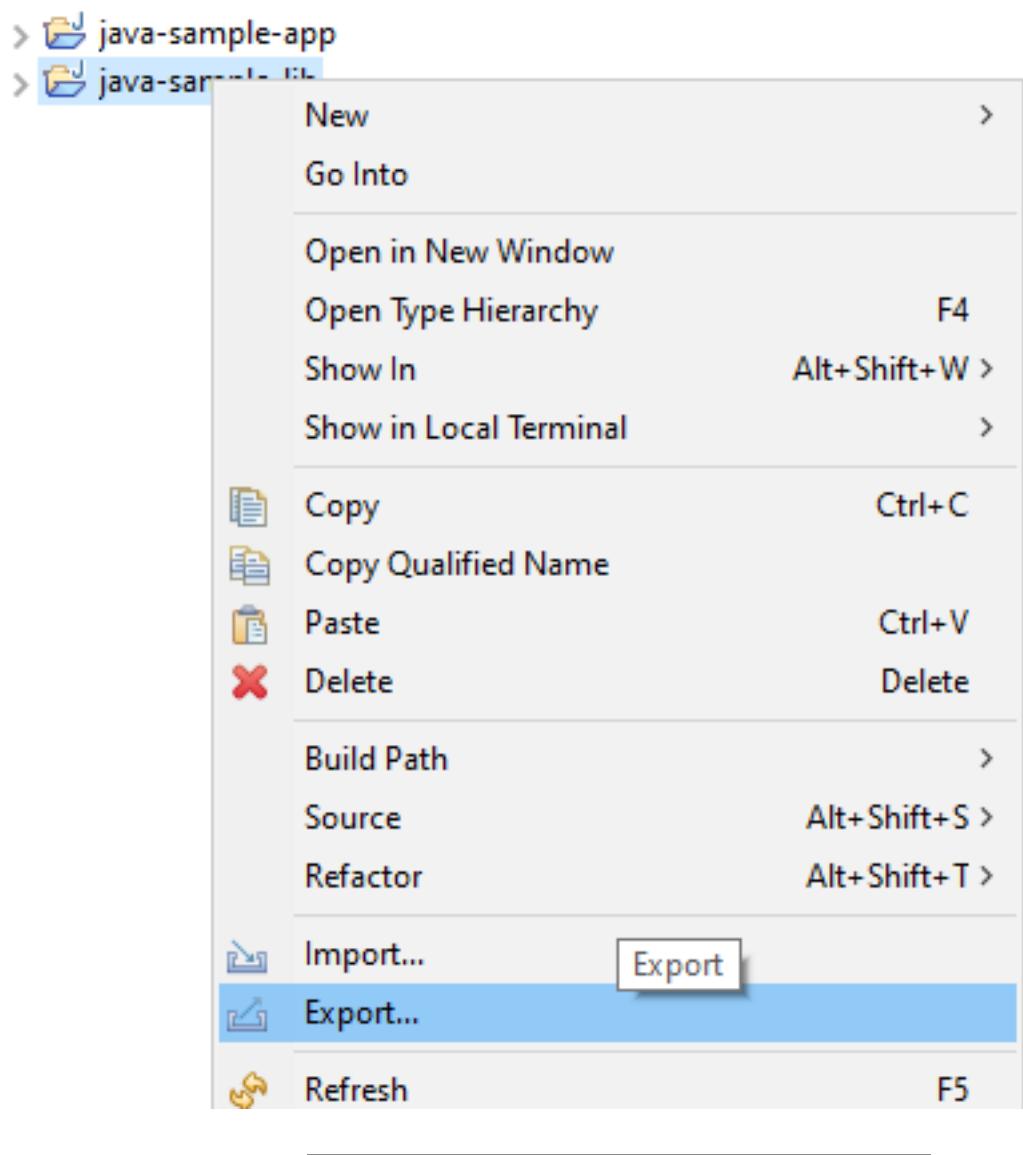
right click to project and add reference



you can enter same configurations from project properties



Lets export our library as a JAR file and then add to our classpath



Select JAR file

Export resources into a JAR file on the local file system

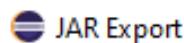
Select an export wizard:

type filter text

- ▼ **Install**
 - Installed Software Items to File
- ▼ **Java**
 - JAR file**
 - Javadoc
 - Runnable JAR file
- ▼ **Run/Debug**
 - Breakpoints
 - Coverage Session
 - Launch Configurations
- ▼ **Team**

we configured output as

C:\Users\ugur.coruh\Desktop\java-export-sample\JavaSampleLib.jar



JAR File Specification

Define which resources should be exported into the JAR.



Select the resources to export:

- > java-sample-app
- > java-sample-lib

- .classpath
- .project

- Export generated class files and resources
- Export all output folders for checked projects
- Export Java source files and resources
- Export refactorings for checked projects. [Select refactorings...](#)

Select the export destination:

JAR file:

Options:

- Compress the contents of the JAR file
- Add directory entries
- Overwrite existing files without warning

