

CE103 Algorithms and Programming I

Week-4

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

Shared Library Development

C Programming (Static Library)

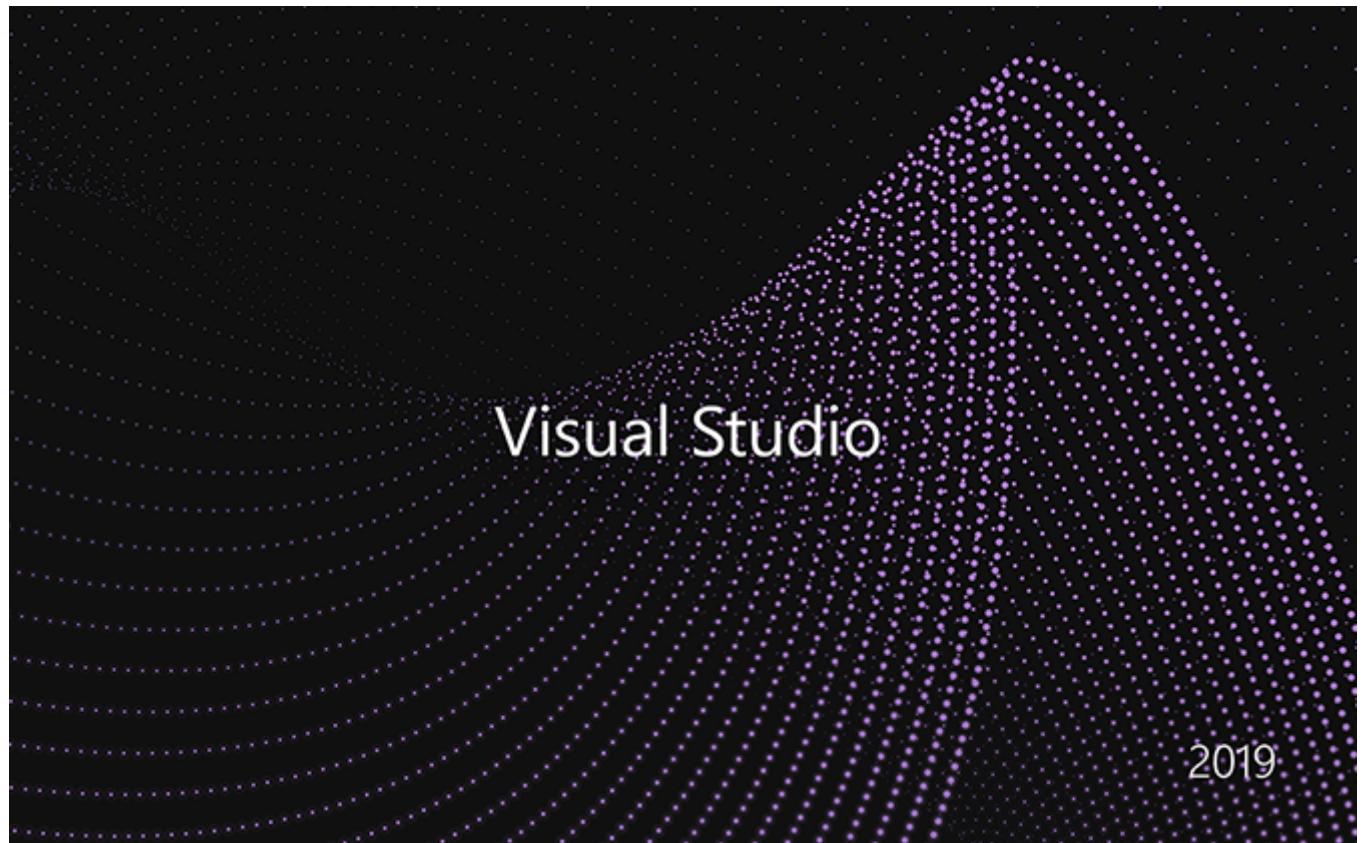
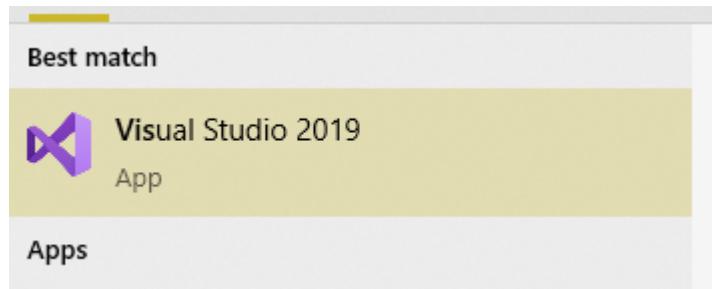
Visual Studio Community Edition

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 İndir](#)

Open visual studio community edition and select create a new project



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\ce11

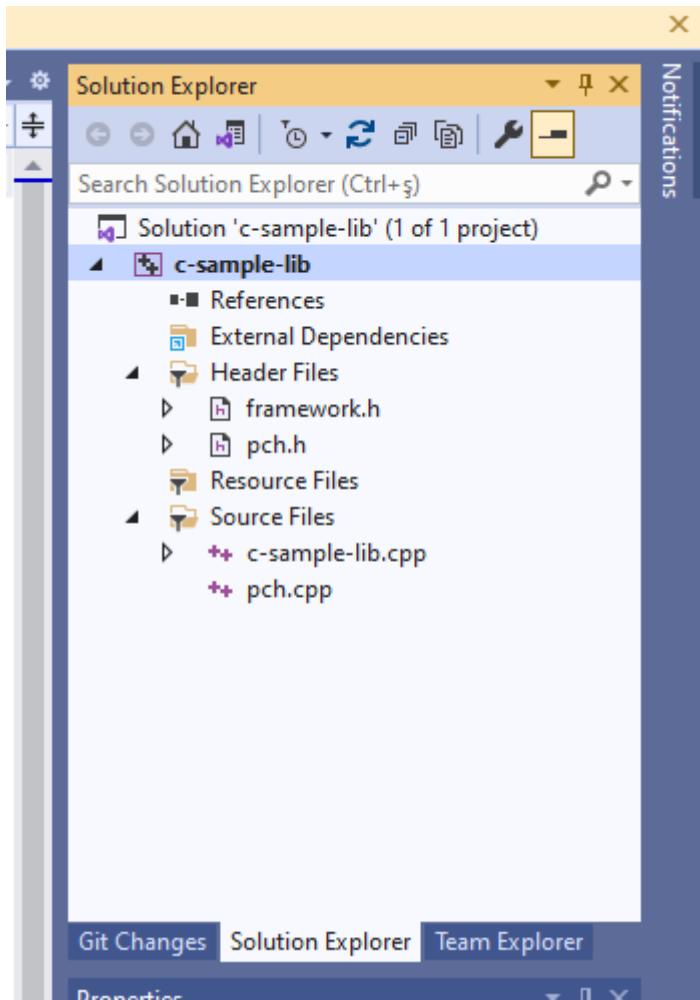
...

Solution name i

c-sample-lib

Place solution and project in the same directory

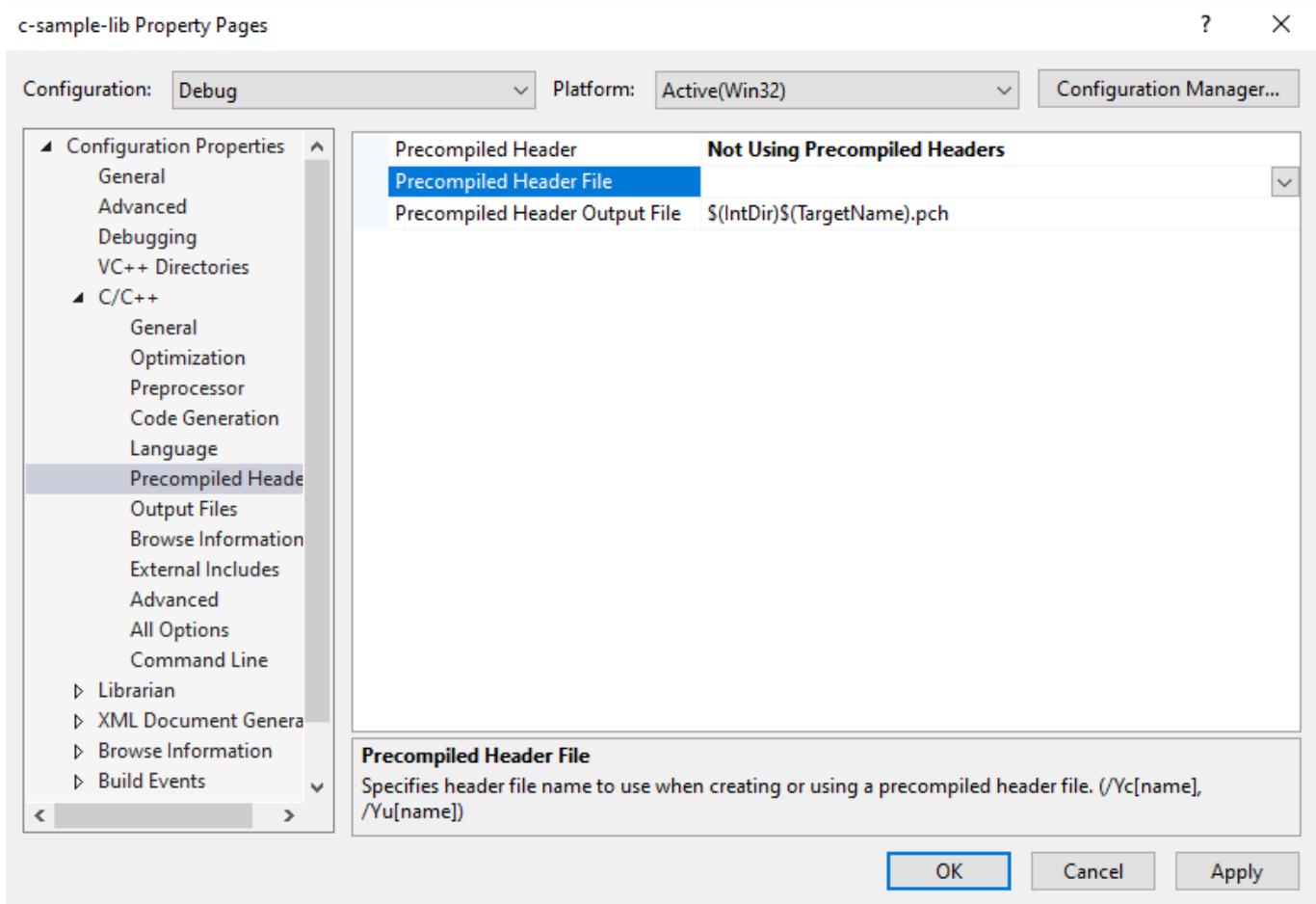
Default configuration come with C++ project types and setting



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

```

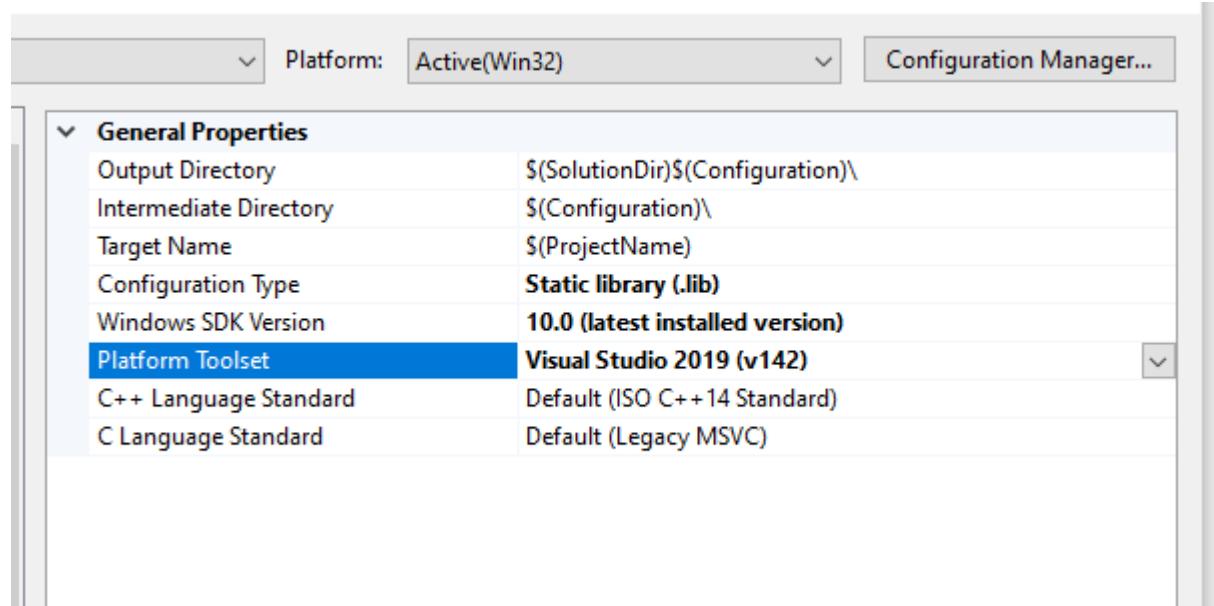
#pragma once

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

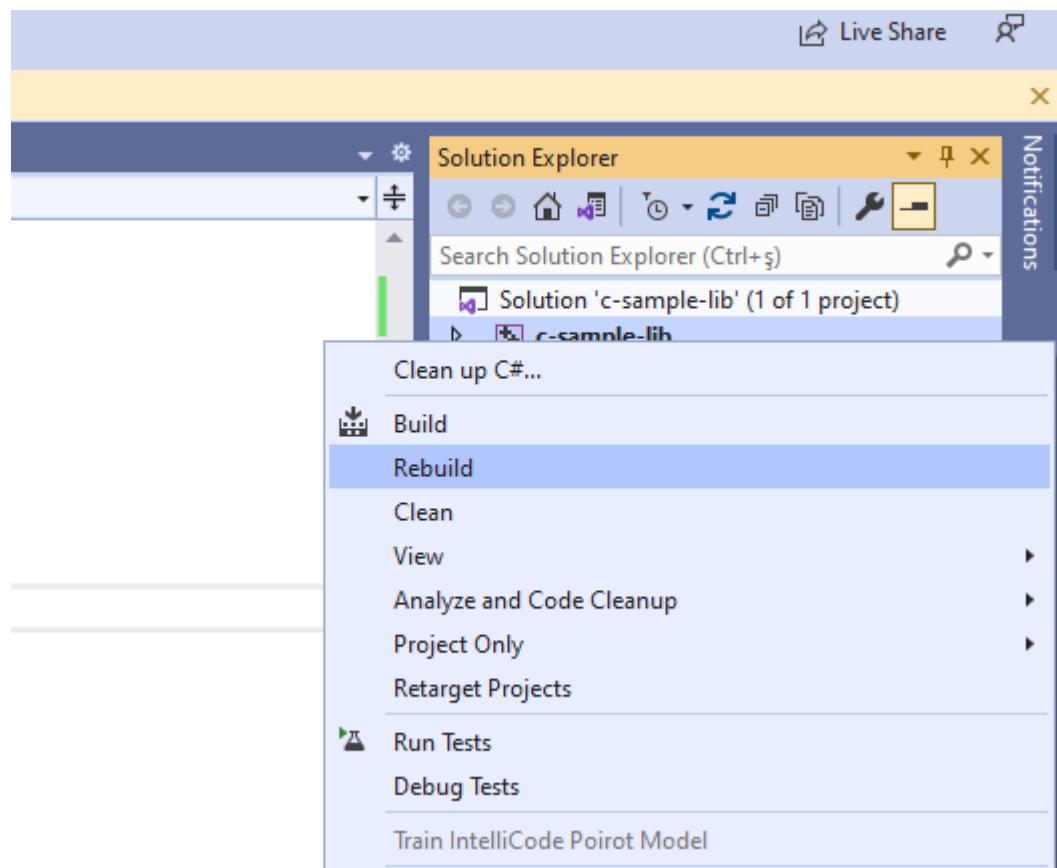
void sayHelloTo(char* name);
int sum(int a, int b);

```

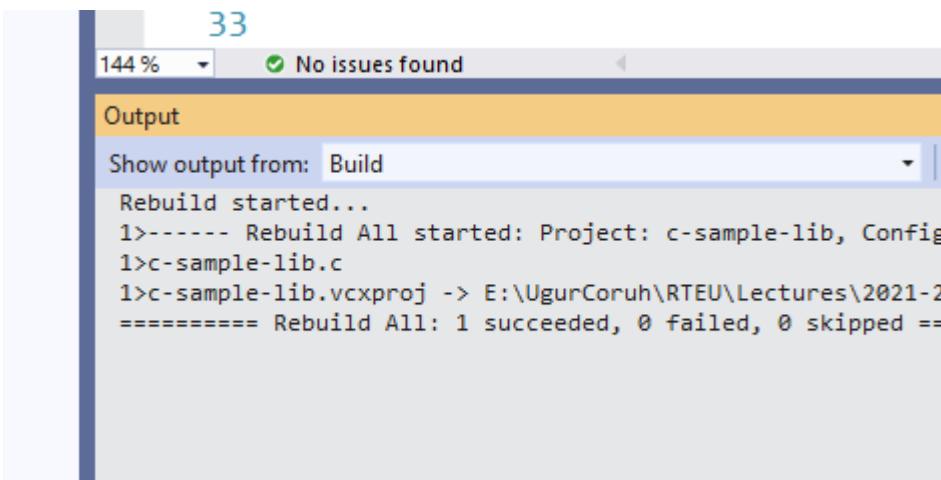
If you check configuration you will see that for C complier we are using Microsoft Environment and Toolkits



Now we can compile our library



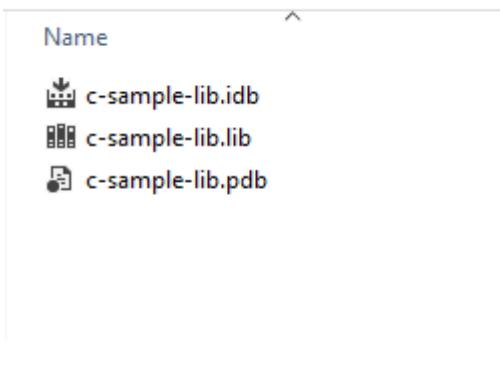
You can follow operation from output window



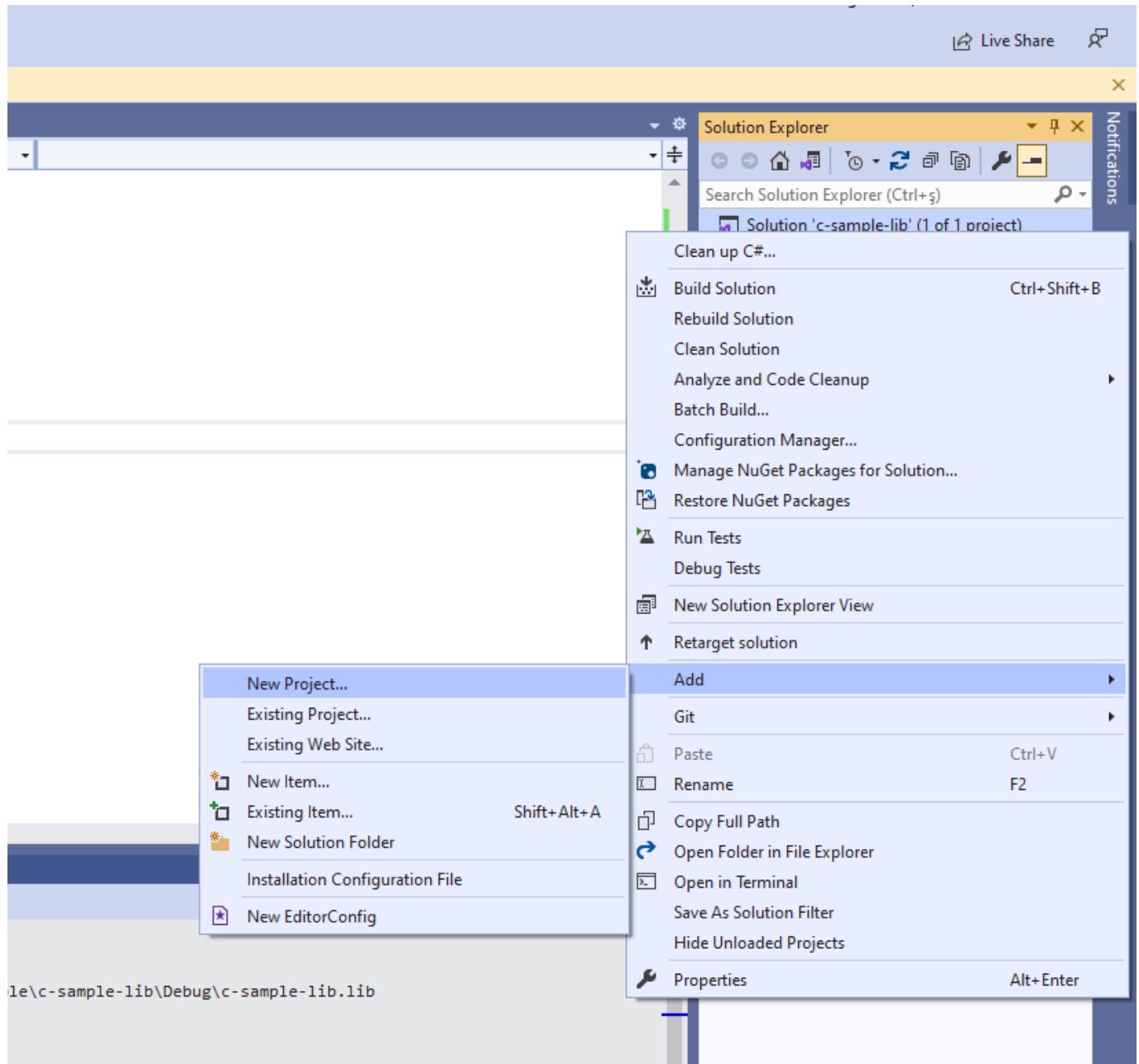
The screenshot shows the Visual Studio Output window with the following content:

```
33
144 % No issues found
Output
Show output from: Build
Rebuild started...
1>----- Rebuild All started: Project: c-sample-lib, Configuration: Debug
1>c-sample-lib.c
1>c-sample-lib.vcxproj -> E:\UgurCoruh\RTEU\Lectures\2021-2
===== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==
```

in debug folder we will see our output



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.
C++ Windows Console

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

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

Windows Desktop Wizard

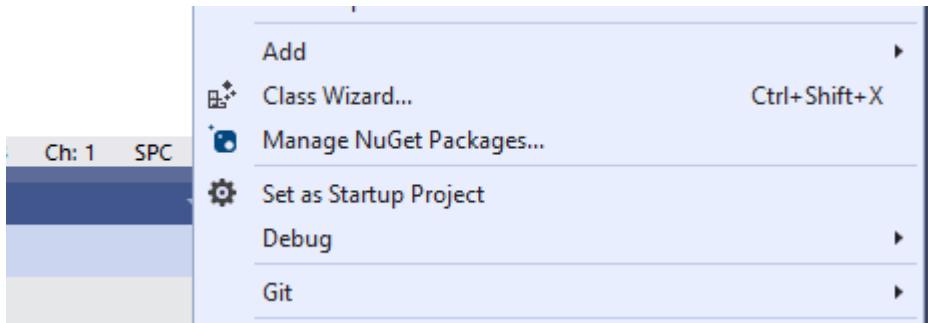
C++ Console Application Selection will generate a C++ console project we can change extension to C to compile our application as C application.

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

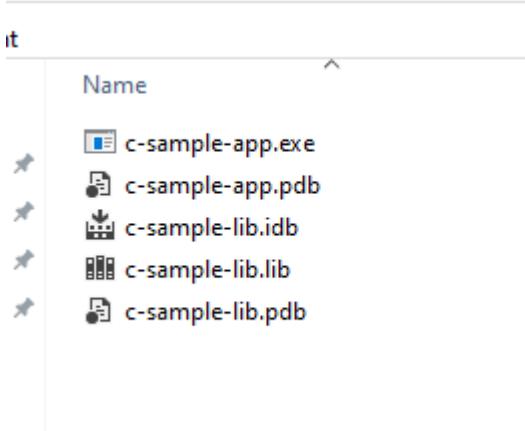
```
#include <stdio.h>

/// <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

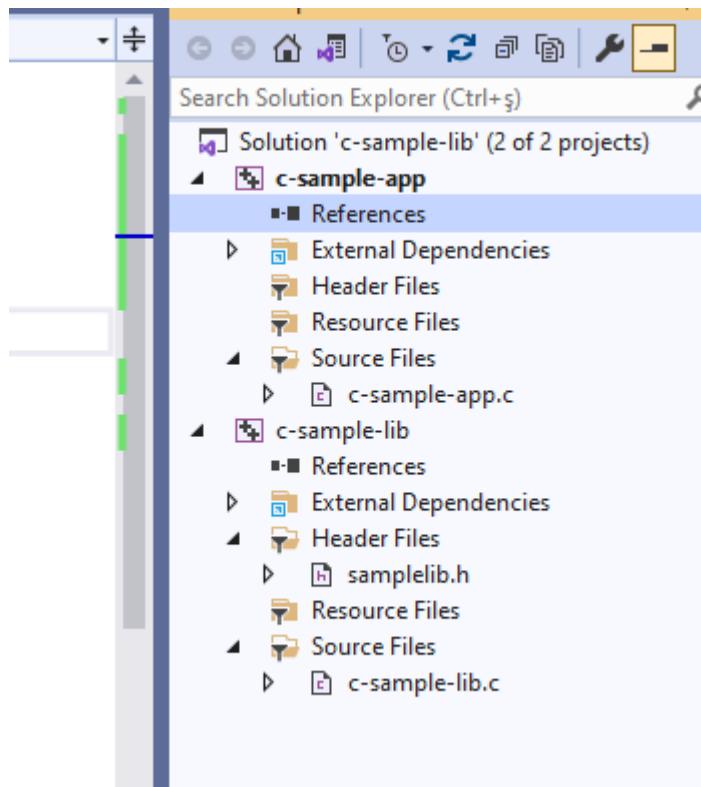


if we run application we will see only "Hello World"

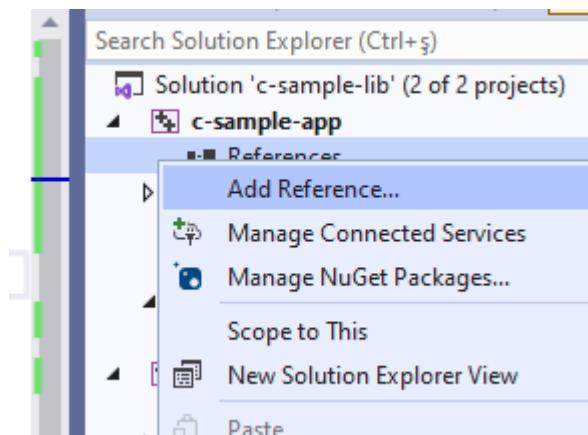
now we will see two options to add library as references in our application and use its functions.

First option

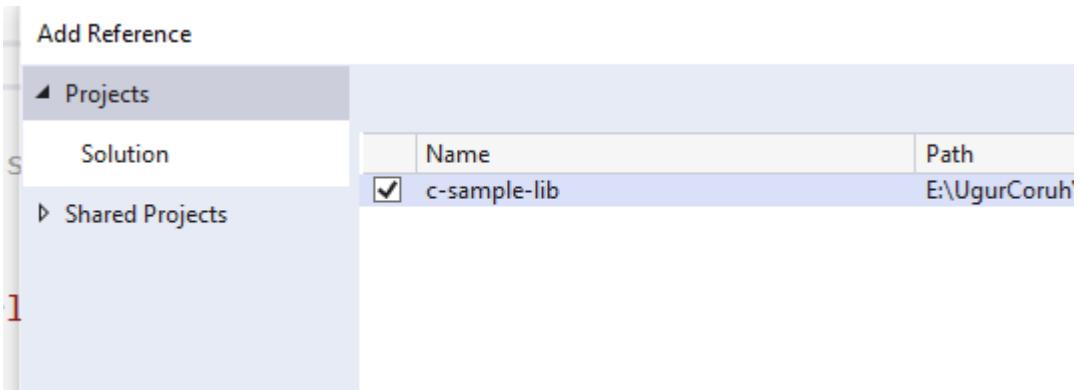
right click references for c-sample-app and add current library as reference



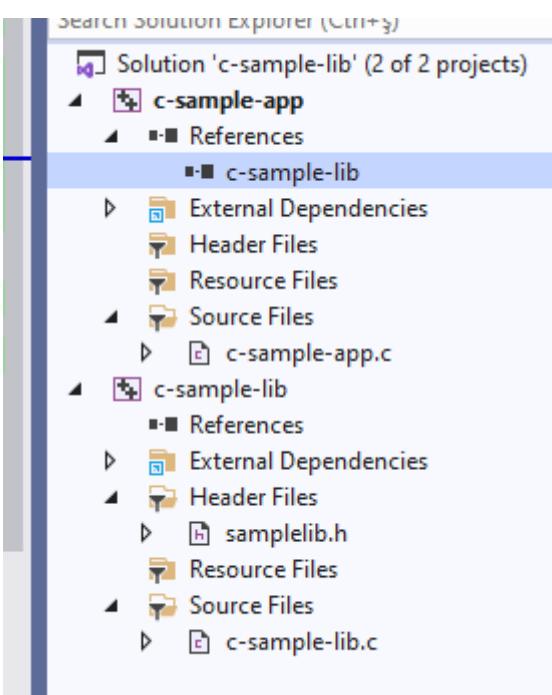
Select Add Reference



Browse for solution and select c-sample-lib



You can check added reference from references section

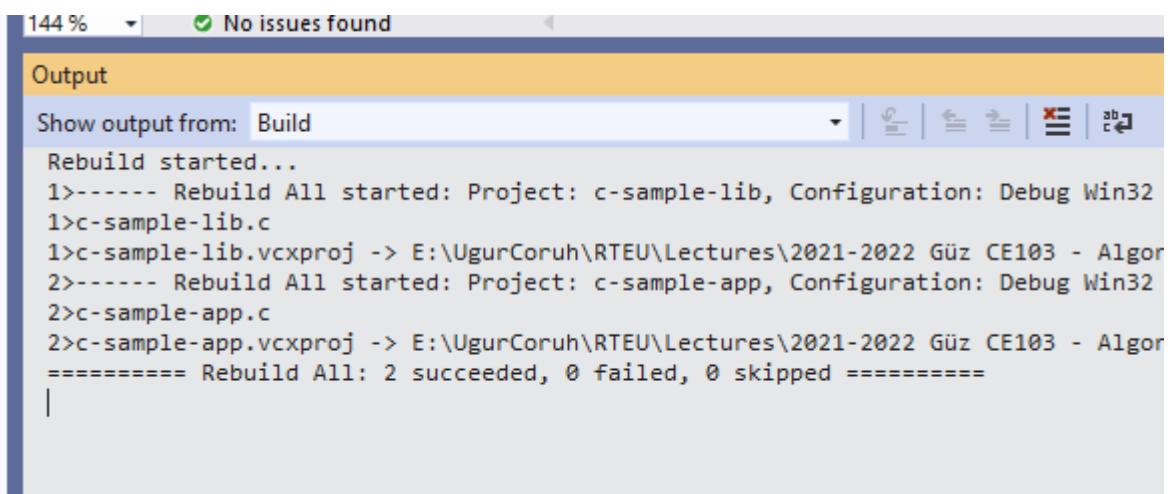


now we can include required headers from c-sample-lib folder and use it.

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



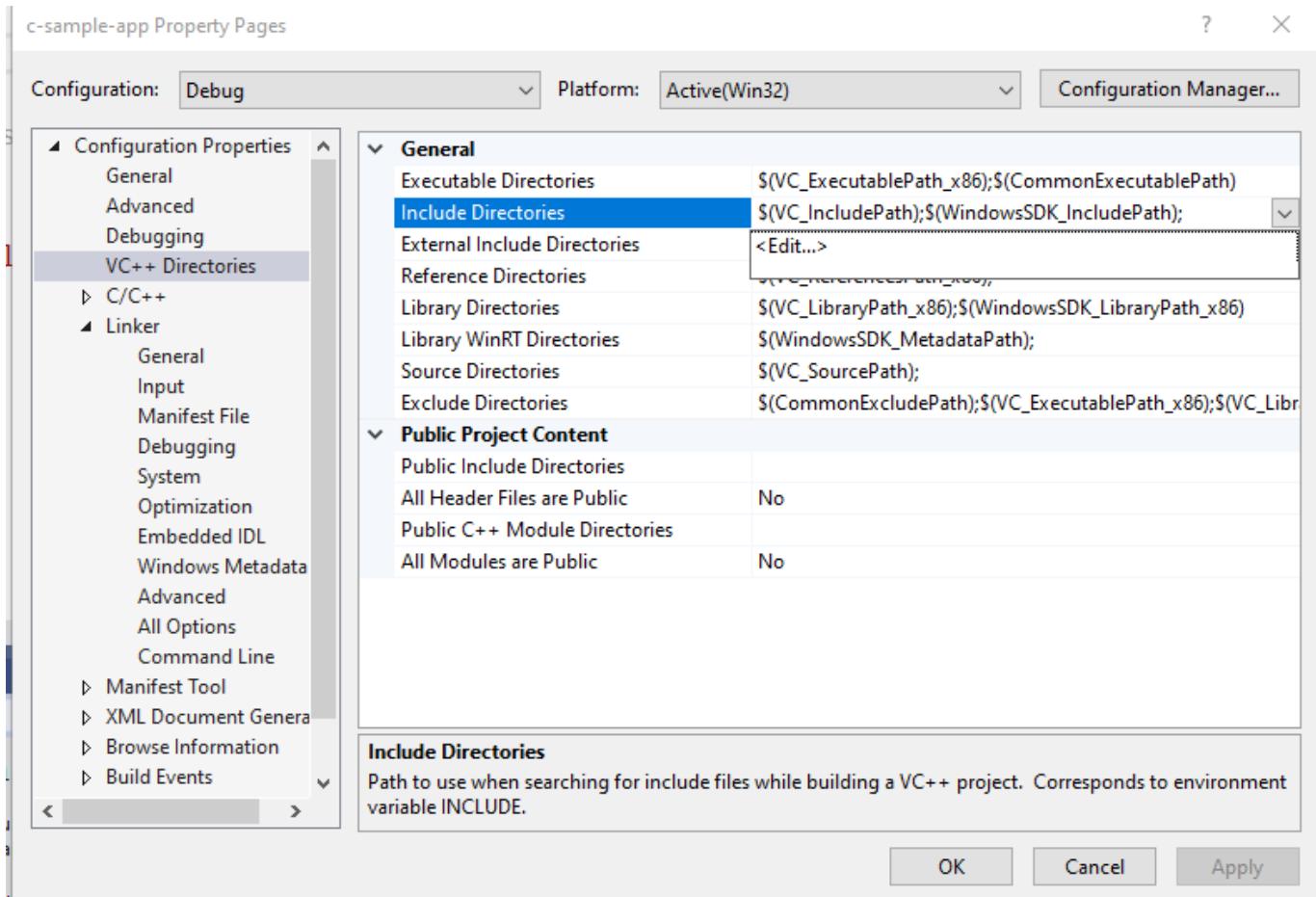
The screenshot shows the Visual Studio Output window with the following log output:

```
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 =====
```

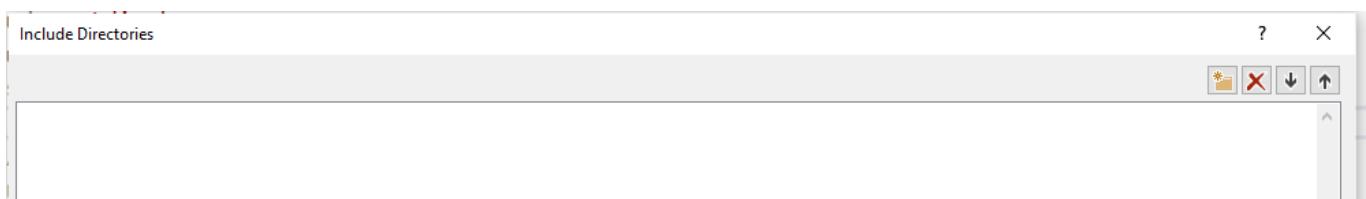
also we can only write header name

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

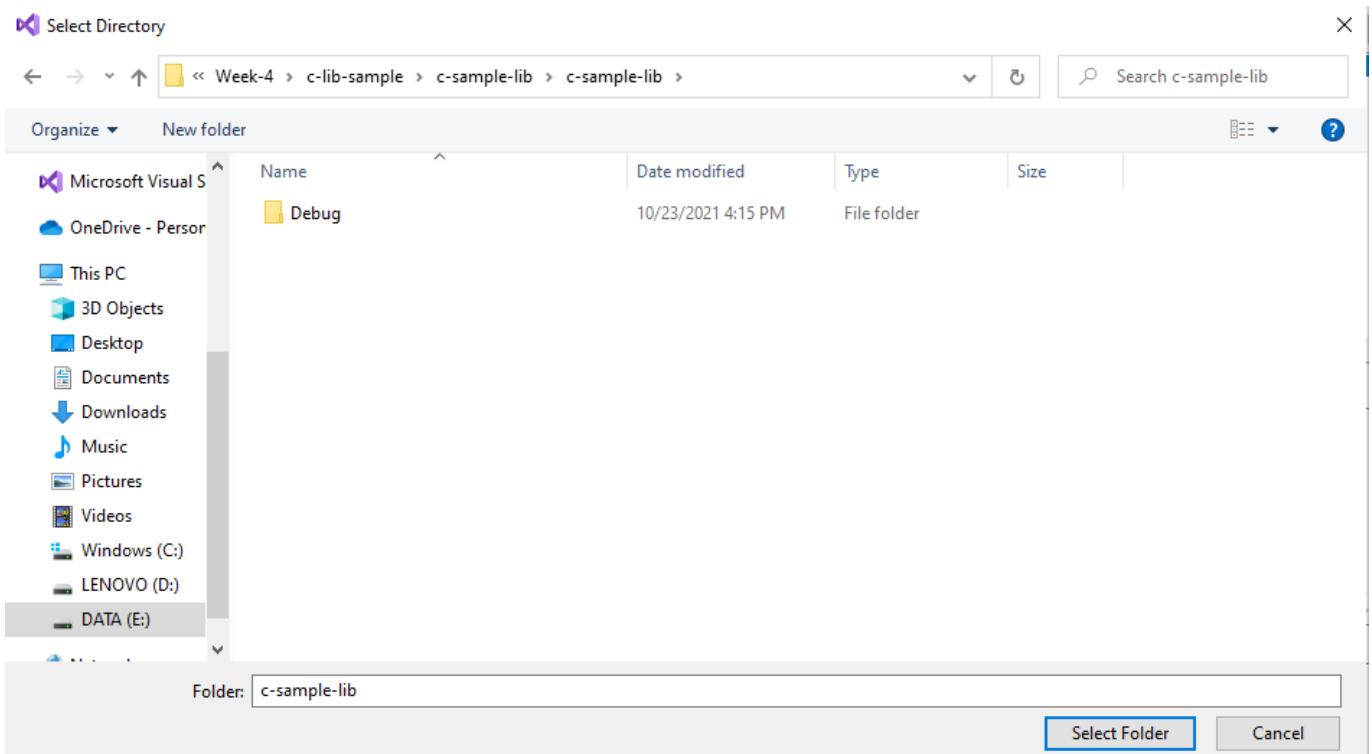
for this we need to configure include directories



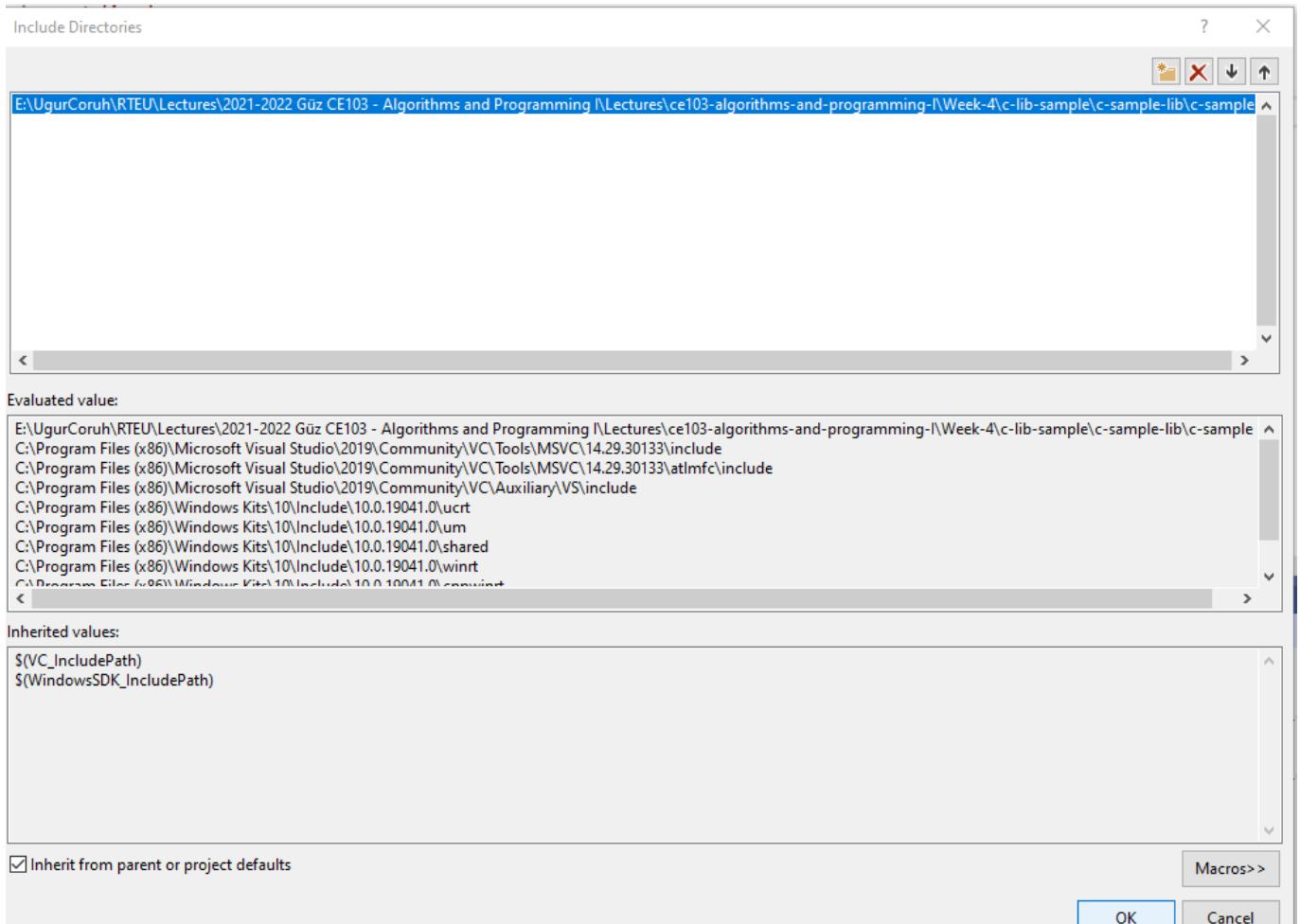
select c-sample-lib header file location



browse for folder



your full path will be added to your configuration

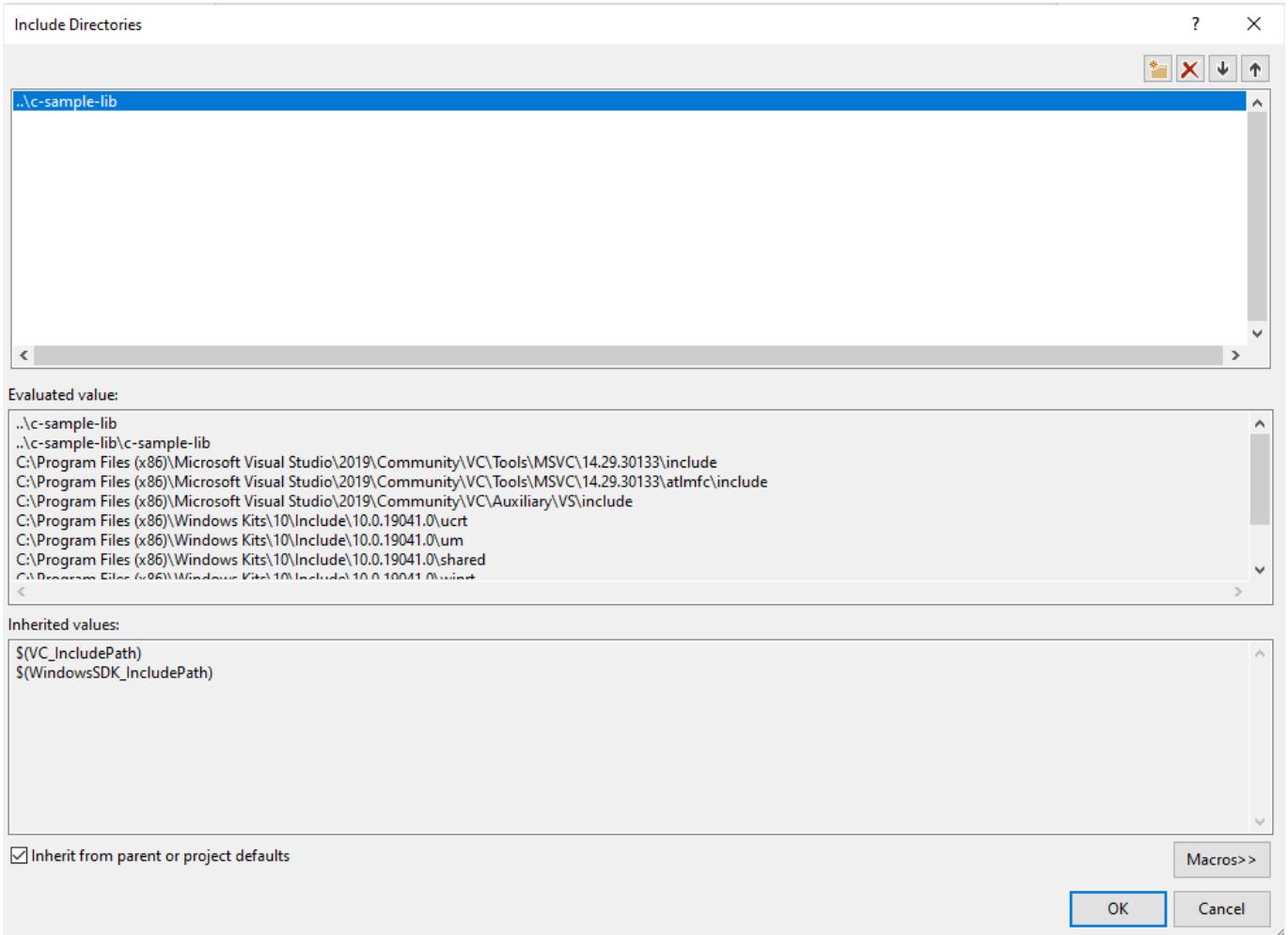


if you add header file paths to your configuration you can use header files by name in your source code

```
#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

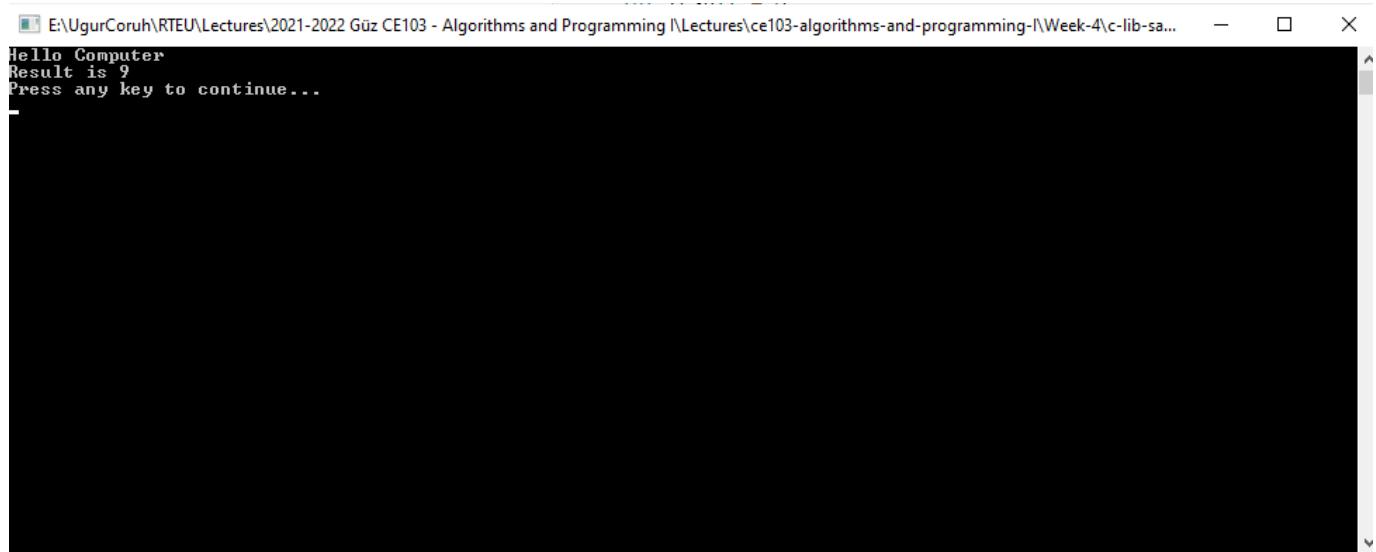
```
.. \c-sample-lib
```



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.



```
E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programming \Lectures\ce103-algorithms-and-programming-\Week-4\c-lib-sa...
Hello Computer
Result is 9
Press any key to continue...
```

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

c-sample-app Property Pages

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

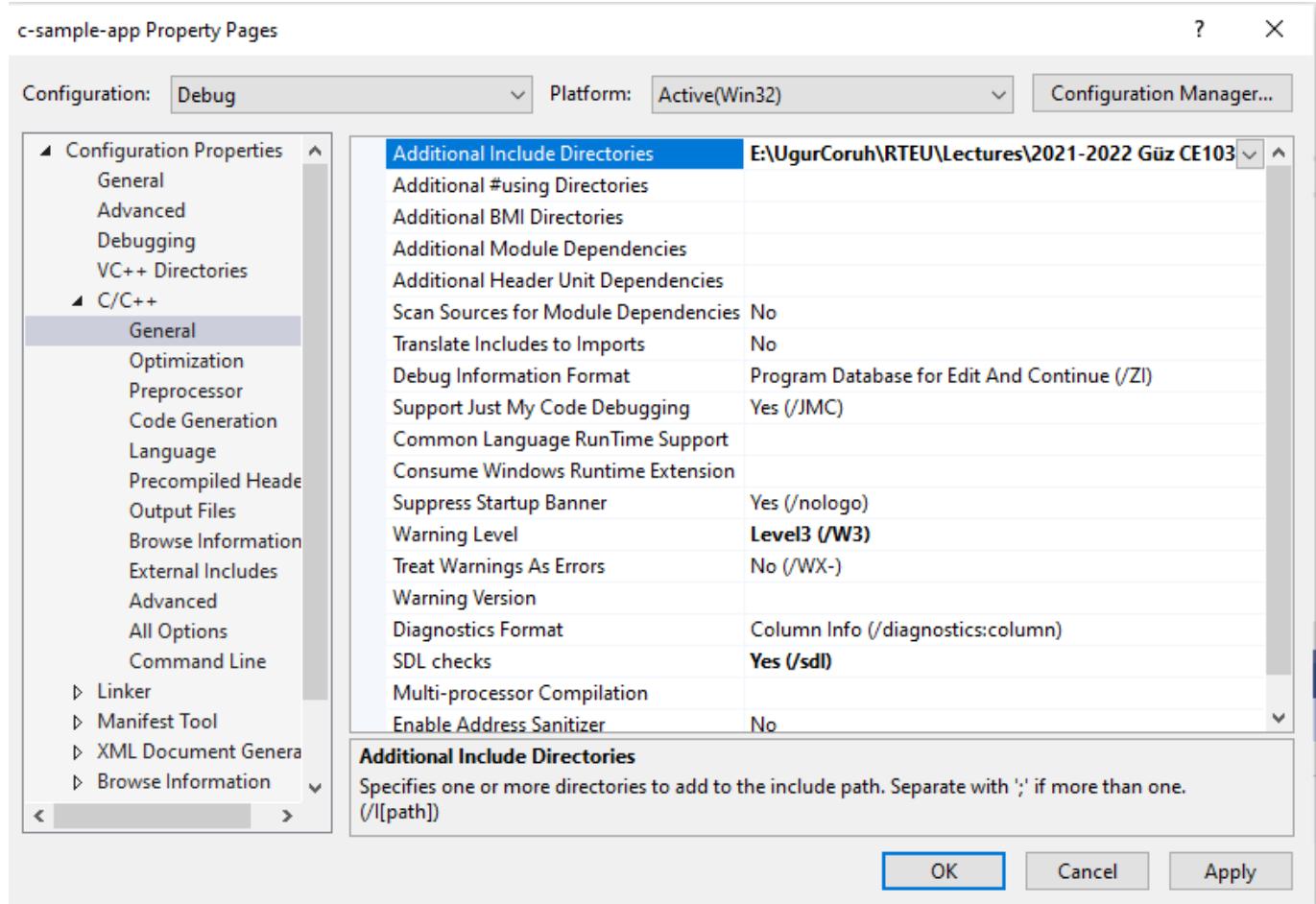
Configuration Properties	Additional Include Directories	..\c-sample-lib\DebugStaticLibDeployment;%Additional Include Directories
General	Additional #using Directories	
Advanced	Additional BMI Directories	
Debugging	Additional Module Dependencies	
VC++ Directories	Additional Header Unit Dependencies	
C/C++	Scan Sources for Module Dependencies	No
General	Translate Includes to Imports	No
Optimization	Debug Information Format	Program Database for Edit And Continue (/ZI)
Preprocessor	Support Just My Code Debugging	Yes (/JMC)
Code Generation	Common Language RunTime Support	
Language	Consume Windows Runtime Extension	
Precompiled Headers	Suppress Startup Banner	Yes (/nologo)
Output Files	Warning Level	Level3 (/W3)
Browse Information	Treat Warnings As Errors	No (/WX-)
External Includes	Warning Version	
Advanced	Diagnostics Format	Column Info (/diagnostics:column)
All Options	SDL checks	Yes (/sdl)
Command Line	Multi-processor Compilation	
Linker	Enable Address Sanitizer	No
Manifest Tool		
XML Document Generator		
Browse Information		

Additional Include Directories
Specifies one or more directories to add to the include path. Separate with ';' if more than one.
(/I[path])