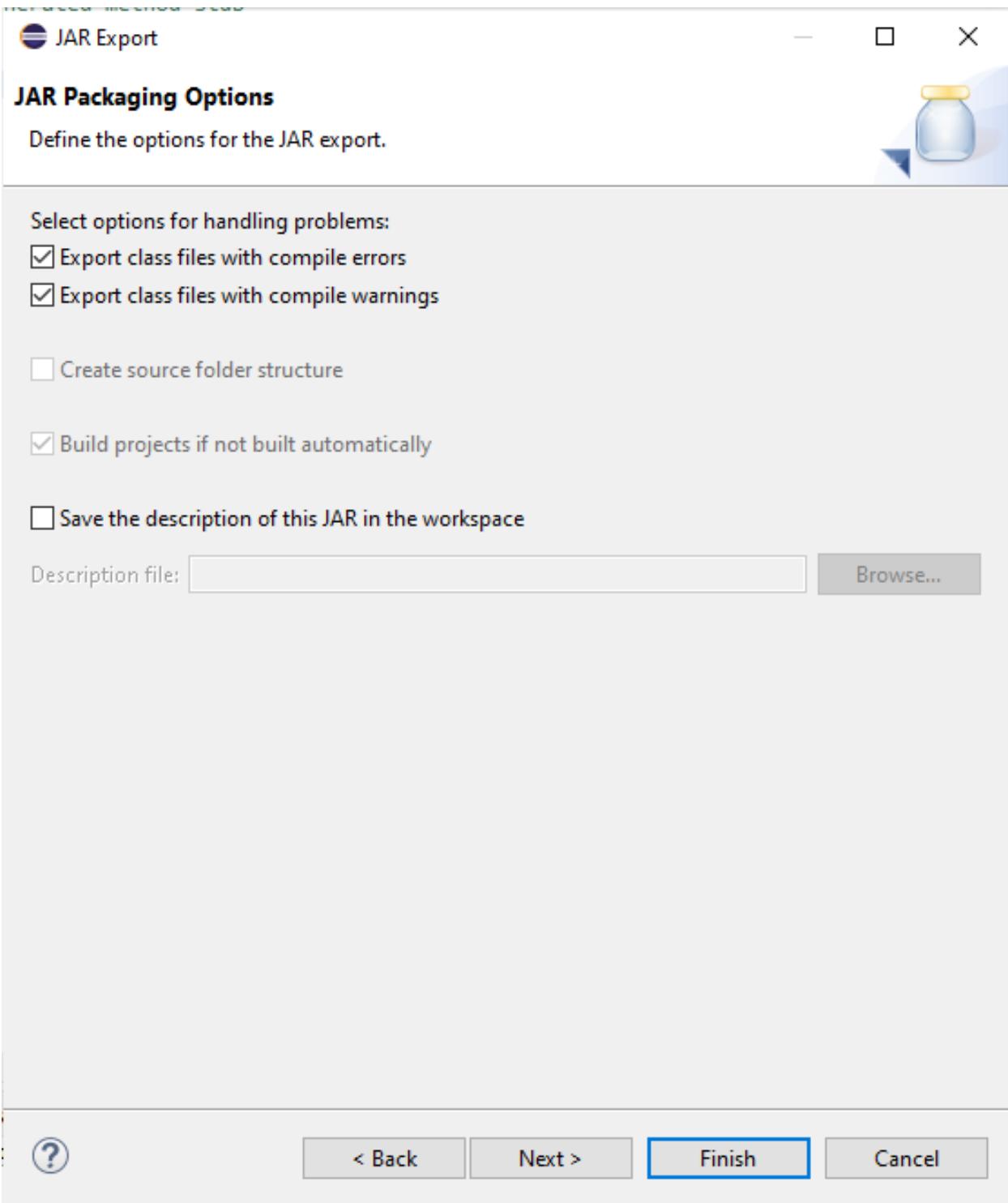


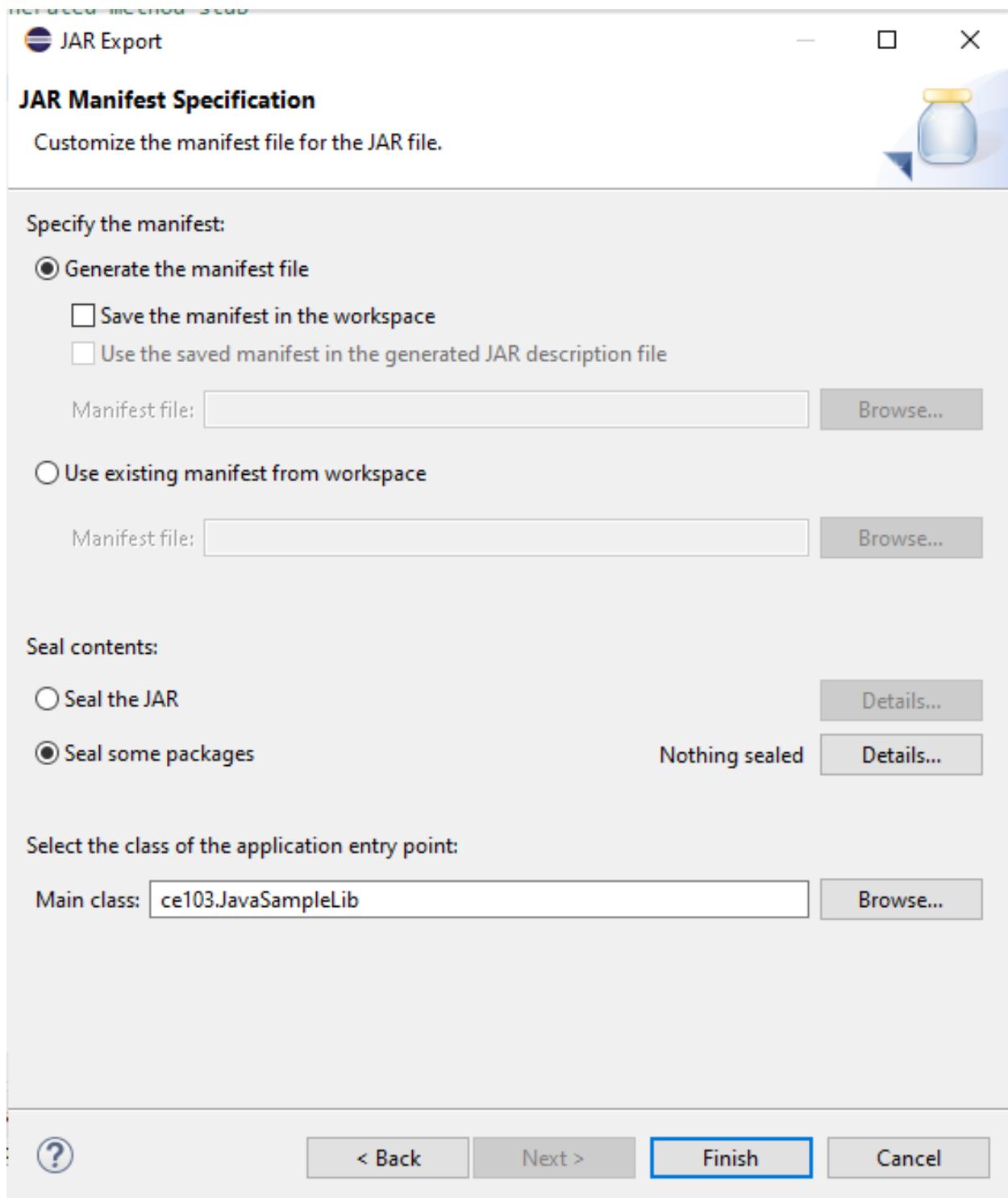
< Back

Next >

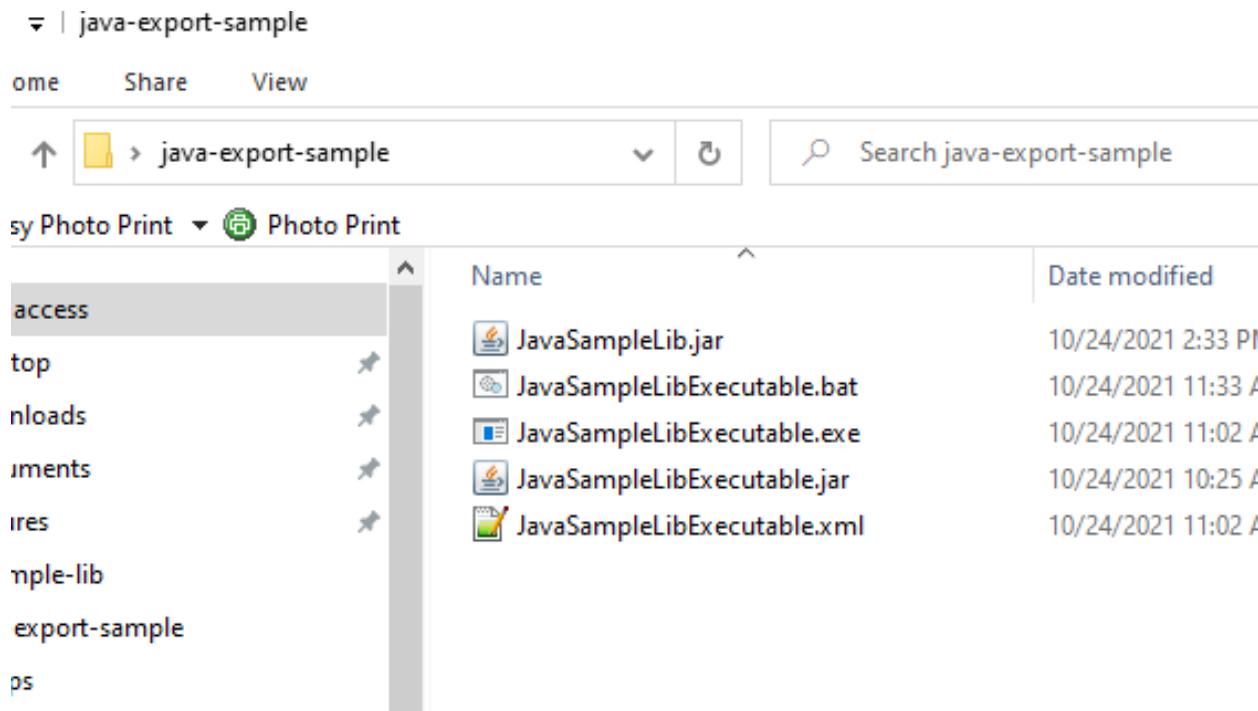
Finish

Cancel



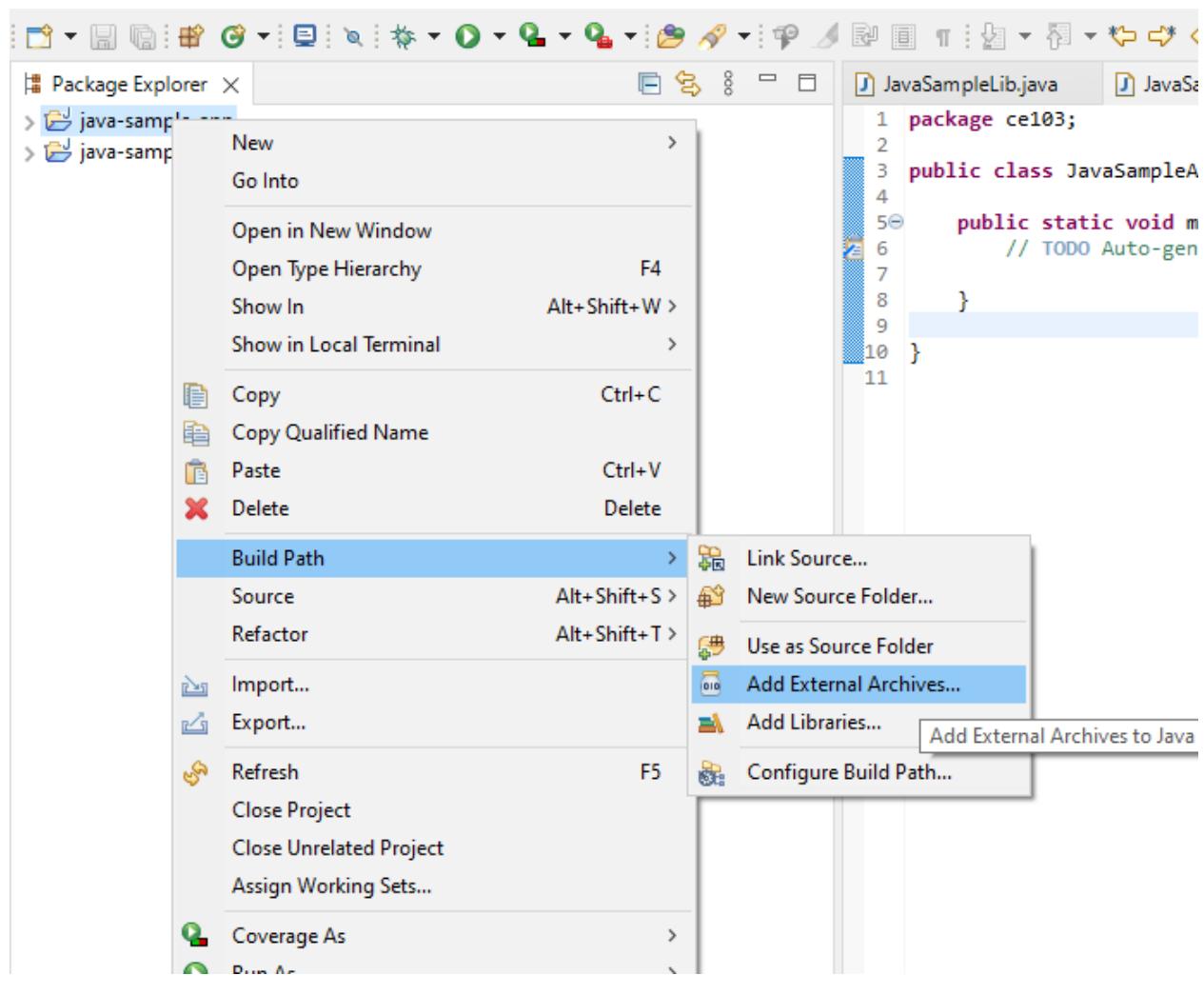


In the same export folder now we have JavaSampleLib.jar

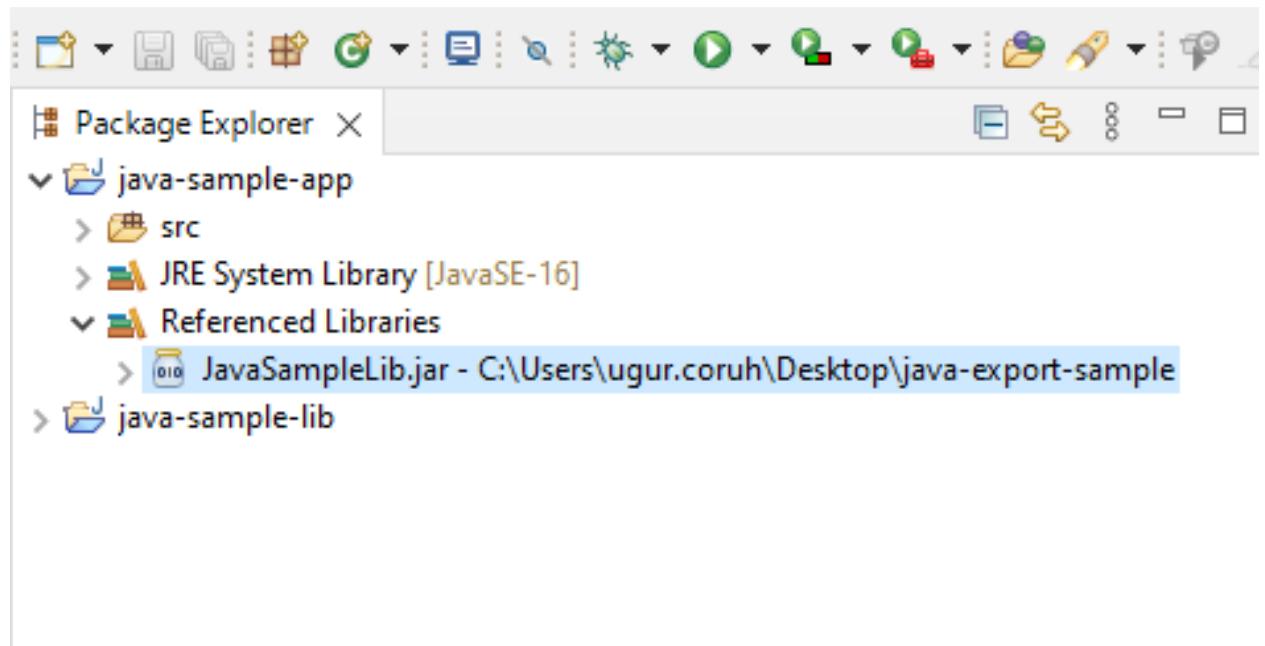


return back to java-sample-app and then add this jar file to our project

Build Path->Add External Archives



you will see its added to reference libraries



in our JavaSampleApp.java we can use the following source codes

```
package ce103;

import java.io.IOException;

public class JavaSampleApp {

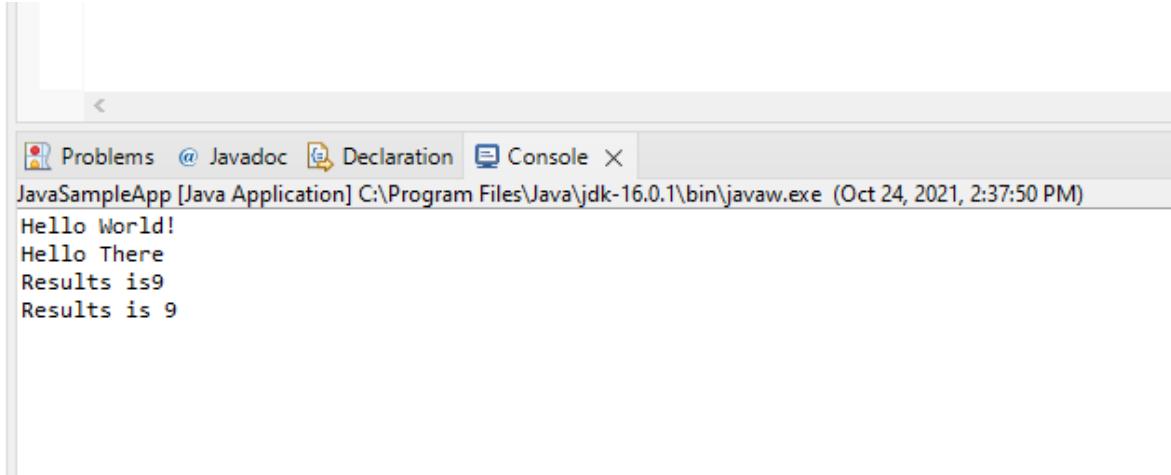
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.out.println("Hello World!");

        JavaSampleLib.sayHelloTo("Computer");
        int result = JavaSampleLib.sum(5, 4);
        System.out.println("Results is" + result);
        System.out.printf("Results is %d \n", result);

        try {
            System.in.read();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

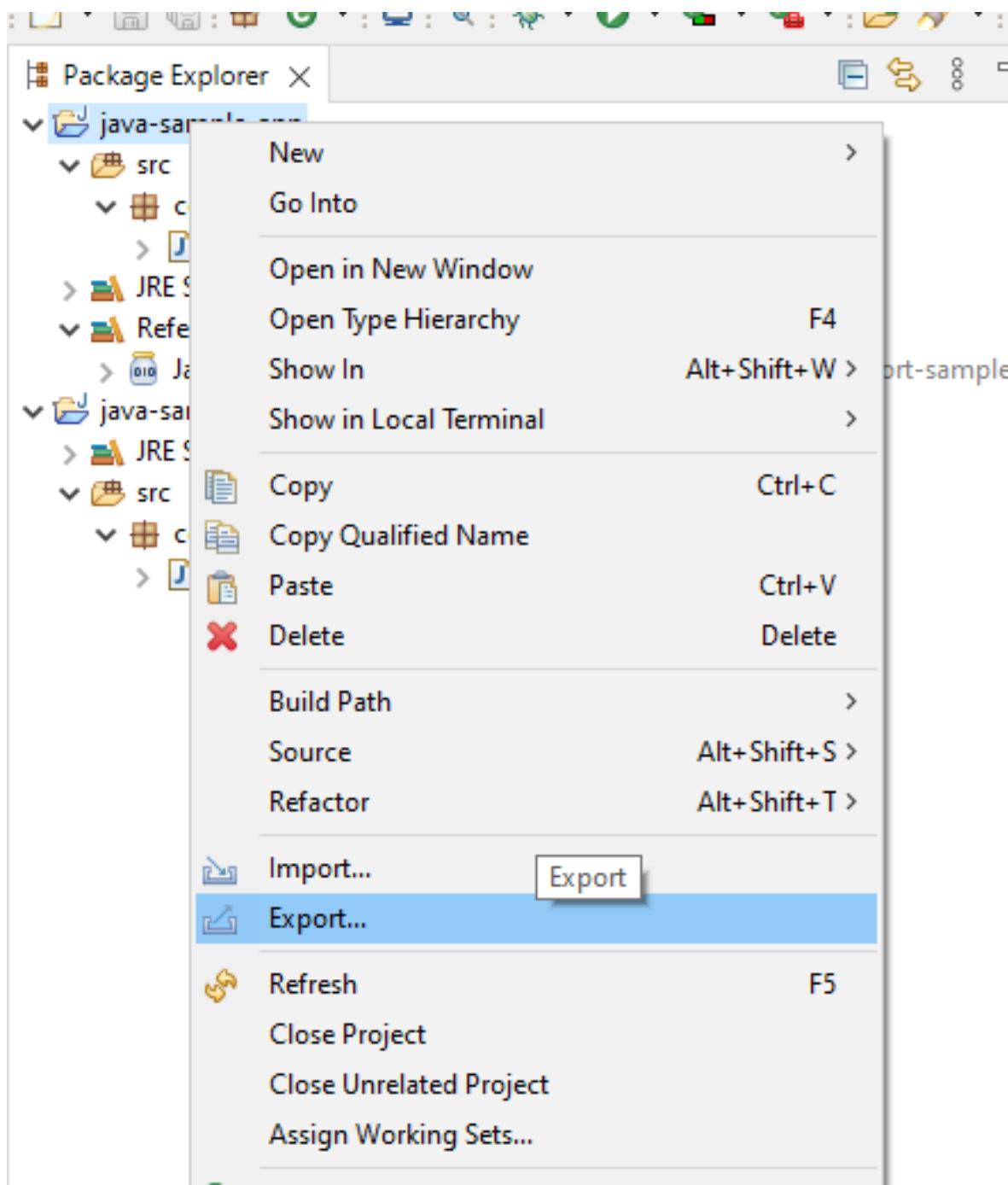
When we run application we will see similar output



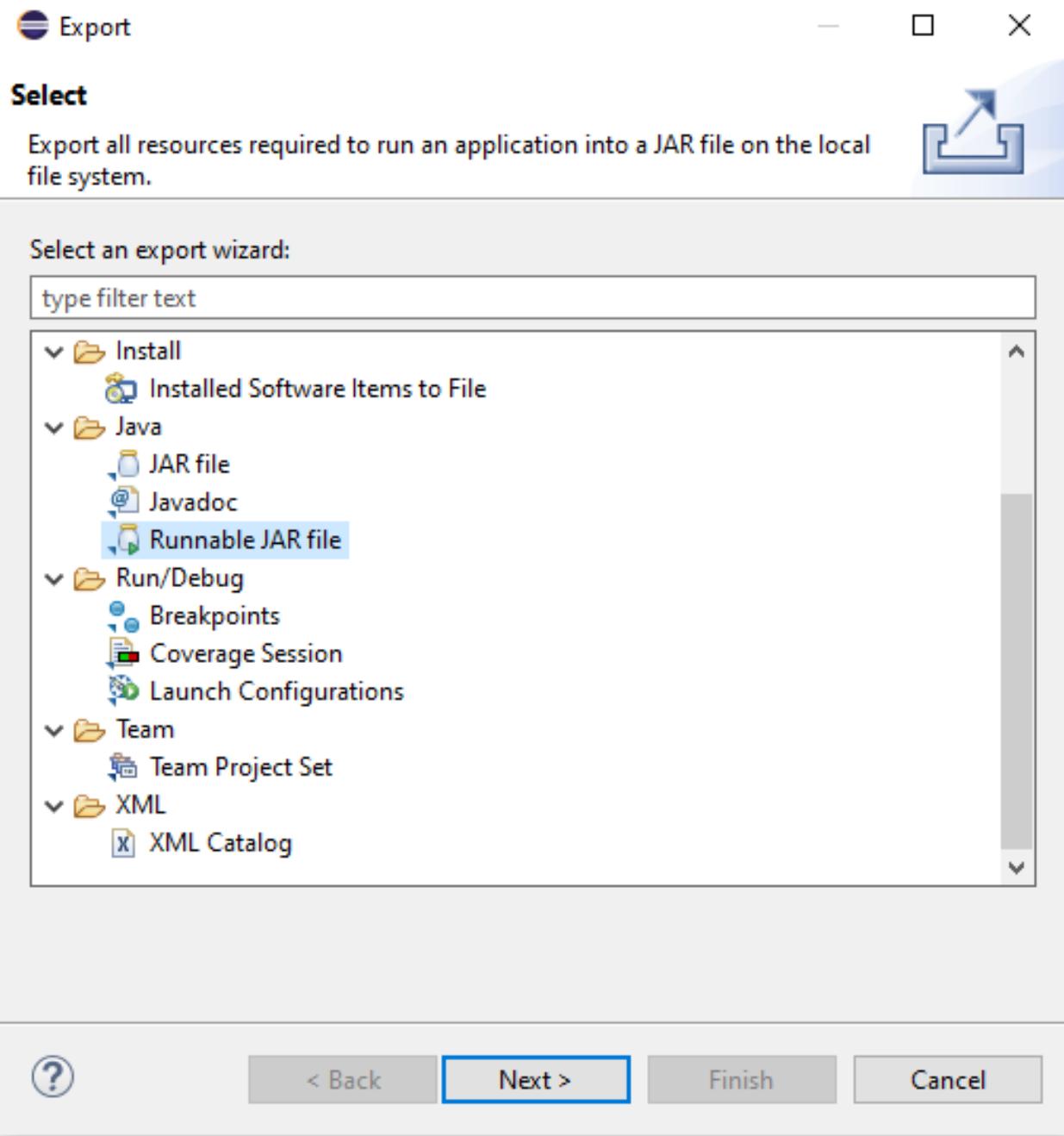
The screenshot shows an IDE interface with a toolbar at the top. Below the toolbar, there are three tabs: 'Problems', 'Declaration', and 'Console'. The 'Console' tab is active, indicated by a blue border. The output window displays the following text:

```
JavaSampleApp [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (Oct 24, 2021, 2:37:50 PM)
Hello World!
Hello There
Results is9
Results is 9
```

Lets export this application with its dependent library

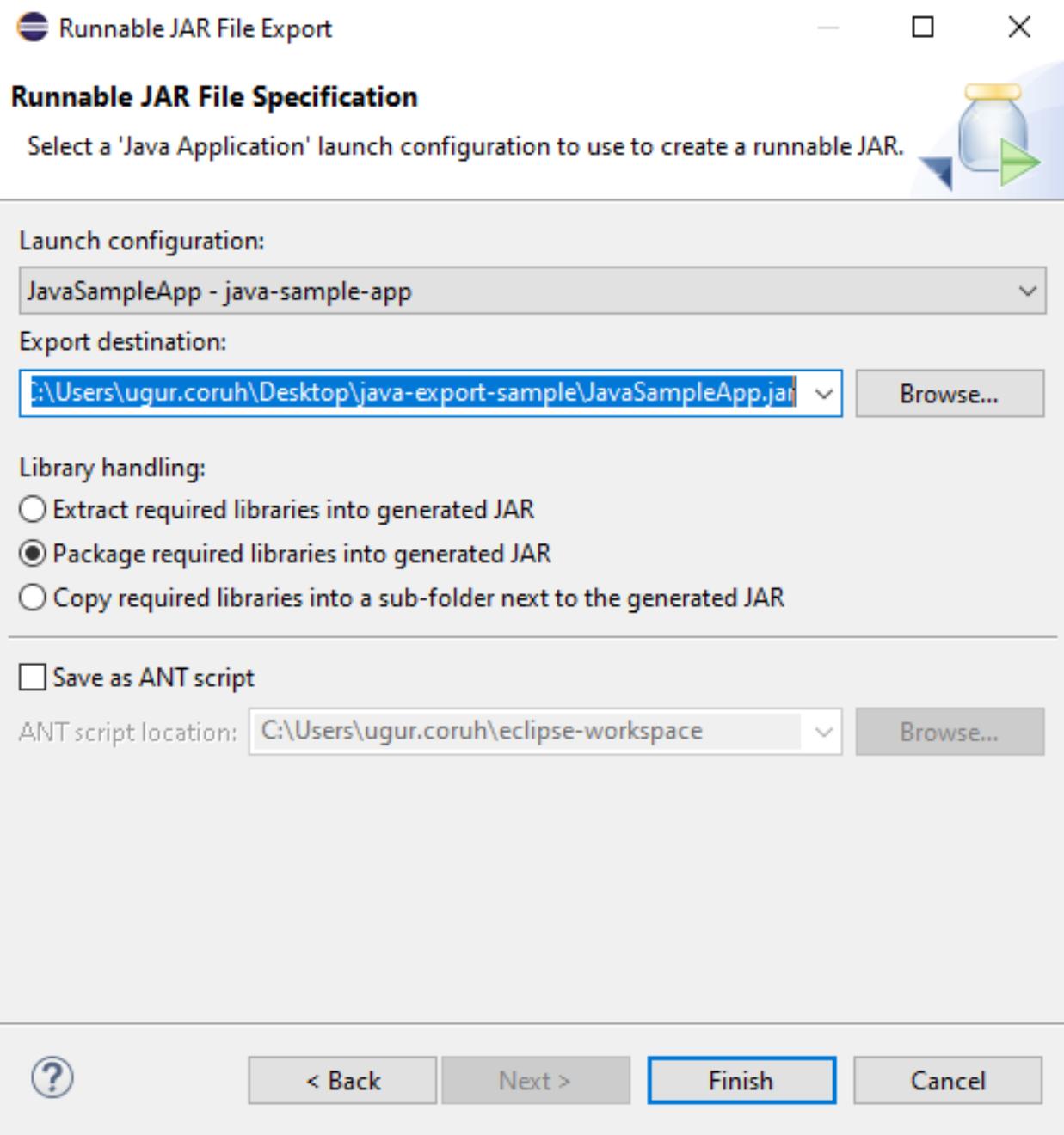


Select runnable jar



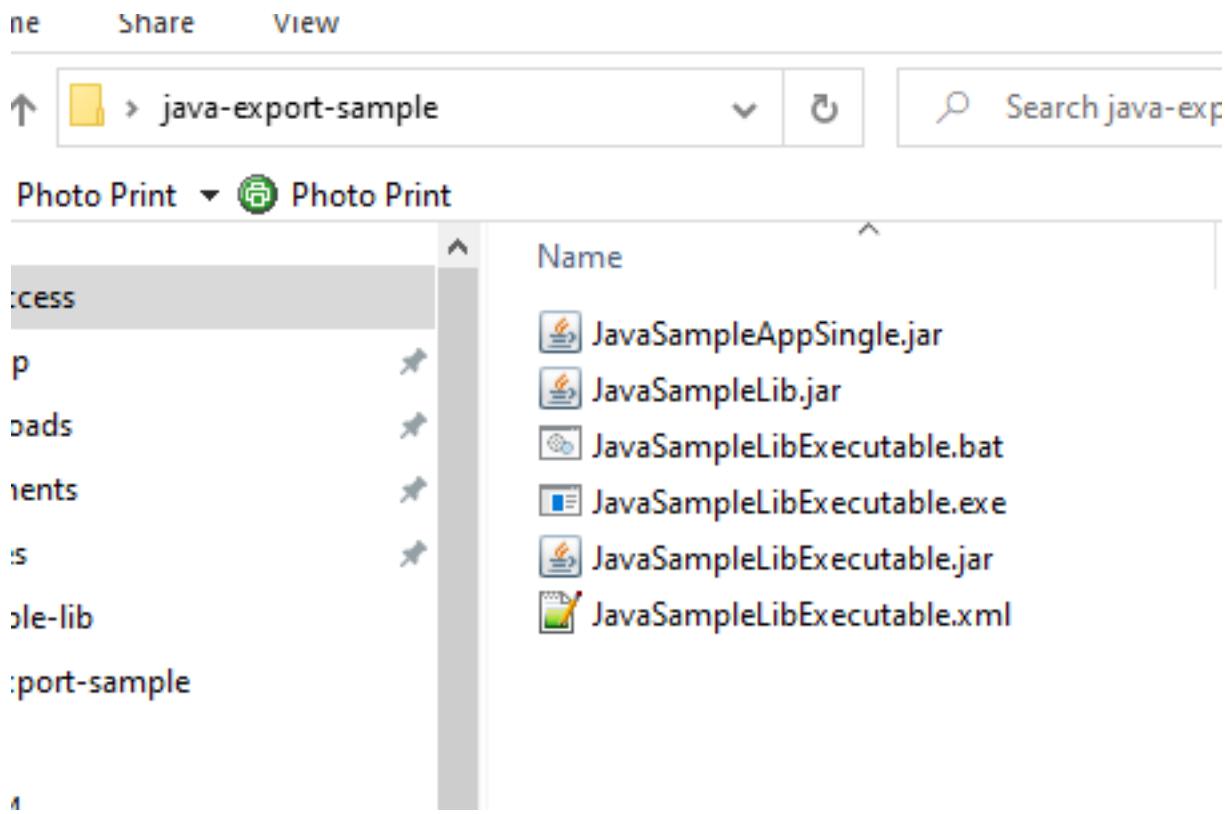
Set Launch configuration and Export destination

C:\Users\ugur.coruh\Desktop\java-export-sample\JavaSampleAppSingle.jar



In this option we will have single jar file

In the export folder we do not see reference libraries

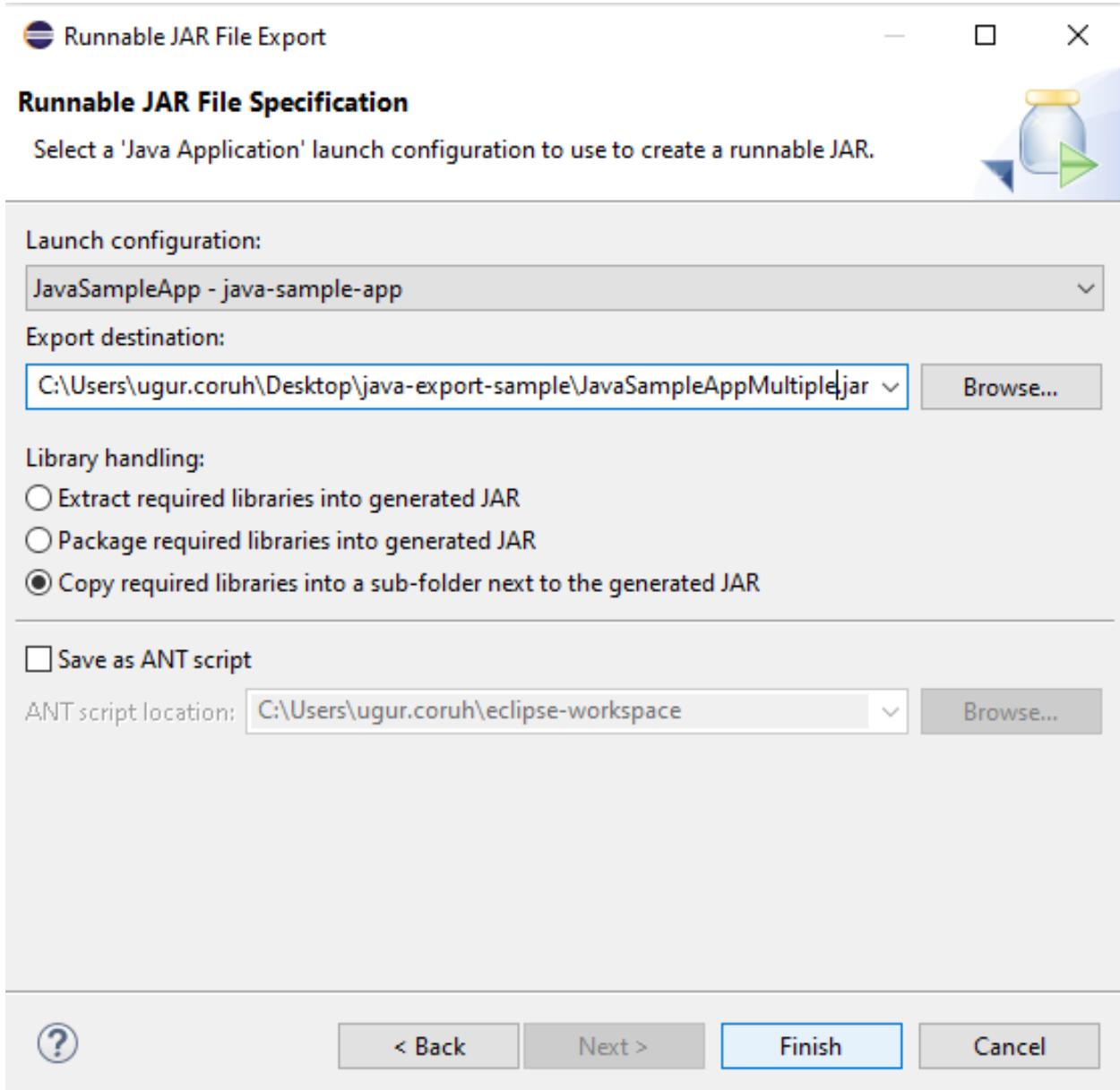


and we can run with command line

```
C:\Users\ugur.coruh\Desktop\java-export-sample>java -jar JavaSampleAppSingle.jar
Hello World!
Hello There
Results is9
Results is 9
```

only change copy required libraries setting and then give a new name for new jar file and export

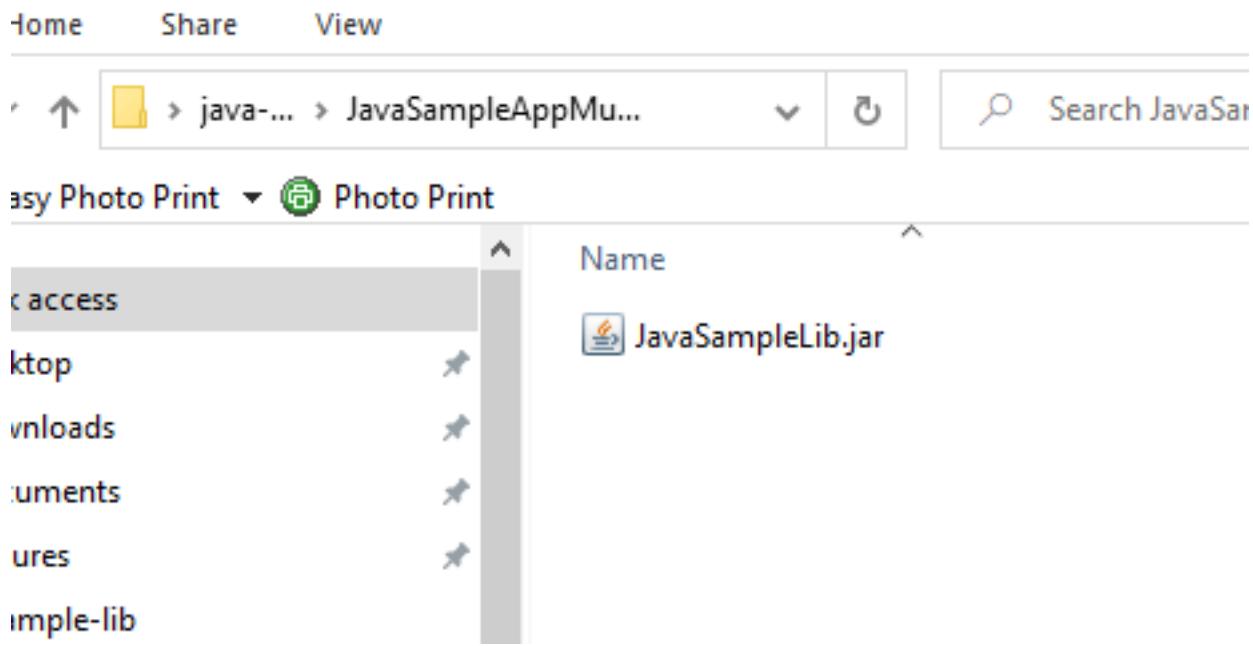
C:\Users\ugur.coruh\Desktop\java-export-sample\JavaSampleAppMultiple.jar



now we have a folder that contains our libraries referenced

	Name
	JavaSampleAppMultiple_lib
	JavaSampleAppMultiple.jar
	JavaSampleAppSingle.jar
	JavaSampleLib.jar

in this file we can find our library



if we test our application we will see it will work

```
C:\Users\ugur.coruh\Desktop\java-export-sample>java -jar JavaSampleAppMultiple.jar
Hello World!
Hello There
Results is 9
Results is 9
```

if we delete JavaSampleLib.jar and then try running application we will get error

```
C:\Users\ugur.coruh\Desktop\java-export-sample>java -jar JavaSampleAppMultiple.jar
Hello World!
Exception in thread "main" java.lang.NoClassDefFoundError: ce103/JavaSampleLib
        at ce103.JavaSampleApp.main(JavaSampleApp.java:12)
Caused by: java.lang.ClassNotFoundException: ce103.JavaSampleLib
        at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.java:636)
        at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:182)
        at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:519)
        ... 1 more
C:\Users\ugur.coruh\Desktop\java-export-sample>
```

0.5 Program Testing

0.6 Unit Test Development

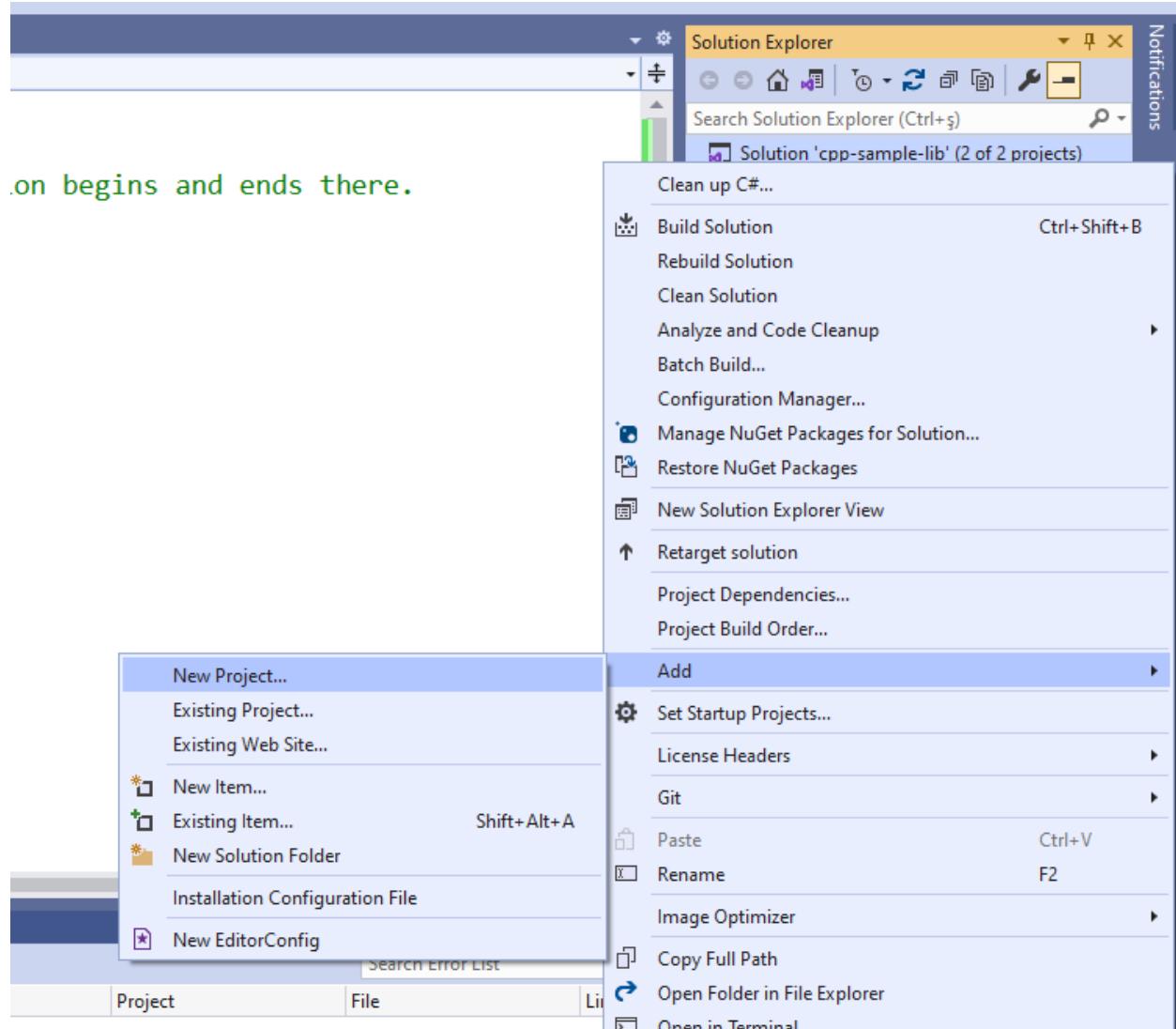
0.6.1 C Unit Tests

0.6.1.1 Visual Studio Community Edition

0.6.2 C++ Unit Tests

0.6.2.1 Visual Studio Community Edition C/C++ için birim testleri yazma - Visual Studio (Windows) | Microsoft Docs¹²

Use cpp-sample-lib project and add



select Native Unit Test

¹²<https://docs.microsoft.com/tr-tr/visualstudio/test/writing-unit-tests-for-c-cpp?view=vs-2019>

The screenshot shows the Microsoft Visual Studio Marketplace search results for the term "Test". At the top, there is a search bar with the placeholder "Search for templates (Alt+S)" and a magnifying glass icon. To the right of the search bar is a "Clear all" link. Below the search bar are three dropdown menus: "C++", "All platforms", and a highlighted "Test" dropdown. A "Not finding what you're looking for? Install more tools and features" link is located in the bottom right corner of the search results area.

Search for templates (Alt+S) Clear all

C++ All platforms Test

Native Unit Test Project
Write C++ unit tests using the native Microsoft CppUnitTest framework.
C++ Windows Test

Google Test
Write C++ unit tests using Google Test. Includes a copy of the Google Test library for use.
C++ Windows Test

Not finding what you're looking for?
[Install more tools and features](#)

set project path and name

The screenshot shows the "Configure your new project" dialog for a "Native Unit Test Project". The title bar displays "Native Unit Test Project" and the "Test" tab is selected. The dialog includes fields for "Project name" (containing "cpp-sample-test") and "Location" (containing "E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programming \Lectures\cell"). There is also a "... " button next to the location field.

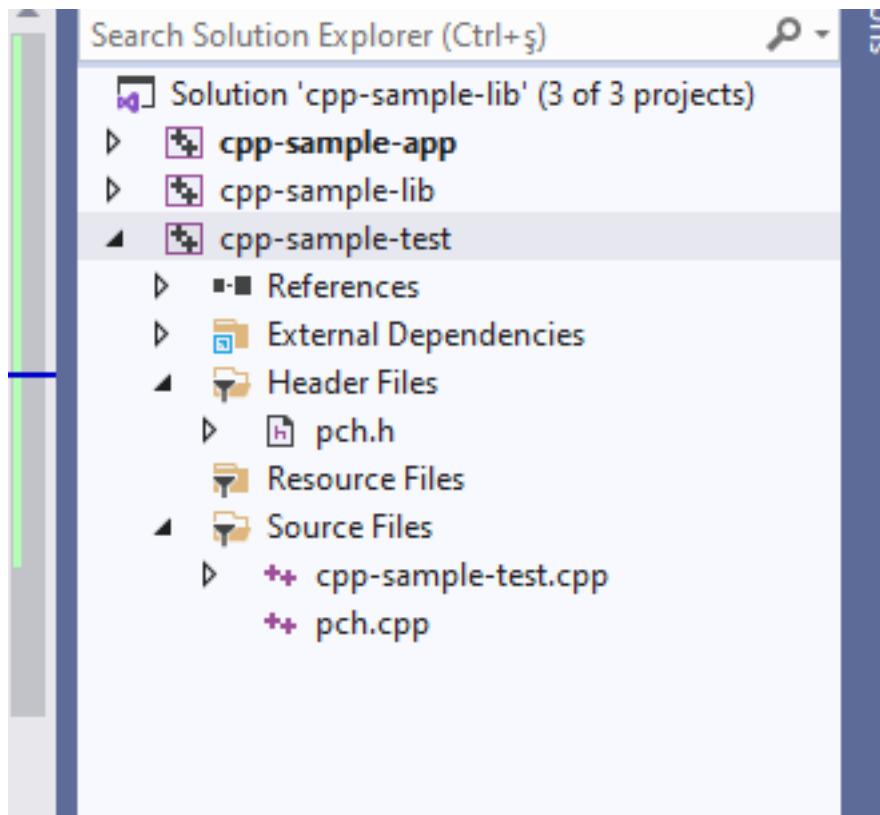
Configure your new project

Native Unit Test Project C++ Windows Test

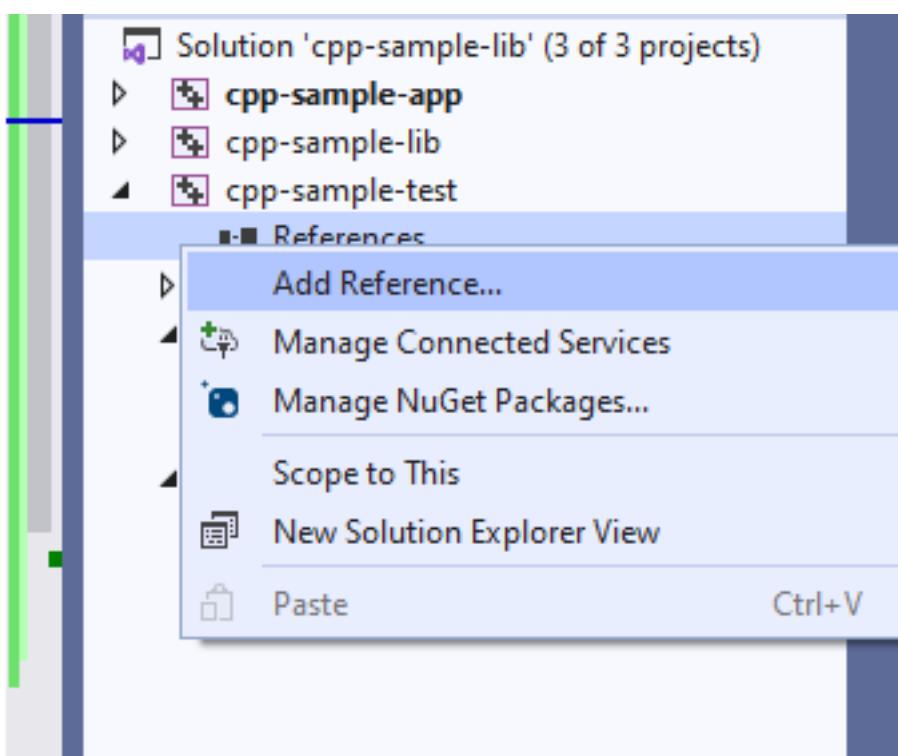
Project name
cpp-sample-test

Location
E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programming \Lectures\cell ...

you will have cpp-sample-test project

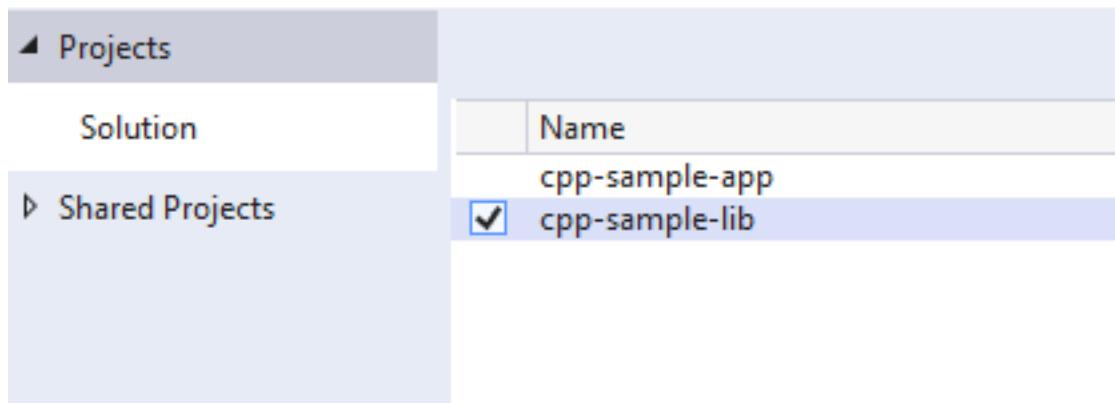


add library project from references



Add cpp-sample-lib to cpp-sample-test project

Add Reference



cpp-sample-test.cpp

```
#include "pch.h"
#include "CppUnitTest.h"
#include "..\cpp-sample-lib\samplelib.h"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace cppsampletest
{
    TEST_CLASS(cppsampletest)
    {
        public:

            TEST_METHOD(TestSumCorrect)
            {
                Assert::AreEqual(9, sum(4, 5));
            }

            TEST_METHOD(TestSumInCorrect)
            {
                Assert::AreEqual(10, sum(4, 5));
            }
    };
}
```

The screenshot shows the Visual Studio IDE interface. At the top, there are tabs for various files: `cpp-sample-test.cpp`, `cpp-sample-app.cpp`, `samplelib.h`, `pch.h`, `pch.cpp`, and `cpp-sample-lib.cpp`. The main editor window displays the following C++ code:

```
8     {
9         TEST_CLASS(cppsampletest)
10        {
11            public:
12
13            TEST_METHOD(TestSumCorrect)
14            {
15                Assert::AreEqual(9, sum(4, 5));
16            }
17
18            TEST_METHOD(TestSumInCorrect)
19            {
20                Assert::AreEqual(10, sum(4, 5));
21            }
22        };
23    }
24
```

The code editor highlights the first test method with a green checkmark icon, indicating it has passed. The second test method is highlighted with a red X icon, indicating it has failed. The Test Explorer window below shows the test results:

Test	Duration	Traits	Error Message
cpp-sample-test (2)	253 ms		
cppsampletest (2)	253 ms		
cppsampletest (2)	253 ms		
TestSumCorrect	< 1 ms		
TestSumInCorrect	253 ms		Assert failed. Expected:<10> Actual:<9>

0.6.3 C# Unit Tests

0.6.4 Visual Studio Community Edition (MSTestV2+.Net)

Install extension fine code coverage

<https://marketplace.visualstudio.com/items?itemName=FortuneNgwenya.FineCodeCoverage>

Create a .Net Framework Library

Search for templates (Alt+S)  Clear all

C# Windows Library

 Class Library (Universal Windows)
A project for creating a managed class library (.dll) for Universal Windows Platform (UWP) apps.
C# Windows Library UWP

 Class Library (.NET Framework)
A project for creating a C# class library (.dll)
C# Windows Library

 WPF Custom Control Library (.NET Framework)
Windows Presentation Foundation custom control library
C# XAML Windows Desktop Library

 WPF User Control Library (.NET Framework)
Windows Presentation Foundation user control library
C# XAML Windows Desktop Library

set project framework and path

Configure your new project

Class Library (.NET Framework) C# Windows Library

Project name

cs-lib-sample

Location

C:\Users\ugur.coruh\Desktop\cs-lib-sample\

...

Solution name i

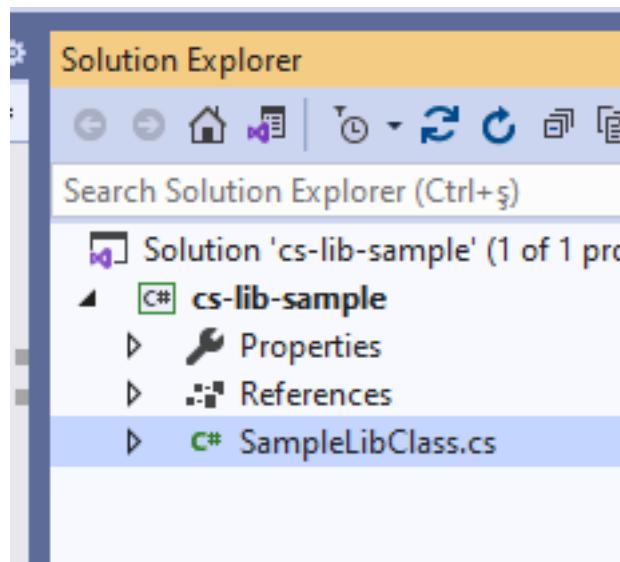
cs-lib-sample

Place solution and project in the same directory

Framework

.NET Framework 3.0

Create library functions



```
using System;
using System.Collections.Generic;
using System.Text;
```

```

namespace cs_lib_sample
{
    public class SampleLibClass
    {
        public static string sayHelloTo(string name)
        {
            string result = String.Empty;

            if (!String.IsNullOrEmpty(name))
            {
                result = "Hello " + name;
            }
            else
            {
                result = "Hello There";
            }

            Console.WriteLine(result);

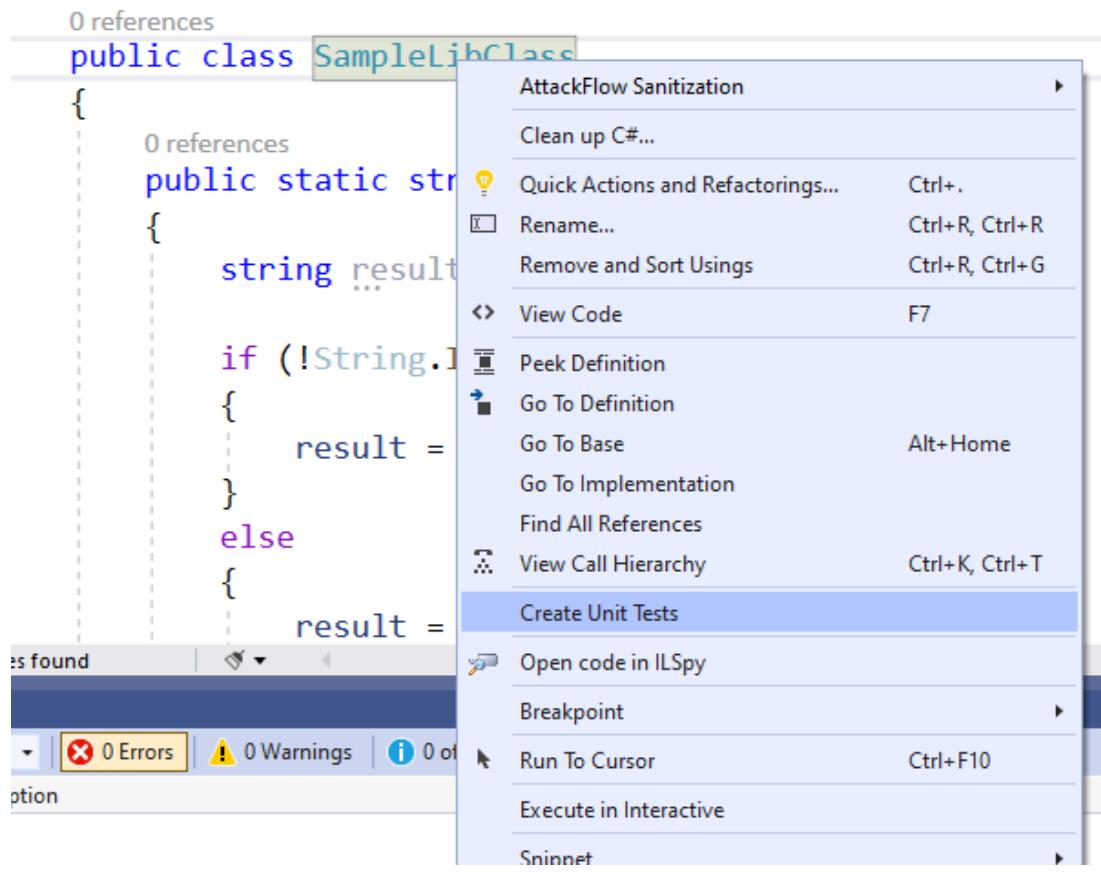
            return result;
        }

        public static int sum(int a, int b)
        {
            int c = 0;
            c = a + b;
            return c;
        }

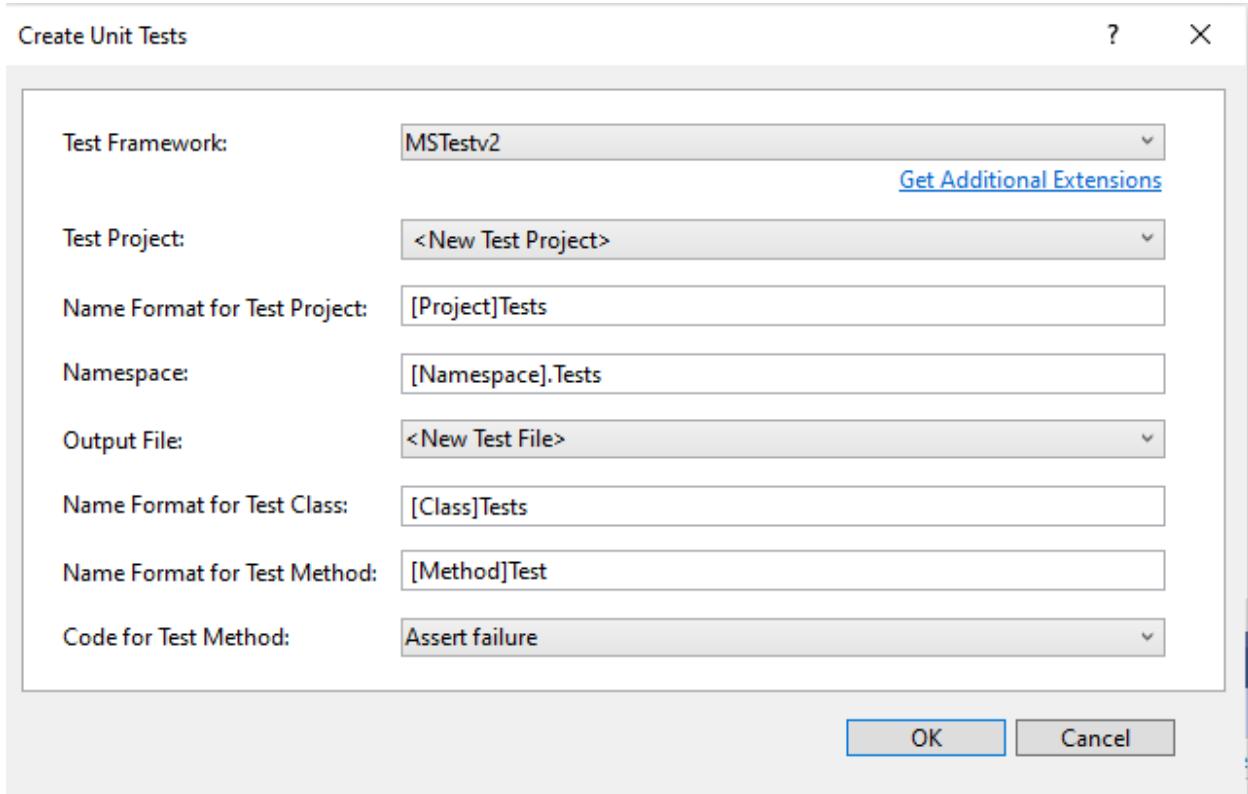
        public int multiply(int a, int b)
        {
            return a * b;
        }
    }
}

```

right click and then create unit test project



press OK



enter test code

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using cs_lib_sample;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace cs_lib_sample.Tests
{
    [TestClass()]
    public class SampleLibClassTests
    {

        [TestMethod()]
        public void testSayHelloTo()
        {

            Assert.AreEqual("Hello Computer", SampleLibClass.sayHelloTo("Computer"), "Regular say hello");
        }
        [TestMethod()]
        public void testSayHelloToWrong()
        {
            Assert.AreEqual("Hello All", SampleLibClass.sayHelloTo("Computer"), "Regular say hello won't work");
        }
    }
}
```

```
[TestMethod()]
public void testSumCorrect()
{
    Assert.AreEqual(9, SampleLibClass.sum(4, 5), "Regular sum should work");
}

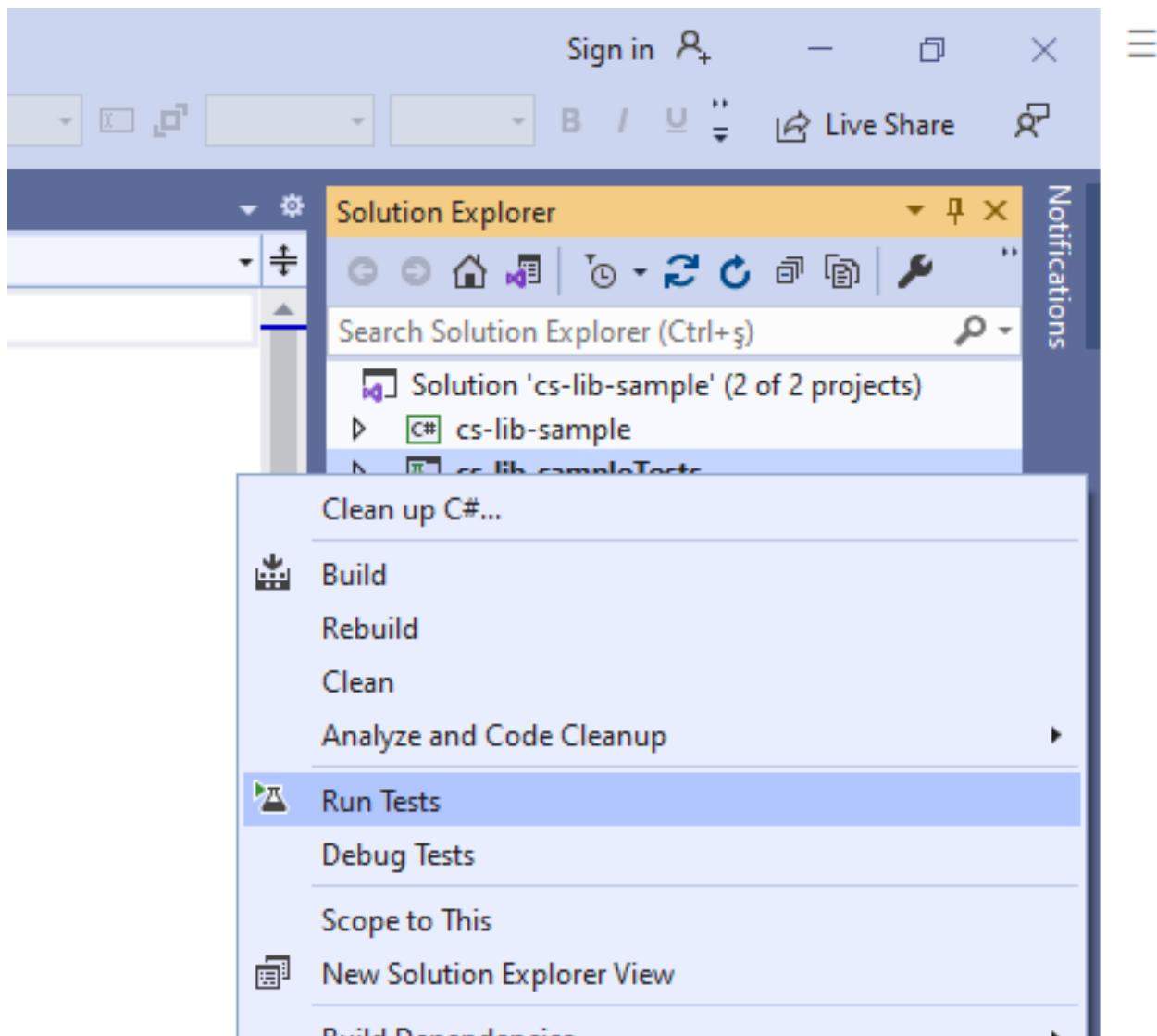
[TestMethod()]
public void testSumWrong()
{
    Assert.AreEqual(10, SampleLibClass.sum(4, 5), "Regular sum shouldn't work");
}

[TestMethod()]
public void testMultiply()
{
    SampleLibClass sampleLib = new SampleLibClass();

    Assert.AreEqual(20, sampleLib.multiply(4, 5), "Regular multiplication should work");
}

}
```

Run tests



you will code coverage and entered or passed branches

```

7  public class SampleLibClass
8  {
9      2 references | 1/2 passing
10     public static string sayHelloTo(string name)
11     {
12         string result = String.Empty;
13
14         if (!String.IsNullOrEmpty(name))
15         {
16             result = "Hello " + name;
17         }
18         else
19         {
20             result = "Hello There";
21         }
22
23         Console.WriteLine(result);
24
25         return result;
26     }
27
28     2 references | 1/2 passing

```

144 % No issues found

Fine Code Coverage

	Covered	Uncovered	Coverable	Total	Line coverage
Name					
- cs-lib-sample	17	3	20	39	85%
SampleLibClass	17	3	20	39	85%
- cs-lib-sampleTests	14	2	16	51	87.5%
SampleLibClassTests	14	2	16	51	87.5%

0.6.4.1 Visual Studio Community Edition (NUnit+.NETCore) use csharp-sample-lib for this example

create and add a unit test project to solution

Search for templates (Alt+S)  Clear all

C# Windows Test

 MSTest Test Project
A project that contains MSTest unit tests that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Test

 NUnit Test Project
A project that contains NUnit tests that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Desktop Test Web

 Unit Test Project (.NET Framework)
A project that contains MSTest unit tests.
C# Windows Test

 xUnit Test Project
A project that contains xUnit.net tests that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Test

 Web Driver Test for Edge (.NET Core)
A project that contains unit tests that can automate UI testing of web sites within Edge browser (using Microsoft's WebDriver API).
C# Windows Web Test

 Web Driver Test for Edge (.NET Framework)
A project that contains unit tests that can automate UI testing of web sites within Edge browser (using Microsoft's WebDriver API).
C# Windows Web Test

 Unit Test App (Universal Windows)
A project to create a unit test app for Universal Windows Platform (UWP) apps using MSTest.
C# Windows UWP Test

Configure your new project

NUnit Test Project C# Linux macOS Windows Desktop Test Web

Project name

csharp-sample-lib-test

Location

E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programming \Lectures\celf

...