OK Cancel Apply

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
```

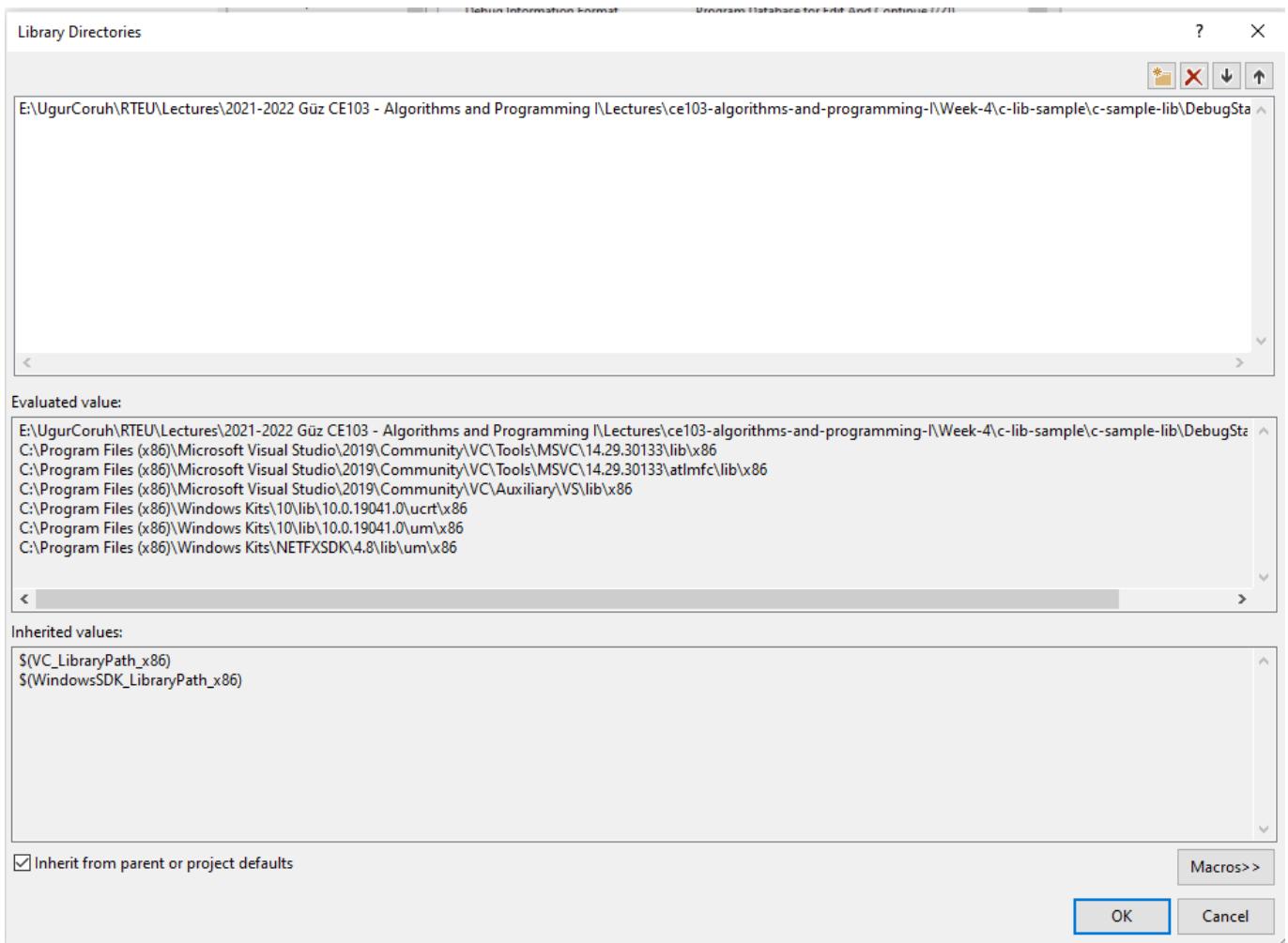
Configuration: Debug Platform: Active(Win32) Configuration Manager...

▲ Configuration Properties General Executable Directories \$(VC_ExecuteablePath_x86);\$(CommonExecuteablePath)
Advanced Include Directories \$(IncludePath)
Debugging External Include Directories \$(ExternalIncludePath)
VC++ Directories Reference Directories \$(VC_ReferencesPath_x86);
▼ C/C++ Library Directories E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - A
General Library WinRT Directories \$(WindowsSDK_MetadataPath);
Optimization Source Directories \$(VC_SourcePath);
Preprocessor Exclude Directories \$(CommonExcludePath);\$(VC_ExecuteablePath_x86);\$(VC_Libr
Code Generation
Language
Precompiled Headers
Output Files
Browse Information
External Includes
Advanced
All Options
Command Line
▷ Linker
▷ Manifest Tool
▷ XML Document Generation
▷ Browse Information

▼ Public Project Content
Public Include Directories No
All Header Files are Public
Public C++ Module Directories No
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

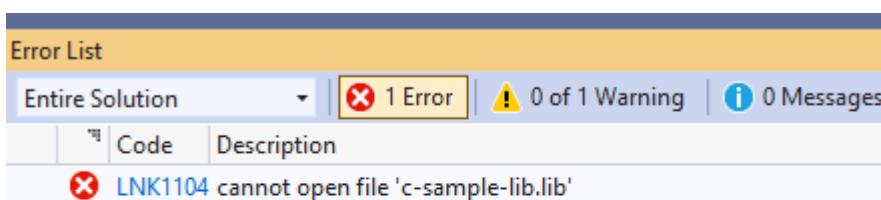
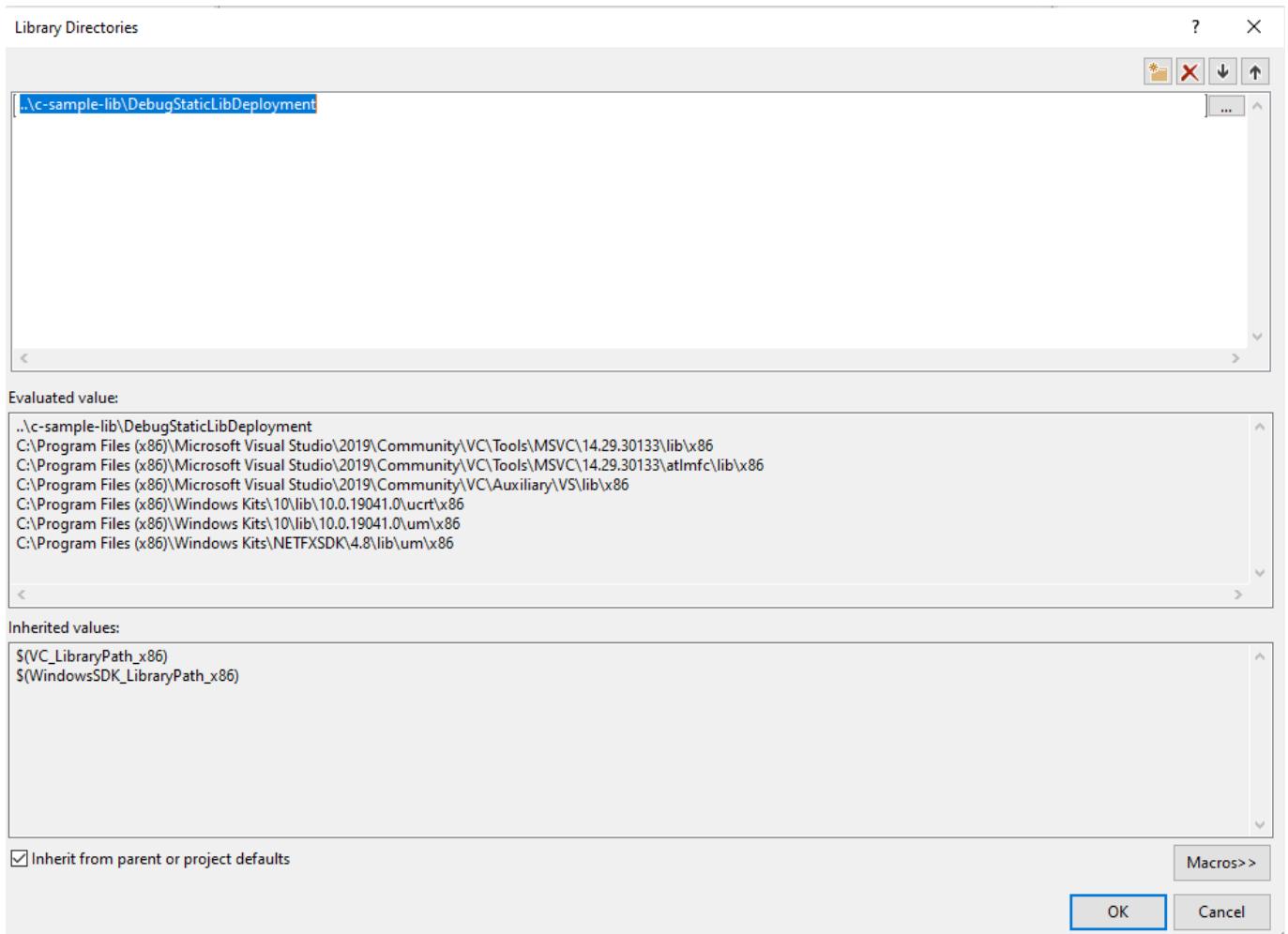
c-sample-app Property Pages

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

▲ Configuration Properties	▼ General
General	Executable Directories \$(VC_ExecutePath_x86);\$(CommonExecutablePath)
Advanced	\$(IncludePath)
Debugging	\$(ExternalIncludePath)
VC++ Directories	\$(VC_ReferencesPath_x86);
C/C++	..\\c-sample-lib\\DebugStaticLibDeployment;\$(LibraryPath)
Linker	\$(WindowsSDK_MetadataPath);
Manifest Tool	
XML Document Generator	\$(VC_SourcePath);
Browse Information	
Build Events	\$(CommonExcludePath);\$(VC_ExecutePath_x86);\$(VC_Libr
Custom Build Step	
Code Analysis	
▼ Public Project Content	
Public Include Directories	No
All Header Files are Public	
Public C++ Module Directories	
All Modules are Public	No

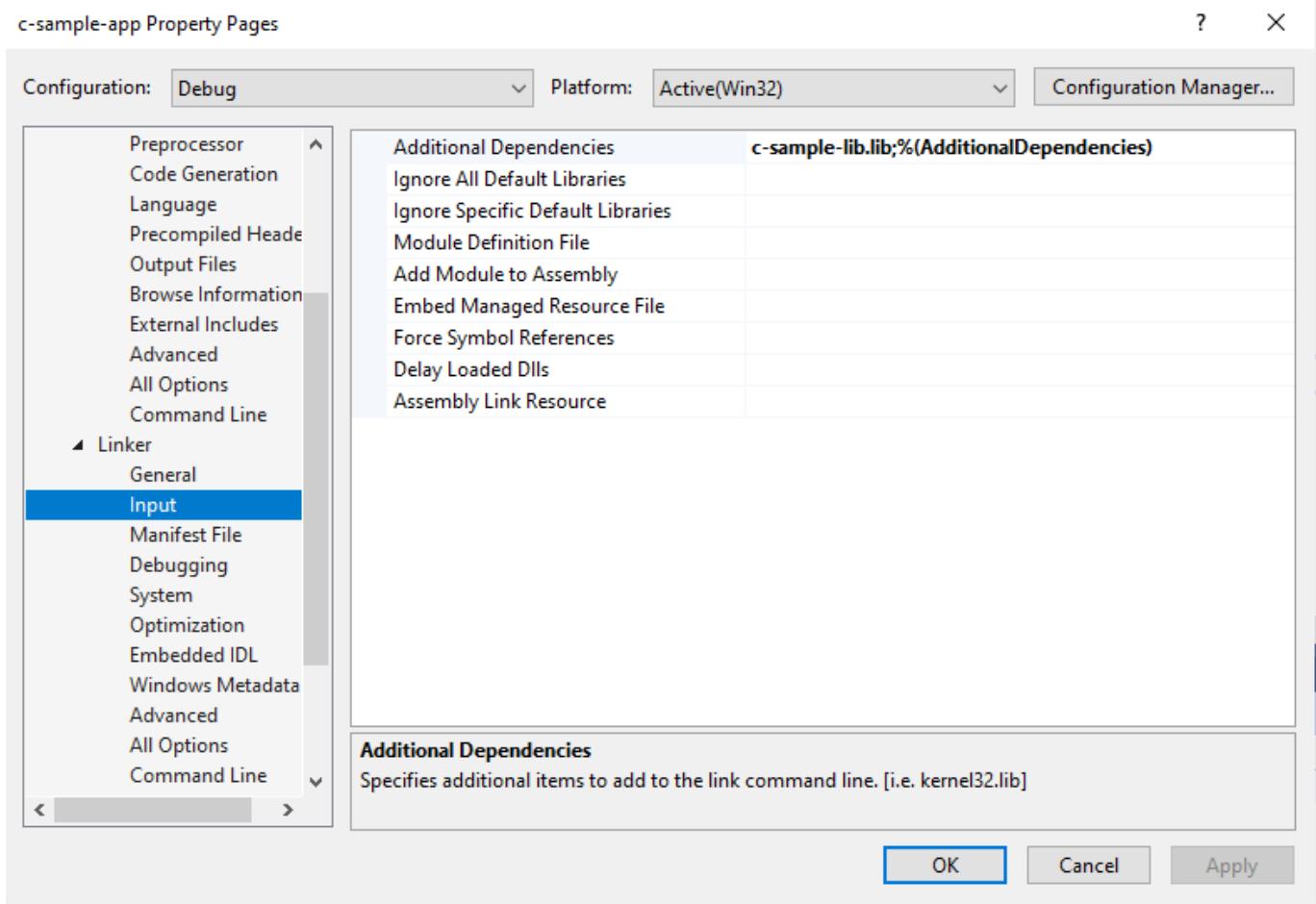
Executable Directories
Path to use when searching for executable files while building a VC++ project. Corresponds to environment variable PATH.

OK Cancel Apply



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

Linker->Input->Additional Dependencies



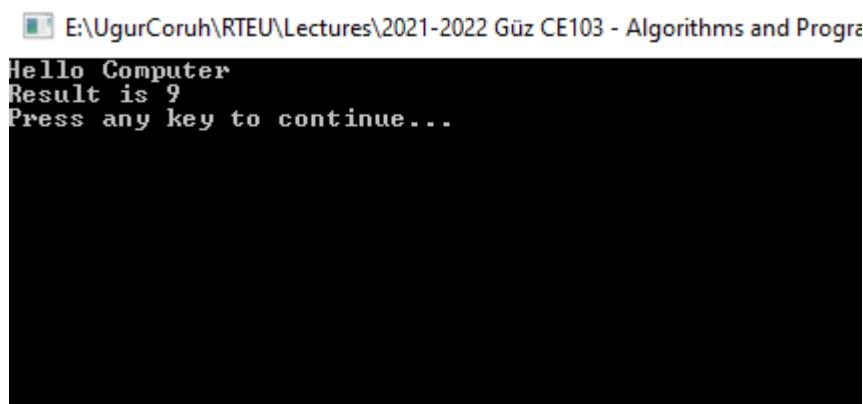
In this case we will compile c-sample-app and we do not need to compile c-sample-lib because we copied output files to different location and they are ready to use.

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 Programing\CE103\HelloWorld> 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

c-sample-app Property Pages

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

Configuration Properties	Output File	<code>\$(OutDir)\$(TargetName)\$(TargetExt)</code>
General	Show Progress	Not Set
Advanced	Version	
Debugging	Enable Incremental Linking	Yes (/INCREMENTAL)
VC++ Directories	Incremental Link Database File	<code>\$(IntDir)\$(TargetName).ilk</code>
C/C++	Suppress Startup Banner	Yes (/NOLOGO)
Linker	Ignore Import Library	No
General	Register Output	No
Input	Per-user Redirection	No
Manifest File	Additional Library Directories	<code>%(AdditionalLibraryDirectories)</code>
Debugging	Link Library Dependencies	Yes
System	Use Library Dependency Inputs	No
Optimization	Link Status	
Embedded IDL	Prevent DLL Binding	
Windows Metadata	Treat Linker Warning As Errors	
Advanced	Force File Output	
All Options	Create Hot Patchable Image	
Command Line	Specify Section Attributes	

Additional Library Directories
Allows the user to override the environmental library path. (`/LIBPATH:folder`)

OK Cancel Apply

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

c-sample-app Property Pages

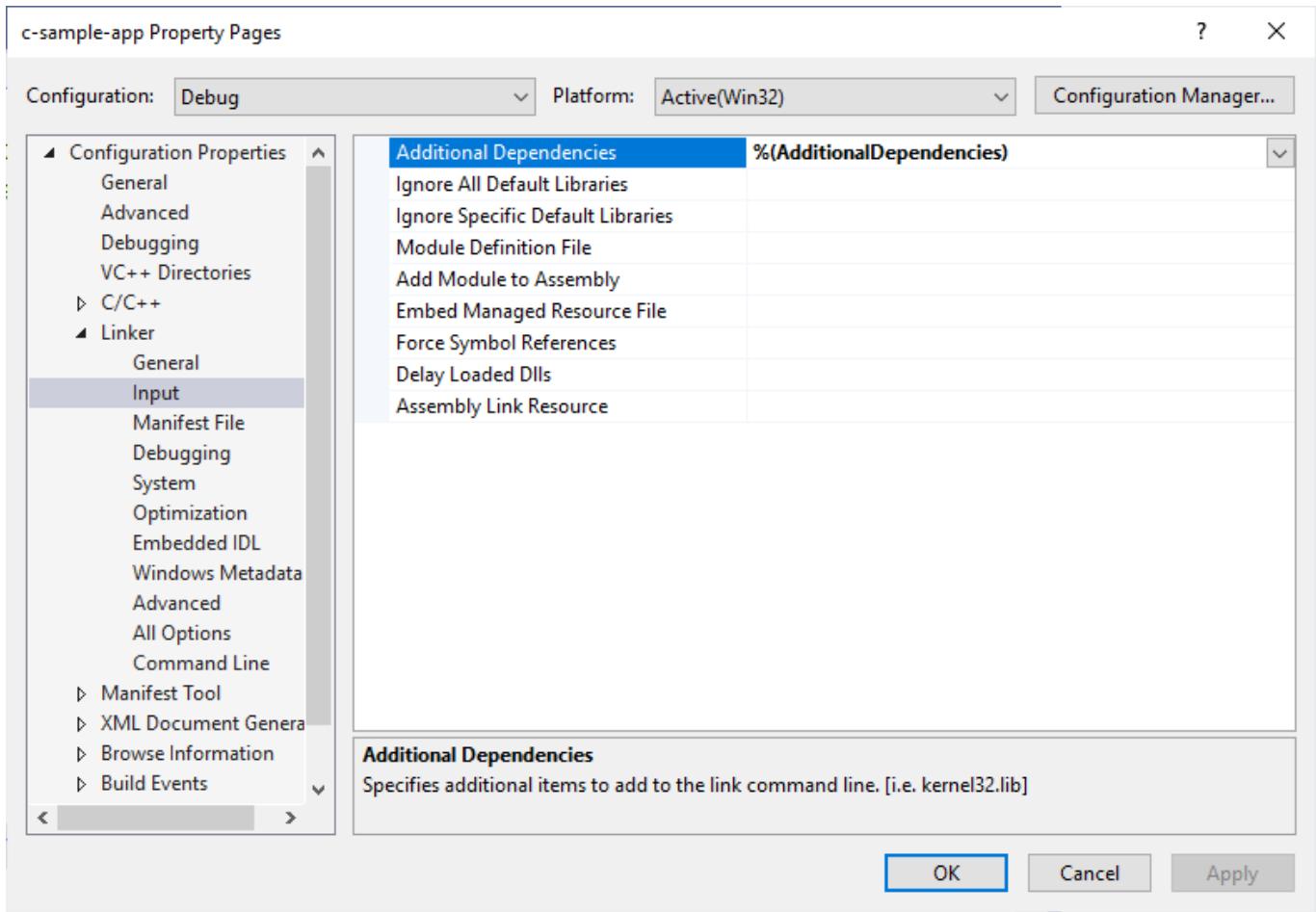
Configuration: Debug Platform: Active(Win32) Configuration Manager...

	Additional Include Directories	%{AdditionalIncludeDirectories}
Additional #using Directories		
Additional BMI Directories		
Additional Module Dependencies		
Additional Header Unit Dependencies		
Scan Sources for Module Dependencies	No	
Translate Includes to Imports	No	
Debug Information Format	Program Database for Edit And Continue (/ZI)	
Support Just My Code Debugging	Yes (/JMC)	
Common Language RunTime Support		
Consume Windows Runtime Extension		
Suppress Startup Banner	Yes (/nologo)	
Warning Level	Level3 (/W3)	
Treat Warnings As Errors	No (/WX-)	
Warning Version		
Diagnostics Format	Column Info (/diagnostics:column)	
SDL checks	Yes (/sdl)	
Multi-processor Compilation		
Enable Address Sanitizer	No	

Additional Include Directories
Specifies one or more directories to add to the include path. Separate with ';' if more than one.
(/I[path])

OK Cancel Apply

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.

C++ Programming (Static Library)

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;
}
```

C/C++ WSL Option

Install WSL

[GitHub - ucoruh/ns3-wsl-win10-setup: ns3 windows 10 WSL2 setup and usage](#)

Create a Linux project

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++

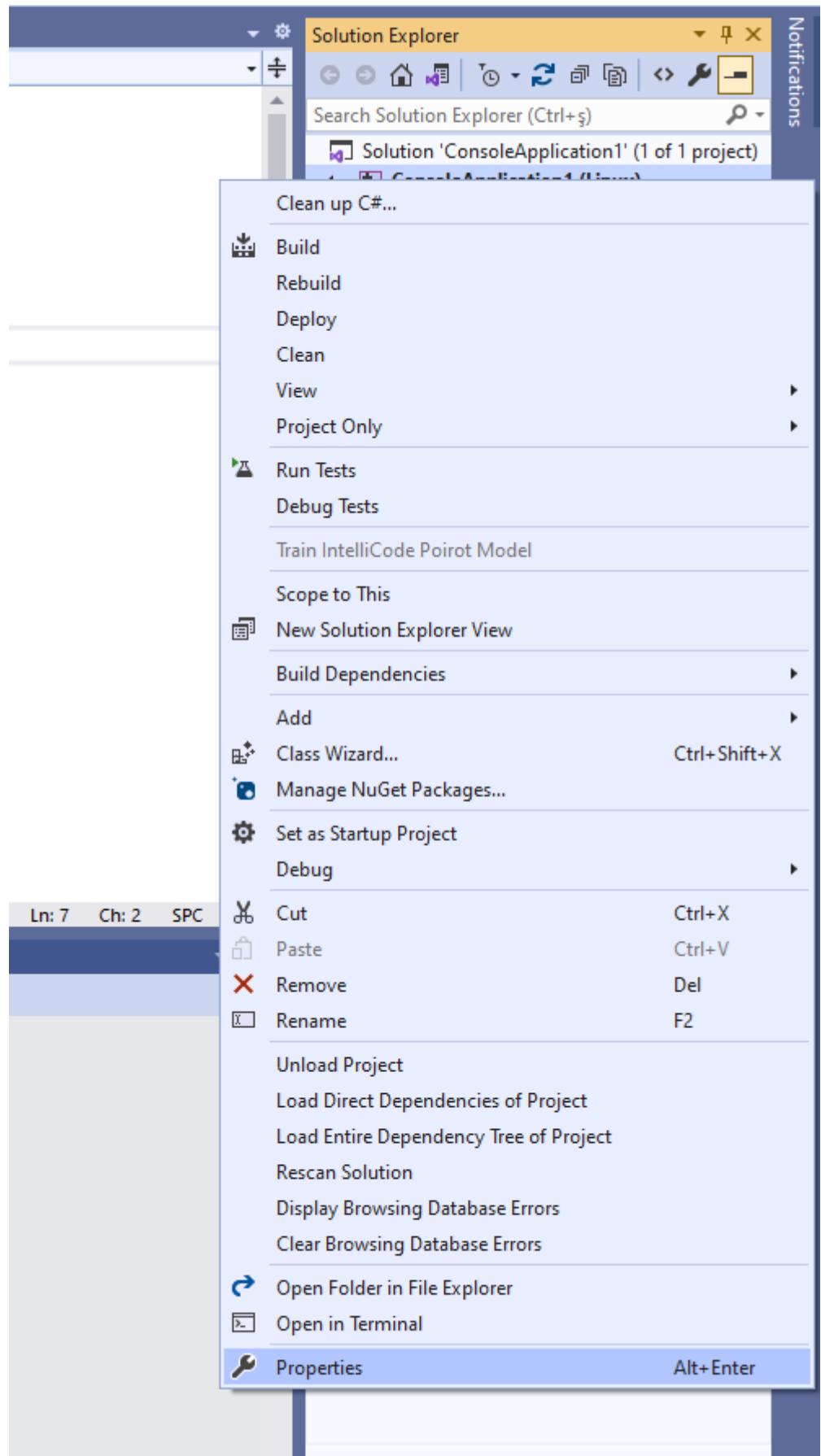
Linux

Console

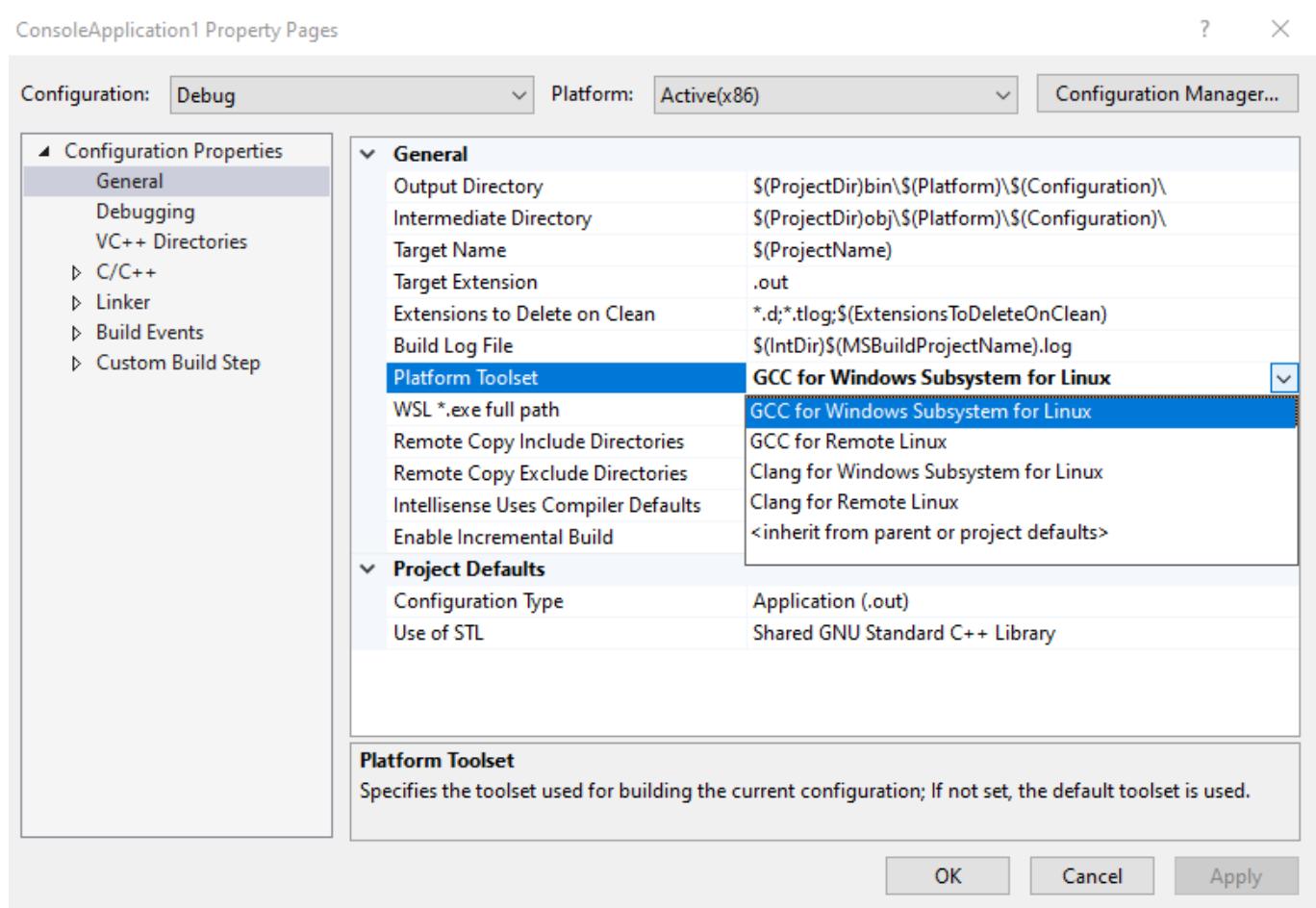


Empty Project

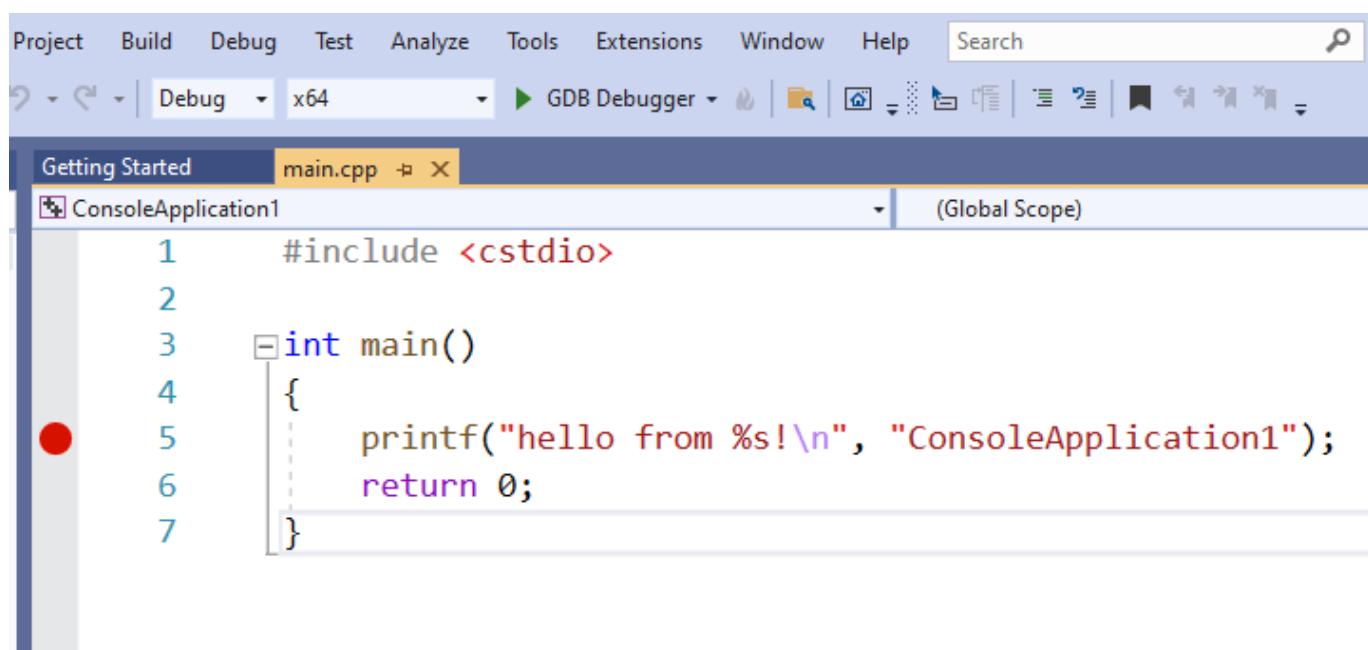
Configure Platform Toolset to WSL



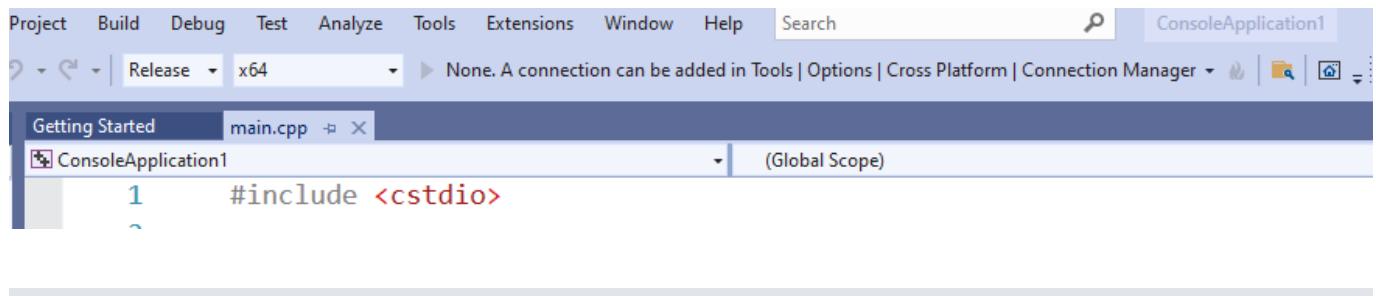
Select GCC for Windows Subsystem for Linux



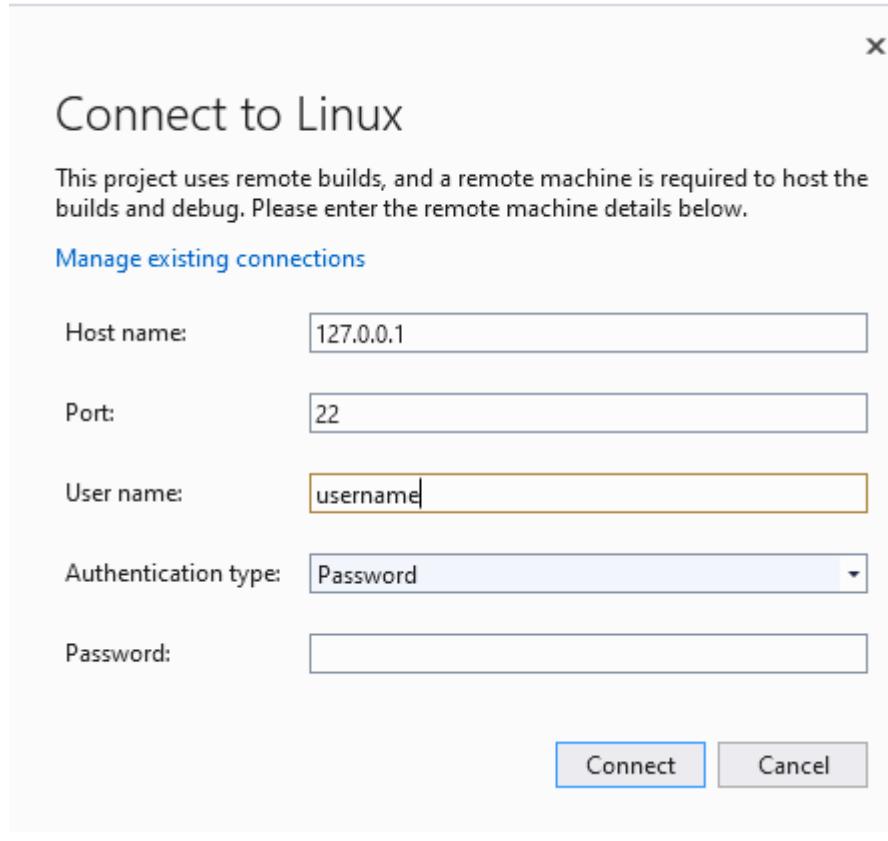
Put a breakpoint and run debugger



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.



Configure SSH parameters



so you have to complete the following steps.

C/C++ Remote Linux Option over SSH

Enable SSH

[SSH on Windows Subsystem for Linux \(WSL\) | Illuminia Studios](#)

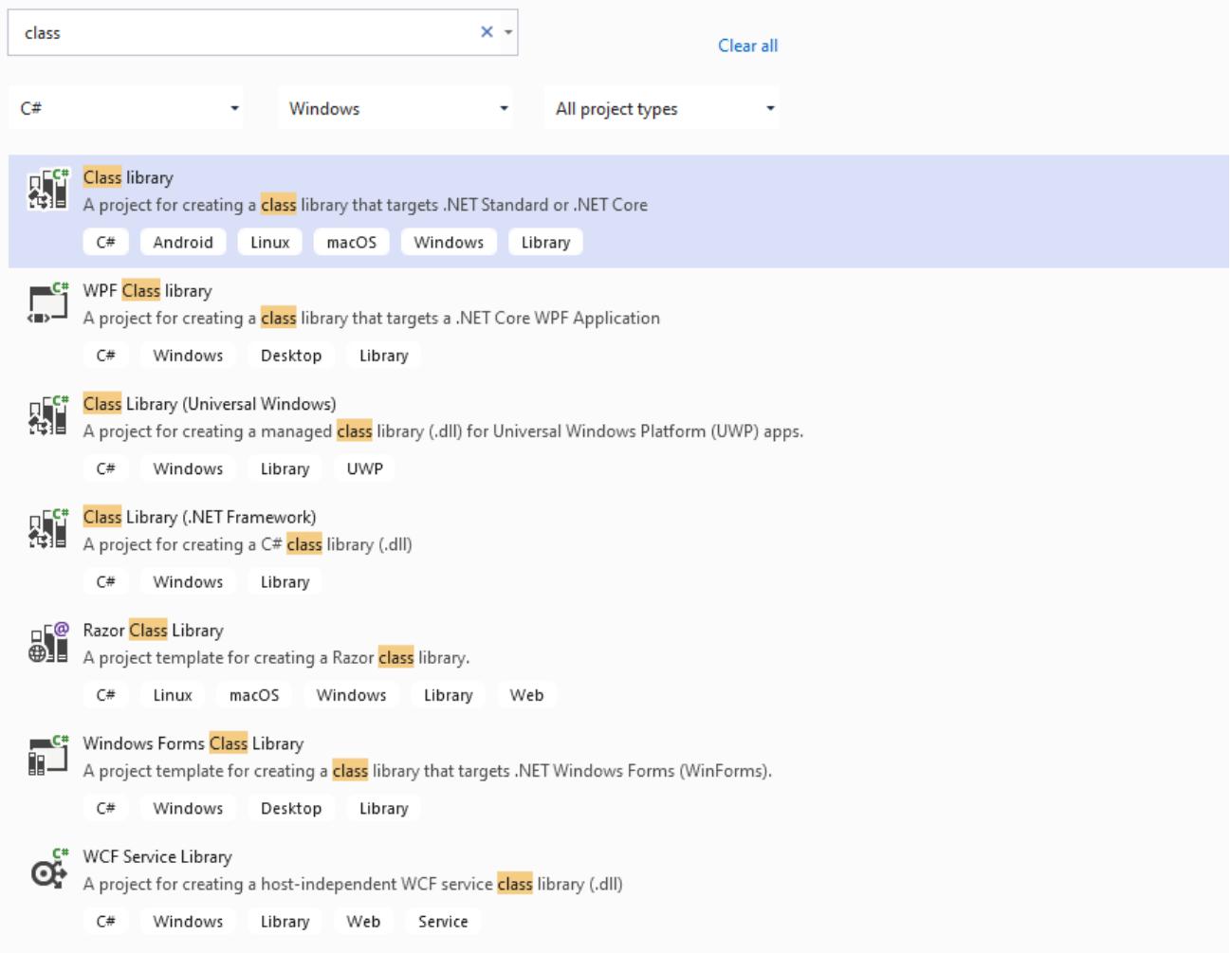
Connect to Remote WSL Environment

C# Programming (Dinamik Library)

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

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 I\Lectures\ce11

...