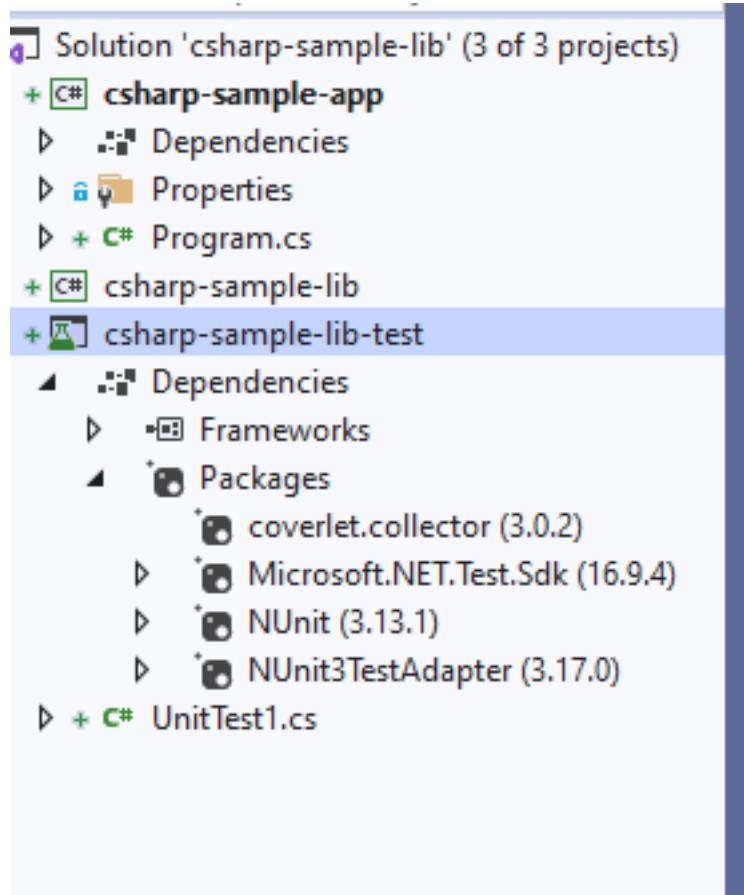
Additional information

NUnit Test Project C# Linux macOS Windows Desktop Test Web

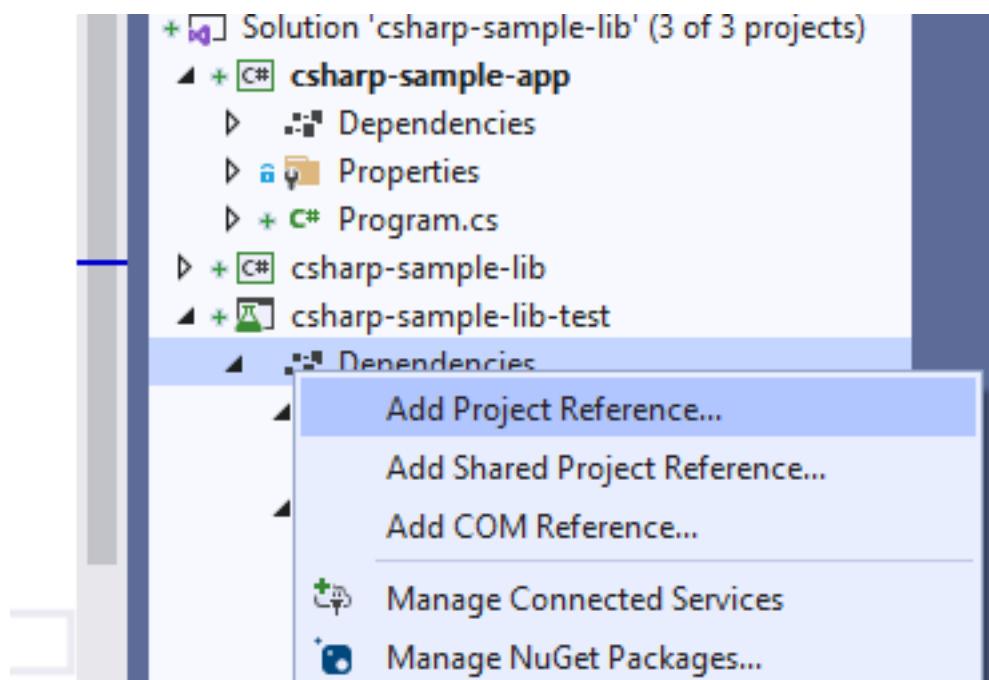
Target Framework 

.NET Core 3.1 (Long-term support)

- .NET Framework 4.0
- .NET Framework 4.5
- .NET Framework 4.5.1
- .NET Framework 4.5.2
- .NET Framework 4.6
- .NET Framework 4.6.1
- .NET Framework 4.6.2
- .NET Framework 4.7
- .NET Framework 4.7.1
- .NET Framework 4.7.2
- .NET Framework 4.8
- .NET Core 1.0 (Out of support)
- .NET Core 1.1 (Out of support)
- .NET Core 2.0 (Out of support)
- .NET Core 2.1 (Long-term support)
- .NET Core 2.2 (Out of support)
- .NET Core 3.0 (Out of support)
- .NET Core 3.1 (Long-term support)
- .NET 5.0 (Current)



Add project reference



Reference Manager - csharp-sample-lib-test

Projects			Search (
Solution	Name	Path	
Shared Projects	csharp-sample-app	E:\UgurCoruh\RTEU\L...	
	<input checked="" type="checkbox"/> csharp-sample-lib	E:\UgurCoruh\RTEU\L...	Name: csharp-

SampleLibraryTestClassss in NUnit Project

```
using csharp_sample_lib;
using NUnit.Framework;

namespace csharp_sample_lib_test
{
    public class SampleLibraryTestClass
    {
        sampleLibClass sampleLib;

        [SetUp]
        public void Setup()
        {
            sampleLib = new sampleLibClass();
        }

        [Test]
        public void testSayHelloTo()
        {
            Assert.AreEqual("Hello Computer", sampleLibClass.sayHelloTo("Computer"), "Regular say hello");
        }

        [Test]
        public void testSayHelloToWrong()
        {
            Assert.AreEqual("Hello All", sampleLibClass.sayHelloTo("Computer"), "Regular say hello won't work");
        }

        [Test]
        public void testSumCorrect()
        {
            Assert.AreEqual(9, sampleLibClass.sum(4, 5), "Regular sum should work");
        }

        [Test]
```

```

public void testSumWrong()
{
    Assert.AreEqual(10, sampleLibClass.sum(4, 5), "Regular sum shouldn't work");
}

[TestMethod]
public void testMultiply()
{
    Assert.AreEqual(20, sampleLib.multiply(4, 5), "Regular multiplication should work");
}
}

```

sample class library

```

using System;

namespace csharp_sample_lib
{
    public class sampleLibClass
    {
        public static string sayHelloTo(string name)
        {
            string result = String.Empty;

            if (!String.IsNullOrEmpty(name))
            {
                result = "Hello " + name;
            }
            else
            {
                result = "Hello There";
            }

            Console.WriteLine(result);

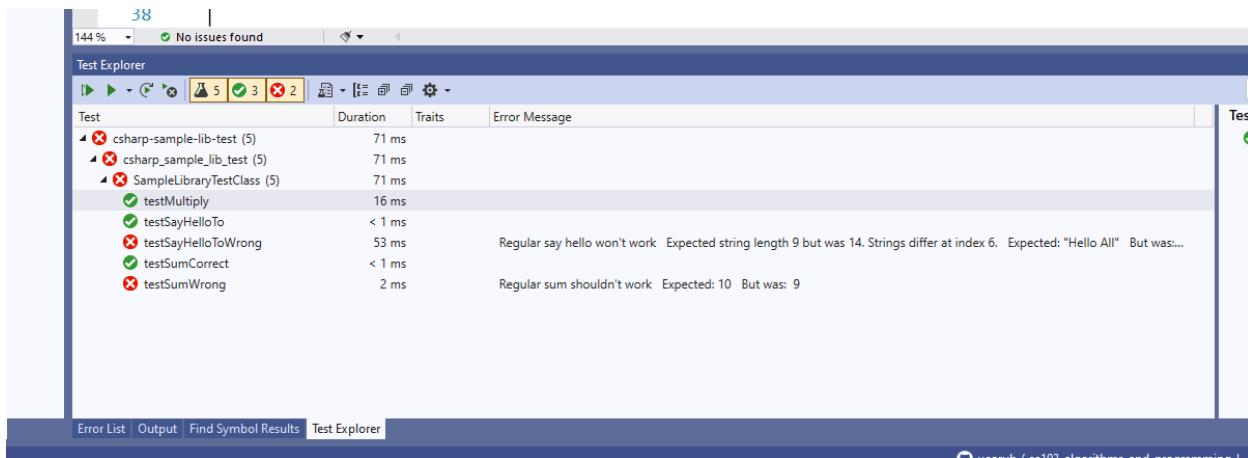
            return result;
        }

        public static int sum(int a, int b)
        {
            int c = 0;
            c = a + b;
            return c;
        }

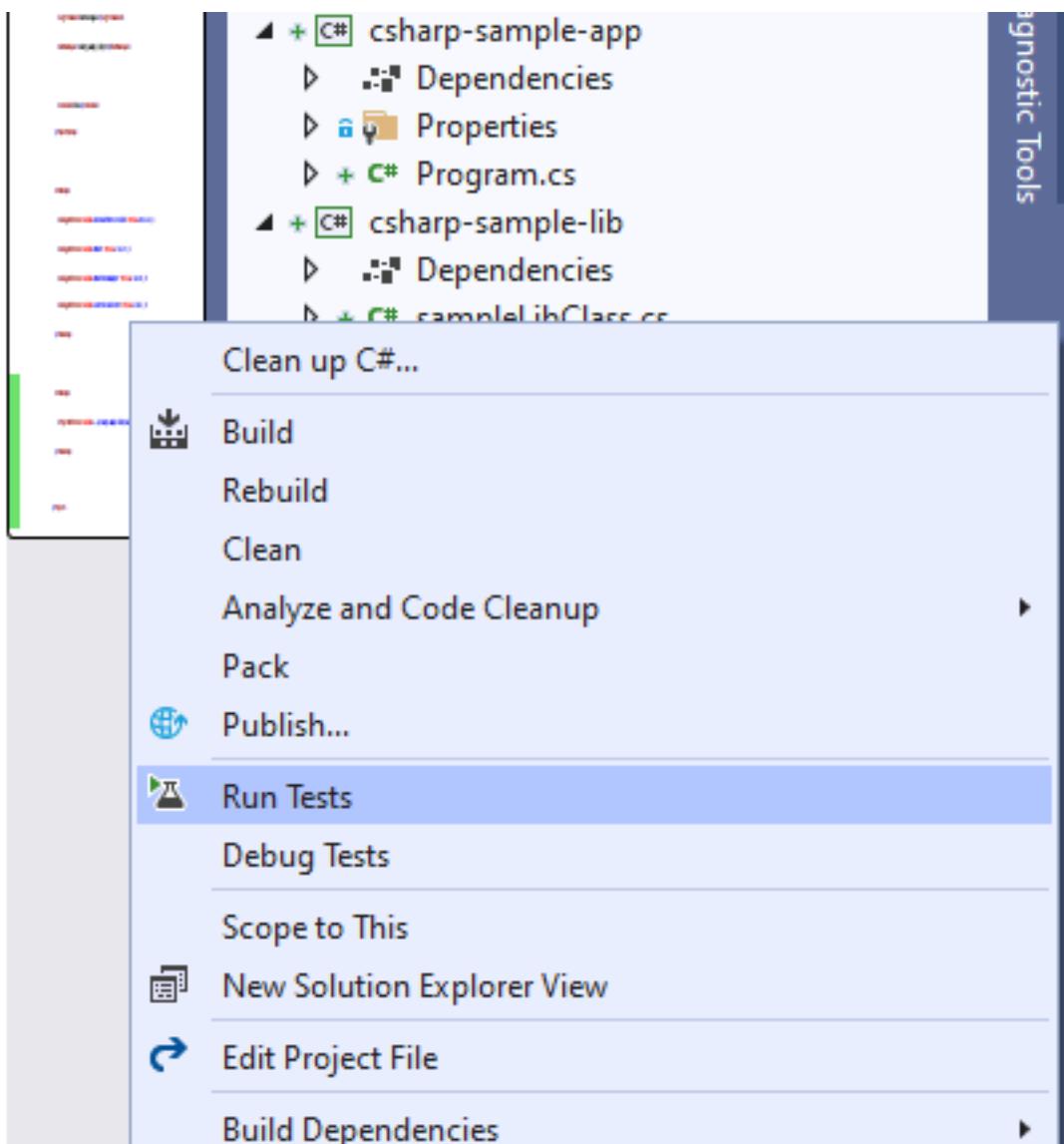
        public int multiply(int a, int b)
        {
            return a * b;
        }
    }
}

```

Open test explorer and run tests

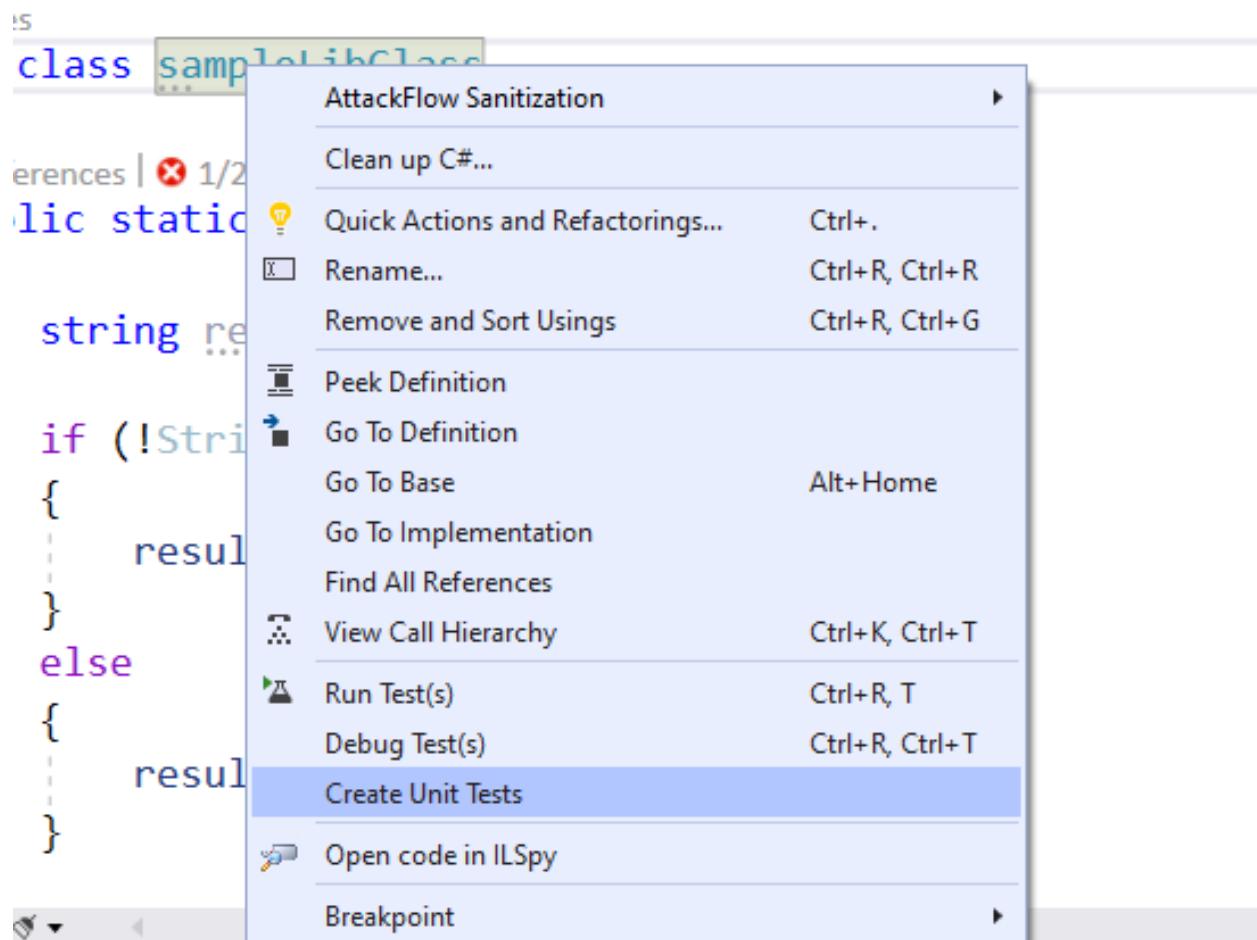


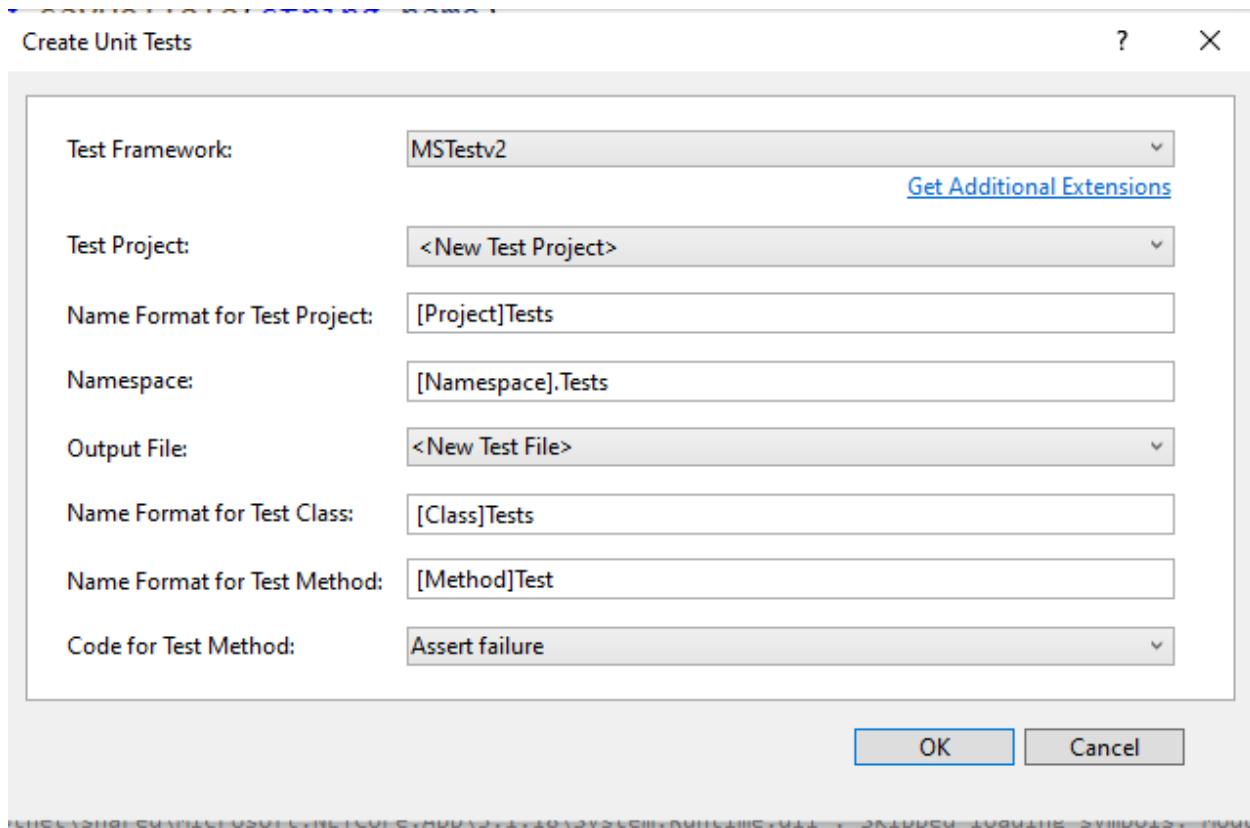
or you can run from project



Also we can create unit test from library class,

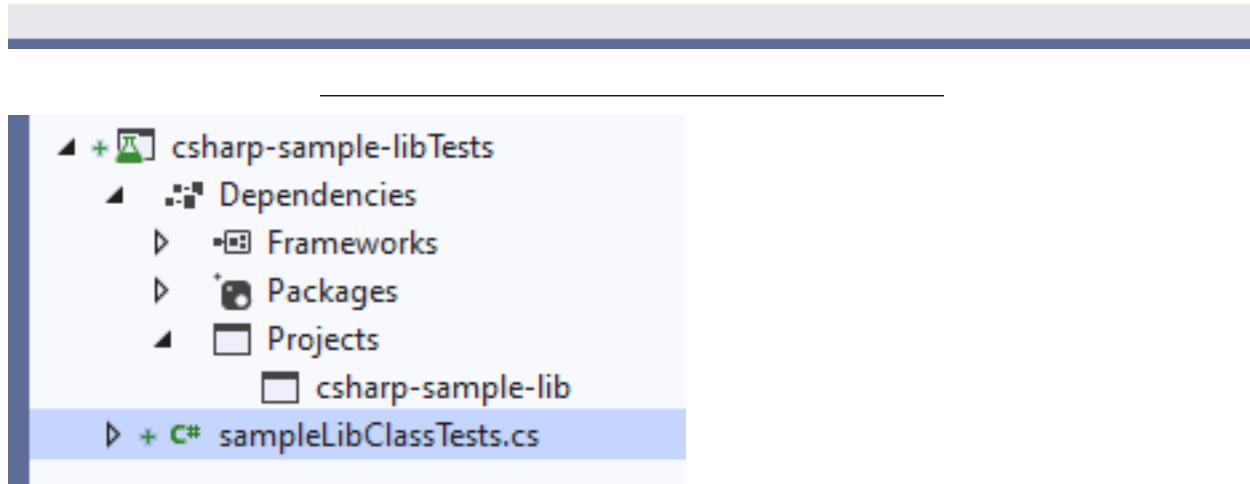
right click the sampleLibClass and select create unit tests but this option do not provide nunit tests.





Create Unit Tests

Creating Unit Tests



```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using csharp_sample_lib;
using System;
using System.Collections.Generic;
using System.Text;

namespace csharp_sample_lib.Tests
{
    [TestClass()]
    public class sampleLibClassTests
    {
        [TestMethod()]
        public void sayHelloToTest()
        {
            Assert.Fail();
        }

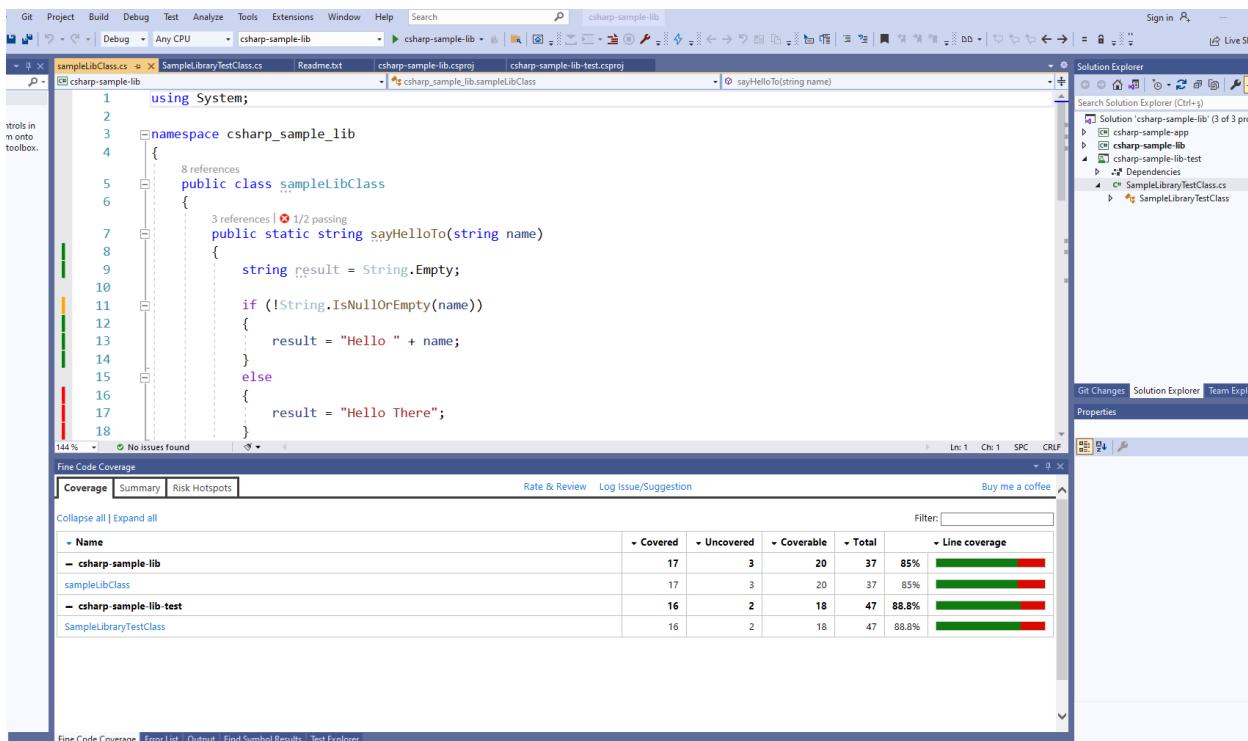
        [TestMethod()]
        public void sumTest()
        {
            Assert.Fail();
        }

        [TestMethod()]
        public void multiplyTest()
        {
            Assert.Fail();
        }
    }
}

```

we will not commit this changes and continue from nunit test project, the fine code coverage also work for nunit test but not provide inline highlighting

if we run tests we will have the following outputs



Inline code highlight is part of enterprise visual studio edition

Analyzing code coverage in Visual Studio - DEV Community¹³

1 TL;DR

Additional information you can use OpenCover + Nunit Runner + Report Generator together to setup a code coverage report but it has complex batch running process. After a few try I decided to use fine code coverage but here is the usage not tested well.

First unit test runner tool doesn't support .Net Core

c# - The NUnit 3 driver encountered an error while executing reflected code (NUnit.Engine.NUnitEngineException)
- Stack Overflow¹⁴

Follow the instructions on the link

¹³<https://dev.to/rruizdev/analizando-cobertura-del-codigo-en-visual-studio-1p27>

¹⁴<https://stackoverflow.com/questions/64611083/the-nunit-3-driver-encountered-an-error-while-executing-reflected-code-nunit-en>

Install OpenCover, ReportGenerator, Nunit,Runners packages then use the package installation folder to get tools that you need

Here is a sample for open cover, select package and copy path

¹⁵<https://github.com/sukhoi1/Useful-Notes/wiki/CMD-OpenCover>

The screenshot shows the Visual Studio Solution Explorer window. The 'Packages' node is expanded, displaying several NuGet packages:

- Frameworks
- Packages
 - coverlet.collector (3.1.0)
 - Microsoft.NET.Test.Sdk (16.11.0)
 - NUnit (3.13.2)
 - NUnit.Runners (3.12.0)
 - NUnit3TestAdapter (4.0.0)
 - OpenCover (4.7.1221) - This package is selected.
 - ReportGenerator (4.8.13)
- Projects
 - nunit-console-run.bat
 - open-cover.bat
 - open-cover-runner.bat
- C# UnitTest1.cs

Below the Solution Explorer, the ribbon tabs are visible: Git Changes, Solution Explorer (selected), and Team Explorer.

A separate 'Properties' window is open, showing the 'OpenCover Package Reference Properties' for the selected 'OpenCover' package. The 'General' section contains the following details:

Aliases	
Excluded Assets	
Generate Path Property	
Included Assets	
Name	OpenCover
Path	C:\Users\ugur.coruh\.nuget\packages\opencover\4.7.1221\lib\net45\
Private Assets	
Suppress Warnings	
Version	4.7.1221

Goto path and tools

C:\Users\ugur.coruh\.nuget\packages\opencover\4.7.1221

You need to setup some batch similar with following

run-test-coverage.bat

```
set pathA=C:\Users\ugur.coruh\.nuget\packages\opencover\4.7.1221\tools
set pathB=C:\Users\ugur.coruh\.nuget\packages\nunit.consolerunner\3.12.0\tools
set pathC=C:\Users\ugur.coruh\.nuget\packages\reportgenerator\4.8.13\tools\netcoreapp3.0
set dllpath=C:\Users\ugur.coruh\Desktop\csharp-sample-lib\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\

"%pathA%\OpenCover.Console.exe" ^
-targetargs:"%dllpath%\csharp-sample-lib-test.dll" ^
-filter:"+[csharp-sample-lib]* -[*test]*" ^
-target:"%pathB%\nunit3-console.exe" ^
-output:"%dllpath%\coverReport.xml" ^
-skipautoprops -register:user && "%pathC%\ReportGenerator.exe" -reports:"%dllpath%\coverReport.xml" -target:"%pathB%\nunit3-console.exe" ^
pause
```

but nunit3-console.exe gives error

```
C:\Users\ugur.coruh\Desktop\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\OpenCover.Console.exe" -targetargs:"C:\Users\ugur.coruh\Desktop\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\csharp-sample-lib-test.dll" -filter:"+[csharp-sample-lib]* -[*test]*" -target:"C:\Users\ugur.coruh\.nuget\packages\nunit.consolerunner\3.12.0\tools\nunit3-console.exe" -output:"C:\Users\ugur.coruh\Desktop\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\coverReport.xml" -skipautoprops -register:user && "C:\Users\ugur.coruh\Desktop\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\ReportGenerator.exe" -reports:"C:\Users\ugur.coruh\Desktop\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\coverReport.xml" -target:"C:\Users\ugur.coruh\.nuget\packages\nunit.consolerunner\3.12.0\tools\nunit3-console.exe"
Launching OpenCover 4.7.1221.0
Process started: C:\Users\ugur.coruh\Desktop\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\OpenCover.Console.exe
.NET Core Version: 3.1.20 (.NET 2.0)
Copyright (c) Microsoft Corporation. All rights reserved.
Monday, October 25, 2021 21:08:06 AM
Runtime Environment
  OS Version: Microsoft Windows NT 6.2.9200.6
  App Version: .NET Framework CLR v4.0.30317.22000
Test Files
  C:\Users\ugur.coruh\Desktop\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\csharp-sample-lib-test.dll
Errors, Failures and Warnings
[1] Error
  at NUnit.Engine.NUnitEngineException : The NUnit 3 driver encountered an error while executing reflected code.
    at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
    at NUnit.Engine.System.InvalidCastException : Unable to cast transparent proxy to type 'System.Web.UI.ICallbackEventHandler'.
    -->NUnit.Engine.NUnitEngineException : The NUnit 3 driver encountered an error while executing reflected code.
  Driver stack trace:
    at NUnit.Engine.Drivers.NUnitFrameworkDriver.CreateObject<T>(String typeName, Object[] args)
    at NUnit.Engine.Drivers.NUnitFrameworkDriver.Load(String testResultsPath, IDictionary<String, settings>)
    at NUnit.Engine.Browsers.DirectTestRunner.LoadDriver(IFrameworkDriver driver, String testFile, TestPackage subPackage)
    at NUnit.Engine.Browsers.DirectTestRunner.Load(IFrameworkDriver driver, String testFile, TestPackage subPackage)
    at NUnit.Engine.Browsers.TestConsoleRunner.LoadPackage()
    at NUnit.Engine.Browsers.TestConsoleRunner.Load()
    at NUnit.Engine.Browsers.DirectTestRunner.RunTests(IEventListener listener, TestFilter filter)
    at NUnit.Engine.Drivers.NUnitFrameworkDriver.RunTests(IEventListener listener, TestFilter filter)
    at System.Runtime.Remoting.Messaging.StackWalkLink.PrivateProcessMessage(IntPtr md, Object[] args, Object server, Object[]& outArgs)
    at System.Runtime.Remoting.Messaging.StackWalkLink.ProcessMessage(IntPtr md, Object[] args, Object server, Object[]& outArgs)
    at System.Runtime.Remoting.Messaging.StackWalkLink.SyncProcessMessage(Message msg)
Exception rethrown at [0]:
  at System.Runtime.Remoting.Proxies.RealProxy.HandleReturnMessage(Message msgObj, IMessage msg)
  at System.Runtime.Remoting.Proxies.RealProxy.PrivateInvoke(MessageData& msgData, Int32 type)
  at NUnit.Engine.ITestingProcessRunner.Run(IEventListener listener, TestFilter filter)
  at NUnit.Engine.Drivers.NUnitFrameworkDriver.RunTests(IEventListener listener, TestFilter filter)
InvalidOperationException
Unable to cast transparent proxy to type 'System.Web.UI.ICallbackEventHandler'.
  at NUnit.Framework.Api.FrameworkController.<LoadTestsAction..ctor>()
  at NUnit.Framework.Api.FrameworkController.<LoadTestsAction..ctor>(FrameworkController controller, Object handler)
Test Run Summary
  Overall result: Failed
  Test Count: 0, Passed: 0, Failed: 0, Warnings: 0, Inconclusive: 0, Skipped: 0
  Start time: 2021-10-24 23:01:09Z
  End time: 2021-10-24 23:01:11Z
  Duration: 2.015 seconds
ResultsCount> saved as TestResult.xml
Warning: This could be for a number of reasons. The most common reason are:
  1) missing PDBs for your assemblies that match the filter please review the
  output file and refer to the Usage guide (Usage.rtf) about filters.
  2) the assembly you are trying to load does not have symbols available. Please
  refer to the Usage guide and the -register switch.
  3) the assembly you are trying to load does not have symbols available. Please
  check your filters. Code coverage may not be enabled, refer to the output XML
  4) the assembly you are trying to load does not have symbols available under test were not
  instrumented. Try using the -wholeAssembly switch.
  5) you are targeting .NET 4.0 or lower and your assemblies are registered with the
  msbuild compiler. Try using the -wholeAssembly switch.
  6) you are using the -wholeAssembly switch and your assemblies are registered with the
  msbuild compiler. Try using the -wholeAssembly switch.
2021-10-25 02:01:12: Arguments: C:\Users\ugur.coruh\Desktop\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\coverReport.xml
2021-10-25 02:01:12: targetdir:C:\Users\ugur.coruh\Desktop\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\coverReport.xml
2021-10-25 02:01:12: Working directory: C:\Users\ugur.coruh\Desktop\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\coverage\index.html
2021-10-25 02:01:13: Report generation took 0.3 seconds
C:\Users\ugur.coruh\Desktop\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\coverReport.xml
Press any key to continue . . .
```

for this compatibility issues I prefer to use fine code coverage extension.

OpenCover related studies

Code coverage of manual or automated tests with OpenCover for .NET applications – Automation Rhapsody¹⁶

Code coverage of .NET Core unit tests with OpenCover – Automation Rhapsody¹⁷

¹⁶<https://automationrhapsody.com/code-coverage-manual-automated-tests-opencover-net-applications/>

¹⁷<https://automationrhapsody.com/code-coverage-net-core-unit-tests-opencover/>

Sample OpenCover report

Summary - Coverage Report¹⁸

1.0.1 Download and Setup OpenCover, NUnit Console, Report Generator without Package Manager

You can also download the tools from github project pages and install on your operating system,

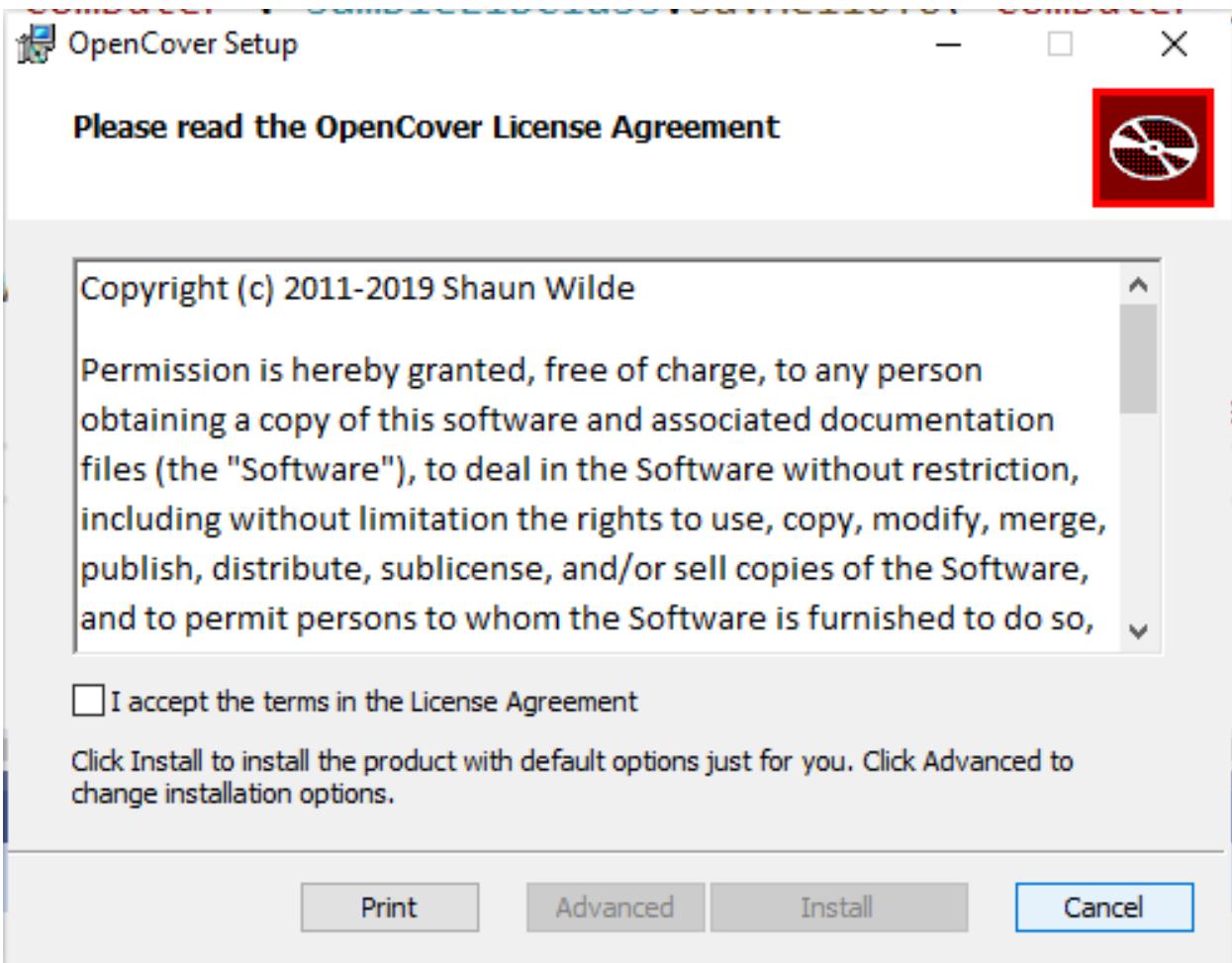
1.0.2 OpenCover

Releases · OpenCover/opencover · GitHub¹⁹



¹⁸<https://automationrhapsody.com/examples/OpenCover-report/>

¹⁹<https://github.com/OpenCover/opencover/releases>



Select advanced and then install for all users



Installation Scope

Choose the installation scope and folder



Install just for you (ugur.coruh)

OpenCover will be installed in a per-user folder and be available just for your user account. You do not need local Administrator privileges.

Install for all users of this machine

OpenCover will be installed in a per-machine folder by default and be available for all users. You can change the default installation folder. You must have local Administrator privileges.

Back

Next

Cancel



OpenCover Setup



Destination Folder

Click Next to install to the default folder or click Change to choose another.



Install OpenCover to:

[Change...](#)

[Back](#)

[Next](#)

[Cancel](#)



OpenCover Setup



Product Features

Select the way you want features to be installed.



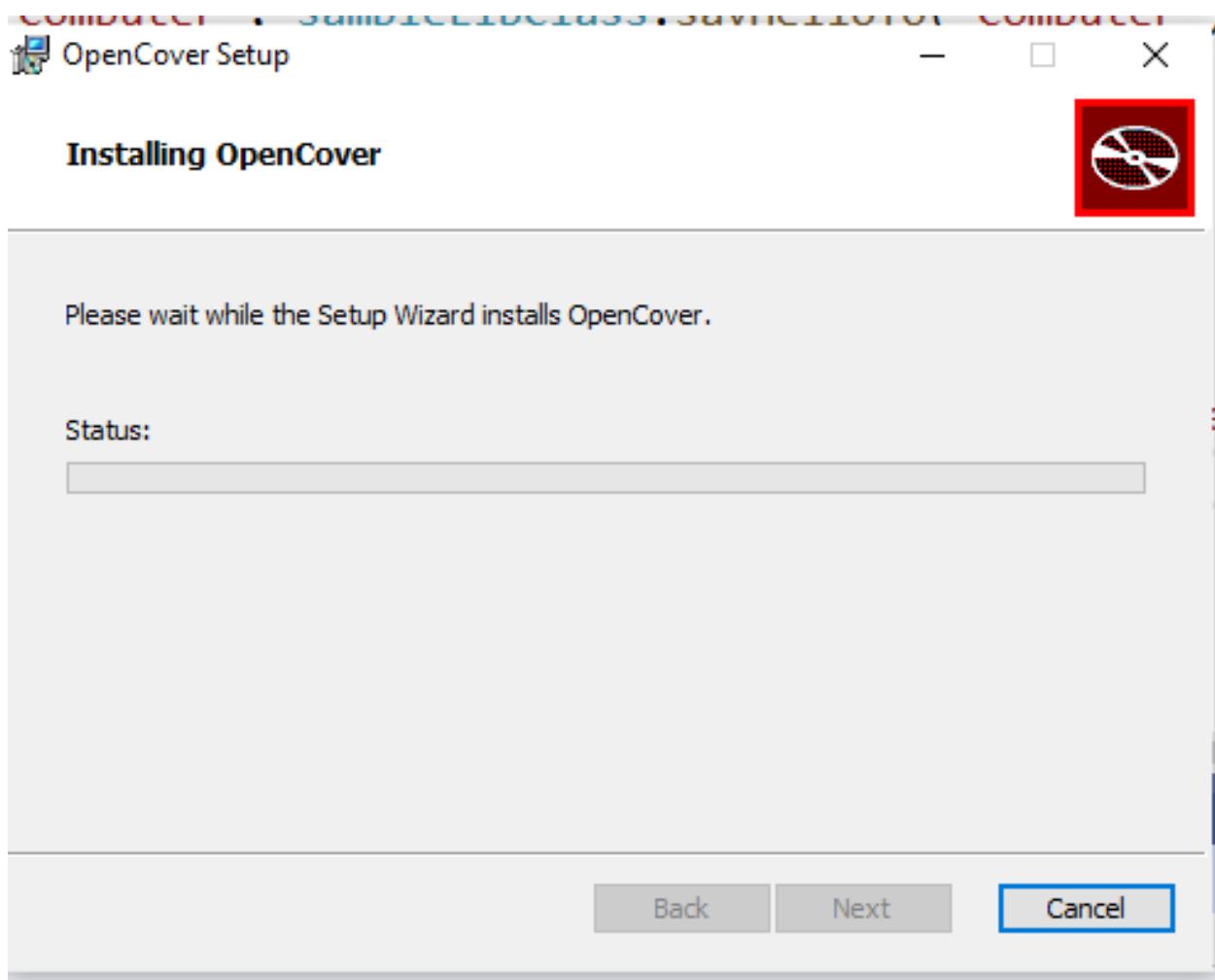
Install 32-bit and 64-bit samples. NOTE: Useful for testing the installation.

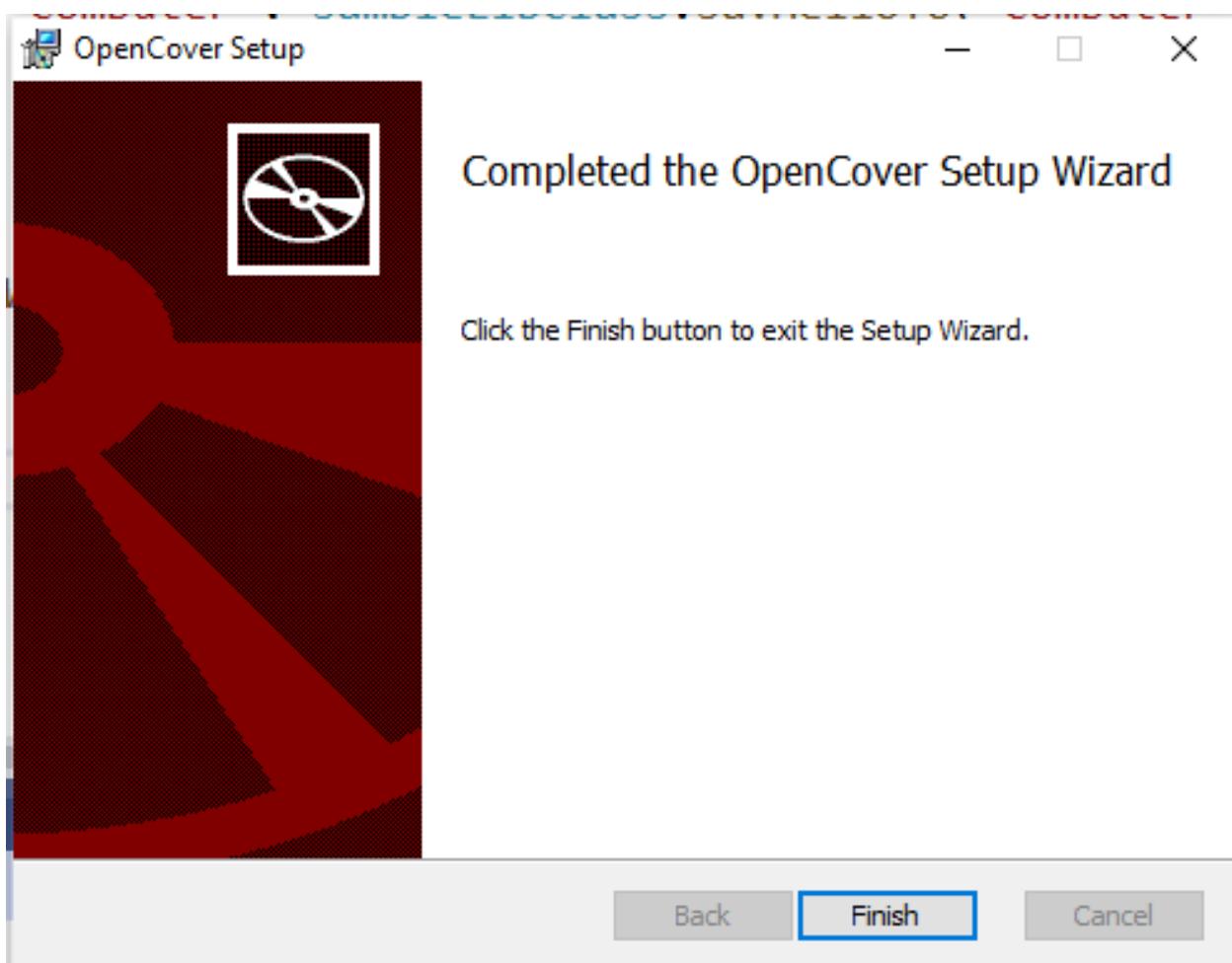
This feature requires 64KB on your hard drive.

Back

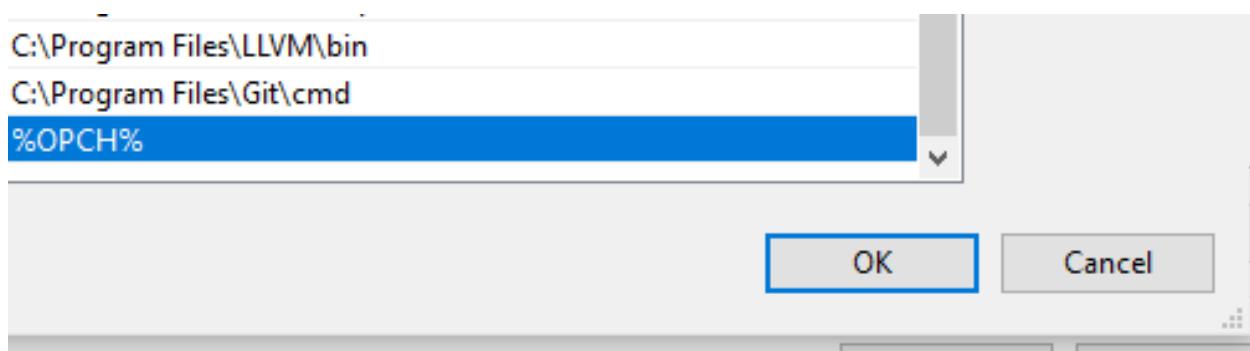
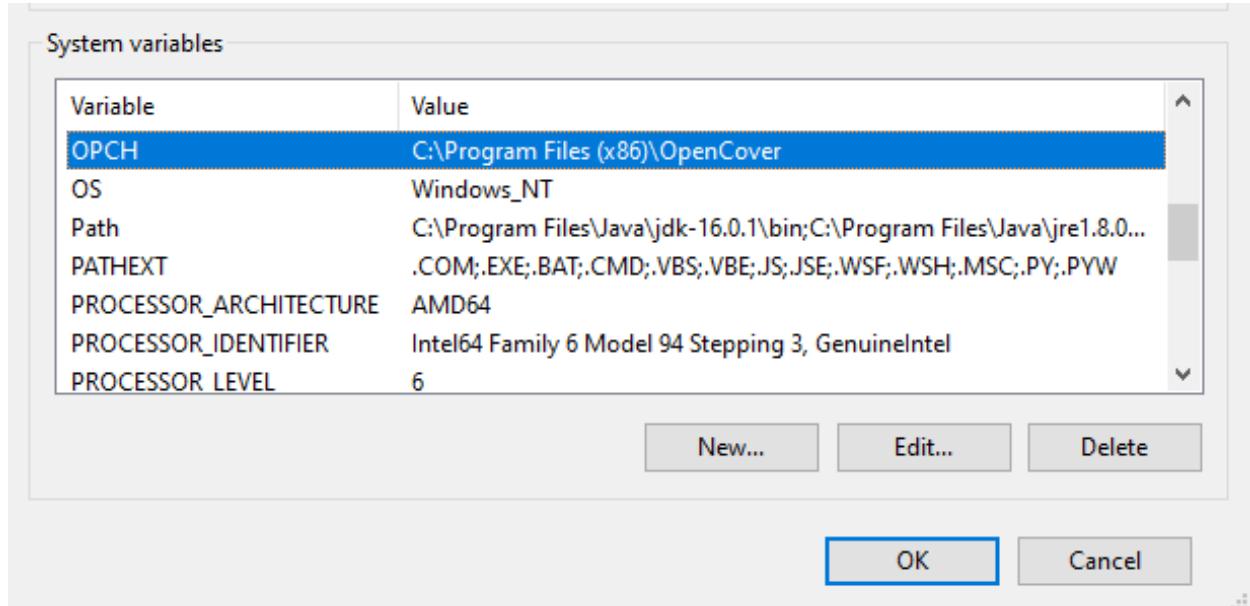
Install

Cancel





	mono.Cecil.xml.dll	9/15/202
	Mono.Cecil.Pdb.dll	9/15/202
	Mono.Cecil.Rocks.dll	9/15/202
	Newtonsoft.Json.dll	11/9/201
	OpenCover.Console.exe	6/19/202
	OpenCover.Console.exe.config	6/19/202
	OpenCover.Console.pdb	6/19/202
	OpenCover.Extensions.dll	6/19/202
	OpenCover.Extensions.nsh	6/10/202



```
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ugur.coruh>OpenCover.Console
Launching OpenCover 4.7.1221.0

Incorrect Arguments: The target argument is required

Usage:
  ["-target:<target application>[]"]
  ["-targetdir:<target directory>[]"]
  ["-searchdirs:<additional PDB directory>[;<additional PDB directory>[]]
  [-targetargs:<arguments for the target process>[]]
```

1.0.3 ReportGenerator

Release ReportGenerator_4.8.13 · danielpalme/ReportGenerator · GitHub²⁰

²⁰<https://github.com/danielpalme/ReportGenerator/releases/tag/v4.8.13>

ReportGenerator_4.8.13 Latest

github-actions released this 27 days ago · 4 commits to master since this release · v4.8.13 · e552cc6

This release requires .NET 4.7 or .NET Core 2.x/3.x/5.x.