Solution name i

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 i

.NET Core 3.1 (Long-term support)

.NET Standard 2.0

.NET Standard 2.1

.NET Core 2.1 (Long-term support)

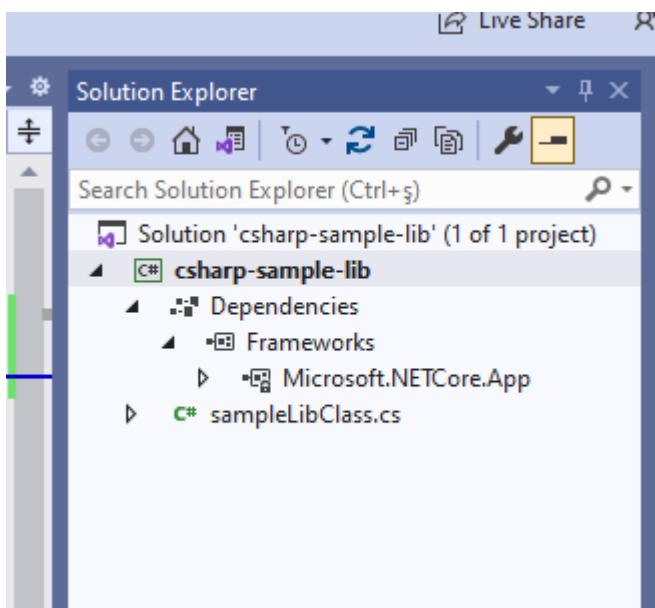
.NET Core 3.1 (Long-term support)

.NET 5.0 (Current)

You will have default empty class library file

```
sampleLibClass.cs  X  csharp-sample-lib  csharp_sample_l
1     using System;
2
3     namespace csharp_sample_lib
4     {
5         public class sampleLibClass
6         {
7             }
8         }
9     }
10
```

In the project you can see .NETcore reference



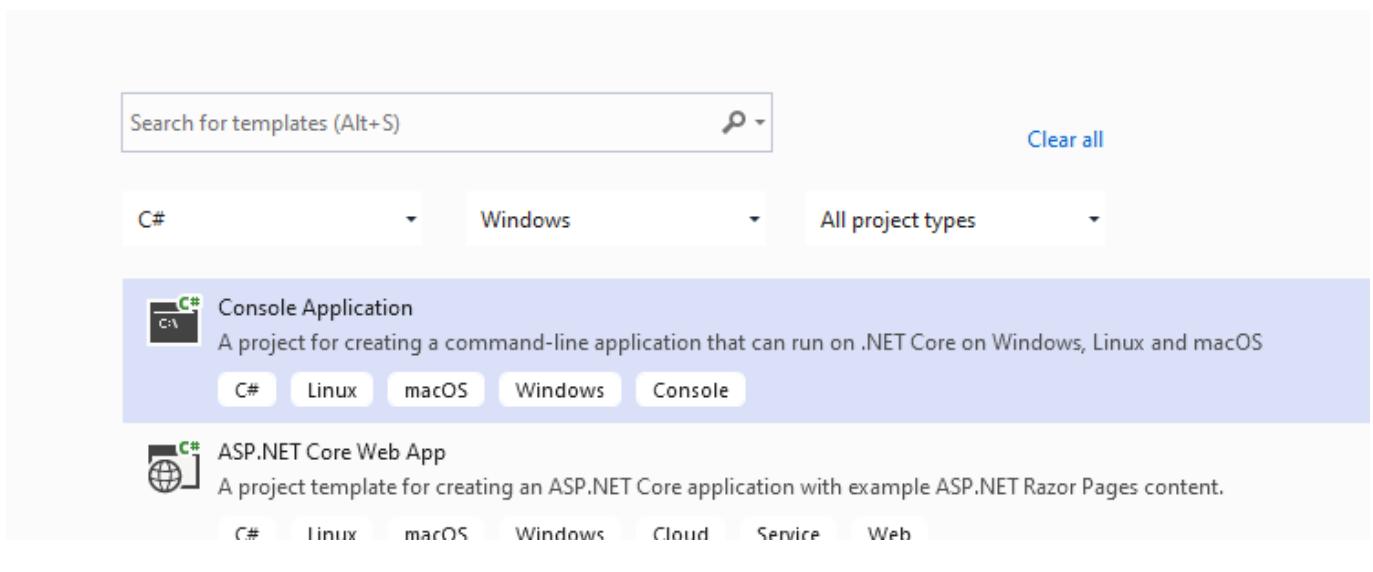
We can build empty class library that generate dll for our application

```
Ln: 7 Ch
```

```
.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

Location



Select .NETCore framework

Additional information

Console Application C# Linux macOS Windows Console

Target Framework 

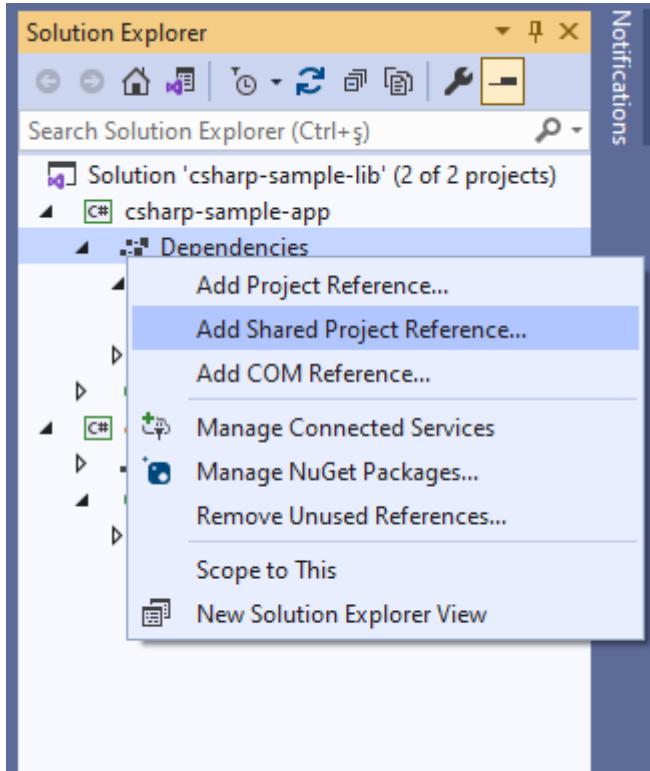
- .NET Core 3.1 (Long-term support)
- .NET Core 2.1 (Long-term support)
- .NET Core 3.1 (Long-term support)
- .NET 5.0 (Current)

You will have the following sample main.cs file

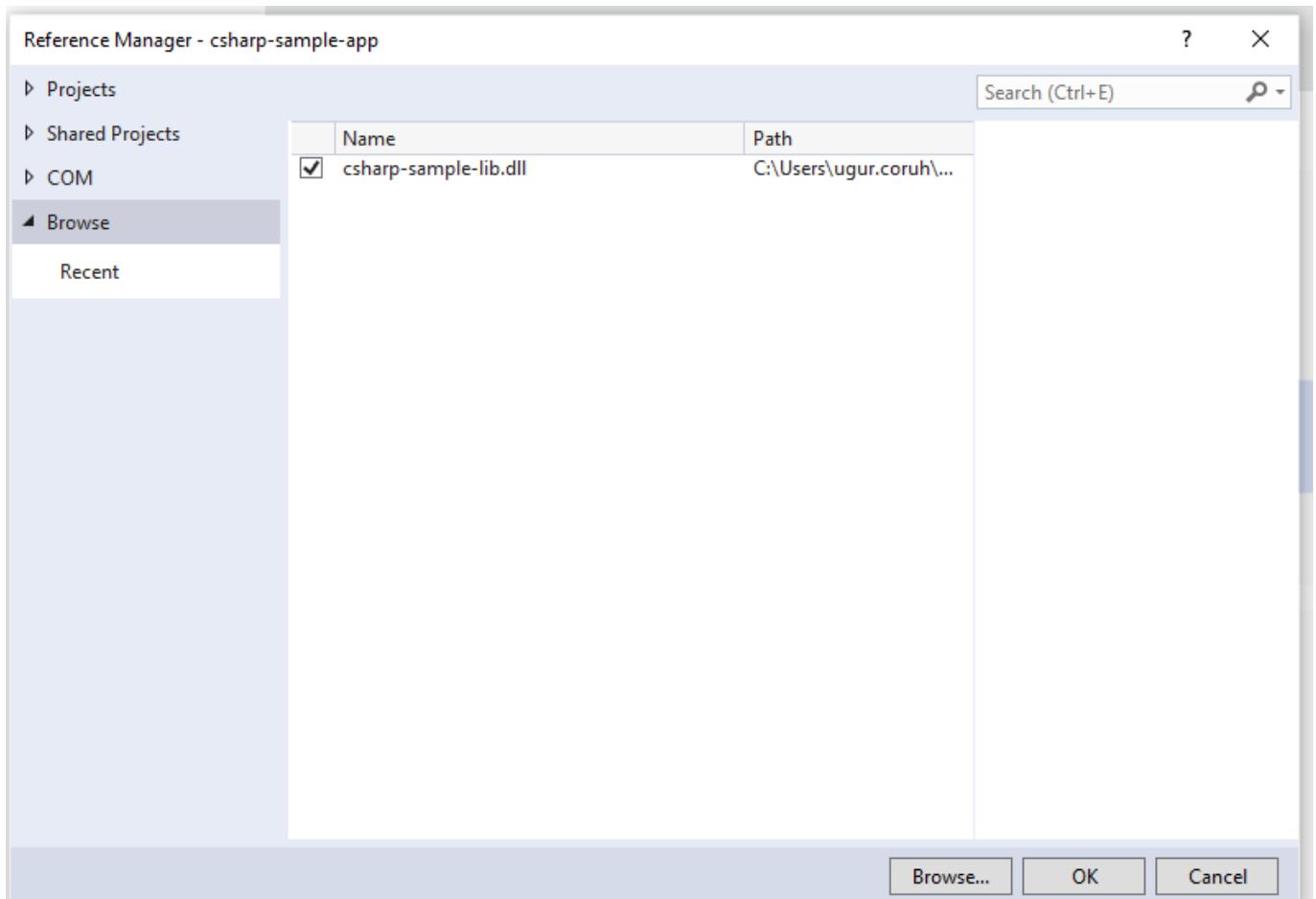
```
using System;

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;
        }
    }
}
```

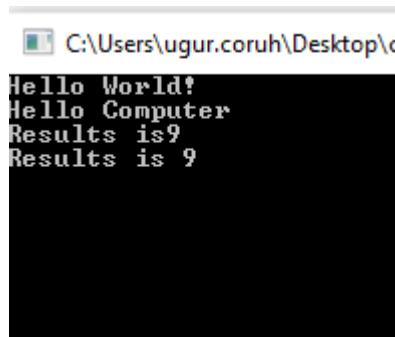
after this operation copy following sample to console application and build app then you can run

```
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



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 – TecAdmin](#)

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
```

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:

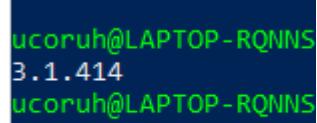
```
sudo apt install dotnet-runtime-2.1
```

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
```

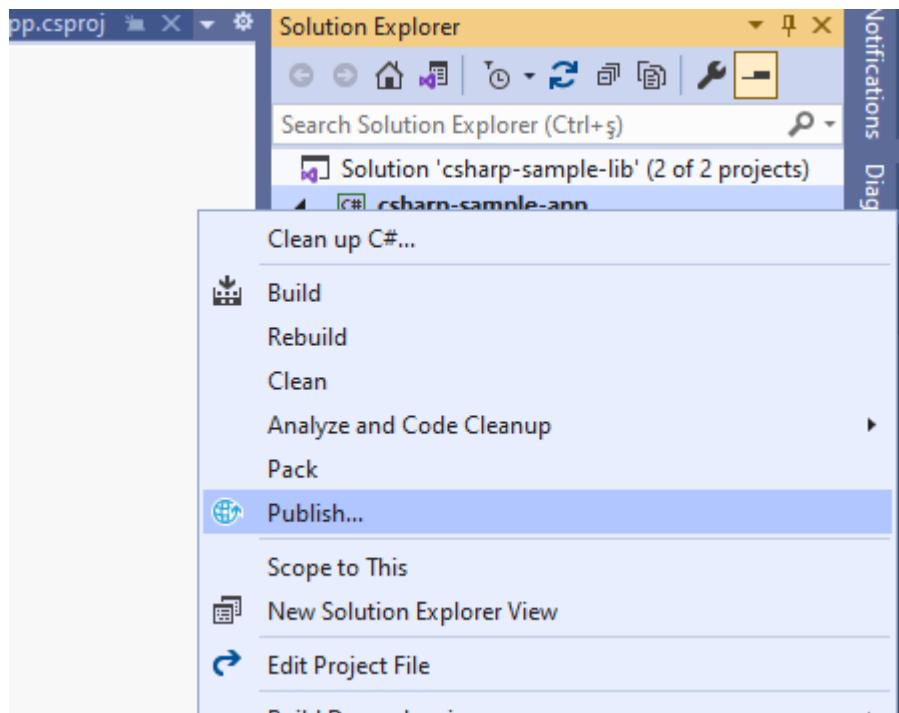


A terminal window showing the command 'dotnet --version' being run. The output shows the user 'ucoruh' at 'LAPTOP-RQNNS' with the version '3.1.414' displayed.

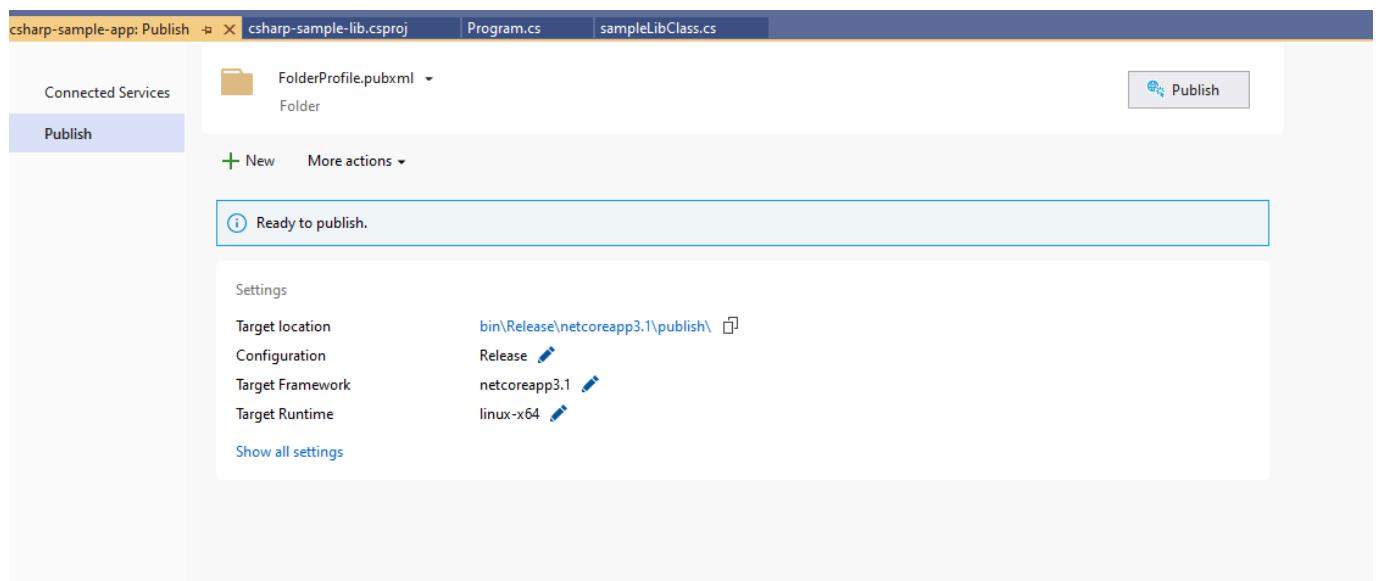
```
ucoruh@LAPTOP-RQNNS: ~  
3.1.414  
ucoruh@LAPTOP-RQNNS: ~
```

Now we will publish our application as single executable

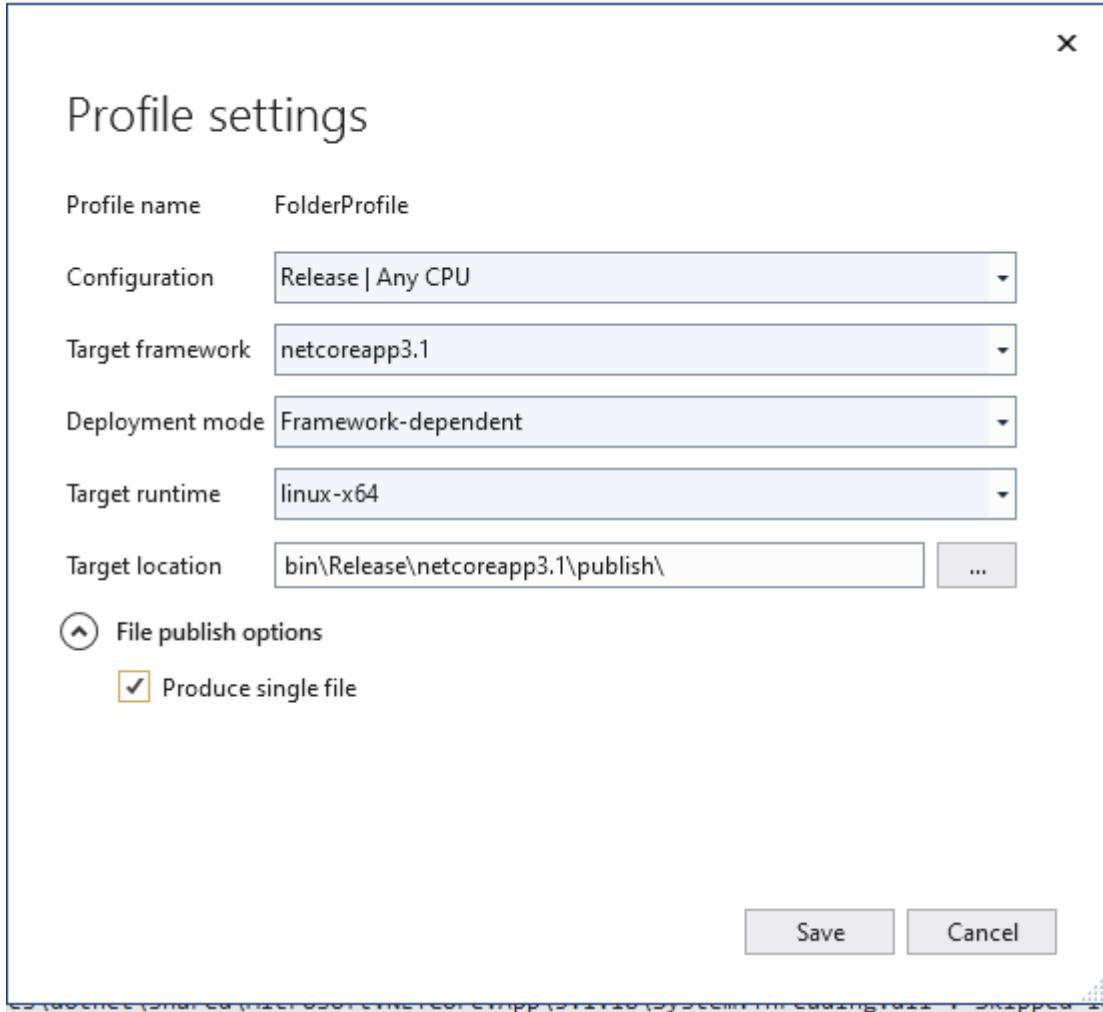
Open publish menu



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

esktop > 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.coruh/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.coruh/Desktop/csharp-lib-sample/csharp-sample-lib/csharp-sample-app/bin/Release/netcoreapp3.1/publish$ ./csharp-sample-app
csharp-sample-app      csharp-sample-app.pdb      csharp-sample-lib.pdb      packages-microsoft-prod.deb
ucoruh@LAPTOP-RQNN9IG:/mnt/c/Users/ugur.coruh/Desktop/csharp-lib-sample/csharp-sample-lib/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 is9
Results is 9
```

Java Programming

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 Packages](#)



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

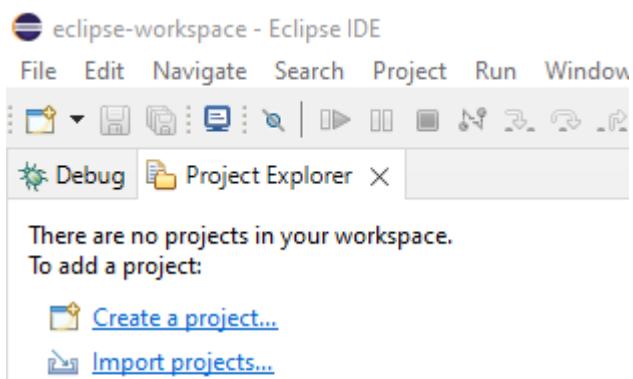
**Installation Folder**

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

New Project

Select a wizard

Create a Java project



Wizards:

type filter text

- ▼ General
 - 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



< Back

Next >

Finish

Cancel

give project name

New Java Project



Create a Java Project



Create a Java project in the workspace or in an external location.

Project name:

Use default location

Location: [Browse...](#)

JRE

Use an execution environment JRE:

Use a project specific JRE:

Use default JRE 'jdk-16.0.1' and workspace compiler preferences

[Configure JREs...](#)

Project layout

Use project folder as root for sources and class files

Create separate folders for sources and class files

[Configure default...](#)

Working sets

Add project to working sets

[New...](#)

Working sets:



[Select...](#)

Module

Create module-info.java file



< Back

Next >

Finish

Cancel

select finish

New Java Project

Java Settings

Define the Java build settings.



Source Projects Libraries Order and Export Module Dependencies

java-sample-lib
src



Details

[Create new source folder](#): use this if you want to add a new source folder to your project.

[Link additional source](#): use this if you have a folder in the file system that should be used as additional source folder.

[Configure inclusion and exclusion filters](#): specify patterns to the inclusion and exclusion filters instead of including and excluding each folder or file manually.

[Remove source folder 'src' from build path](#): Children of the folder will not be seen by the compiler anymore and will not be included when building the project.

Allow output folders for source folders

Default output folder:

java-sample-lib/bin

[Browse...](#)



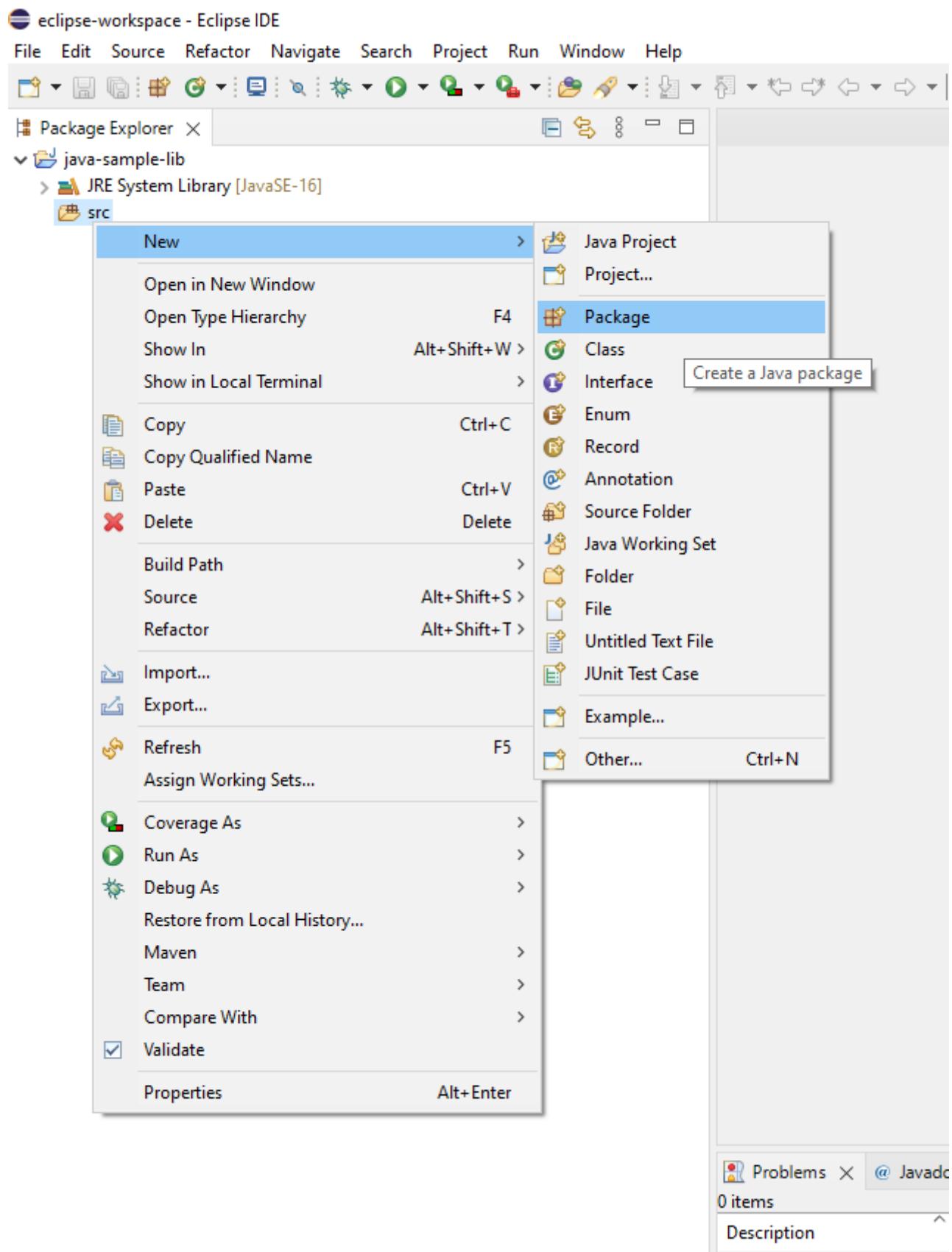
< Back

Next >

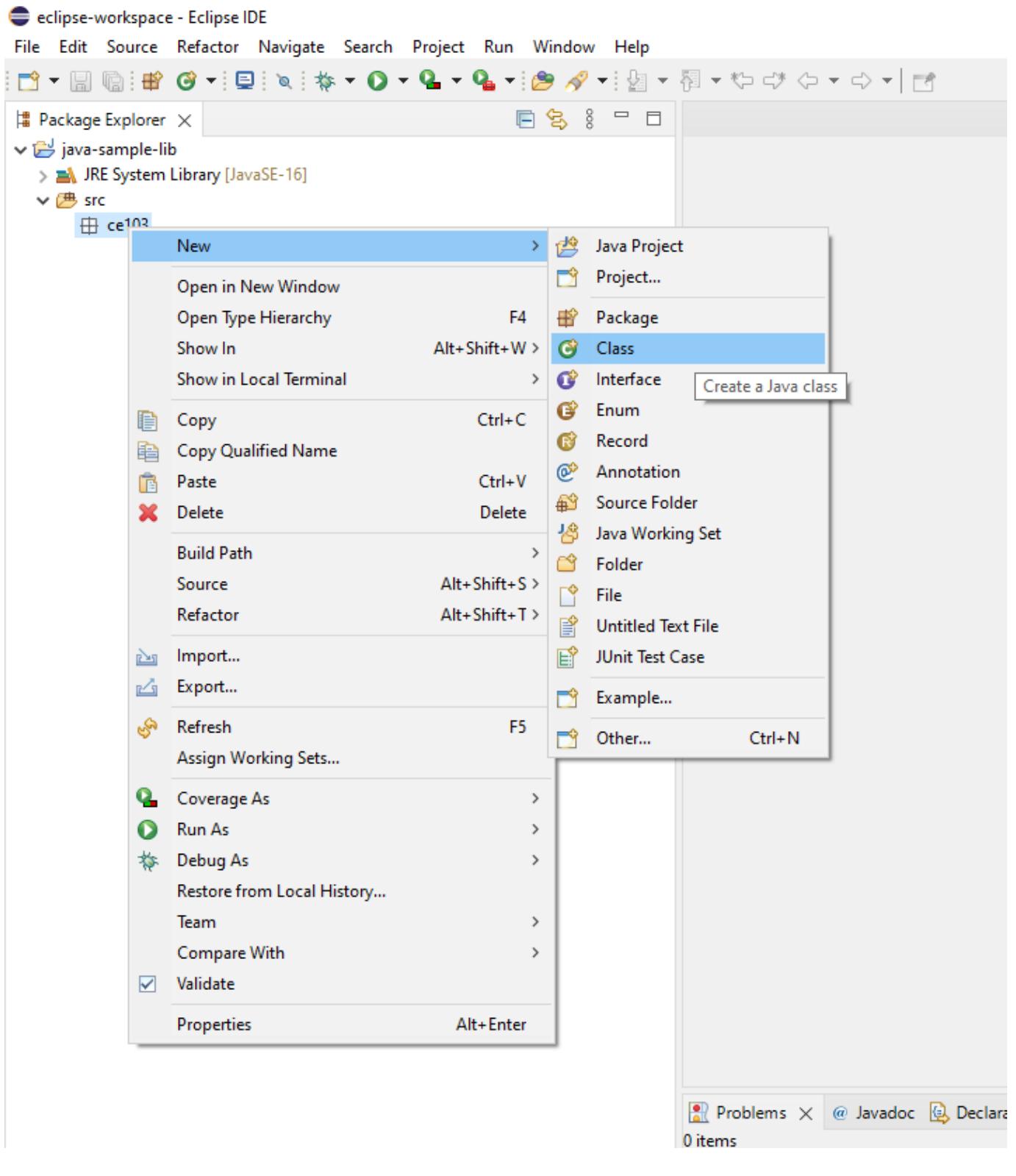
[Finish](#)

[Cancel](#)

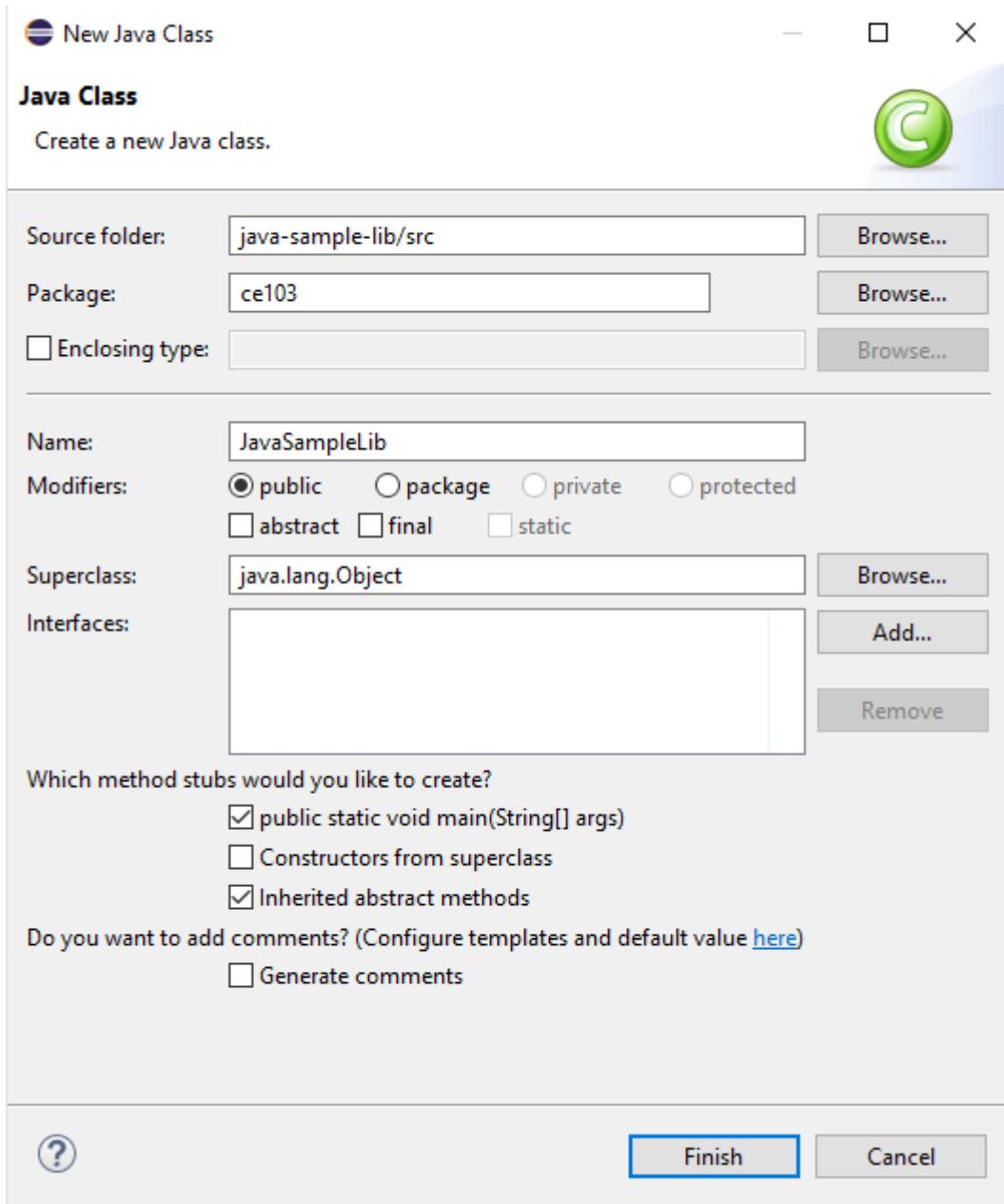
first we need to add a default package to keep everything organized



then we can create our class that includes our functions



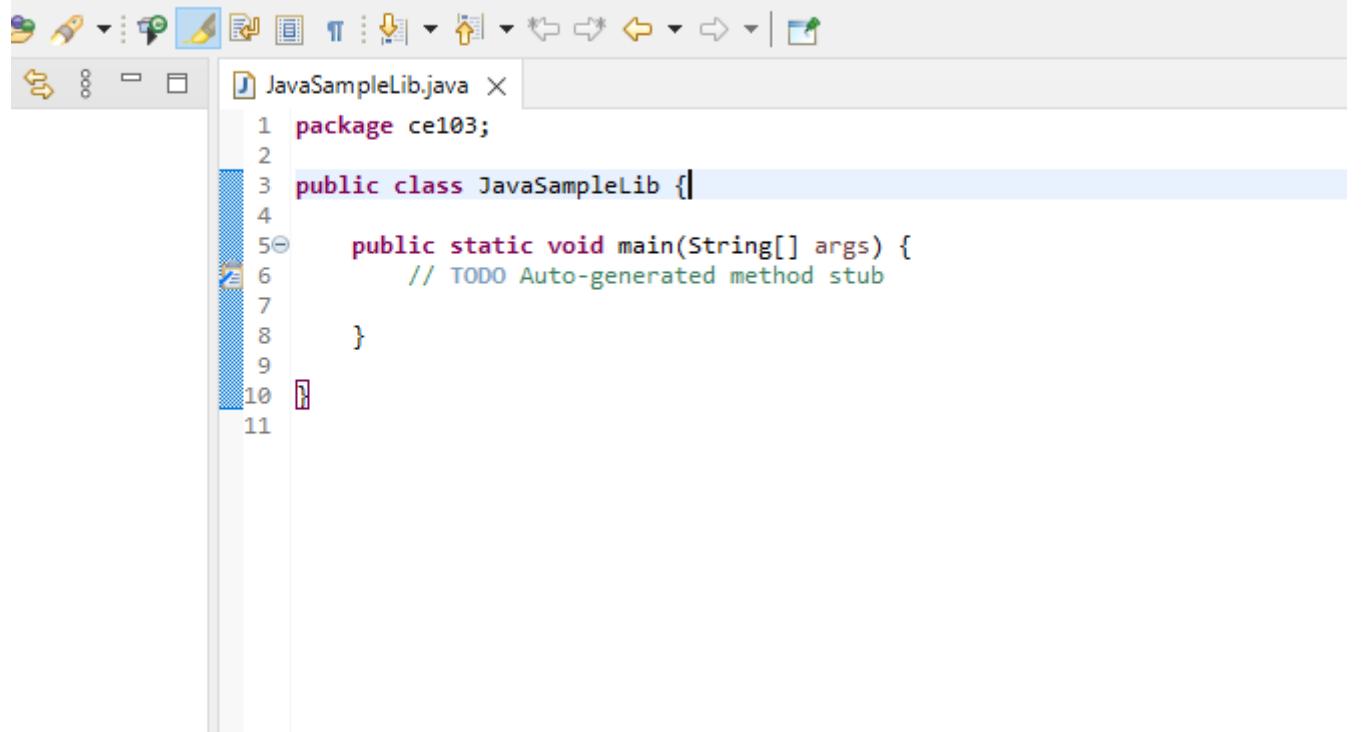
give class a name



you will have following class with main

Eclipse IDE

Window Help



The screenshot shows the Eclipse IDE interface with the JavaSampleLib.java file open in the editor. The code is as follows:

```
1 package ce103;
2
3 public class JavaSampleLib {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10}
11
```

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;
    }

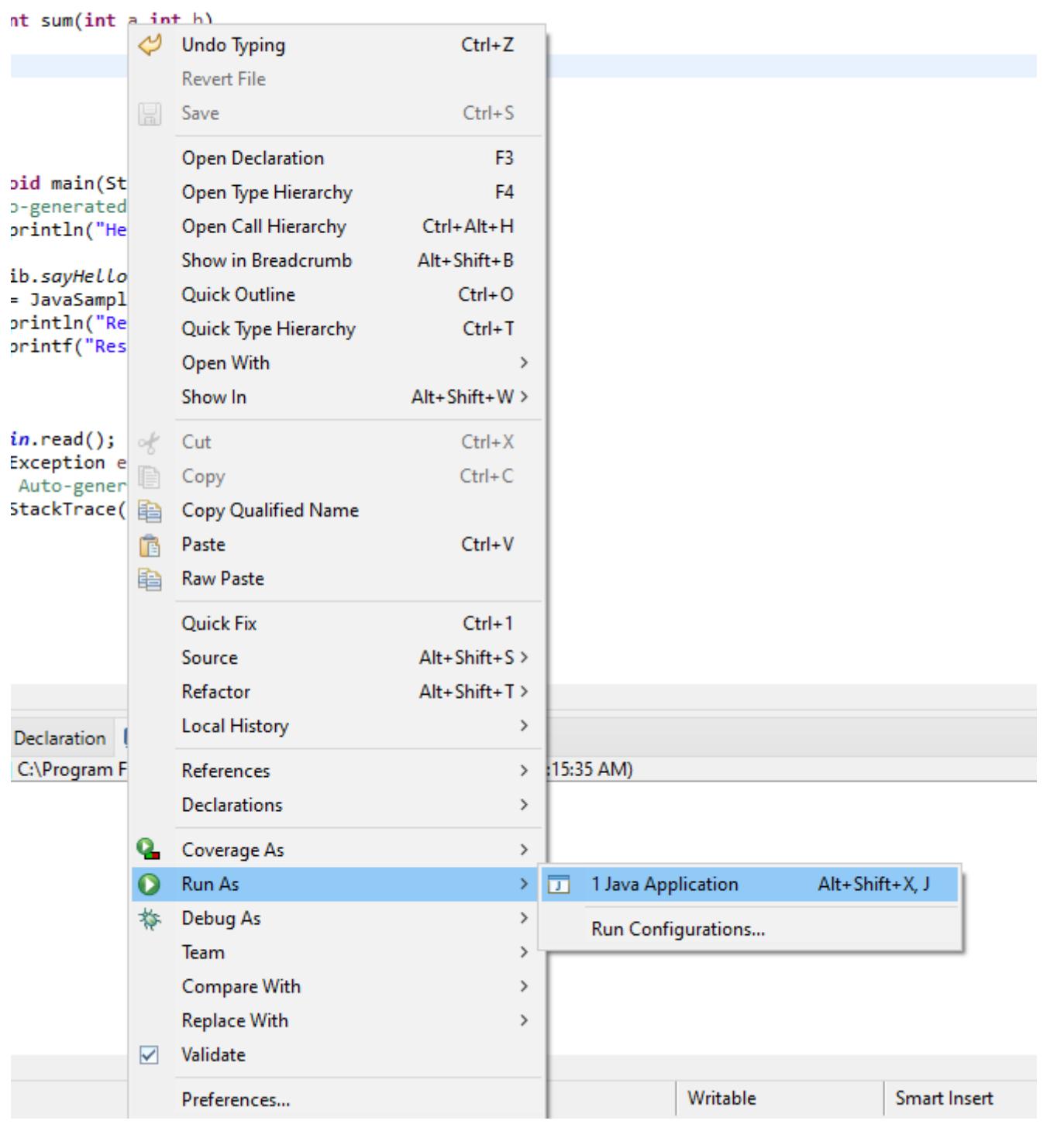
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Hello World!");

        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



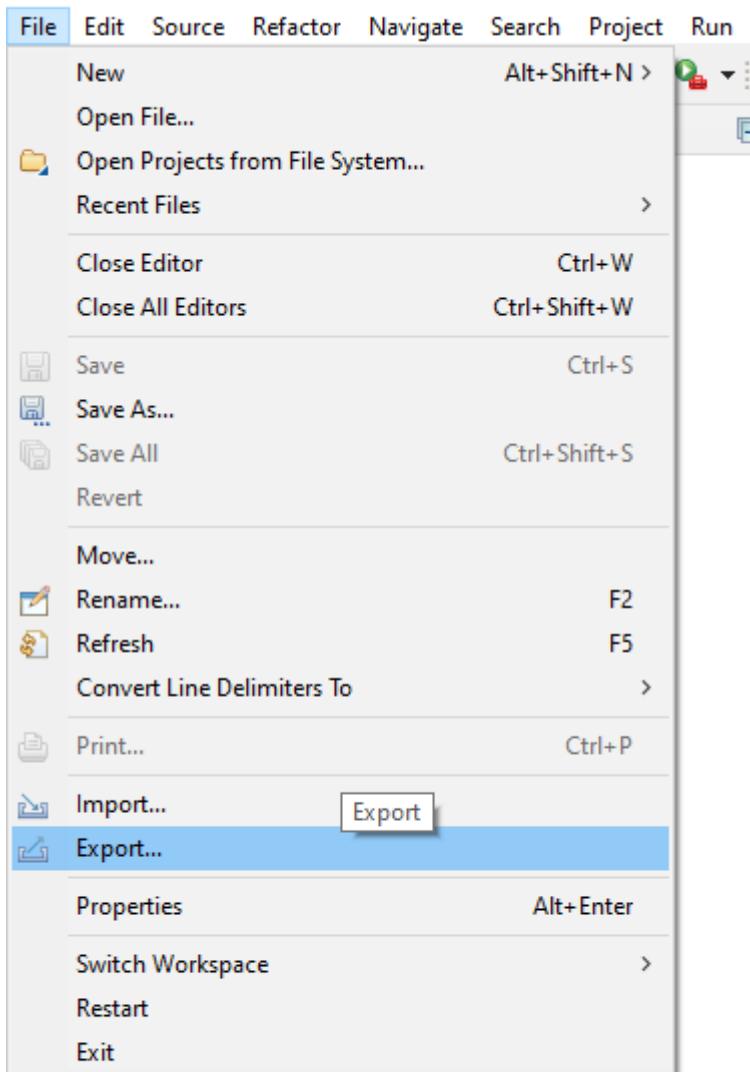
we can see output from console as below

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

- Top Bar:** Eclipse-workspace - java-sample-lib/src/ce103/JavaSampleLib.java - Eclipse IDE. Includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Standard Eclipse toolbar icons.
- Package Explorer:** Shows the project structure: java-sample-lib (selected), JRE System Library [JavaSE-16], src, ce103, JavaSampleLib.java.
- Editor:** Displays the Java code for JavaSampleLib.java. The code includes methods for printing "Hello World!", summing integers, and handling input from System.in. The main method calls these functions and prints their results.
- Console:** Shows the output of the program's 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.



Select Java->Runnable JAR File

 Export



Select

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



Select an export wizard:

-  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



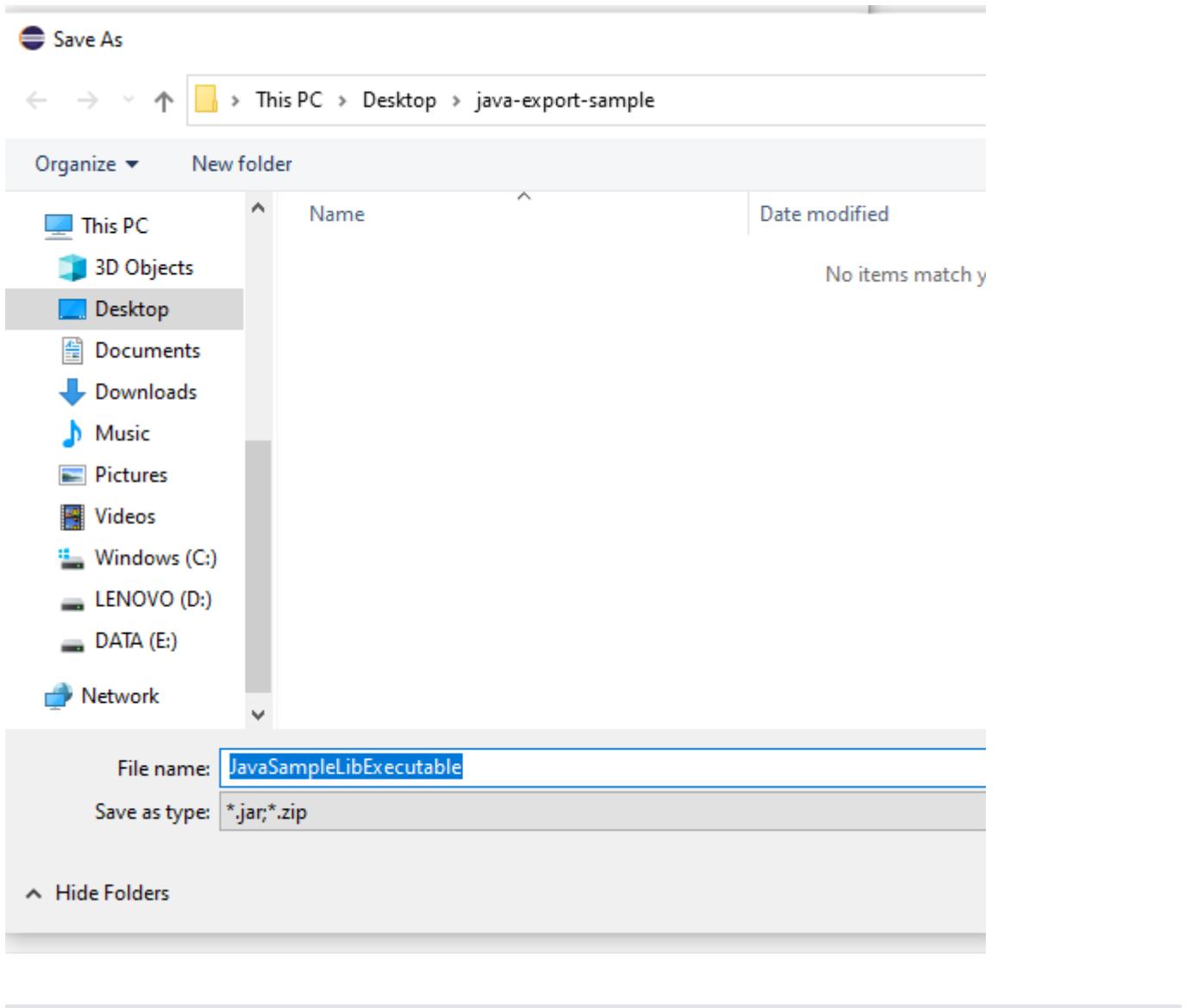
< Back

Next >

Finish

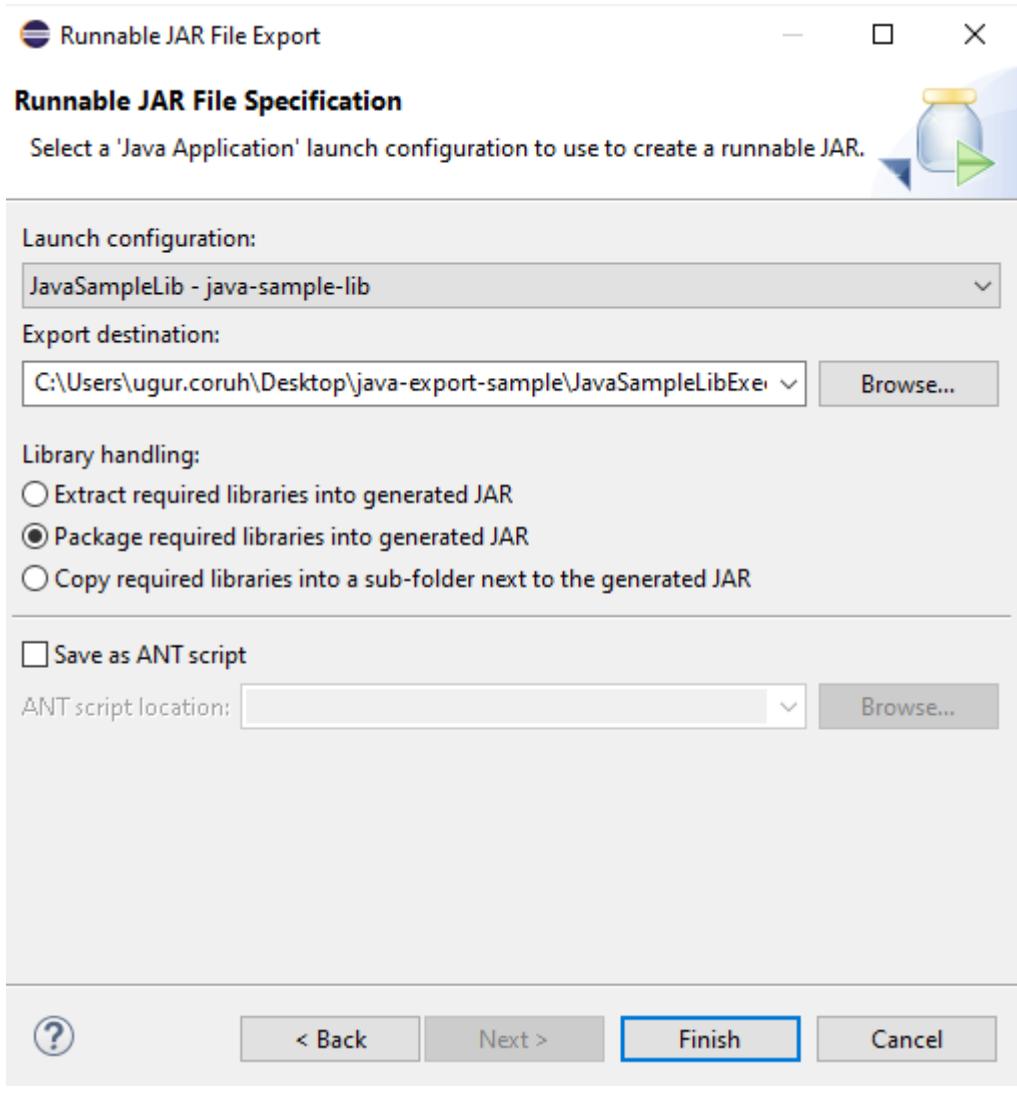
Cancel

click next and set output path for jar file

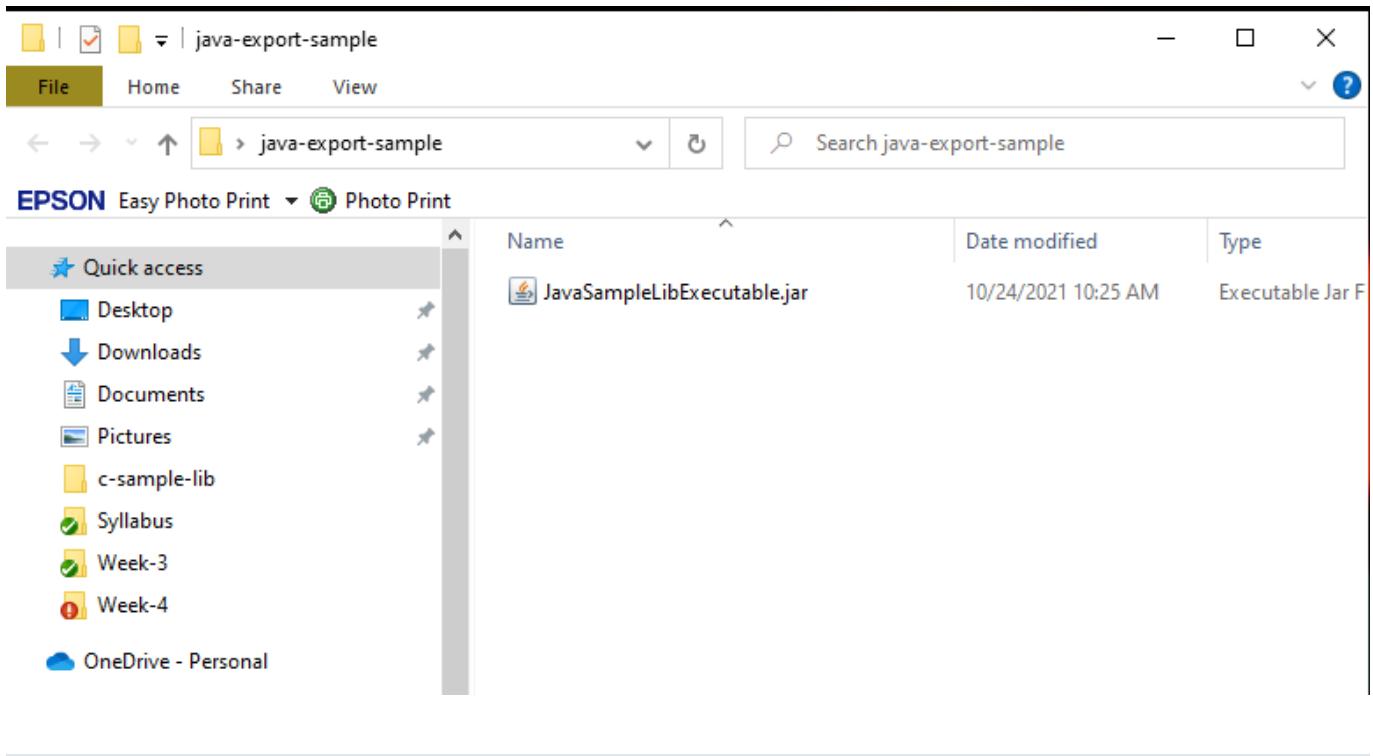


If our project has several external dependency then we can extract this required files (jar, so, dll) in seperated folder or we can combine them and generate a single executable jar

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>
```

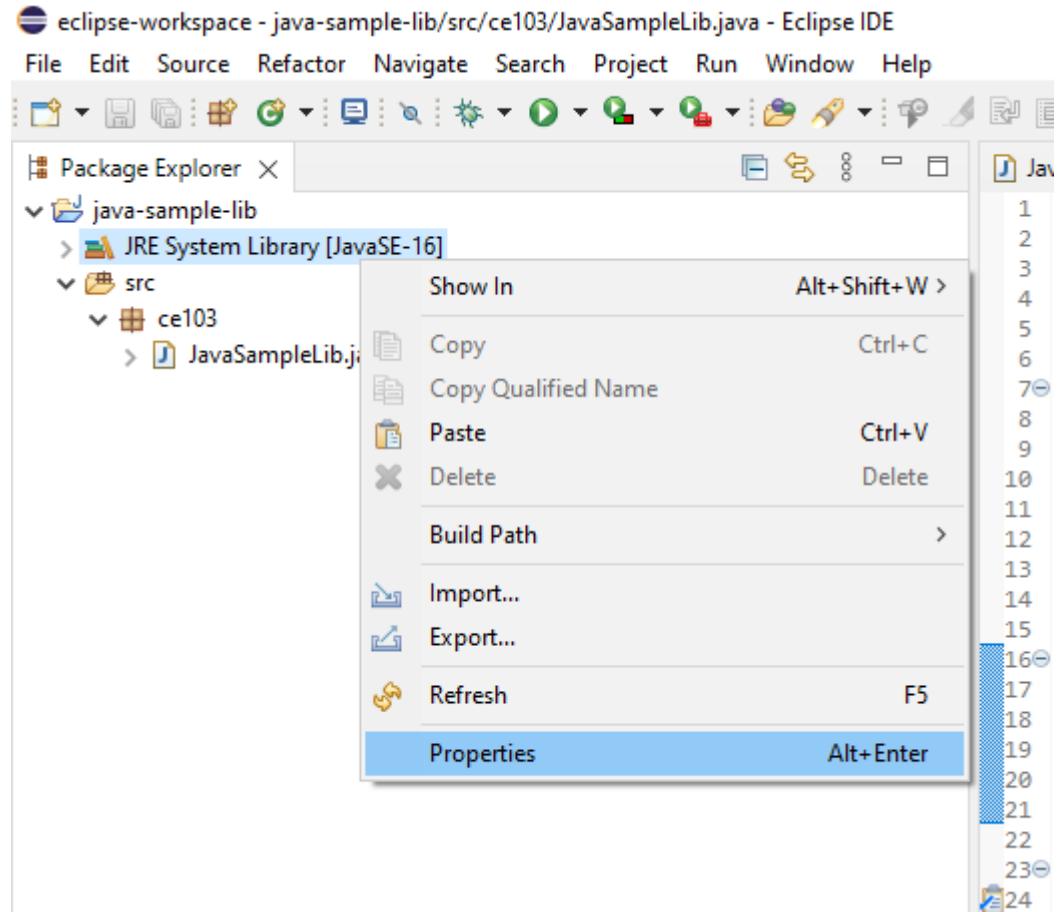
In my case eclipse build JDK is newer than that I installed and set for my OS