Changes:

- #441: Added method coverage to reports
- #445: Added support for better custom logging
- #450: Conditional file numbers in class report

Assets 3

ReportGenerator_4.8.13.zip	13.2 MB
Source code (zip)	
Source code (tar.gz)	

Share View

This PC > Windows (C:) > ReportGenerator_4.8.13 >

Print Photo Print

Name
net5.0
net47
netcoreapp2.0
netcoreapp2.1
netcoreapp3.0
LICENSE.txt
Readme.txt

1.0.4 NUnit Console

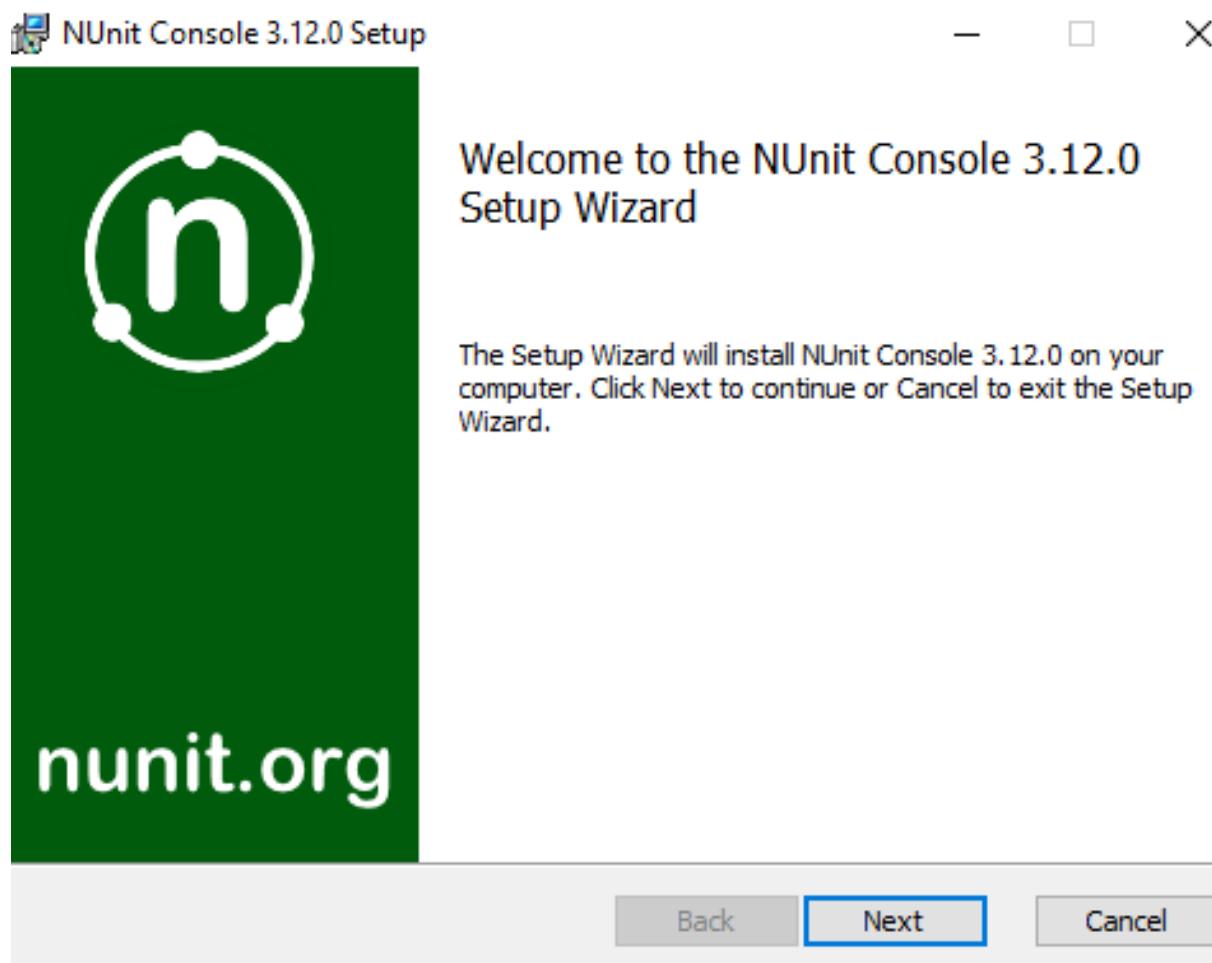
Downloads²¹

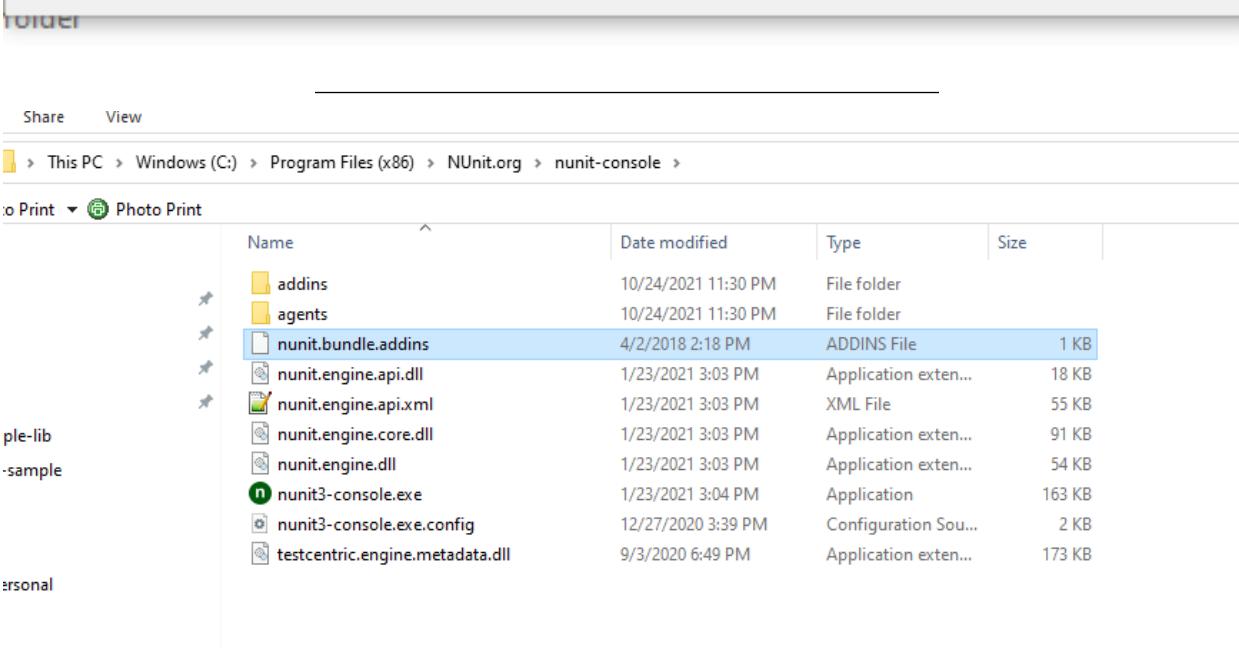
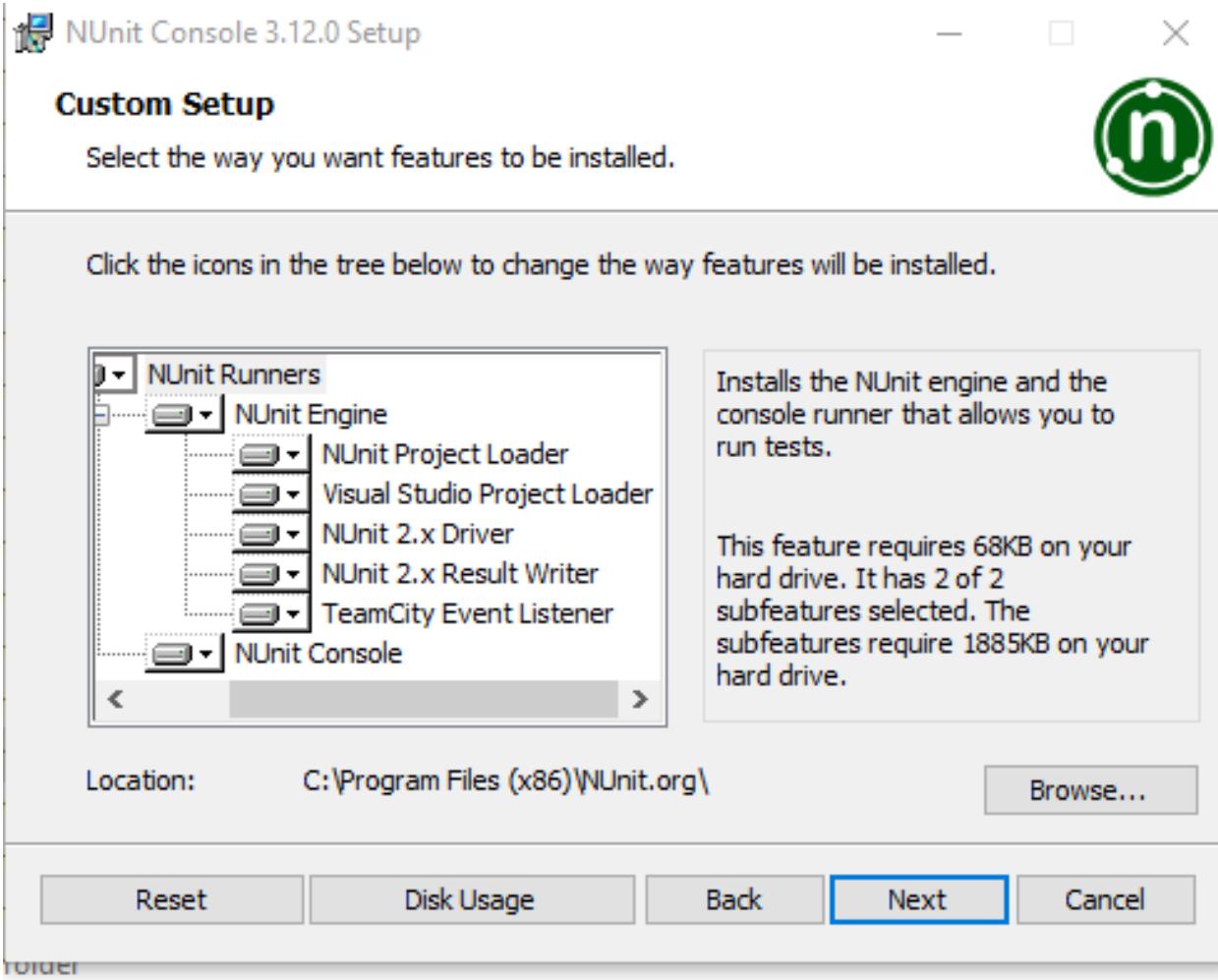
²¹<https://nunit.org/download/>

The screenshot shows the NUnit.org website's 'Downloads' section. At the top, there's a navigation bar with links for News, Download, Documentation, Contact, Twitter, Slack, and GitHub. Below the navigation is a heading 'Downloads' with a download icon. A section titled 'Download Types' explains that the preferred way to download NUnit is through the NuGet package manager, and that the latest releases can be found on the relevant GitHub releases pages. To the right, there are two tables: 'Latest NUNIT 3 Releases' and 'Latest NUNIT 2 Release'. The 'Latest NUNIT 3 Releases' table includes rows for NUNIT 3.13.2 (April 27, 2021), NUNIT Console 3.12 (January 17, 2021), NUNIT Test Adapter 3.17 (July 11, 2020), NUNIT Test Generator 2.3 (September 20, 2019), and NUNIT 3 Template for dotnet new CLI. The 'Latest NUNIT 2 Release' table includes rows for NUNIT 2.7.1 (August 19, 2019) and NUNIT Test Adapter 2.2 (June 5, 2019).

The screenshot shows the GitHub repository assets page for the 'NUNIT 3.12.0' release. The title 'Assets 10' is at the top. Below it is a table listing ten assets:

Asset	Size
nunit-console-runner.3.12.0.nupkg	733 KB
NUnit.Console-3.12.0.msi	1.04 MB
NUnit.Console-3.12.0.zip	14.4 MB
NUnit.Console.3.12.0.nupkg	19.2 KB
NUnit.ConsoleRunner.3.12.0.nupkg	746 KB
NUnit.Engine.3.12.0.nupkg	1 MB
NUnit.Engine.Api.3.12.0.nupkg	42.8 KB
NUnit.Runners.3.12.0.nupkg	19.3 KB
Source code (zip)	
Source code (tar.gz)	





1.0.5 NUnit + MSTest Batch Report Generation (Not Tested)

OpenCover and ReportGenerator Unit Test Coverage in Visual Studio 2013 and 2015 – CodeHelper.Net²²

OpenCover and ReportGenerator Unit Test Coverage in Visual Studio 2013 and 2015 - CodeProject²³

1.0.6 Java Unit Tests

1.0.6.1 Eclipse IDE (JUnit4 , JUnit5) In this sample we will create two example for similar library

Please check the following links

JUnit 5 tutorial - Learn how to write unit tests²⁴

JUnit 5²⁵

JUnit 5 User Guide²⁶

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

JUnit Hello World Example - Examples Java Code Geeks - 2021²⁷

<https://yasinmemic.medium.com/java-ile-unit-test-yazmak-birim-test-ca15cf0d024b>

1.0.6.2 Java Application + JUnit In normal java application we can right click the project java-sample-lib and add Junit case

²²<http://codehelper.net/unit-testing/opencover-and-reportgenerator-unit-test-coverage-in-visual-studio-2013-and-2015/>

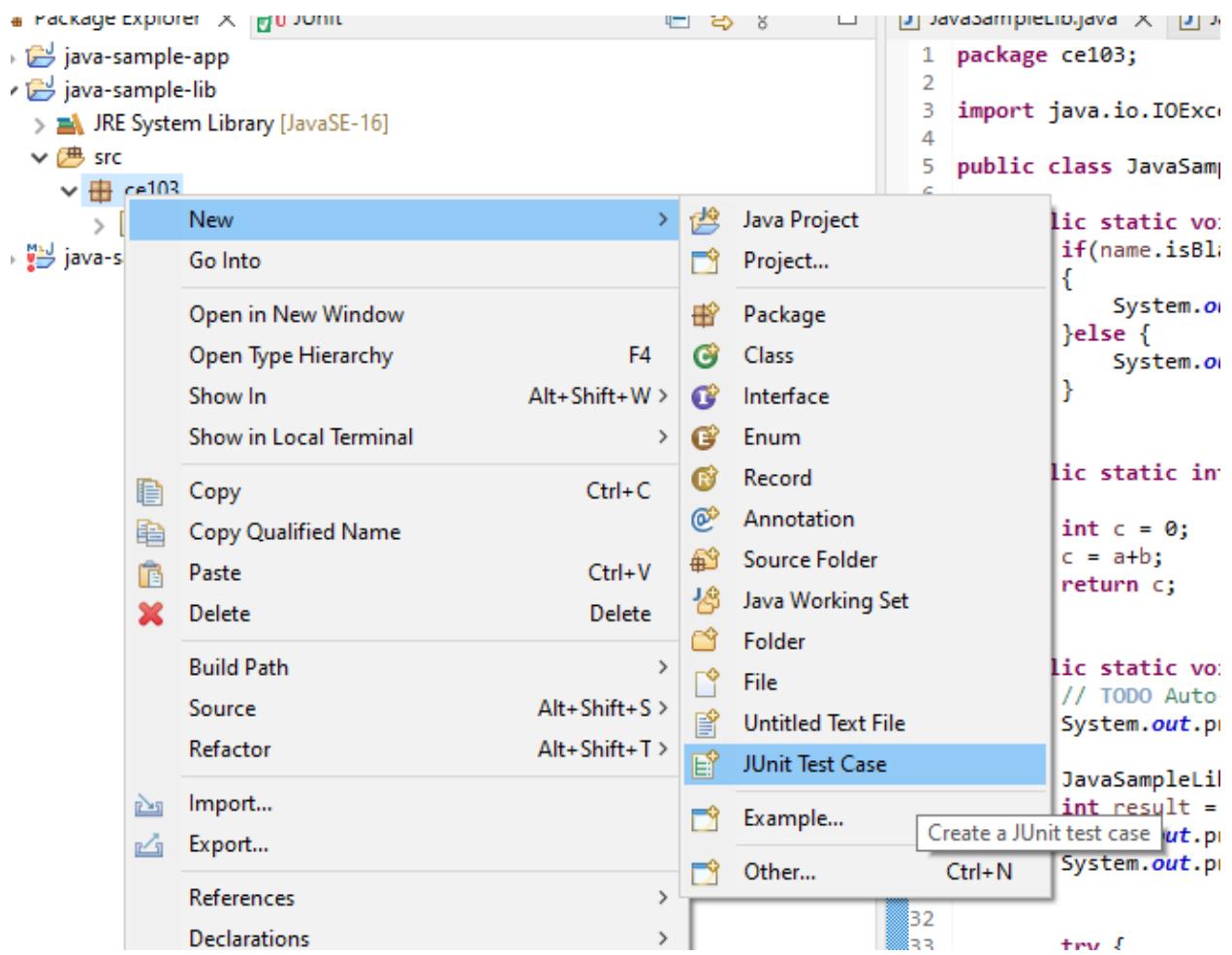
²³<https://www.codeproject.com/Articles/1276980/OpenCover-and-ReportGenerator-Unit-Test-Coverage-i>

²⁴<https://www.vogella.com/tutorials/JUnit/article.html>

²⁵<https://junit.org/junit5/>

²⁶<https://junit.org/junit5/docs/current/user-guide/>

²⁷<https://examples.javacodegeeks.com/core-java/junit/junit-hello-world-example/>





JUnit Test Case

Select the name of the new JUnit test case. Specify the class under test to select methods to be tested on the next page.

New JUnit 3 test New JUnit 4 test New JUnit Jupiter test

Source folder:

Package:

Name:

Superclass:

Which method stubs would you like to create?

@BeforeAll setUpBeforeClass() @AfterAll tearDownAfterClass()

@BeforeEach setUp() @AfterEach tearDown()

constructor

Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

Class under test:

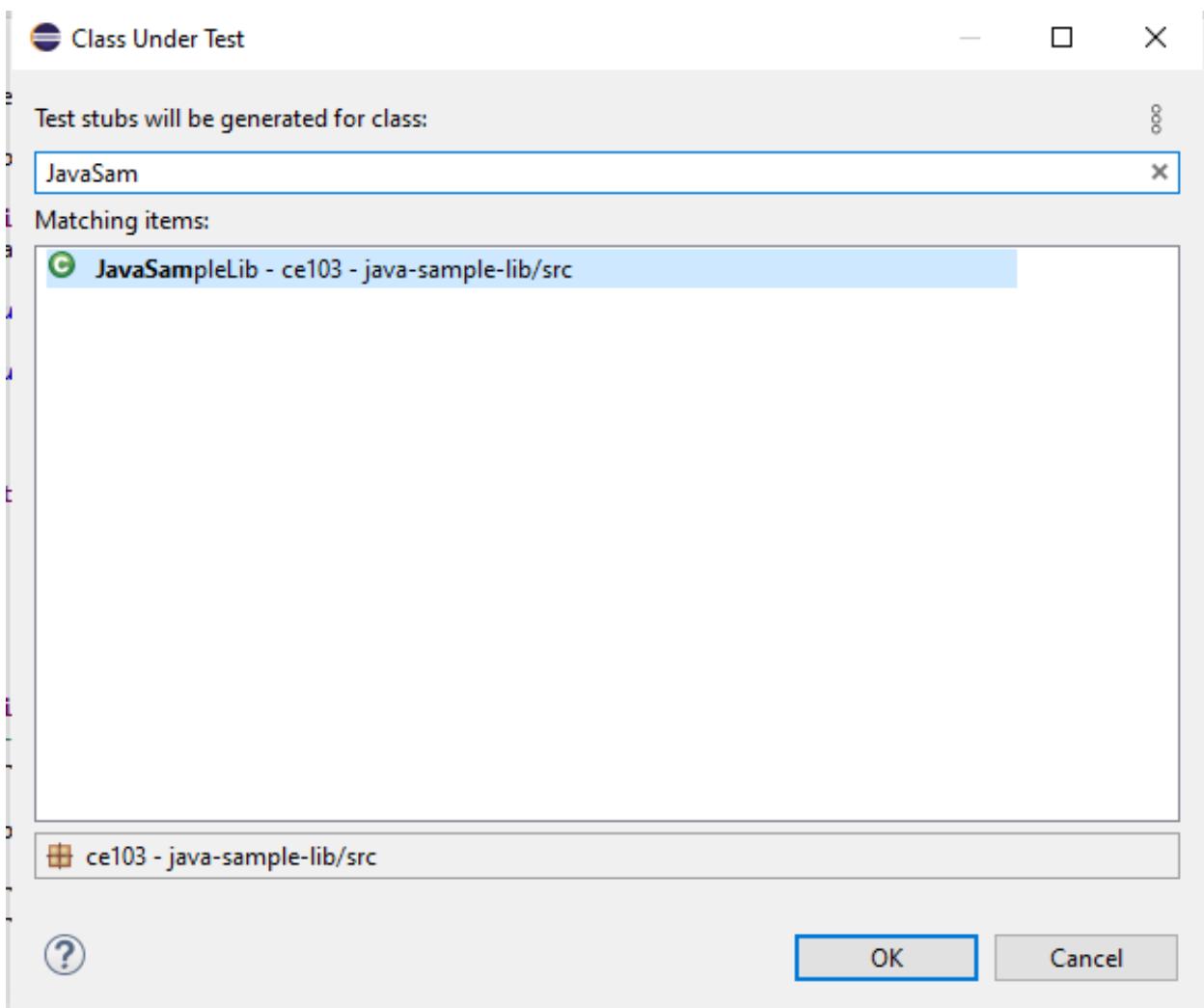


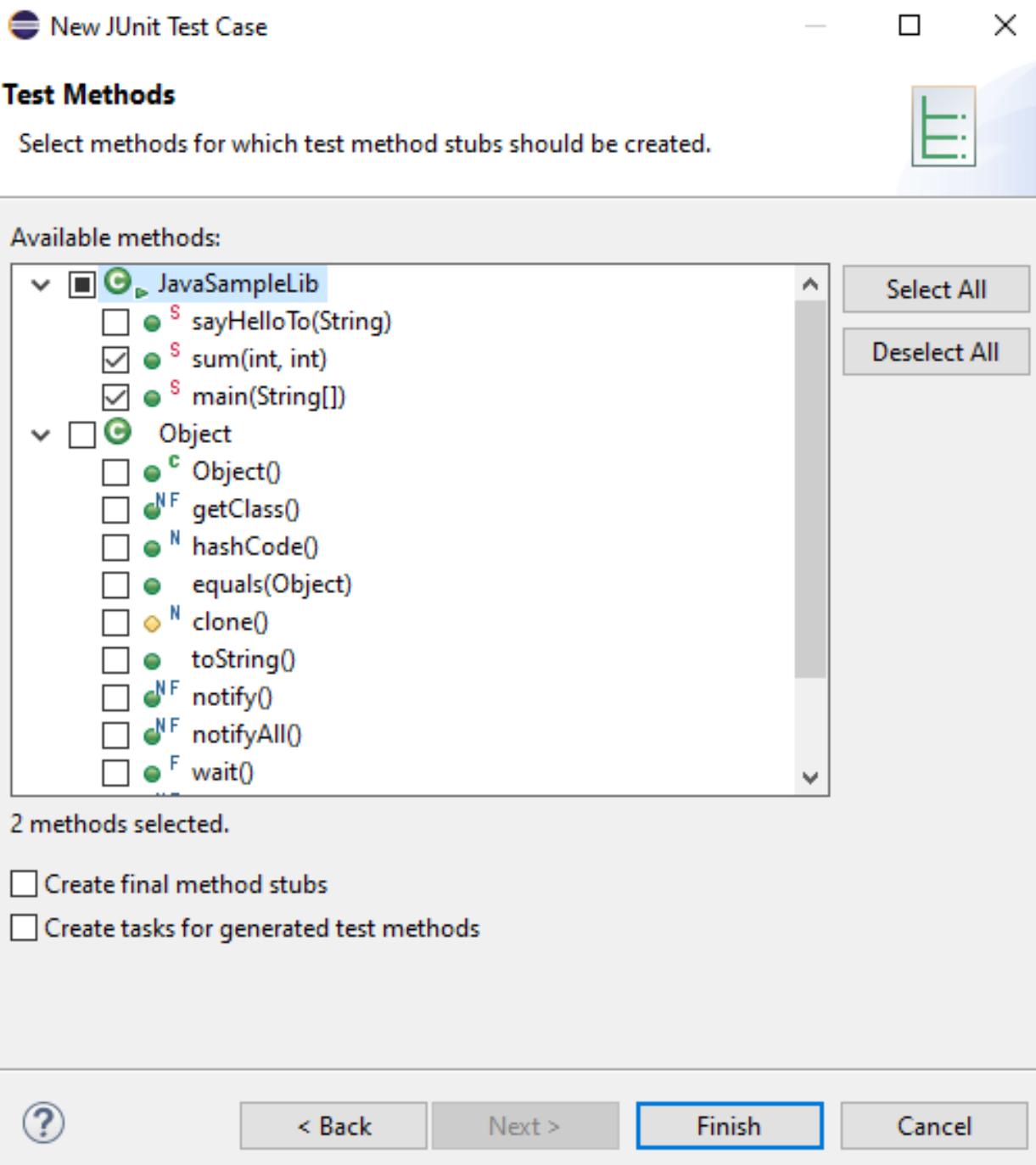
< Back

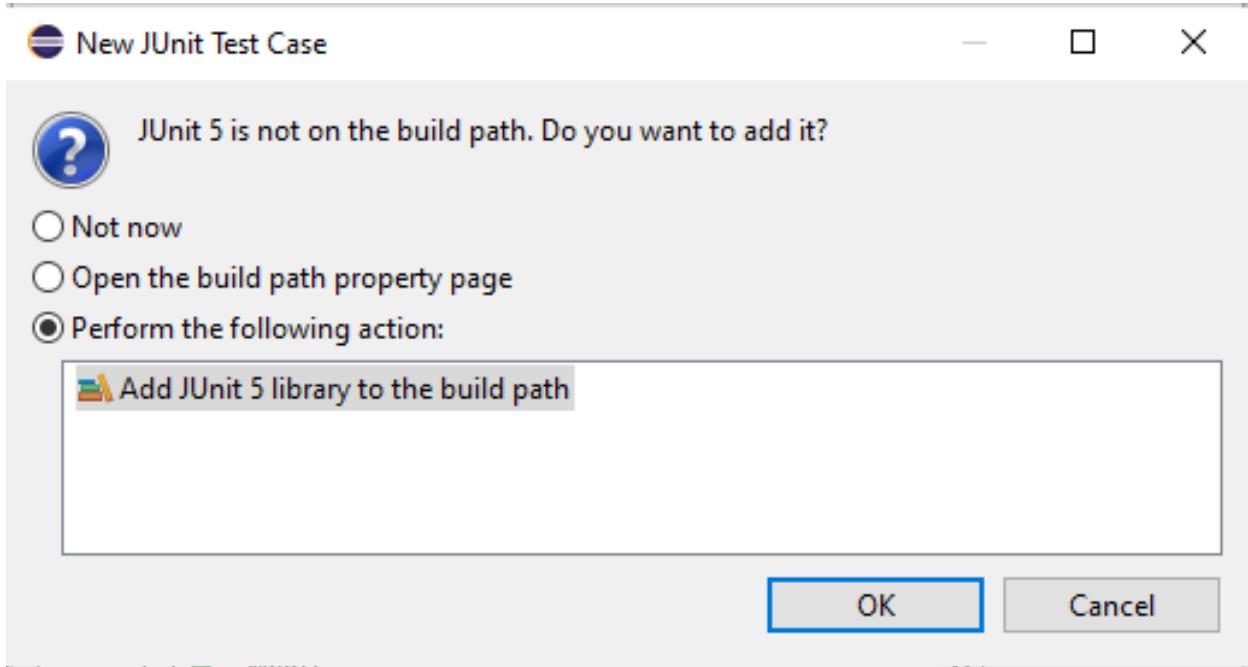
Next >

Finish

Cancel







and you will have the following test class

```

1 package ce103;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 class JavaSampleLibTest {
6
7     @BeforeAll
8     static void setUpBeforeClass() throws Exception {
9
10    }
11
12    @AfterAll
13    static void tearDownAfterClass() throws Exception {
14
15    }
16
17    @BeforeEach
18    void setUp() throws Exception {
19
20    }
21
22    @AfterEach
23    void tearDown() throws Exception {
24
25    }
26
27    @Test
28    void testSum() {
29        fail("Not yet implemented");
30    }
31
32    @Test
33    void testMain() {
34        fail("Not yet implemented");
35    }
36
37    }
38
39 }
40