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

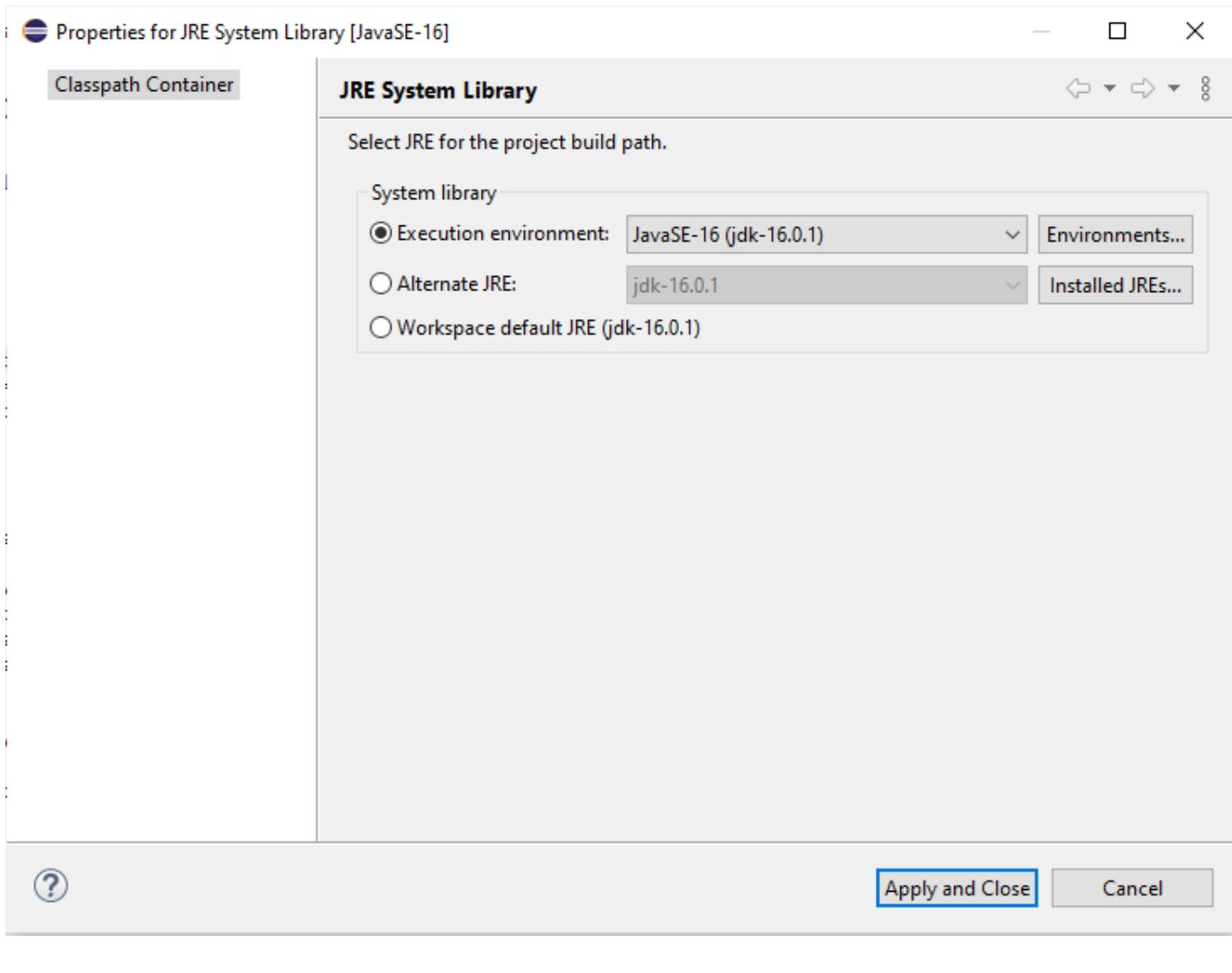
```
C:\Users\ugur.coruh\Desktop\java-export-sample>java -showversion
java version "1.8.0_231"
Java(TM) SE Runtime Environment (build 1.8.0_231-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.231-b11, mixed mode)

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

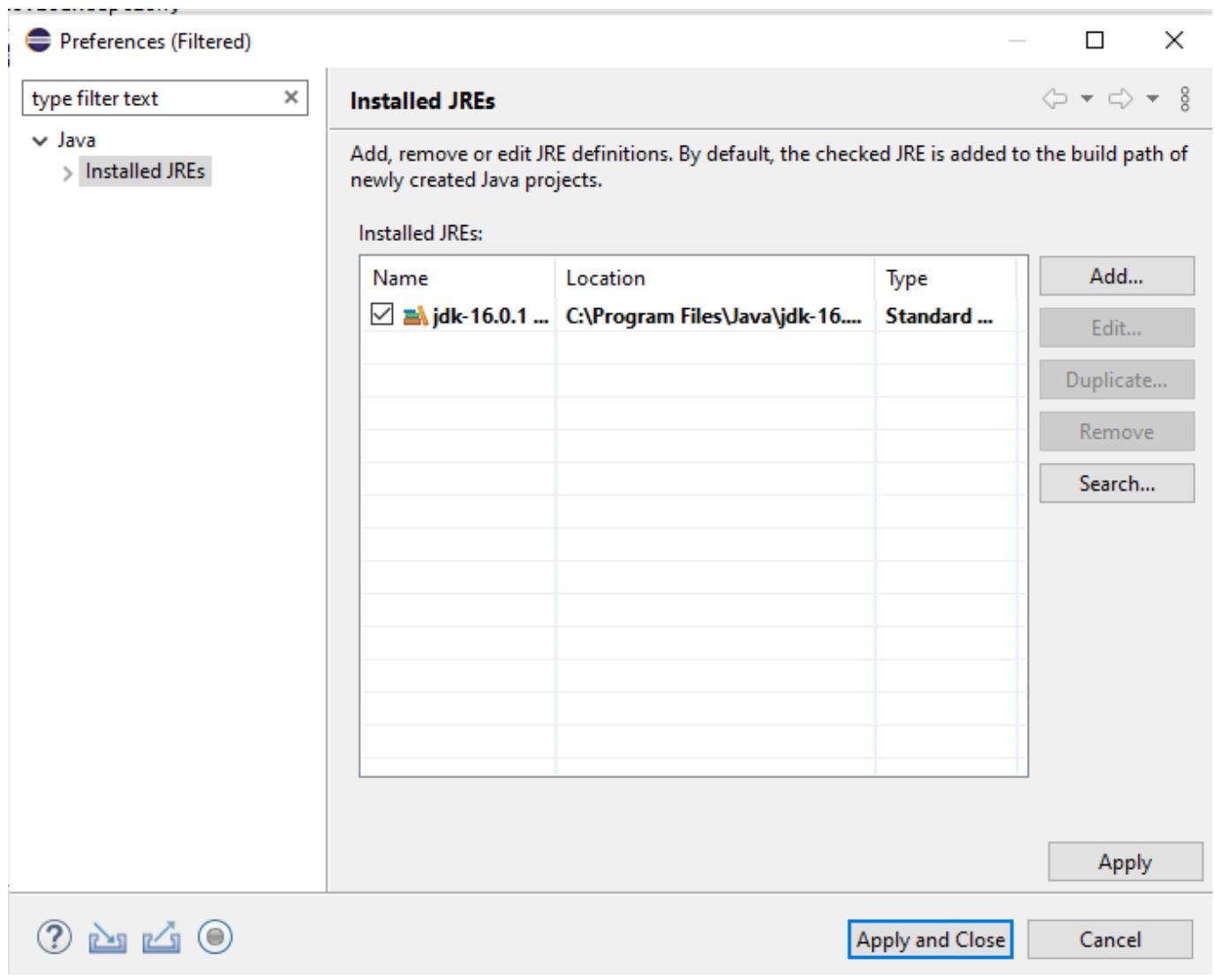
We can found installed and builded JDK for our application from Eclipse setting



select environments



select installed JRE or JDK



you can see installed JRE or JDK home

C:\Program Files\Java\jdk-16.0.1



JRE Definition

Specify attributes for a JRE



JRE home:

[Directory...](#)

JRE name:

Default VM arguments:

[Variables...](#)

JRE system libraries:

> C:\Program Files\Java\jdk-16.0.1\lib\jrt-fs.jar

[Add External JARs...](#)

[Javadoc Location...](#)

[Source Attachment...](#)

[External annotations...](#)

[Remove](#)

[Up](#)

[Down](#)

[Restore Default](#)



[Finish](#)

[Cancel](#)

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 >

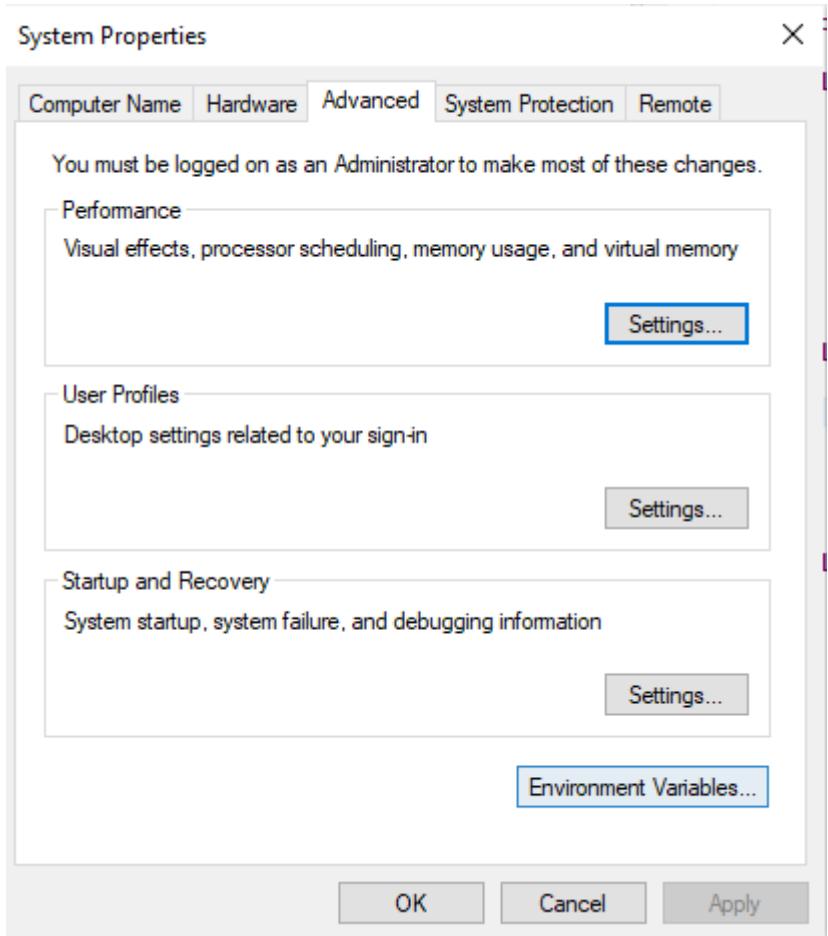
 Taskbar notification area >

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

Search the web

 syste - See web results >

Apps (7+)



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\

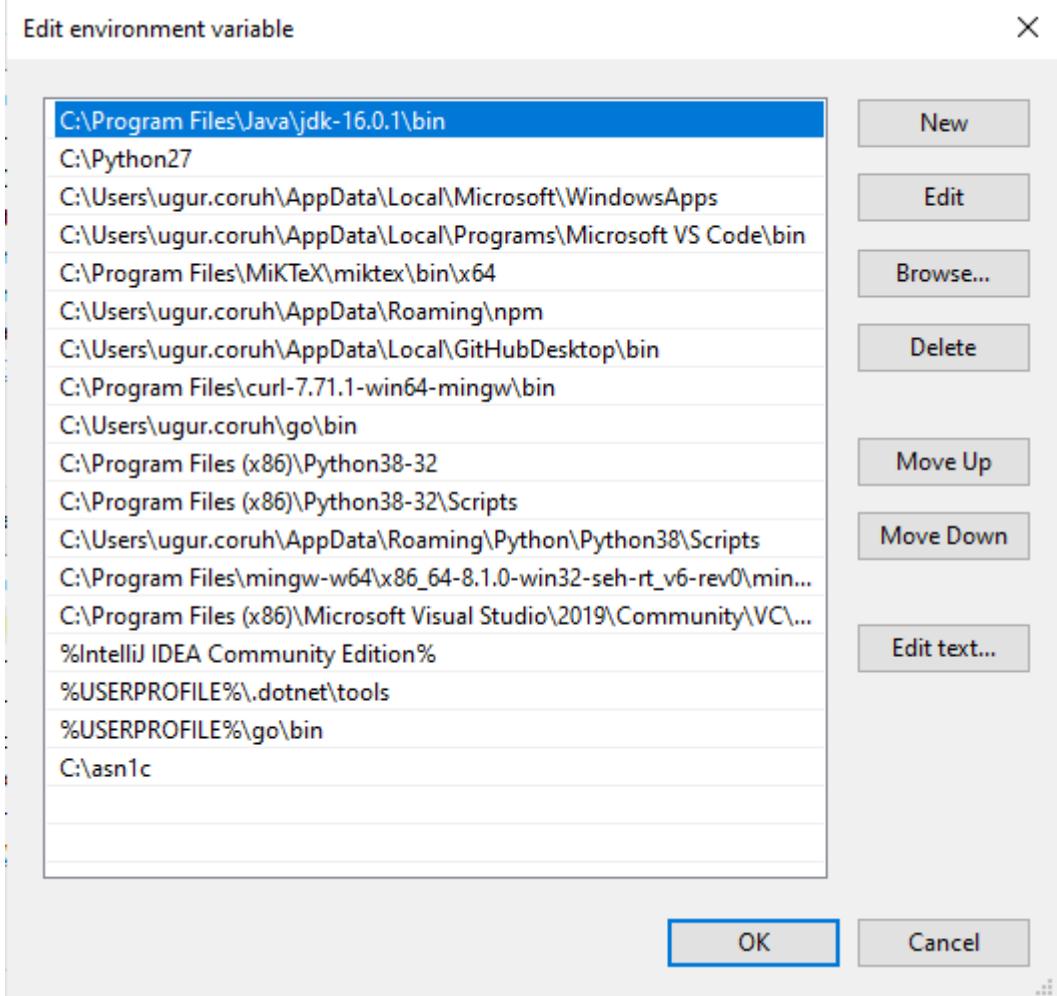
New...

Edit...

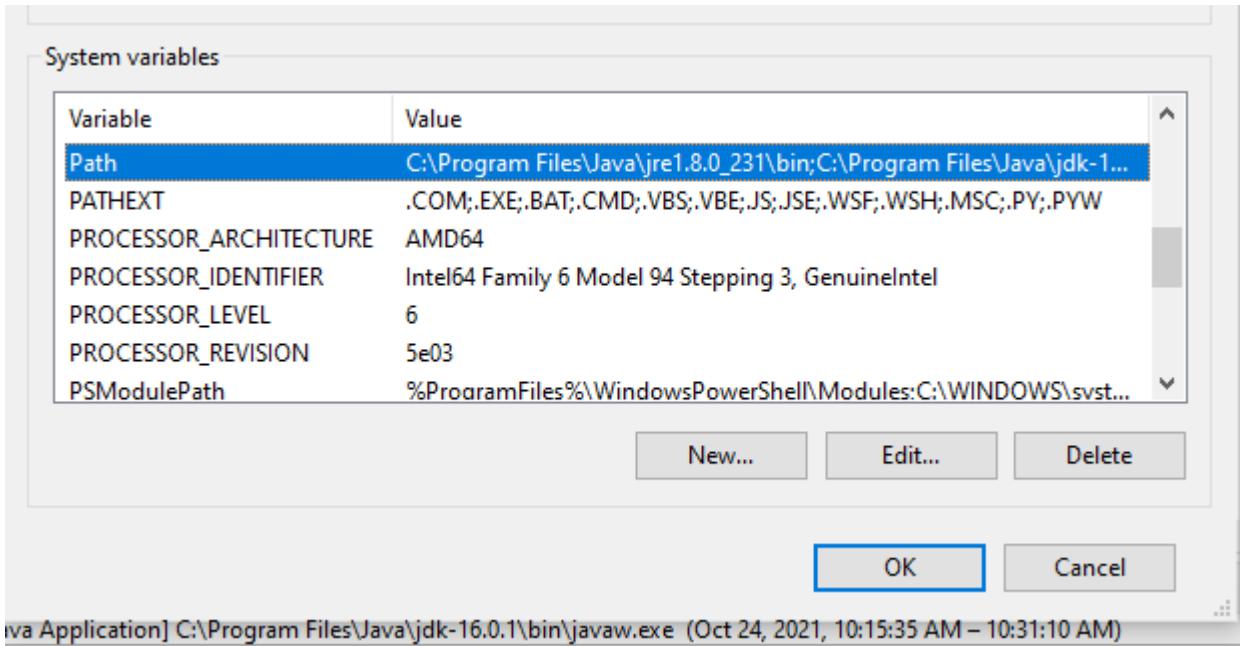
Delete

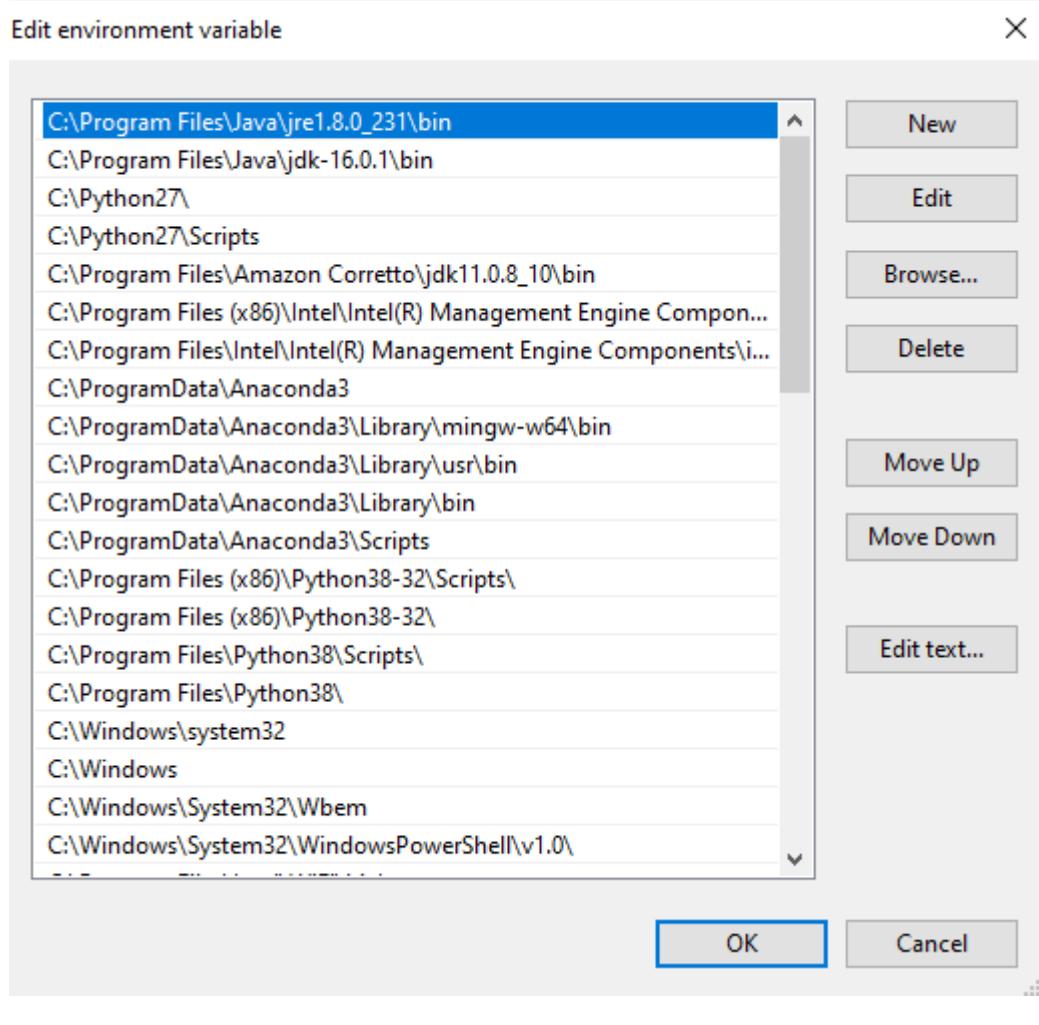
OK

Cancel

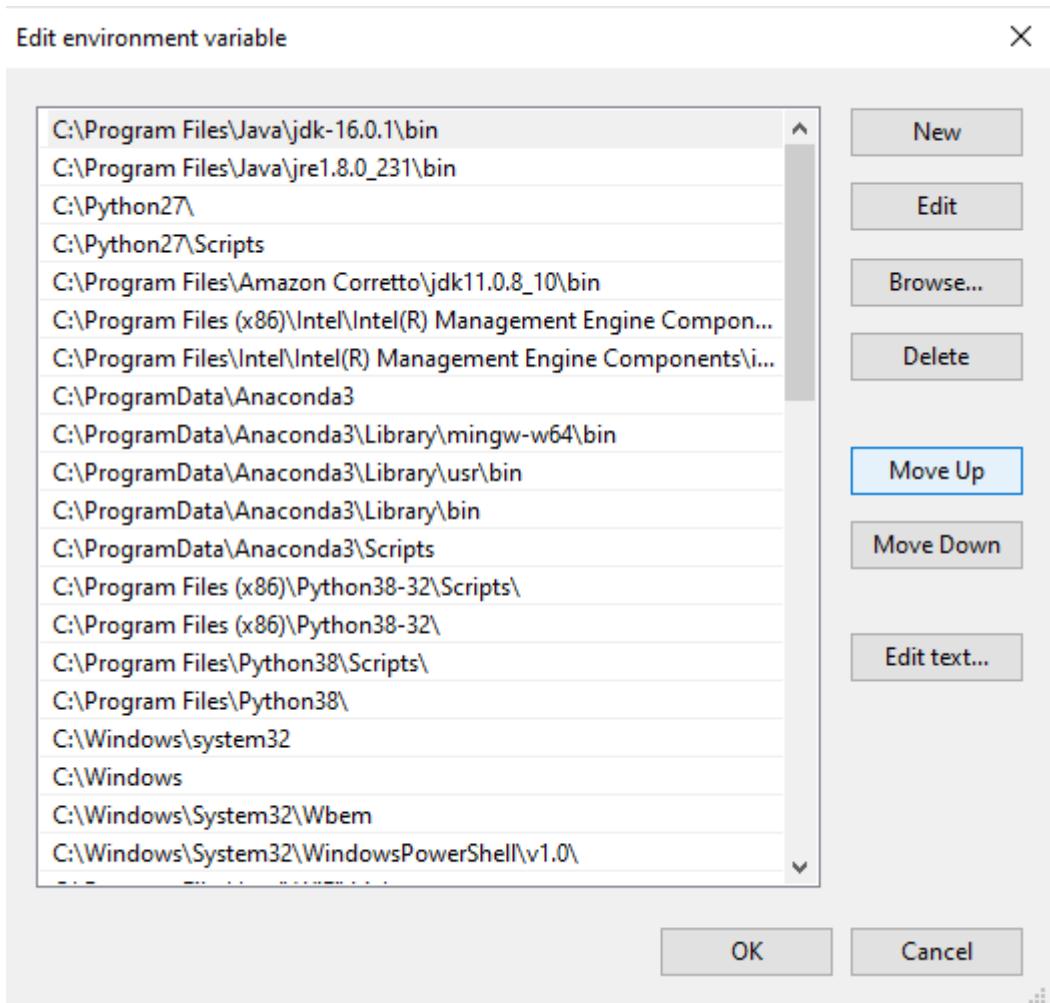


Check system settings





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>
```

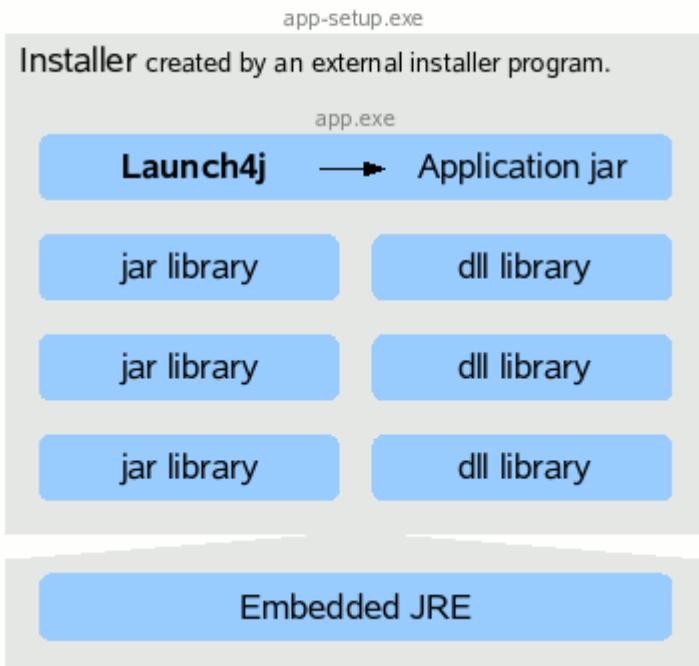
and now if we enter and run application as follow we will see output

```
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

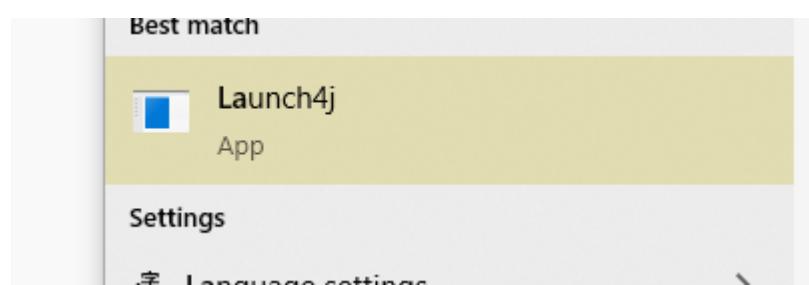
[Launch4j - Cross-platform Java executable wrapper](#)



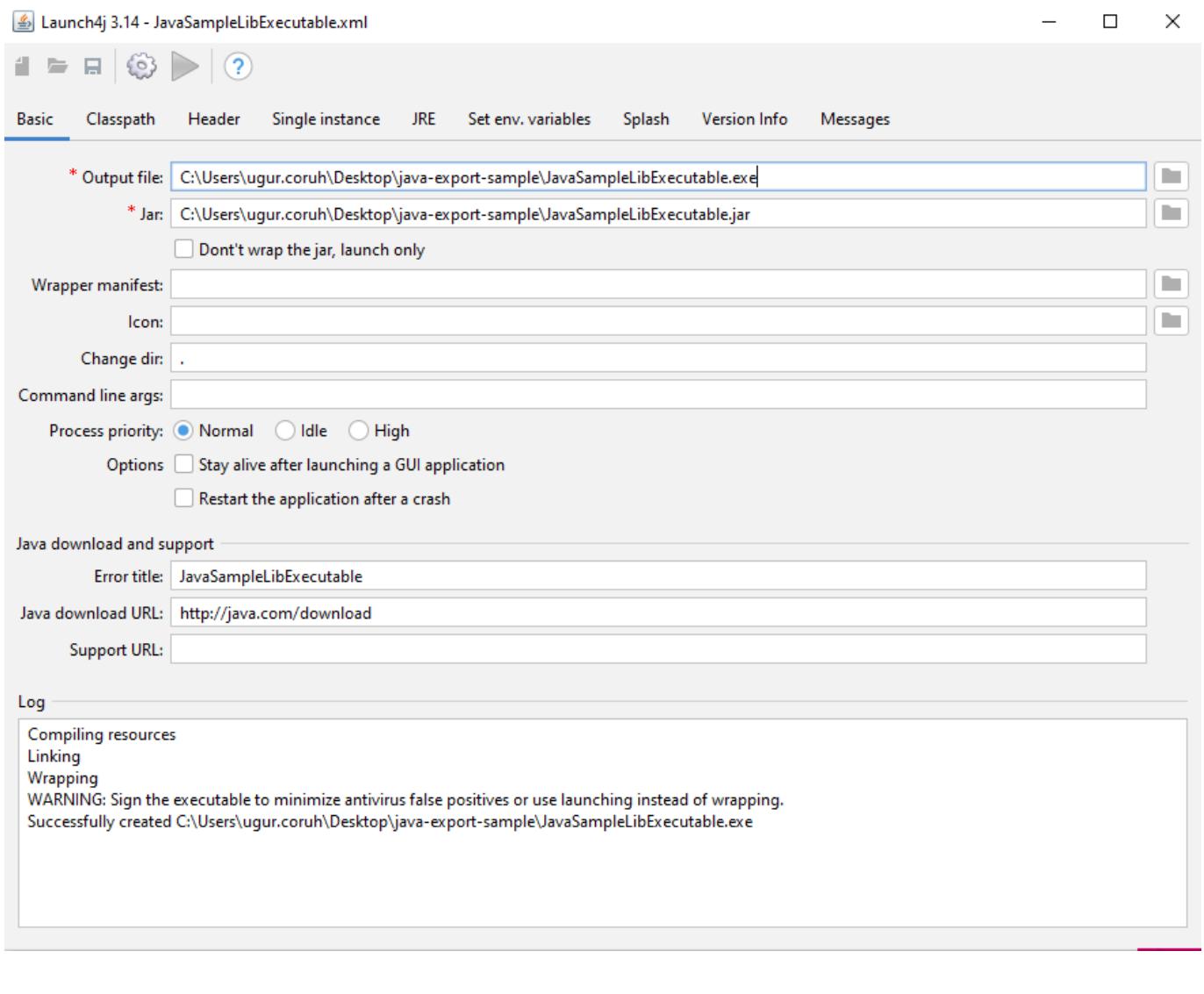
you can watch this tutorial also

[How to convert jar to exe using Launch4J Full explanation - YouTube](#)

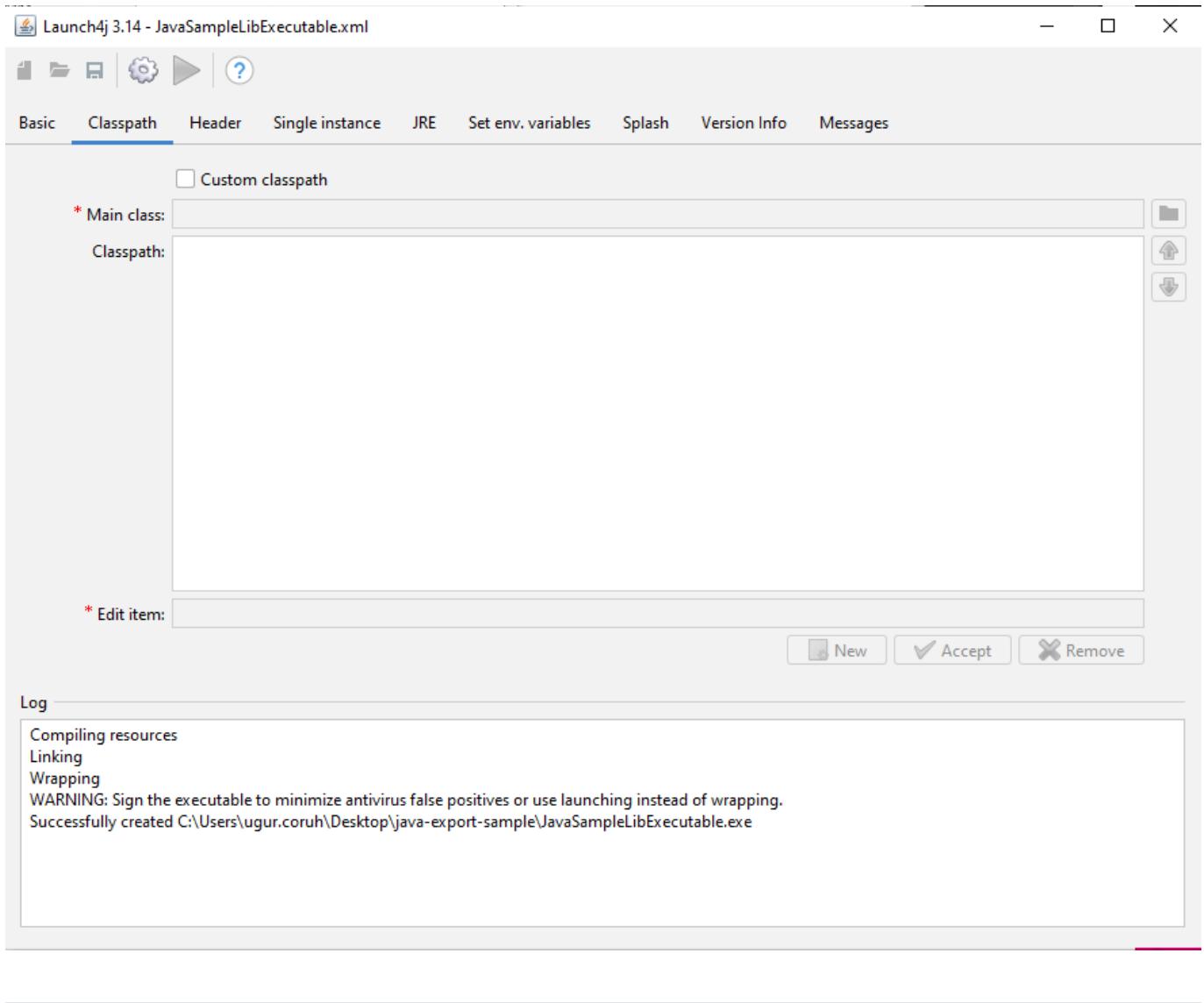
Download and install launch4j and open application



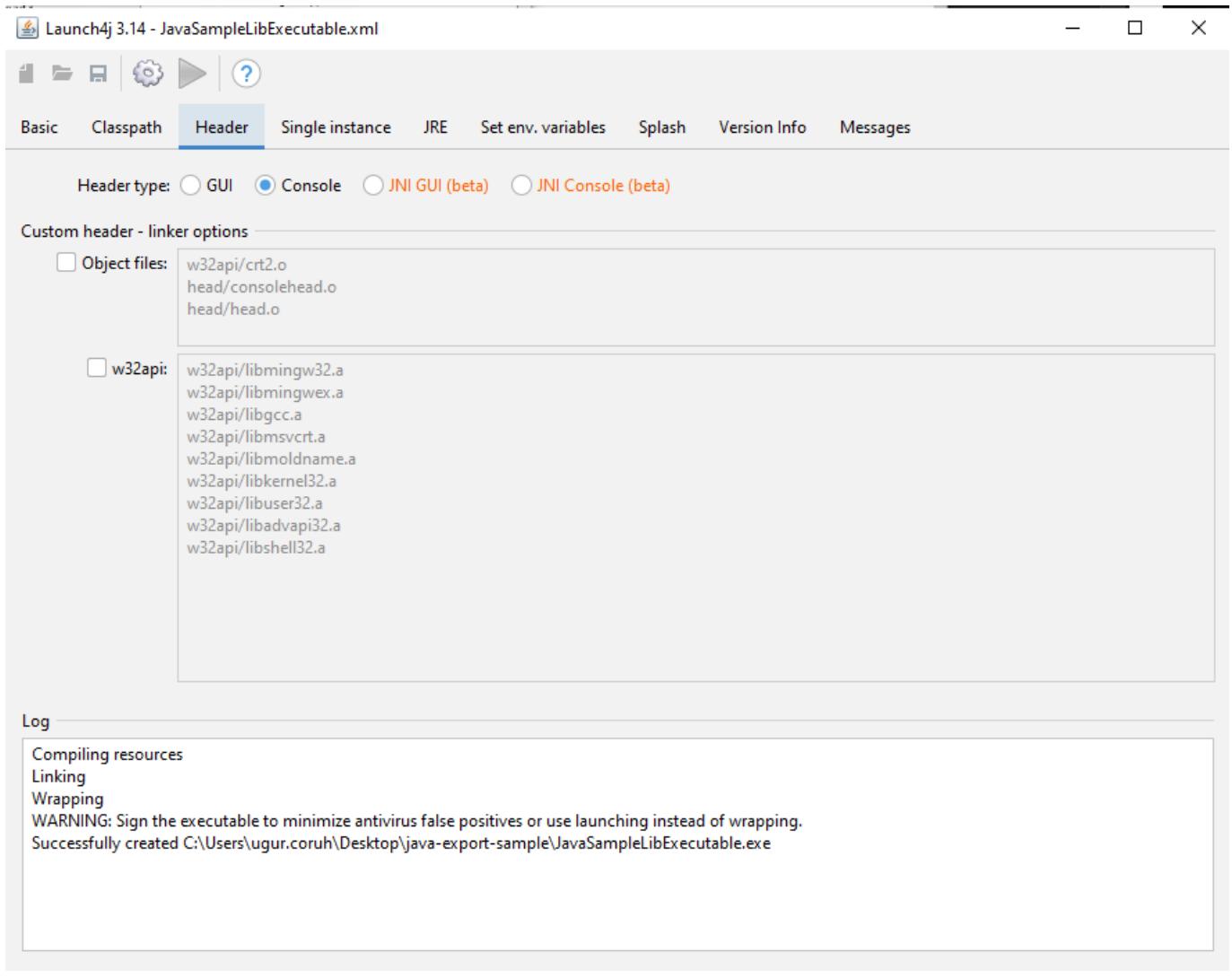
configure your application settings similar to below select jar file and exe output path



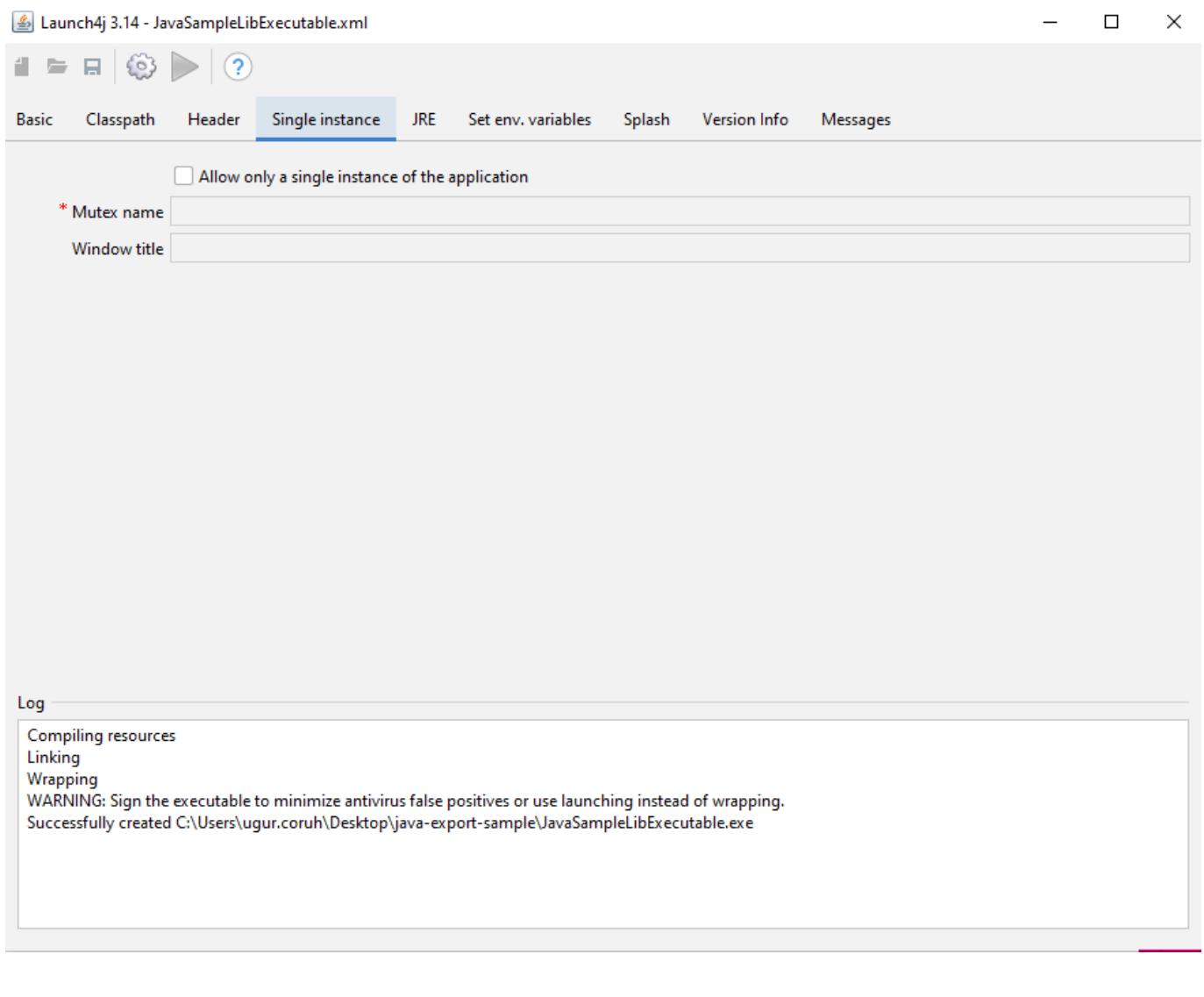
we can customize main class if have multiple main class



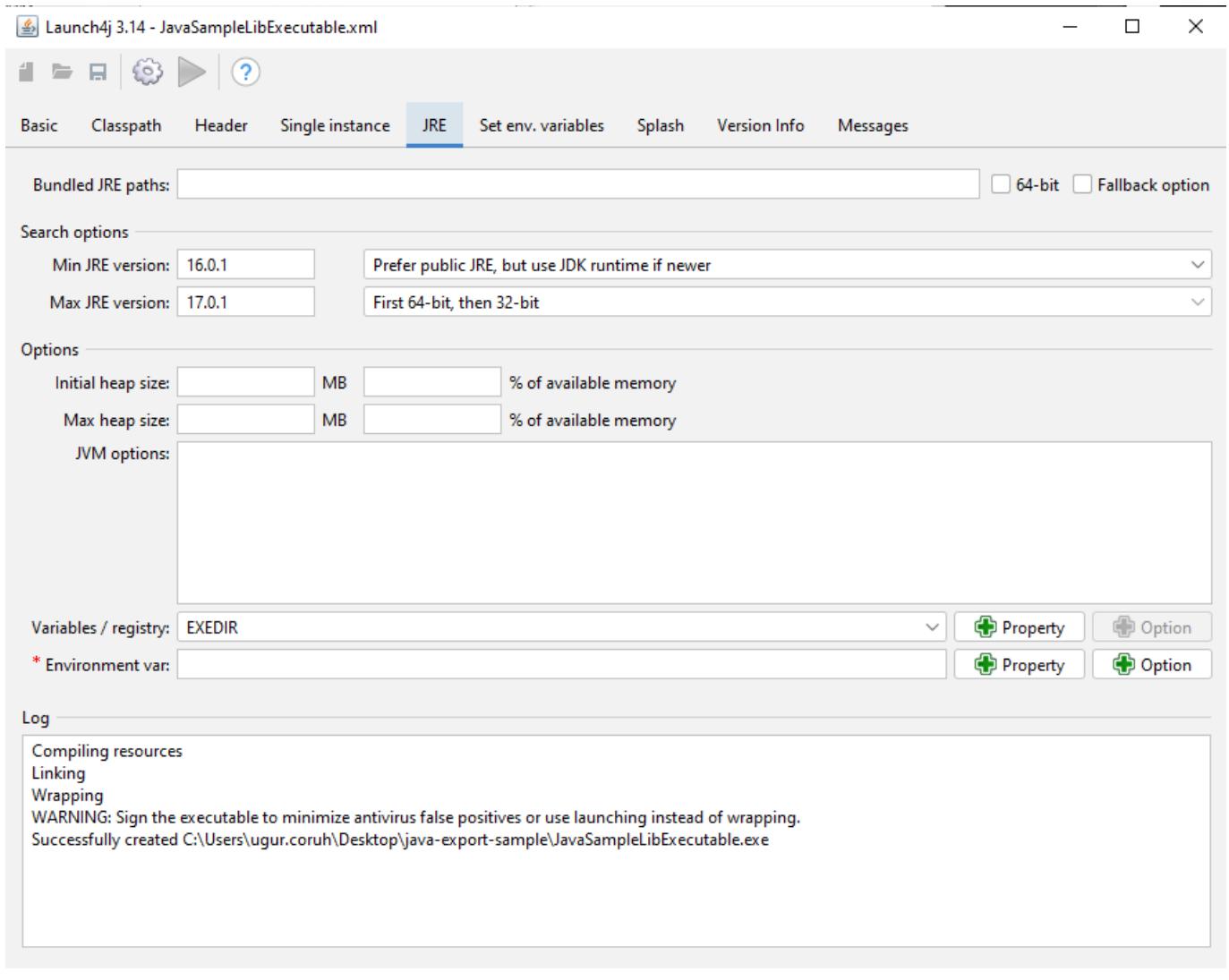
select console from setting for this application



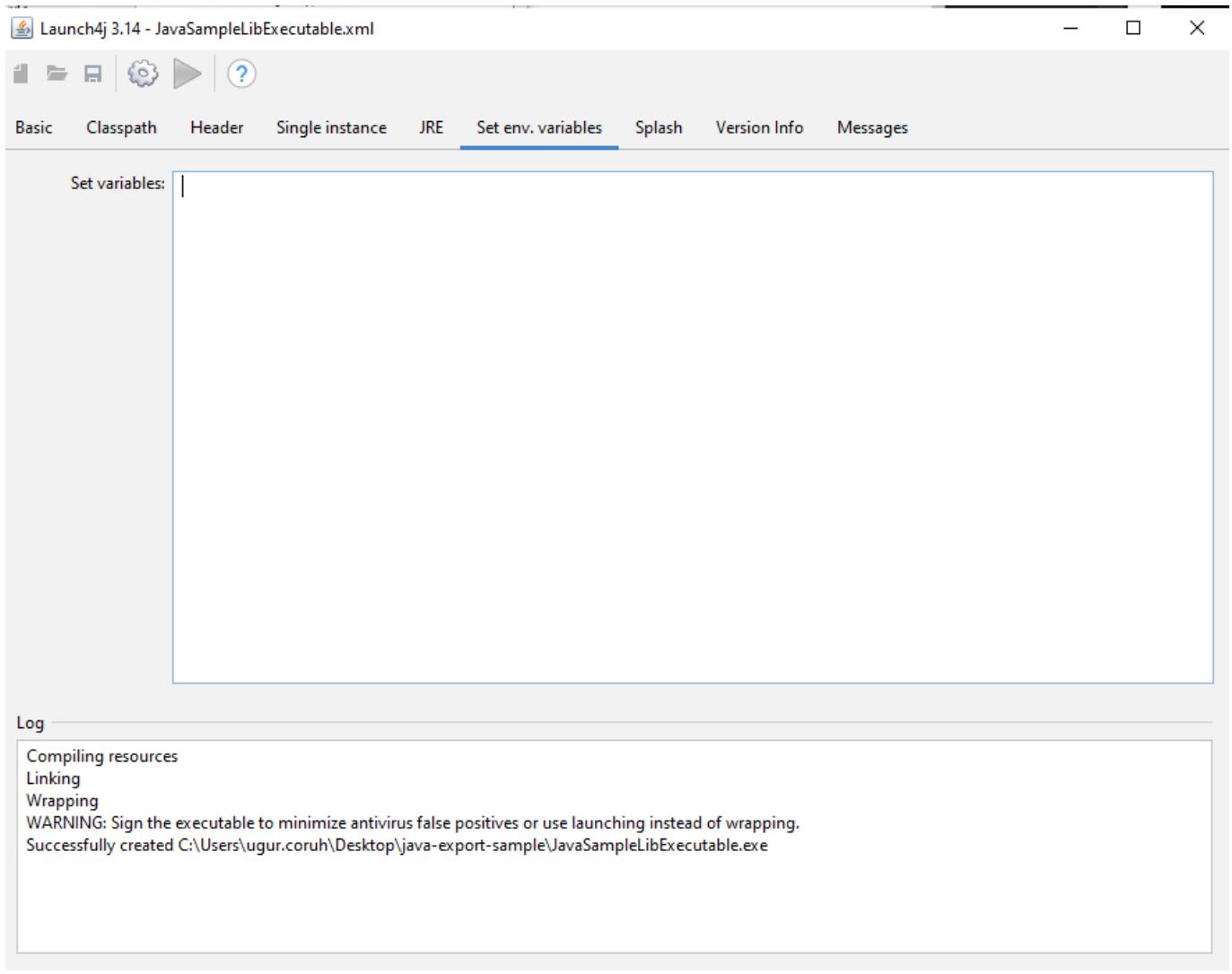
we can provide a single running application, this setting avoid to run multiple instances



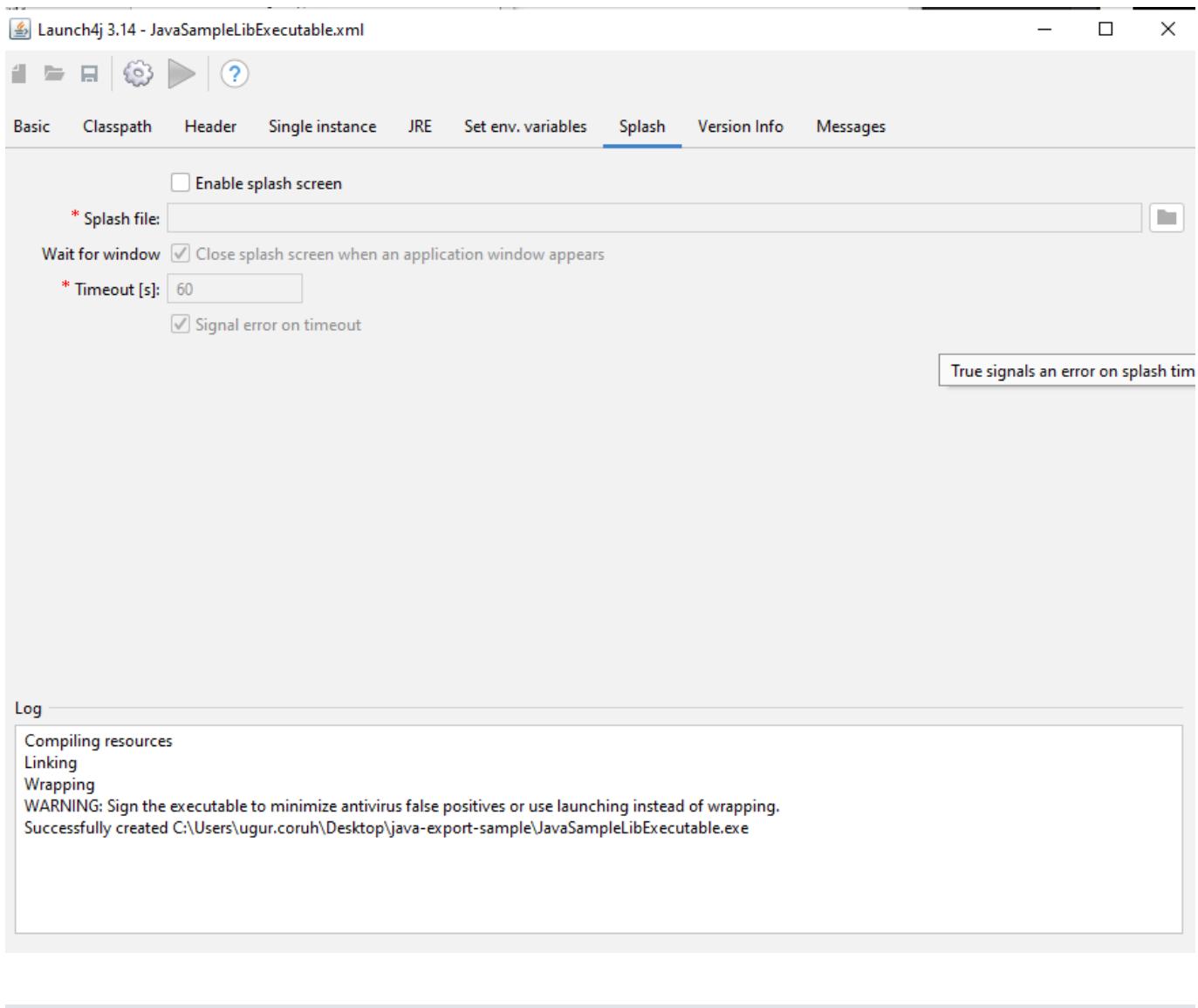
we need to set runtime environment versions



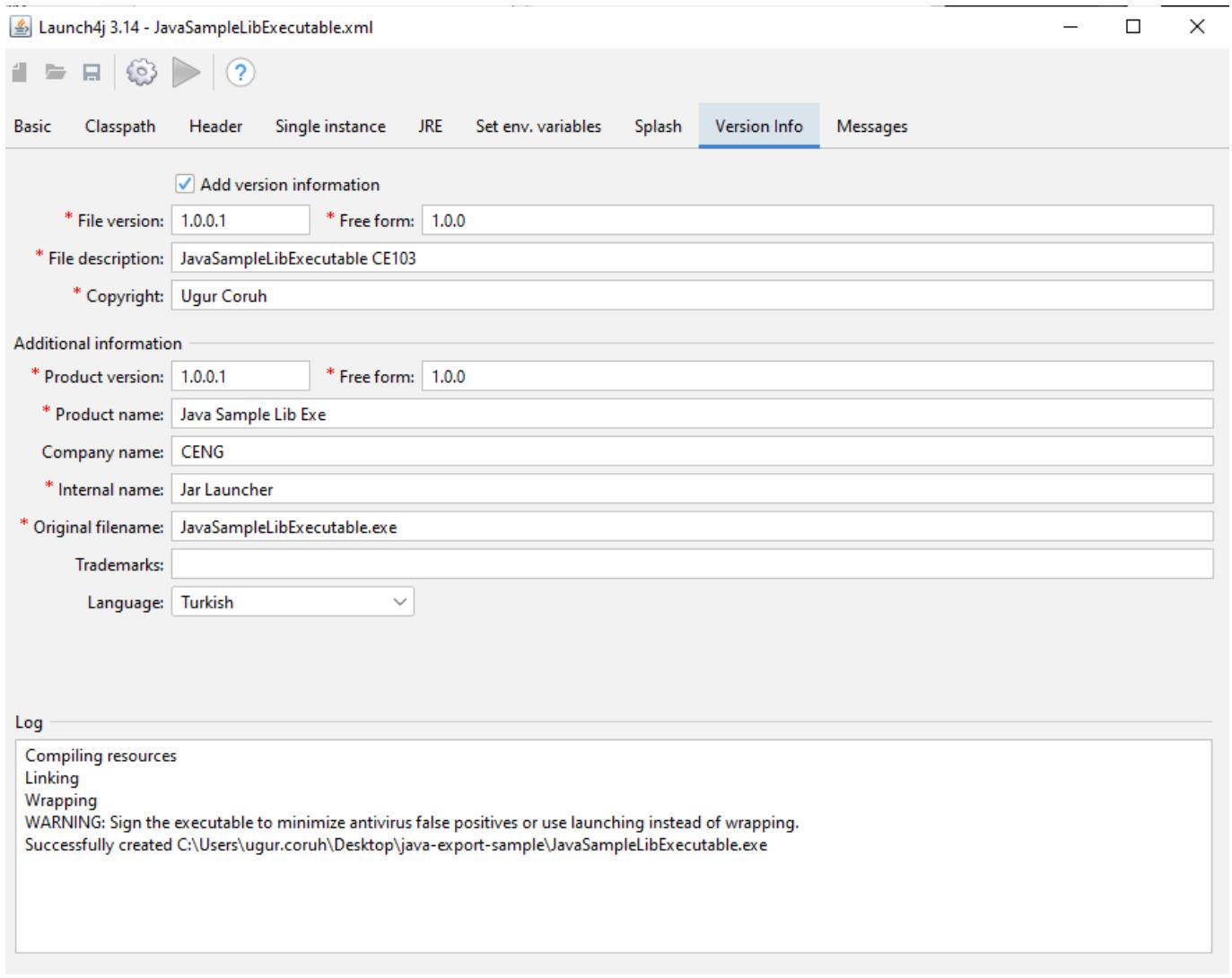
you can set system parameters before running application



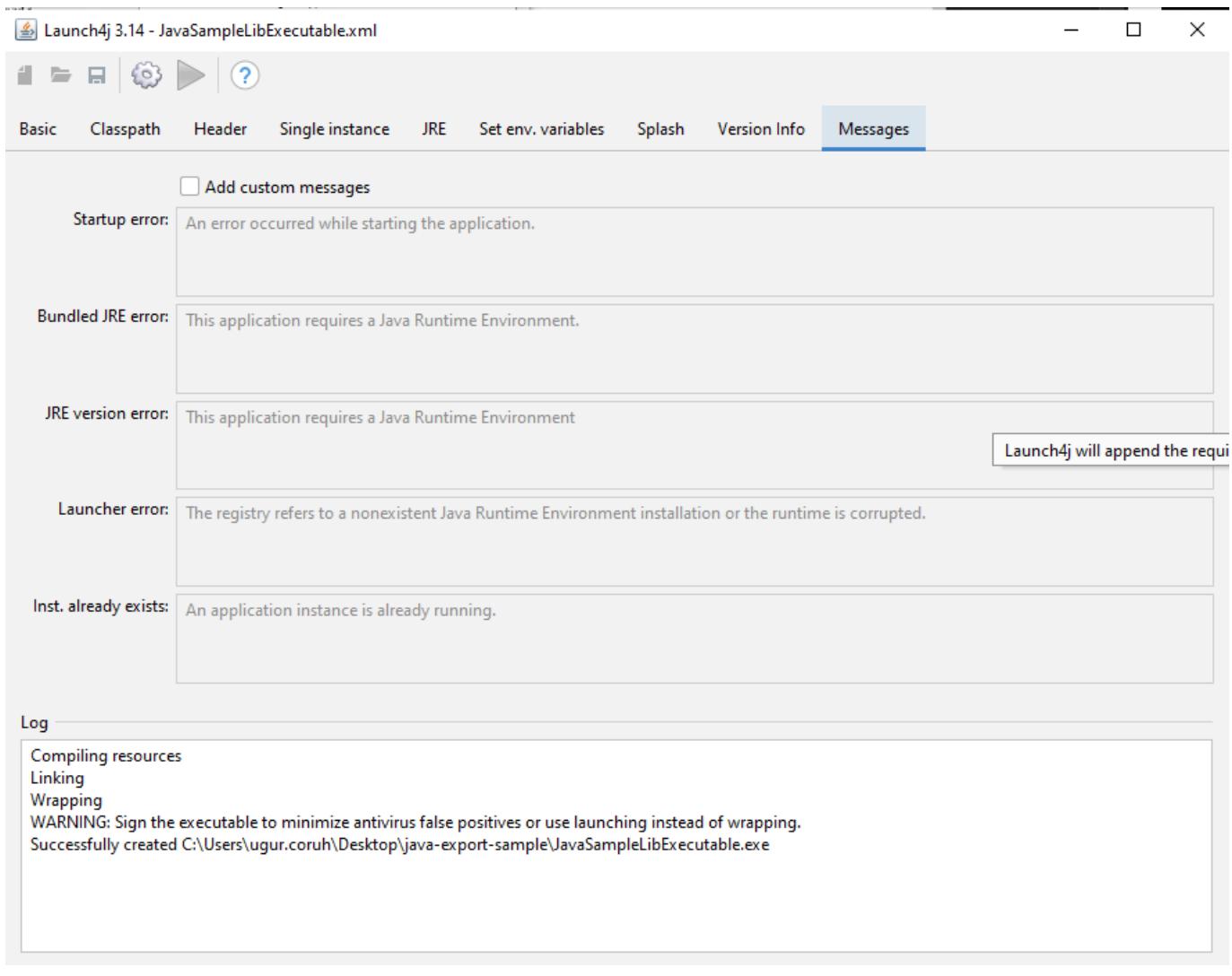
with splash screen you can show a splash screen image for your application



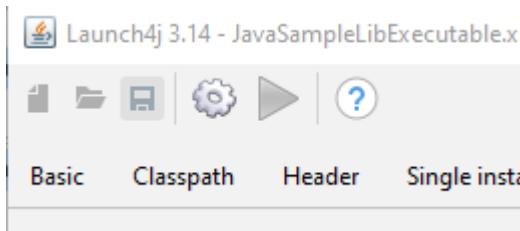
File attributes such as version product information is configured from version info tab



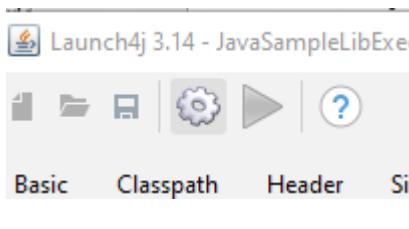
if your application runtime condition has an error then you can show this customized messages also



with this options save configuration file xml



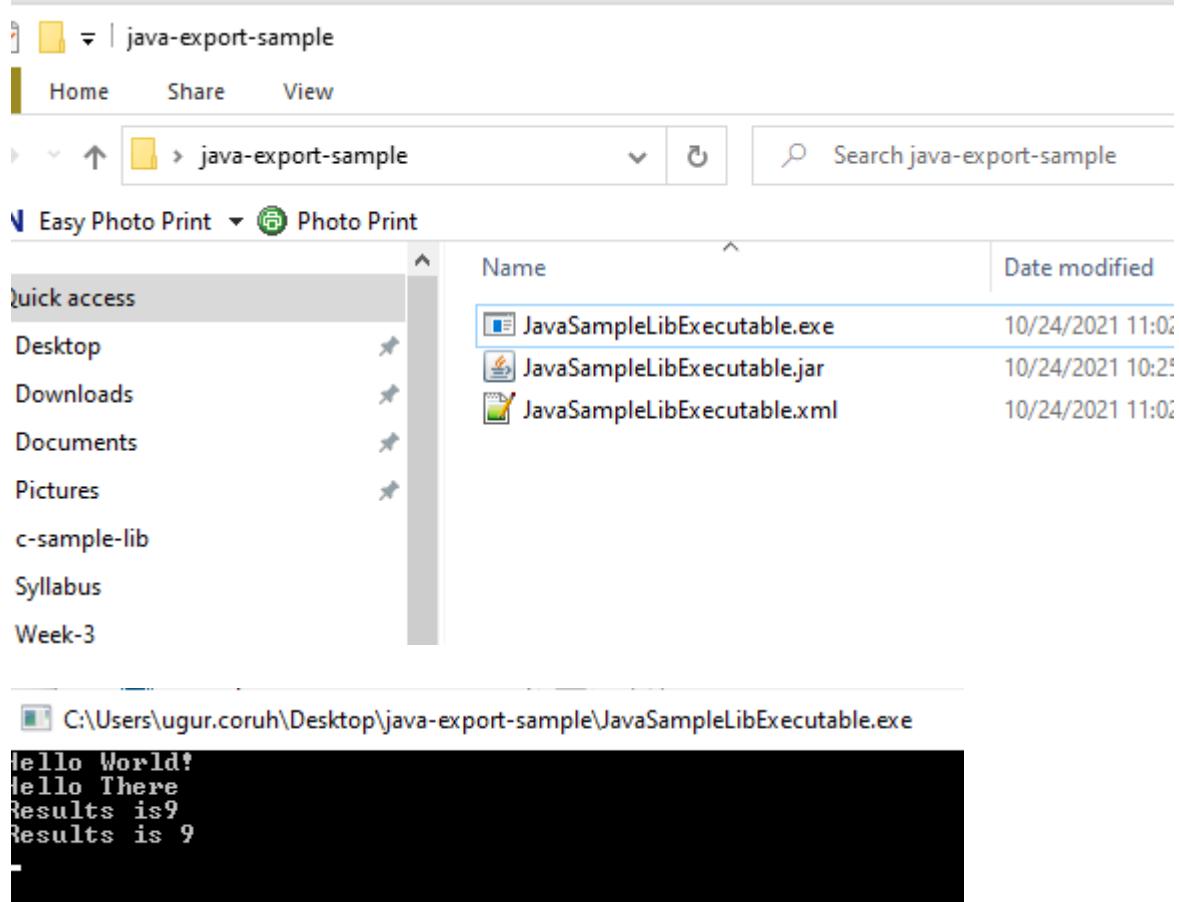
and compile settings



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
```

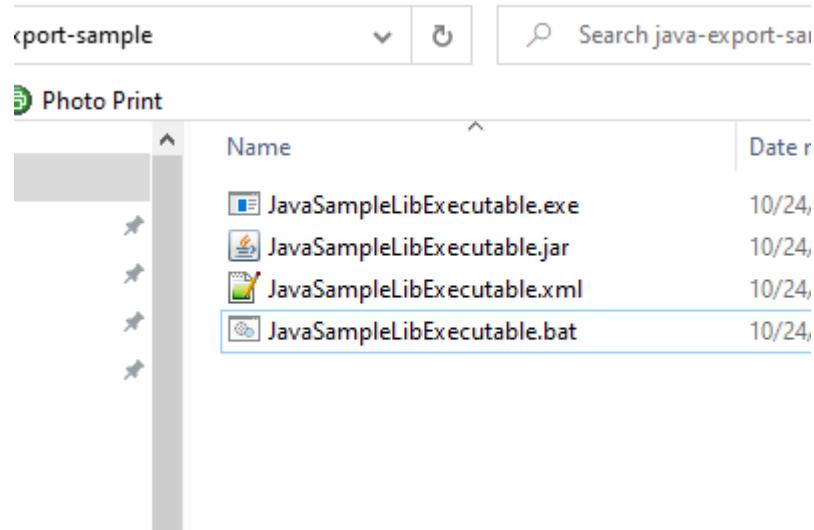
now we can run exe by click



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\ugur.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

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

File Edit Source Refactor Navigate Search Project Run Window Help

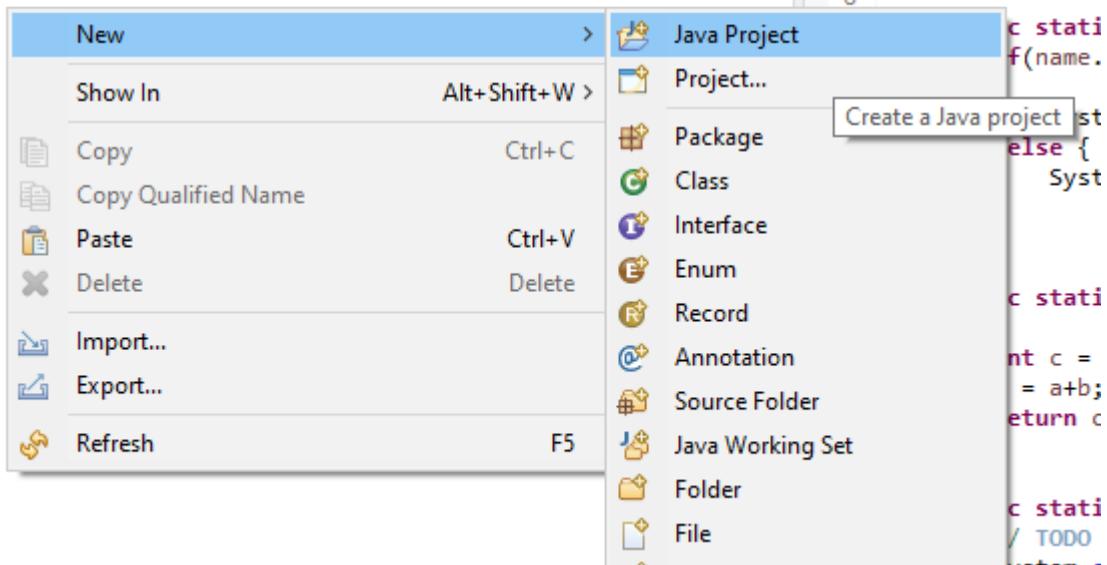


Package Explorer X

> java-sample-lib

JavaSampleLib.java X

```
1 package ce103;
2
3 import java.io.I
4
5 public class Jav
6
```



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.



Project name:

Use default location

Location:

JRE

Use an execution environment JRE:

Use a project specific JRE:

Use default JRE 'jdk-16.0.1' and workspace compiler preferences

[Configure JREs...](#)

Project layout

Use project folder as root for sources and class files

Create separate folders for sources and class files

[Configure default...](#)

Working sets

Add project to working sets

Working sets:

Module

Create module-info.java file



< Back

Next >

 New Java Project

Java Settings

Define the Java build settings.



Source Projects Libraries Order and Export Module Dependencies

java-sample-app

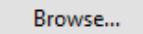
 src

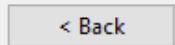
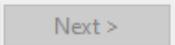
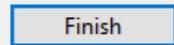
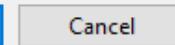
Details

-  [Create new source folder](#): use this if you want to add a new source folder to your project.
-  [Link additional source](#): use this if you have a folder in the file system that should be used as additional source folder.
-  [Add project 'java-sample-app' to build path](#): Add the project to the build path if the project is the root of packages and source files. Entries on the build path are visible to the compiler and used for building.

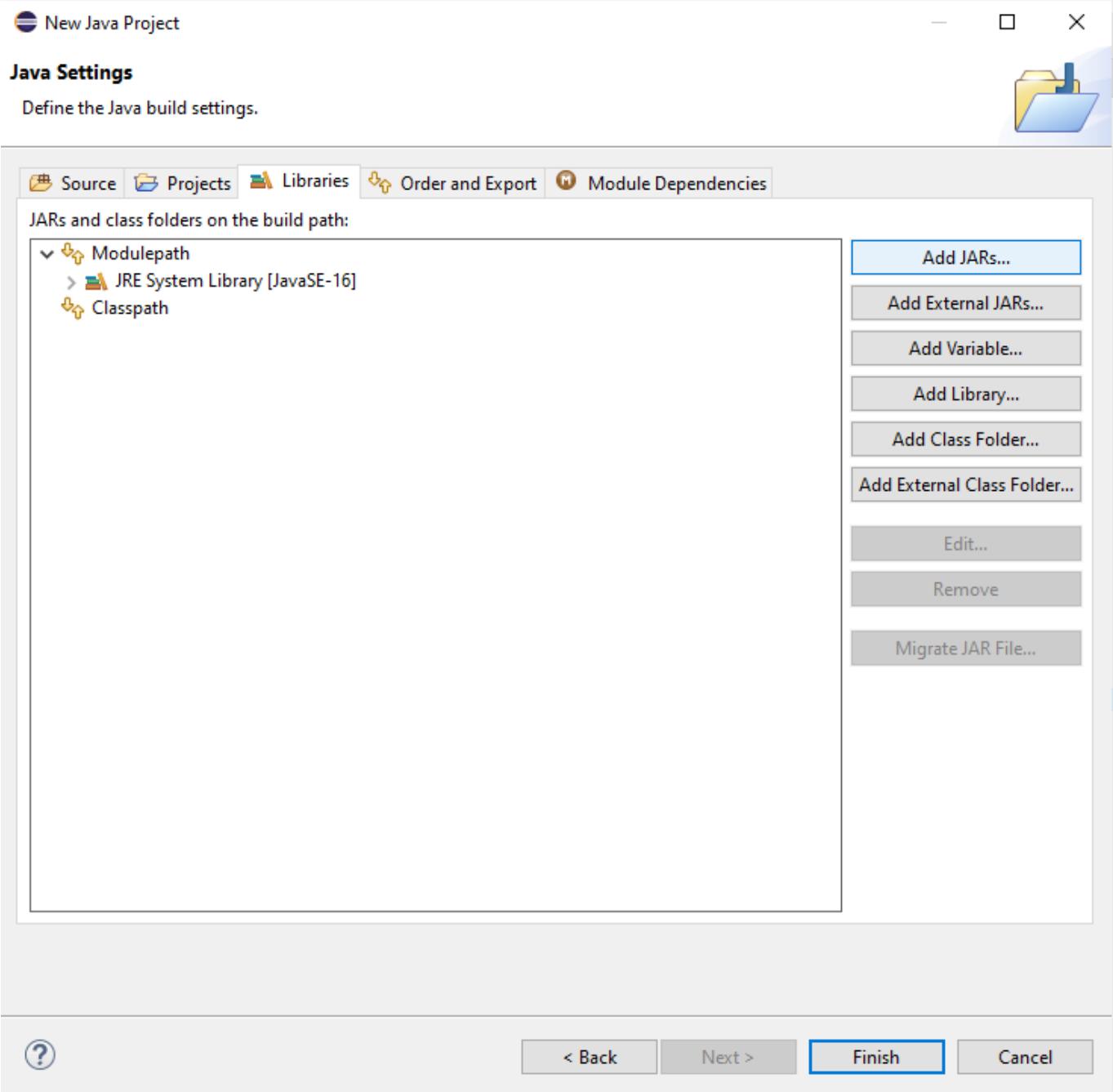
Allow output folders for source folders

Default output folder:

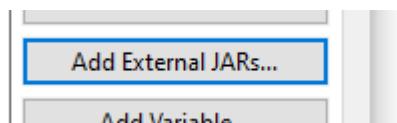


you can set libraries in this step from but our library should exported for our solution

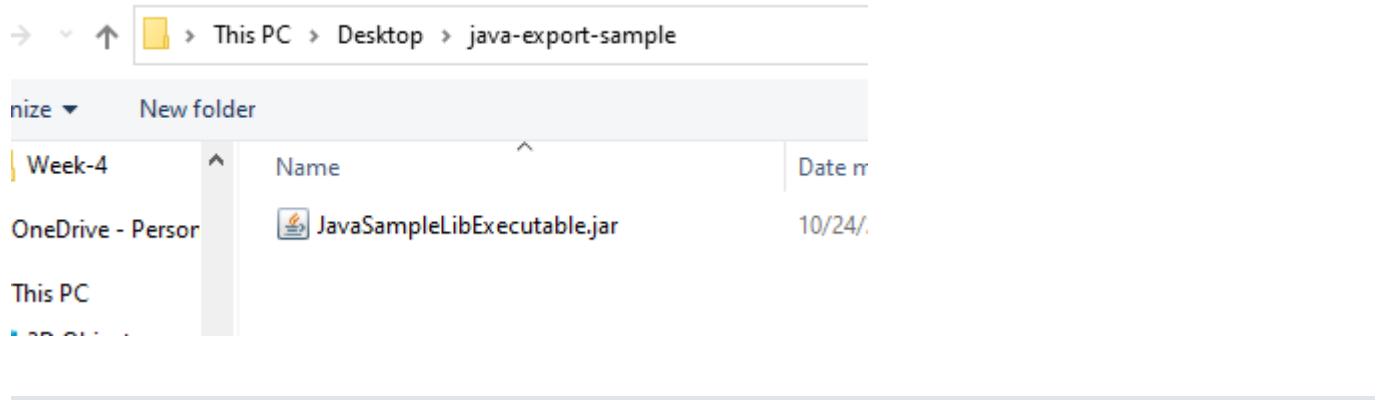


Select Add External JARs...