```

Now we will create tests that check our function flowchart and return values

We need to cover all code branches that we coded

I have updated JavaSampleLib.java as follow to check outputs

JavaSampleLib.java

```

package ce103;

public class JavaSampleLib {

    public static String sayHelloTo(String name) {

        String output = "";

        if(!name.isBlank() && !name.isEmpty()){
            output = "Hello "+name;
        }else {
            output = "Hello There";
        }

        System.out.println(output);

        return output;
    }

    public static int sum(int a,int b)
    {
        int c = 0;
        c = a+b;
        return c;
    }

    public int multiply(int a, int b) {
        return a * b;
    }

//    public static void main(String[] args) {
//        // TODO Auto-generated method stub
//        System.out.println("Hello World!");
//
//        // JavaSampleLib.sayHelloTo("Computer");
//        int result = JavaSampleLib.sum(5, 4);
//        System.out.println("Results is" + result);
//        System.out.printf("Results is %d \n", result);
//
//        //
//        //
//        try {
//            System.in.read();
//        } catch (IOException e) {
//            // TODO Auto-generated catch block
//            e.printStackTrace();
//        }
//    }
//
//}
```

and JavaSampleLibTest.java

```

package ce103;

import static org.junit.jupiter.api.Assertions.*;
```

```

import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.RepeatedTest;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.MethodSource;

class JavaSampleLibTest {

    JavaSampleLib sampleLib;

    @BeforeAll
    static void setUpBeforeClass() throws Exception {
    }

    @AfterAll
    static void tearDownAfterClass() throws Exception {
    }

    @BeforeEach
    void setUp() throws Exception {
        sampleLib = new JavaSampleLib();
    }

    @AfterEach
    void tearDown() throws Exception {
    }

    @Test
    @DisplayName("Simple Say Hello should work")
    void testSayHelloTo() {
        assertEquals("Hello Computer", JavaSampleLib.sayHelloTo("Computer"), "Regular say hello should work");
    }

    @Test
    @DisplayName("Simple Say Hello shouldn't work")
    void testSayHelloToWrong() {
        assertEquals("Hello All", JavaSampleLib.sayHelloTo("Computer"), "Regular say hello won't work");
    }

    @Test
    @DisplayName("Simple sum should work")
    void testSumCorrect() {
        assertEquals(9, JavaSampleLib.sum(4, 5), "Regular sum should work");
    }

    @Test
    @DisplayName("Simple sum shouldn't work")
    void testSumWrong() {
        assertEquals(10, JavaSampleLib.sum(4, 5), "Regular sum shouldn't work");
    }
}

```

```

@Test
@DisplayName("Simple multiplication should work")
void testMultiply() {
    assertEquals(20, sampleLib.multiply(4, 5), "Regular multiplication should work");
}

@RepeatedTest(5)
@DisplayName("Ensure correct handling of zero")
void testMultiplyWithZero() {
    assertEquals(0, sampleLib.multiply(0, 5), "Multiple with zero should be zero");
    assertEquals(0, sampleLib.multiply(5, 0), "Multiple with zero should be zero");
}

public static int[][] data() {
    return new int[][] { { 1, 2, 2 }, { 5, 3, 15 }, { 121, 4, 484 }, { 2, 2, 2 } };
}

@ParameterizedTest
@MethodSource(value = "data")
void testWithStringParameter(int[] data) {
    JavaSampleLib tester = new JavaSampleLib();
    int m1 = data[0];
    int m2 = data[1];
    int expected = data[2];
    assertEquals(expected, tester.multiply(m1, m2));
}
}

```

if we run tests

The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays two projects: 'java-sample-app' and 'java-sample-lib'. Inside 'java-sample-lib', there is a 'src' folder containing 'ce103' and 'JavaSampleLibTest.java'. A context menu is open over 'JavaSampleLibTest.java', with 'Run As' selected. The code editor on the right shows Java code for a JUnit test class:

```

1 package ce103;
2
3 import static org.junit.jupiter.api.*;
4
5 import org.junit.jupiter.api.AfterEach;
6 import org.junit.jupiter.api.AfterAll;
7 import org.junit.jupiter.api.BeforeEach;
8 import org.junit.jupiter.api.BeforeAll;
9 import org.junit.jupiter.api.DisplayName;
10 import org.junit.jupiter.api.Repeat;
11 import org.junit.jupiter.api.Test;
12 import org.junit.jupiter.params.ParameterizedTest;
13 import org.junit.jupiter.params.provider.*;
14
15 class JavaSampleLibTest {
16
17     JavaSampleLib sampleLib;
18
19     @BeforeAll
20     static void setUpBeforeClass() {
21     }
22
23     @AfterAll
24     static void tearDownAfterClass() {
25     }
26
27     @BeforeEach
28     void setUp() throws Exception {
29         sampleLib = new JavaSampleLib();
30     }
31
32     @AfterEach
33     void tearDown() throws Exception {
34     }
35
36     @Test
37     @DisplayName("Simple Say Hello")
38     void testSayHelloTo() {
39         assertEquals("Hello Computer", sampleLib.sayHello("Computer"));
40     }
41
42     @ParameterizedTest
43     @MethodSource("dataProvider")
44     void testSayHelloWrong(@Value("Computer") String name, @Value("Hello Computer") String expected) {
45         assertEquals(expected, sampleLib.sayHello(name));
46     }
47
48     @Test
49     void dataProvider() {
50         yield().method("testSayHelloTo");
51         yield().method("testSayHelloWrong");
52     }
53
54     static class DataProvider {
55         static Stream<Arguments> testSayHelloTo() {
56             return Stream.of(
57                 Arguments.arguments("Computer", "Hello Computer"),
58                 Arguments.arguments("World", "Hello World")
59             );
60         }
61
62         static Stream<Arguments> testSayHelloWrong() {
63             return Stream.of(
64                 Arguments.arguments("Computer", "Hello Computer"),
65                 Arguments.arguments("World", "Hello World")
66             );
67         }
68     }
69 }

```

The JUnit Test view at the bottom shows one test result: 'Hello Computer'.

we will see all results there

The screenshot shows the Eclipse IDE interface with the JUnit View open. The toolbar at the top includes icons for file operations, search, and navigation. The JUnit tab is selected in the perspective bar. Below the toolbar, there are buttons for navigating through test results: a yellow arrow pointing down, a yellow arrow pointing up, a red X, a blue square, a blue lock, a green checkmark, a blue checkmark, a blue square with a minus sign, and a blue square with a plus sign.

Finished after 0.451 seconds

Runs: 14/14 Errors: 0 Failures: 3

JavaSampleLibTest [Runner: JUnit 5] (0.120 s)

- Simple sum shouldn't work (0.000 s)
- testWithStringParameter(int[]) (0.049 s)
 - [1] [1, 2, 2] (0.049 s)
 - [2] [5, 3, 15] (0.003 s)
 - [3] [121, 4, 484] (0.002 s)
 - [4] [2, 2, 2] (0.005 s)
- Simple sum should work (0.001 s)
- Simple Say Hello shouldn't work (0.004 s)
- Simple multiplication should work (0.001 s)
- Simple Say Hello should work (0.001 s)
- Ensure correct handling of zero (0.001 s)
 - repetition 1 of 5 (0.001 s)
 - repetition 2 of 5 (0.001 s)
 - repetition 3 of 5 (0.002 s)
 - repetition 4 of 5 (0.002 s)
 - repetition 5 of 5 (0.001 s)

Failure Trace

also we can see the code coverage of tests

Screenshot of the Eclipse IDE Coverage view showing code coverage for JavaSampleLibTest (Oct 24, 2021 5:07:01 PM).

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
java-sample-lib	92.4 %	182	15	197
src	92.4 %	182	15	197
ce103	92.4 %	182	15	197
JavaSampleLibTest.java	91.8 %	145	13	158
JavaSampleLib.java	94.9 %	37	2	39
JavaSampleLib	94.9 %	37	2	39
sayHelloTo(String)	91.7 %	22	2	24
sum(int, int)	100.0 %	8	0	8
multiply(int, int)	100.0 %	4	0	4

when we open our source code (just close and open again another case highlighting will not work) you will see tested part of your codes

```
1 package ce103;
2
3 public class JavaSampleLib {
4
5     public static String sayHelloTo(String name) {
6
7         String output = "";
8
9         if(!name.isBlank() && !name.isEmpty()){
10             output = "Hello "+name;
11         }else {
12             output = "Hello There";
13         }
14
15         System.out.println(output);
16
17         return output;
18     }
19
20     public static int sum(int a,int b)
21     {
22         int c = 0;
23         c = a+b;
24         return c;
25     }
26
27     public int multiply(int a, int b) {
28         return a * b;
29     }
30 }
```

1.0.6.3 Maven Java Application + JUnit Lets create Maven project with tests

Create a maven project

File -> New -> Maven Project



Select a wizard

Create a Maven project



Wizards:

type filter text

- Project
- Gradle
- Gradle Project
- Java
 - Java Project
 - Java Project from Existing Ant Buildfile
- Maven
 - Check out Maven Projects from SCM
 - Maven Module
 - Maven Project
- Examples
 - Editing and validating XML files

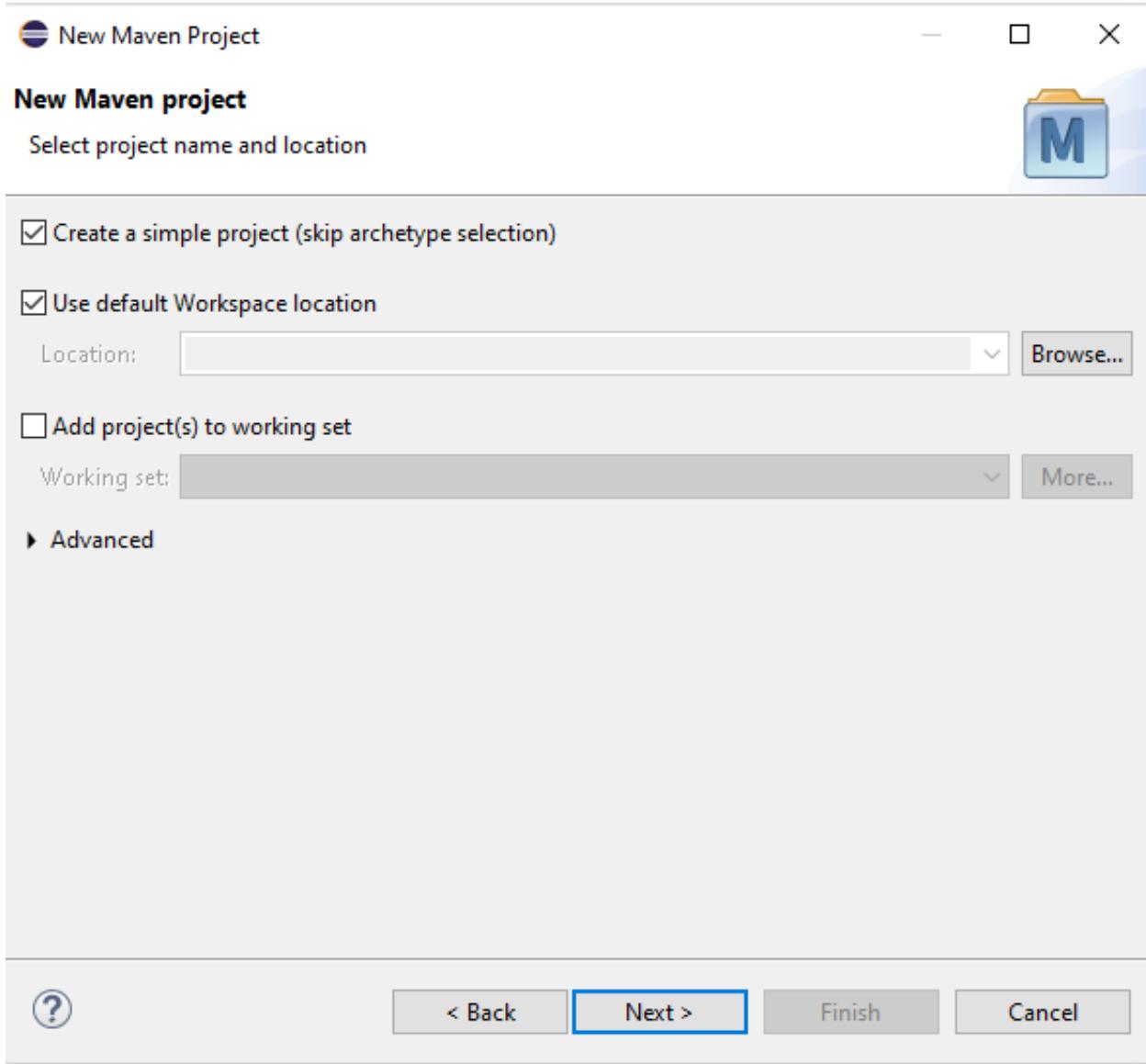


< Back

Next >

Finish

Cancel



Lets convert our sample java-sample-lib directories to standard folder structure for test and app division

Maven – Introduction to the Standard Directory Layout²⁸

Also for intro you can use this

JUnit Hello World Example - Examples Java Code Geeks - 2021²⁹

Eclipse

Maven

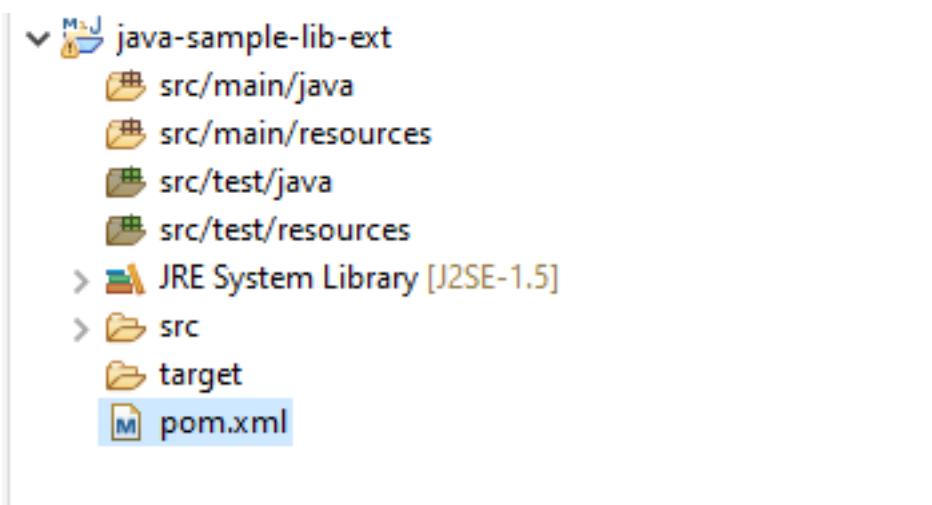
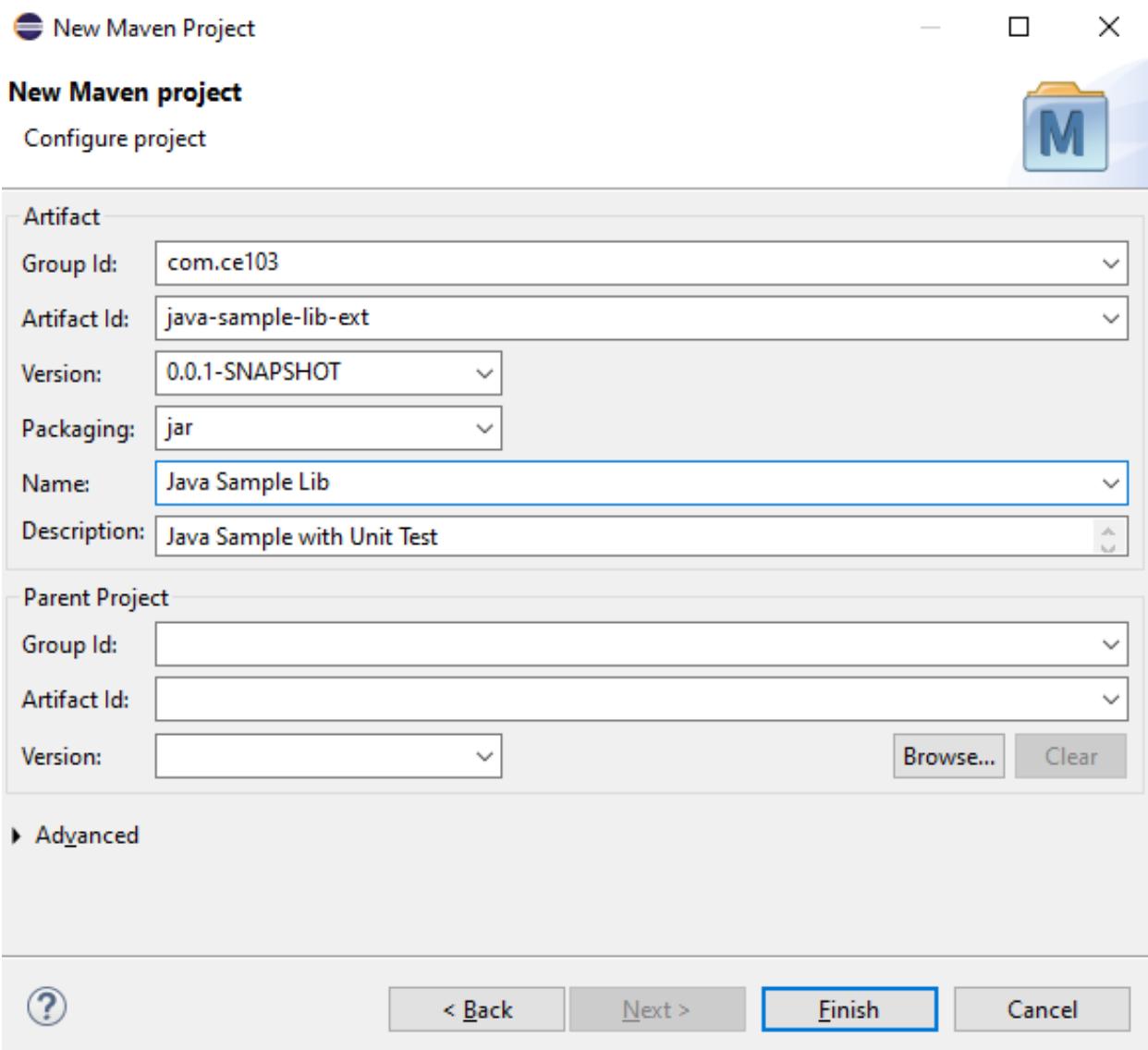
Java

JUnit 4.12 (pulled by Maven automatically)

Lets give new sample java-sample-lib-mvnbut in this time we will create a maven project

²⁸<http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html>

²⁹<https://examples.javacodegeeks.com/core-java/junit/junit-hello-world-example/>



pom.xml file

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.ce103</groupId>
  <artifactId>java-sample-lib-ext</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Java Sample Lib</name>
  <description>Java Sample with Unit Test</description>
</project>
```

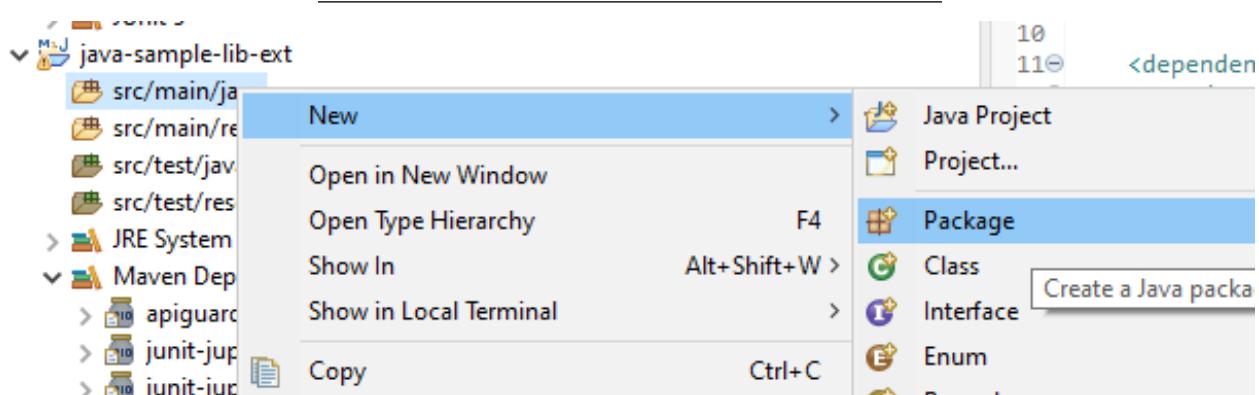
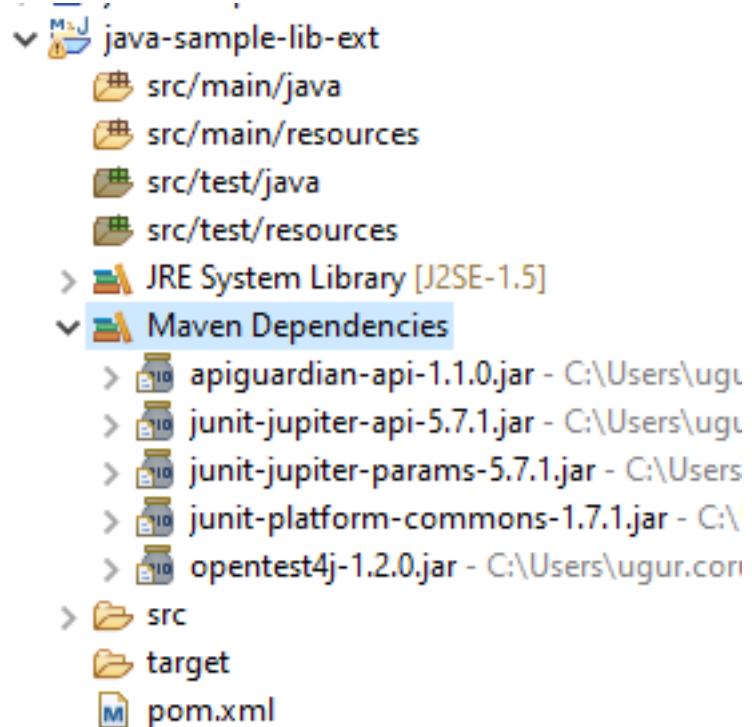
we will add JUnit 5 for our project

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.ce103</groupId>
  <artifactId>java-sample-lib-ext</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Java Sample Lib</name>
  <description>Java Sample with Unit Test</description>

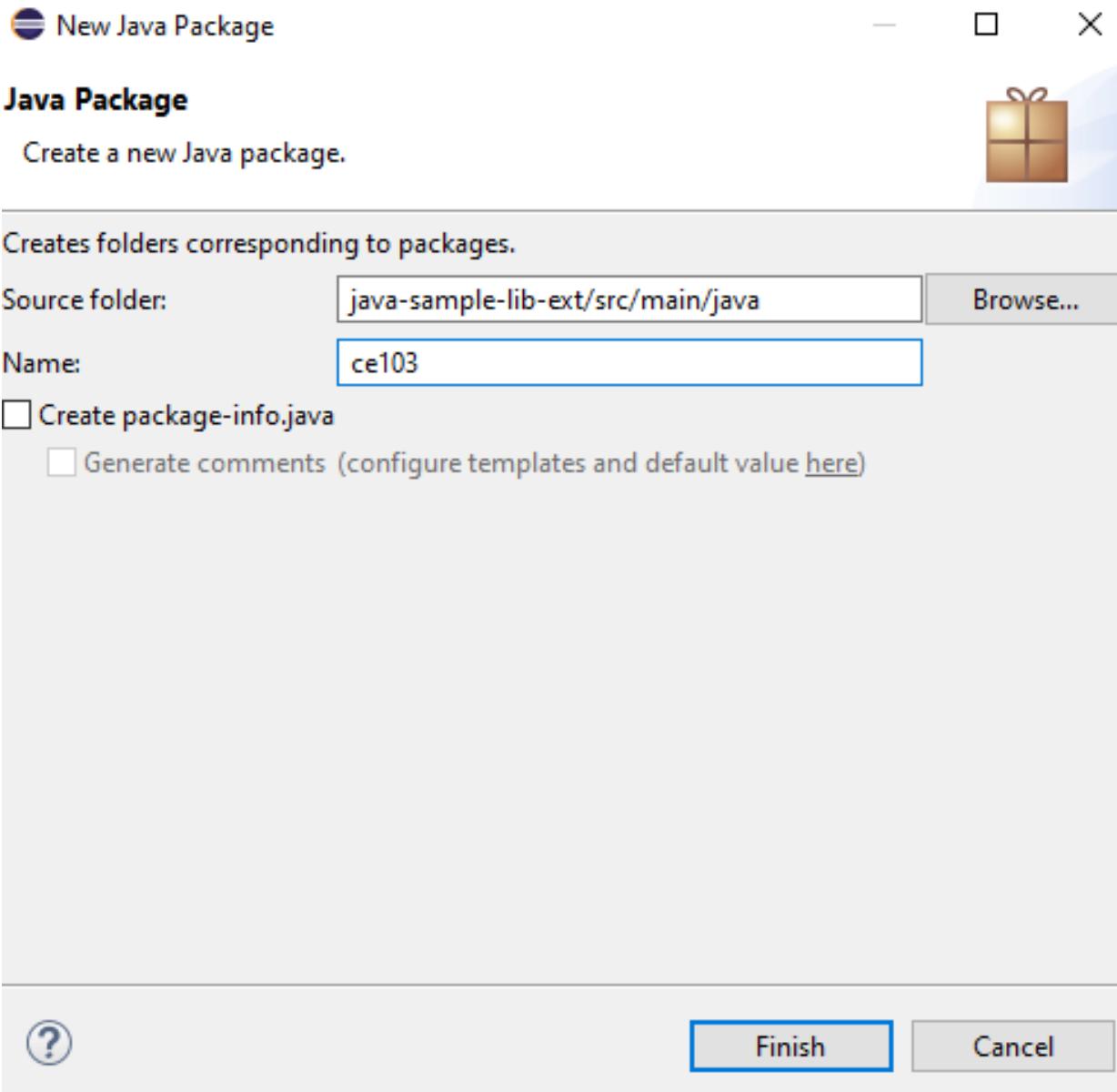
  <dependencies>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-params</artifactId>
      <version>5.7.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

</project>
```

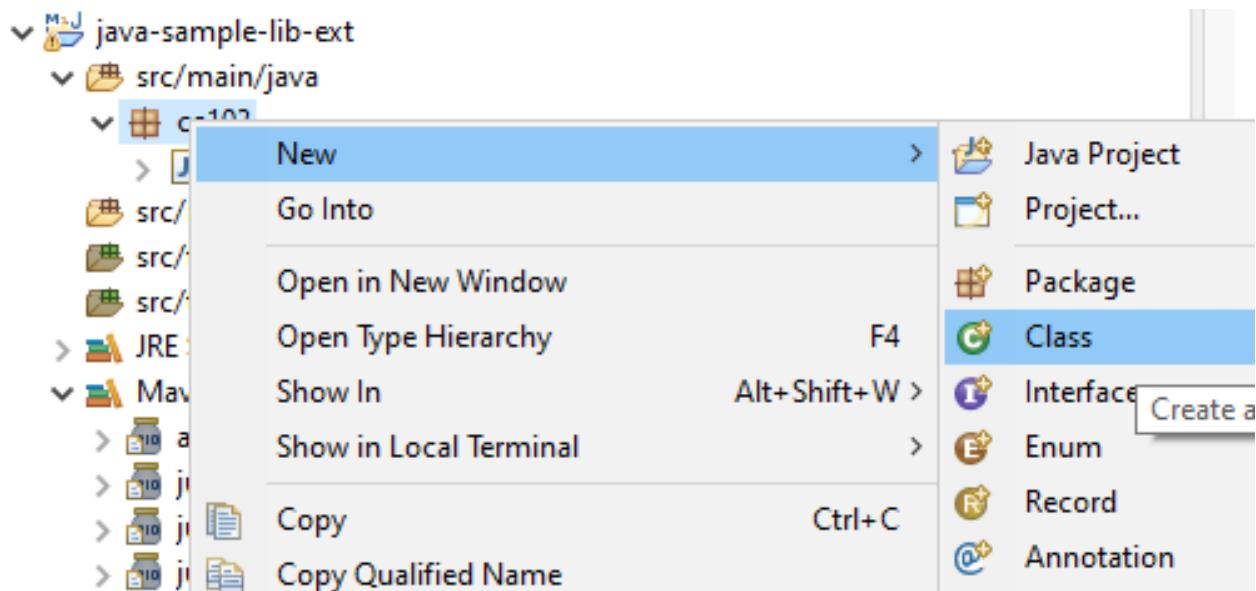
it will automatically download libraries

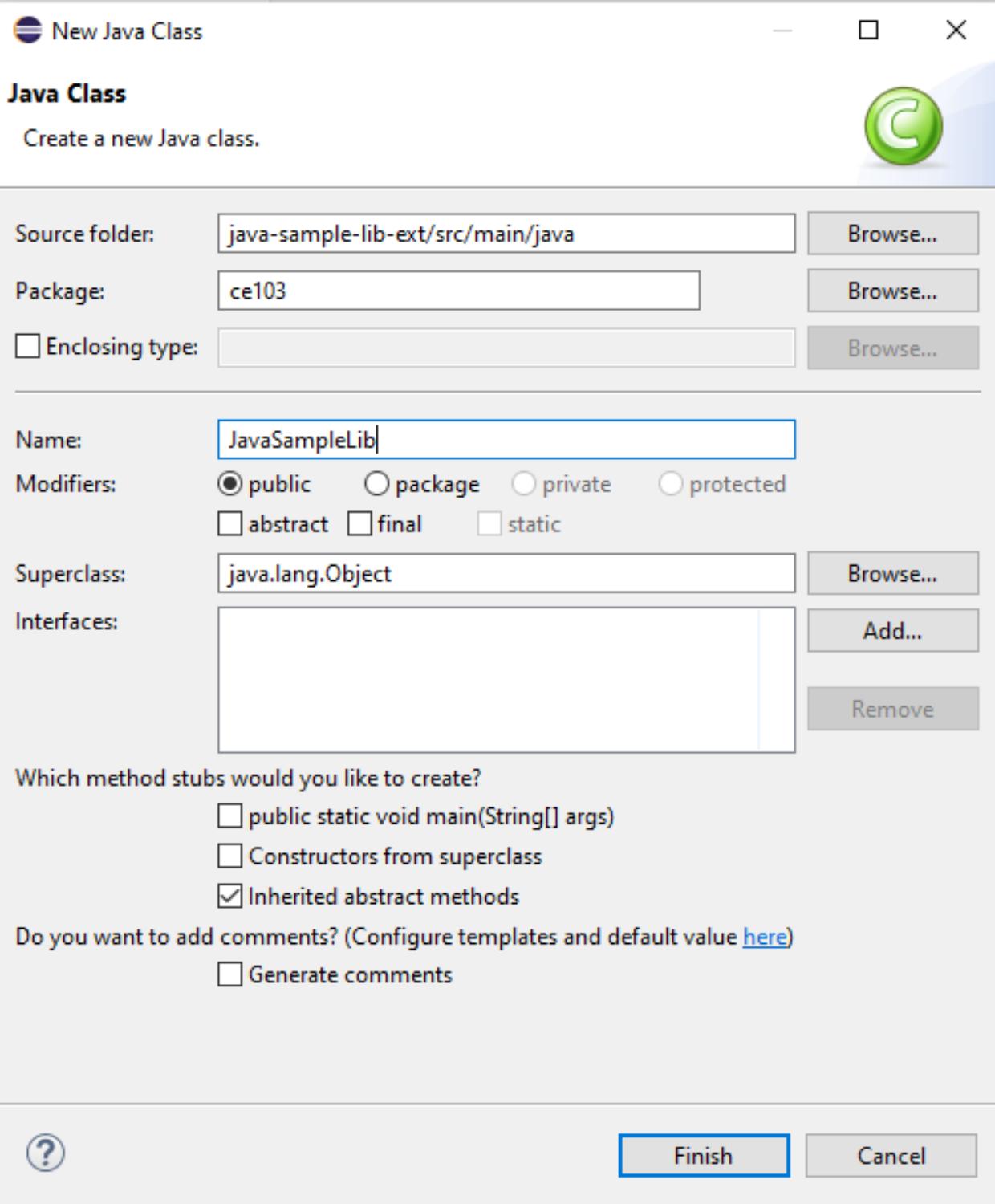


Create java sample library in ce103 package, first create java package



In this package create library class





copy content from other library

```
package ce103;

public class JavaSampleLib {

    public static String sayHelloTo(String name) {
```

```

        String output = "";

        if(!name.isBlank() && !name.isEmpty()){
            output = "Hello "+name;
        }else {
            output = "Hello There";
        }

        System.out.println(output);

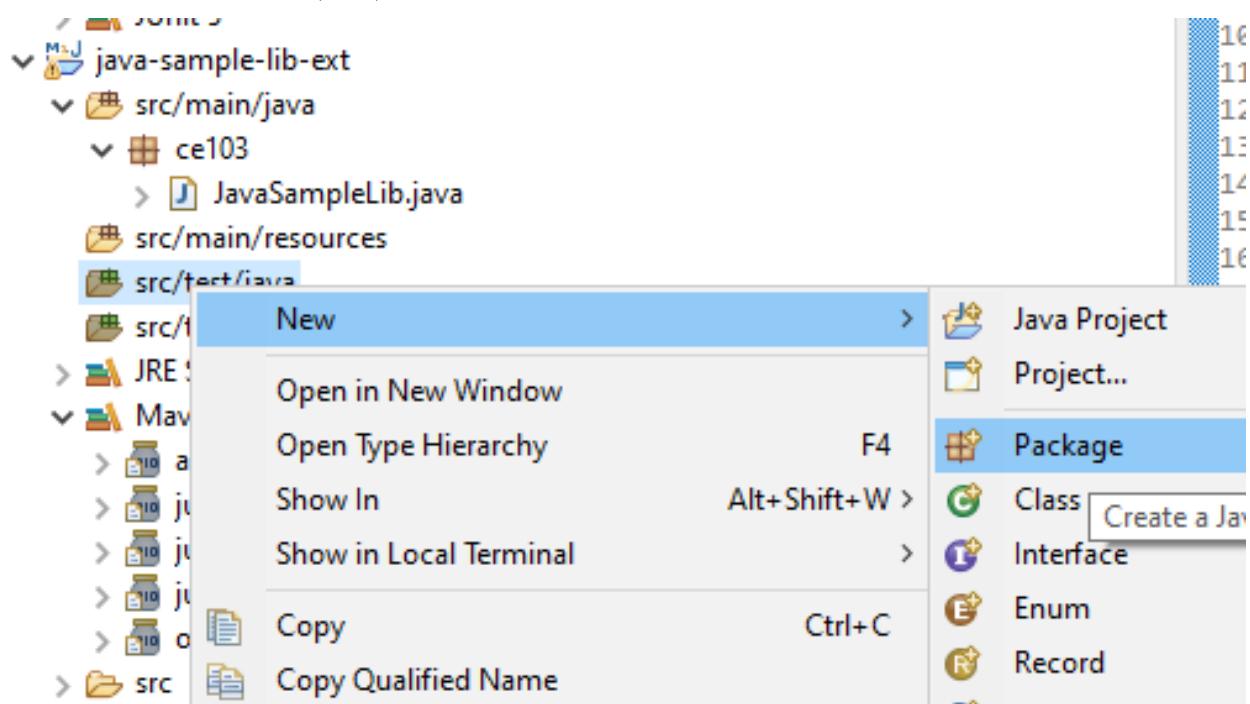
        return output;
    }

    public static int sum(int a,int b)
    {
        int c = 0;
        c = a+b;
        return c;
    }

    public int multiply(int a, int b) {
        return a * b;
    }
}

```

Now lets create tests inf src/test/java





New Java Package



Java Package



Create a new Java package.

Creates folders corresponding to packages.

Source folder:

java-sample-lib-ext/src/test/java

[Browse...](#)

Name:

ce103

Create package-info.java

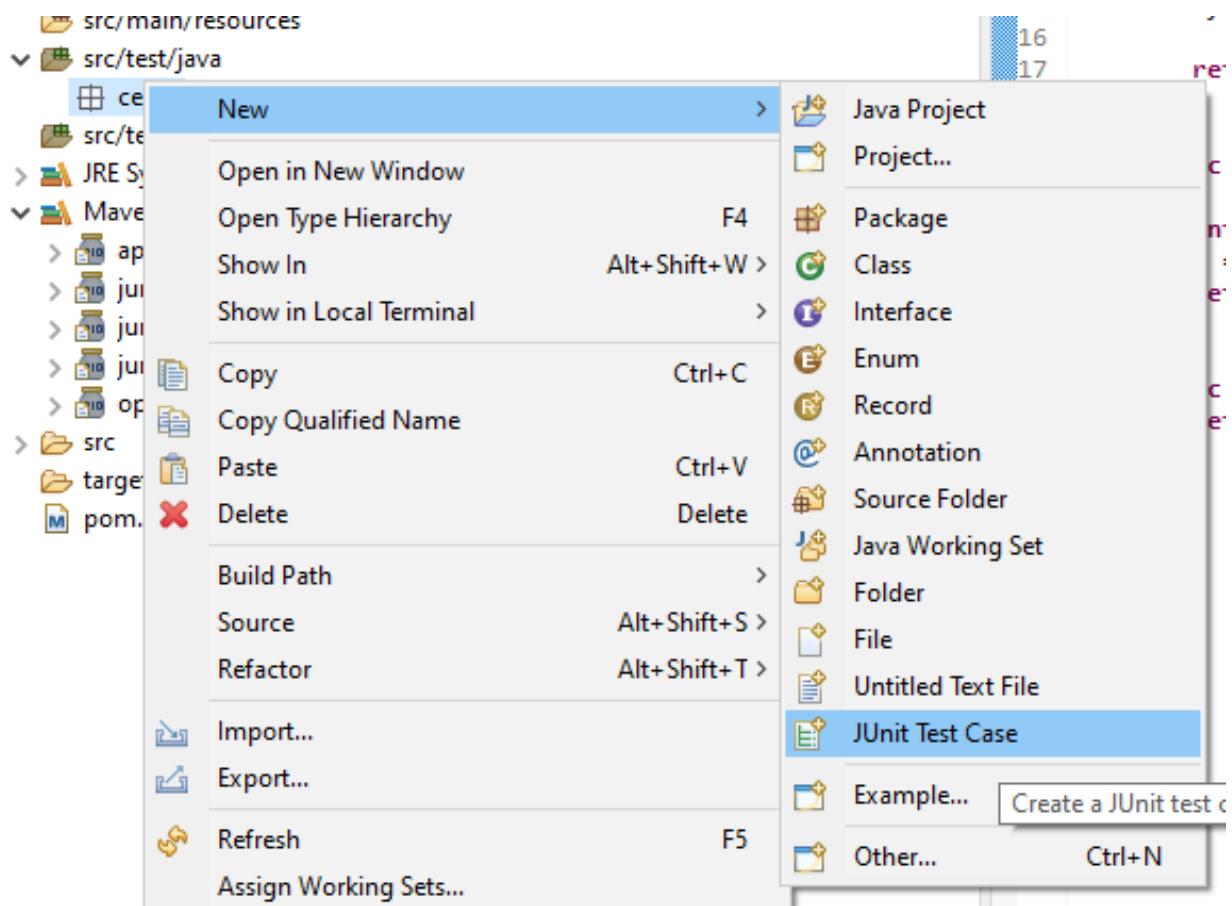
Generate comments (configure templates and default value [here](#))



[Finish](#)

[Cancel](#)

create a JUnit Case



New JUnit Test Case

JUnit Test Case

Select the name of the new JUnit test case. Specify the class under test to select methods to be tested on the next page.



New JUnit 3 test New JUnit 4 test New JUnit Jupiter test

Source folder:

Package:

Name:

Superclass:

Which method stubs would you like to create?

- @BeforeAll setUpBeforeClass() @AfterAll tearDownAfterClass()
 @BeforeEach setUp() @AfterEach tearDown()
 constructor

Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

Class under test:

JUnit 5 requires a Java 8 project. [Configure](#) project compliance and the project build path.

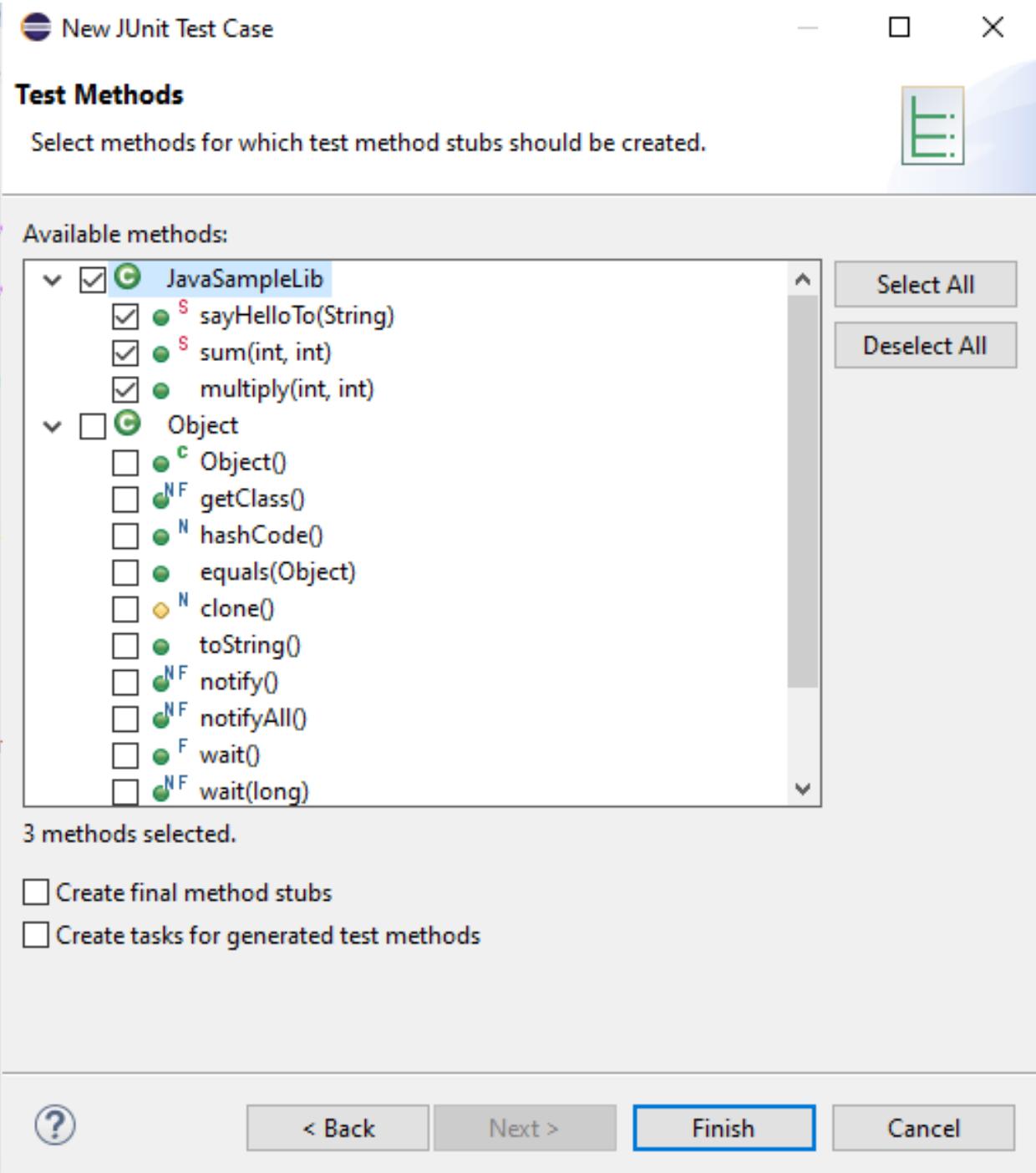


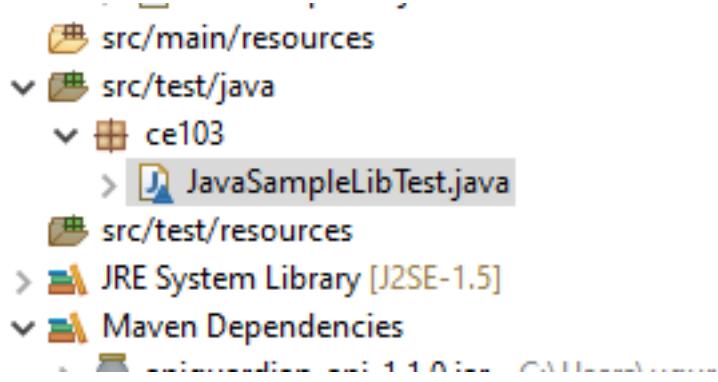
< Back

Next >

Finish

Cancel





you will simple template

```
package ce103;

import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

class JavaSampleLibTest {

    @BeforeAll
    static void setUpBeforeClass() throws Exception {
    }

    @AfterAll
    static void tearDownAfterClass() throws Exception {
    }

    @BeforeEach
    void setUp() throws Exception {
    }

    @AfterEach
    void tearDown() throws Exception {
    }

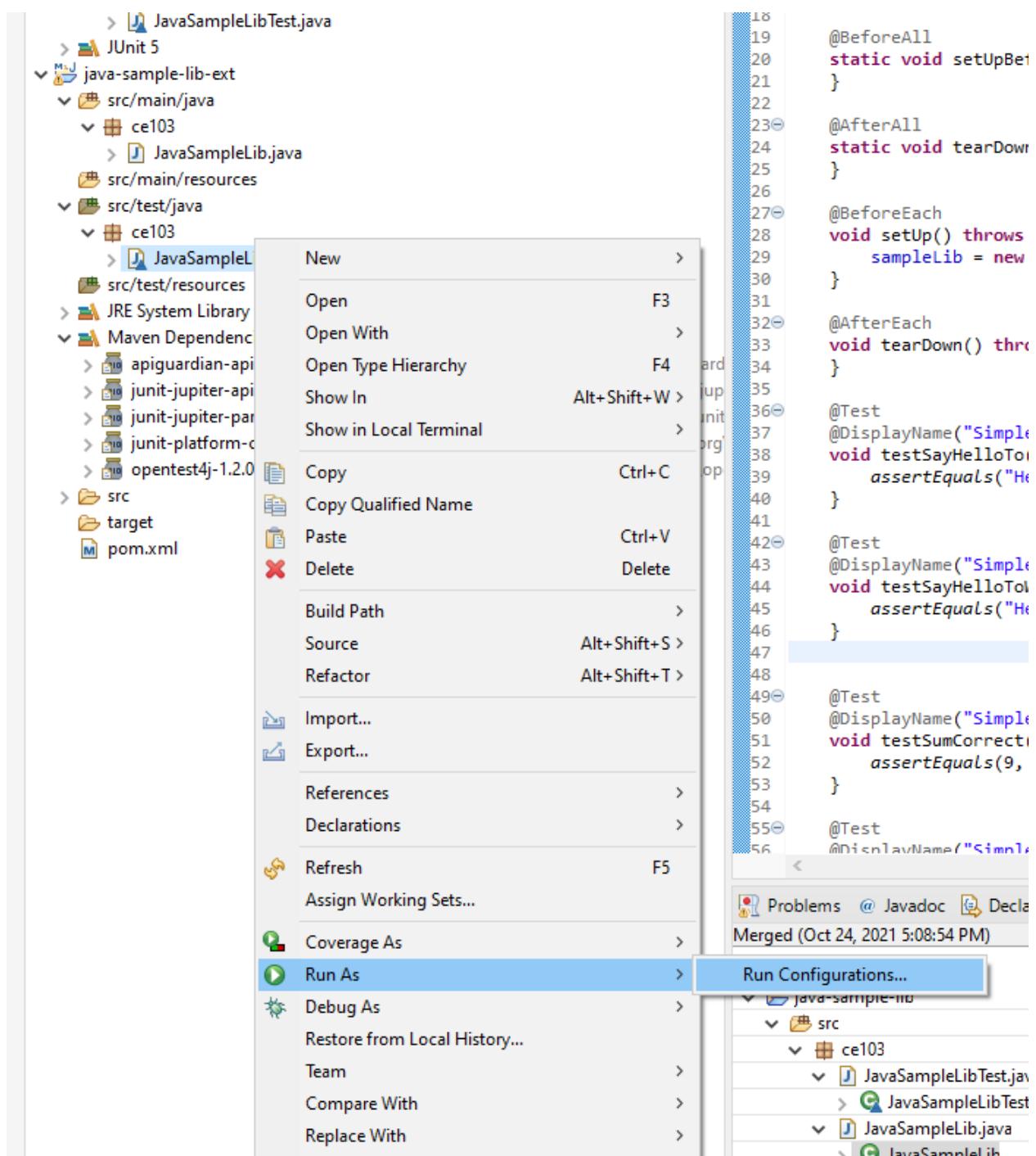
    @Test
    void testSayHelloTo() {
        fail("Not yet implemented");
    }

    @Test
    void testSum() {
        fail("Not yet implemented");
    }

    @Test
    void testMultiply() {
        fail("Not yet implemented");
    }
}
```

}

now lets copy tests from other projects



Package Explorer JUnit X

Finished after 0.407 seconds

Runs: 14/14 Errors: 0 Failures: 3

JavaSampleLibTest [Runner: JUnit 5] (0.129 s)

- Simple sum shouldn't work (0.000 s)
- testWithStringParameter(int[]) (0.049 s)
 - [1] [1, 2, 2] (0.049 s)
 - [2] [5, 3, 15] (0.002 s)
 - [3] [121, 4, 484] (0.001 s)
 - [4] [2, 2, 2] (0.003 s)
- Simple sum should work (0.003 s)
- Simple Say Hello shouldn't work (0.005 s)
- Simple multiplication should work (0.003 s)
- Simple Say Hello should work (0.002 s)
- Ensure correct handling of zero (0.002 s)
 - repetition 1 of 5 (0.002 s)
 - repetition 2 of 5 (0.001 s)
 - repetition 3 of 5 (0.001 s)
 - repetition 4 of 5 (0.002 s)
 - repetition 5 of 5 (0.002 s)

The screenshot shows the Eclipse IDE interface with the following components:

- Top Bar:** eclipse-workspace - java-sample-lib-ext/src/main/java/ce103/JavaSampleLib.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- Package Explorer:** Shows JavaSampleLibTest [Runner: JUnit 5] (0.278 s) with 14/14 runs, 0 errors, and 3 failures.
- Java Sample Library Code:**

```

1 package ce103;
2
3 public class JavaSampleLib {
4     public static String sayHelloTo(String name) {
5         String output = "";
6
7             if(!name.isBlank() && !name.isEmpty()){
8                 output = "Hello "+name;
9             } else {
10                 output = "Hello There";
11             }
12
13             System.out.println(output);
14
15             return output;
16         }
17
18         public static int sum(int a,int b)
19         {
20             int c = 0;
21             c = a+b;
22             return c;
23         }
24
25         public int multiply(int a, int b) {
26             return a * b;
27         }
28     }
29
30
31
32 }
```
- Failure Trace:** Shows an assertion error: org.opentest4j.AssertionFailedError: Regular sum shouldn't work ==> expected: <10> but was: <11> at ce103.JavaSampleLibTest.testSumWrong(JavaSampleLibTest.java:58)
- Bottom Right:** Coverage report for JavaSampleLibTest (Oct 24, 2021 5:39:34 PM).

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
java-sample-lib-ext	92.4 %	182	15	197
src/test/java	91.8 %	145	13	158
ce103	91.8 %	145	13	158
src/main/java	94.9 %	37	2	39
ce103	94.9 %	37	2	39
JavaSampleLib.java	94.9 %	37	2	39

That's a part of java unit testing...

1.1 TDD (Test Driven Development)

1.2 Test and Deployment Automation Management

1.2.0.1 Travis-CI + C

1.2.0.2 Travis-CI + Cpp

1.2.0.3 Travis-CI + C

1.2.0.4 Travis-CI + Java

2 References

GitHub - MicrosoftDocs/cpp-docs: C++ Documentation³⁰

³⁰<https://github.com/MicrosoftDocs/cpp-docs>