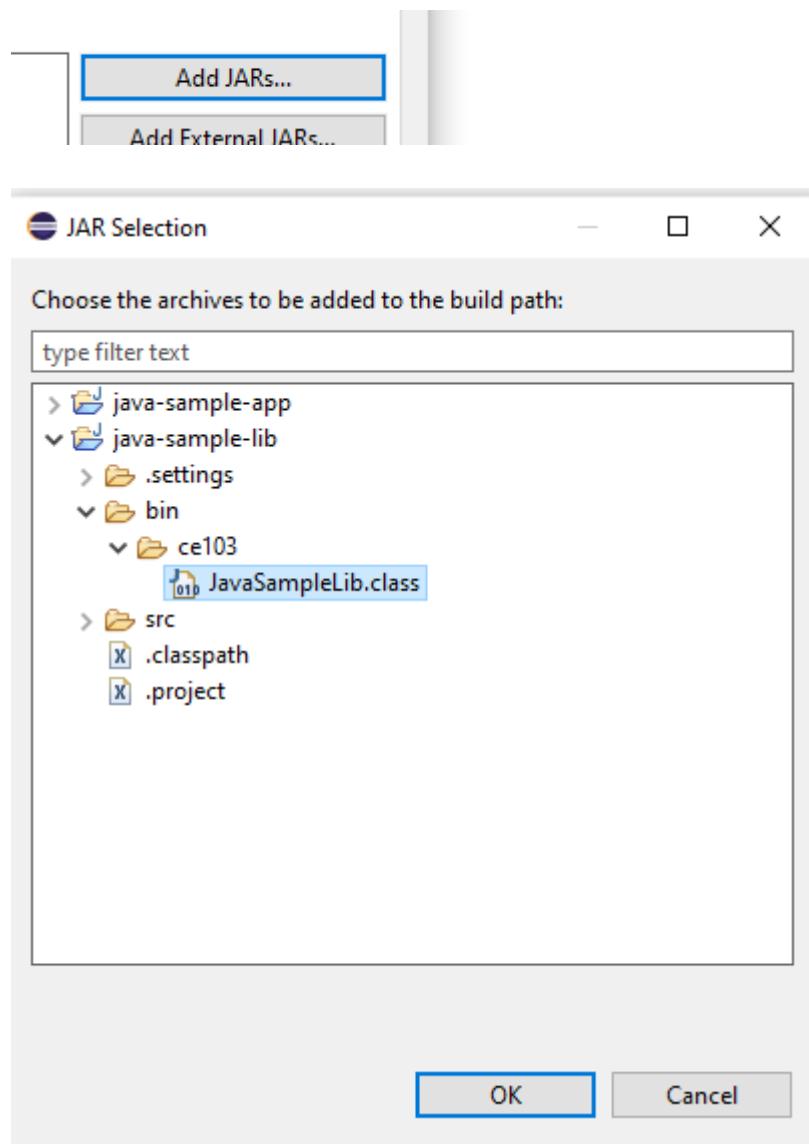


Open Exported jar folder and select

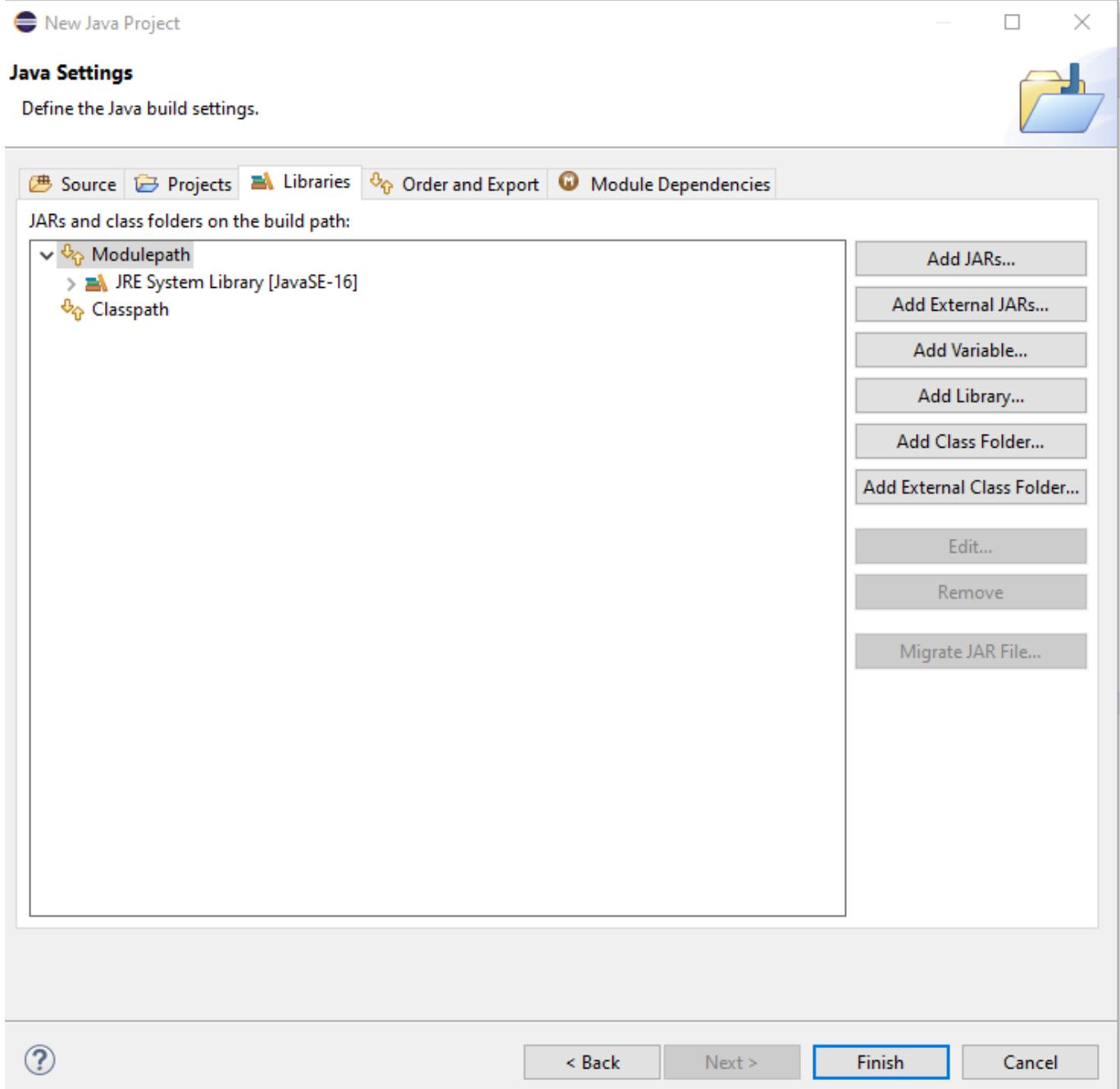
Selection



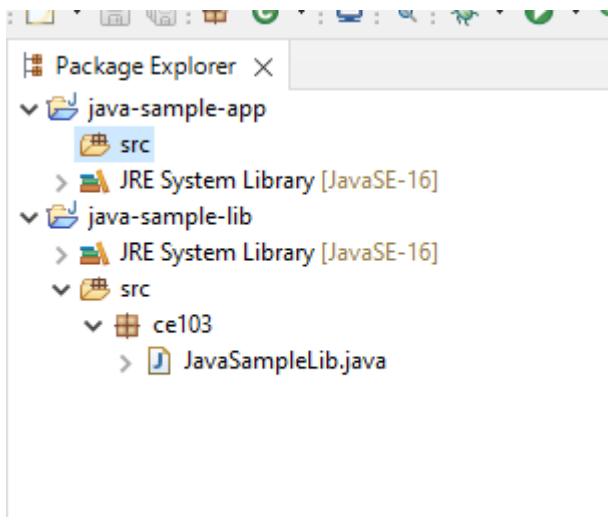
Or we can select by Add jar from current workspace



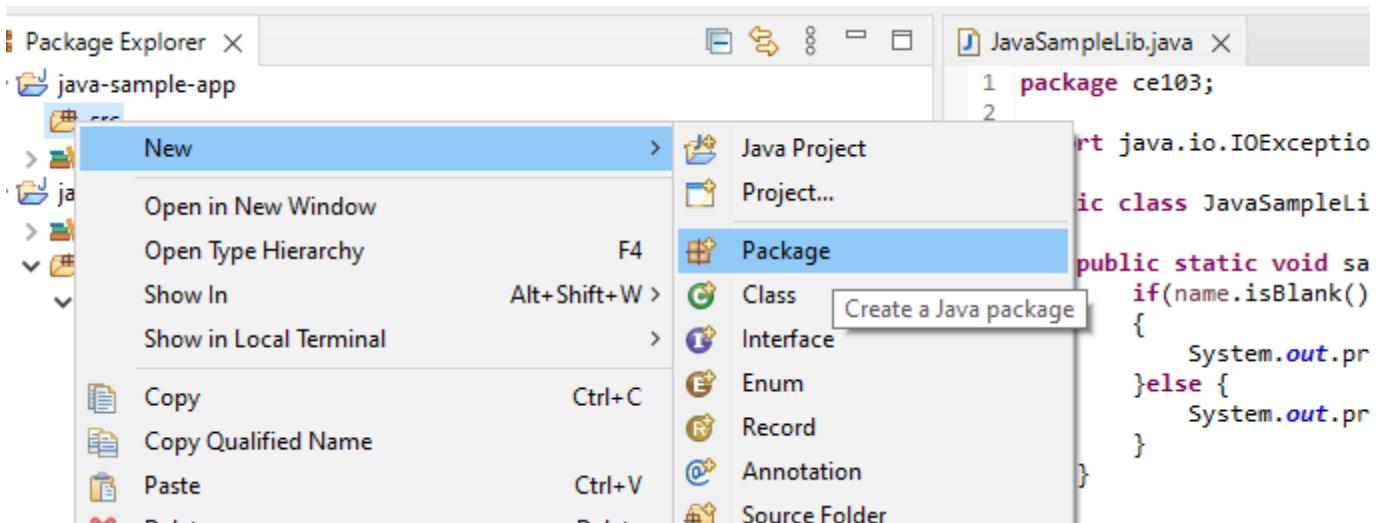
but in this step I won't add anything I'll add references later

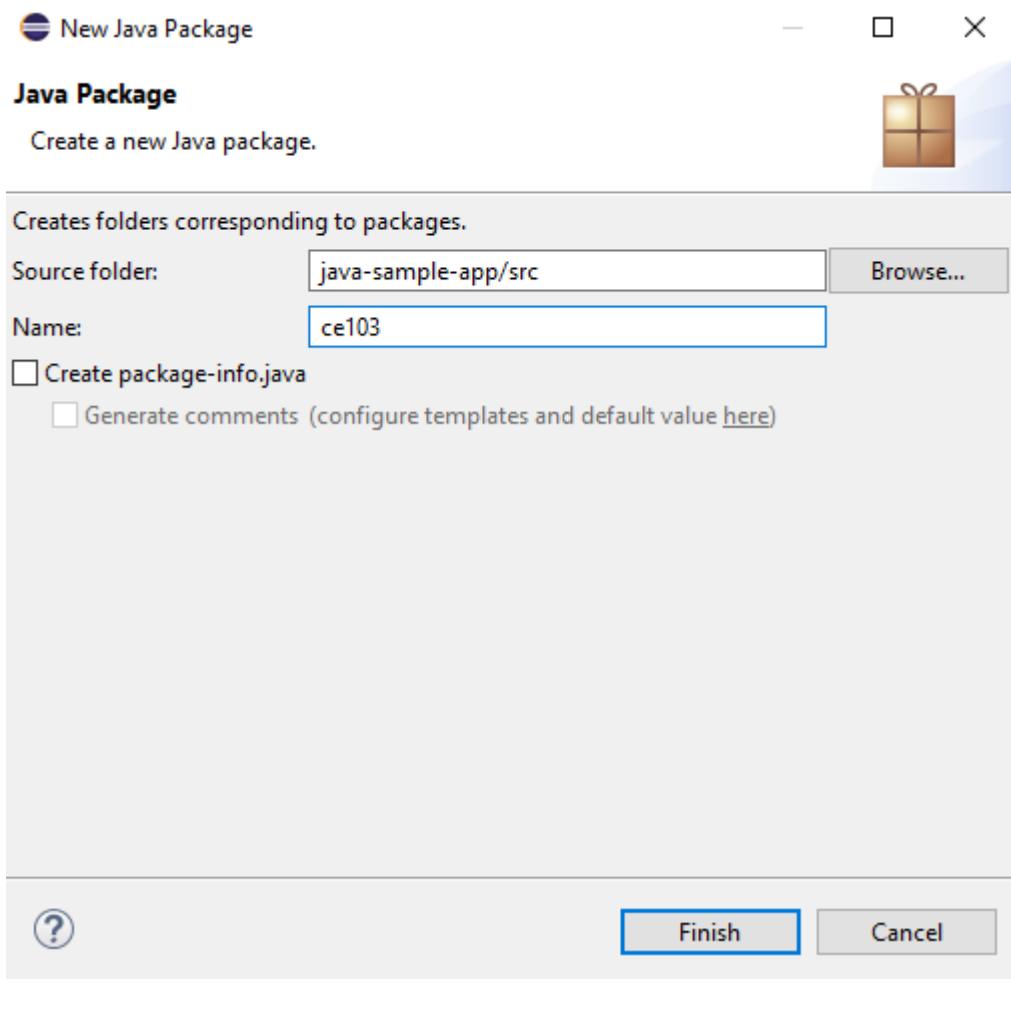


we will have the following project

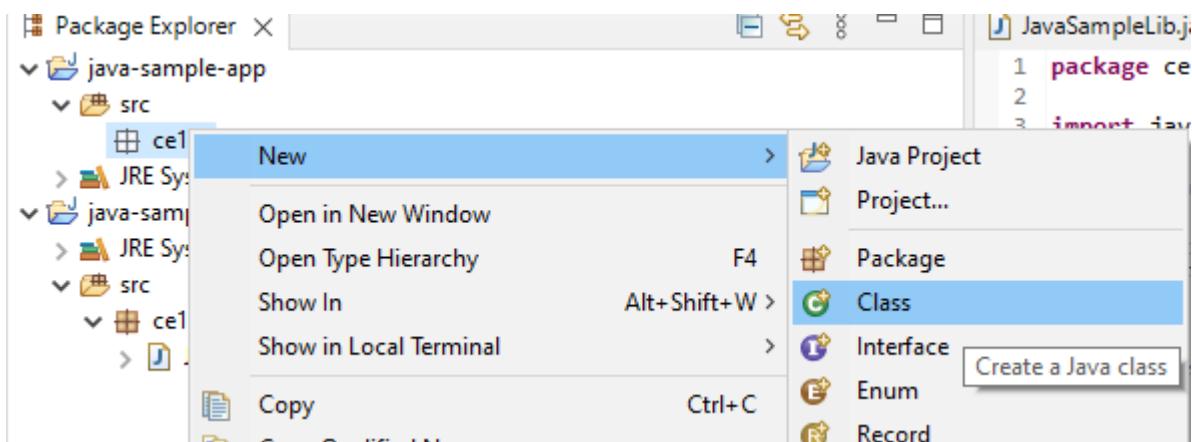


lets create a package

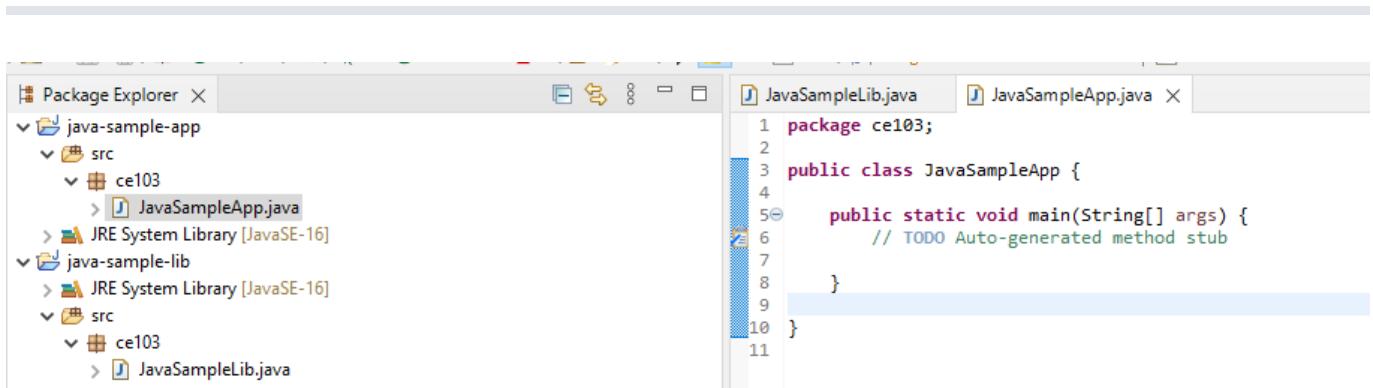
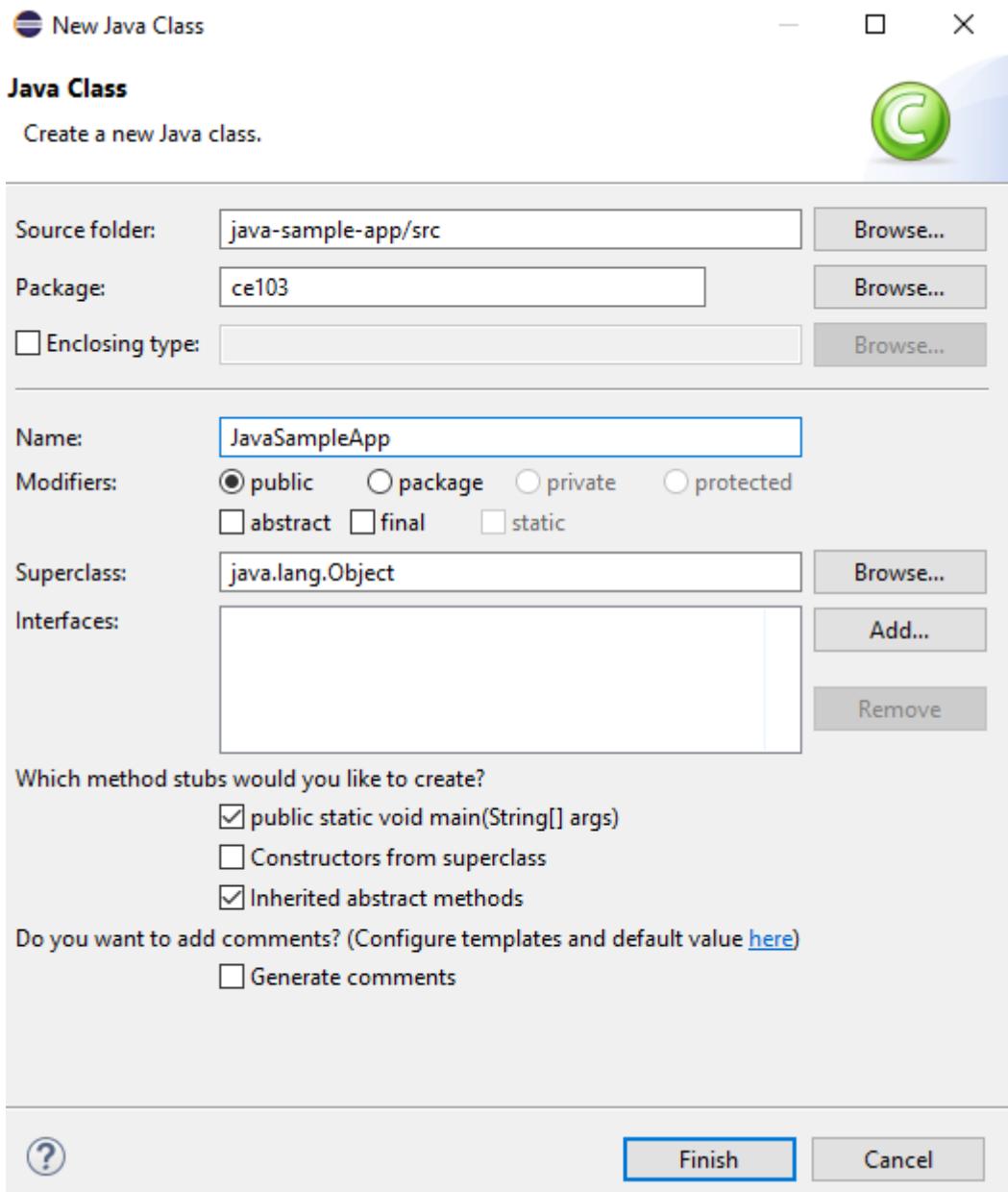




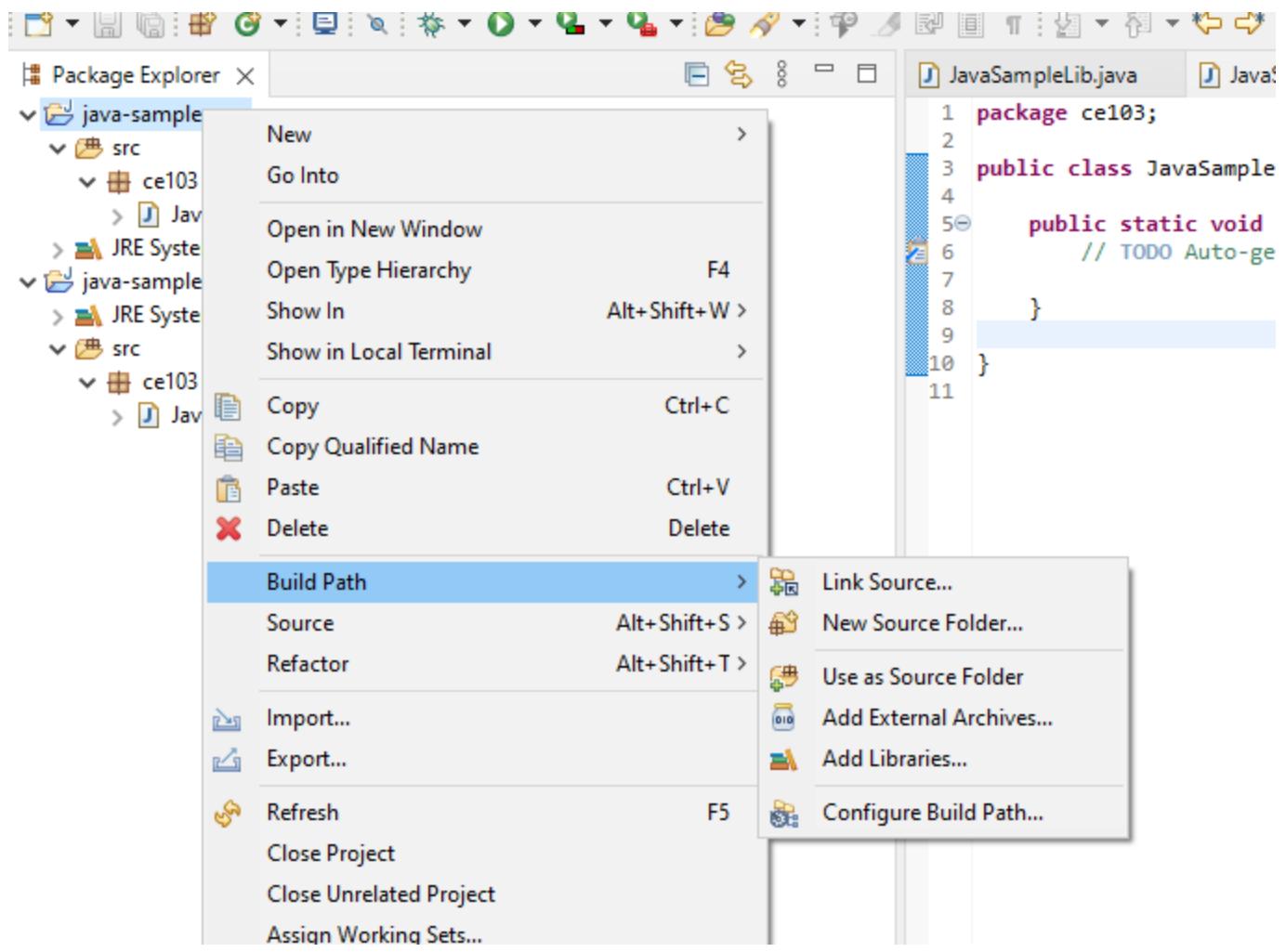
and lets create a main class for our application



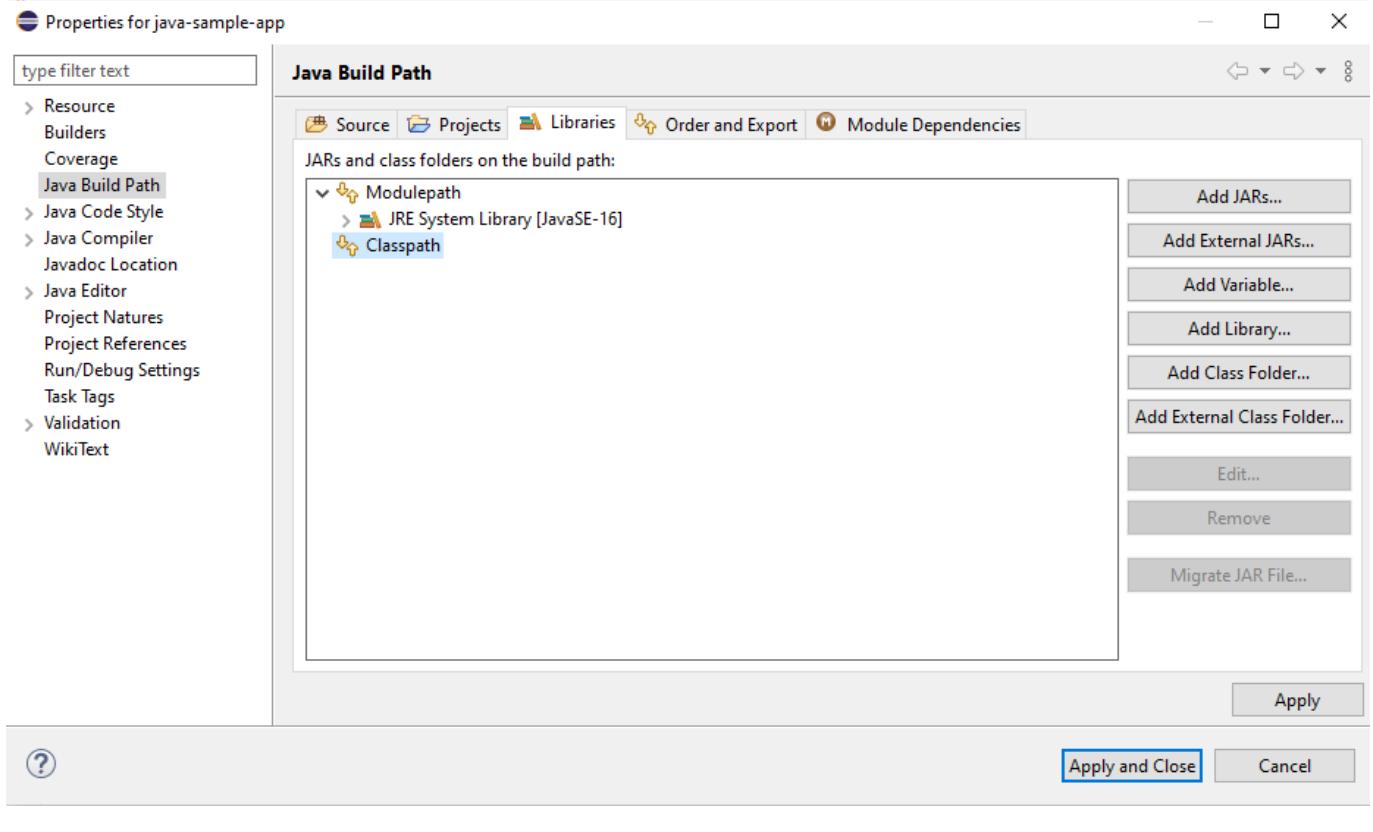
check create main function



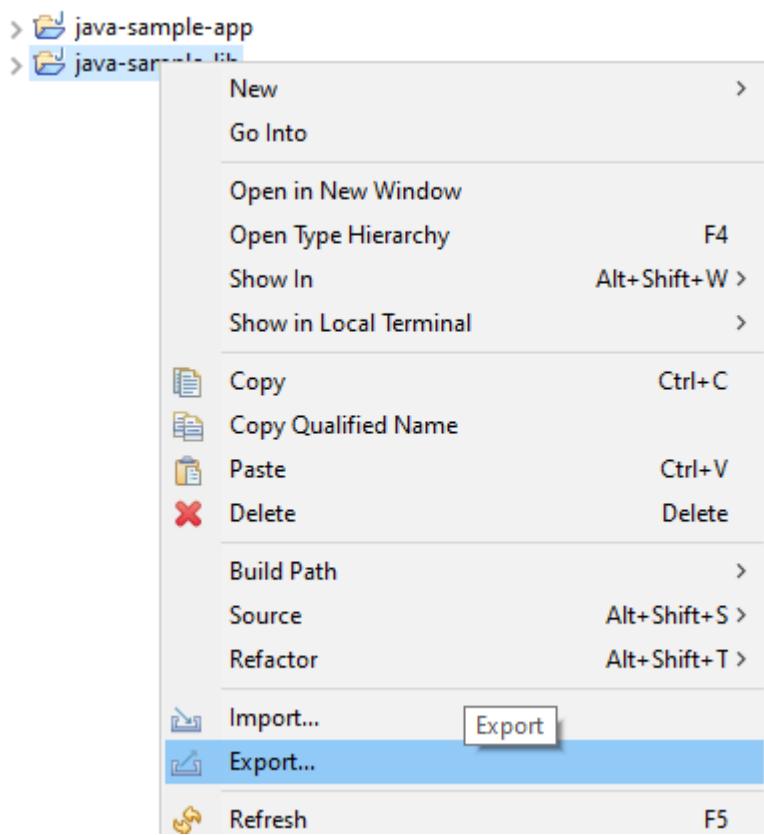
right click to project and add reference



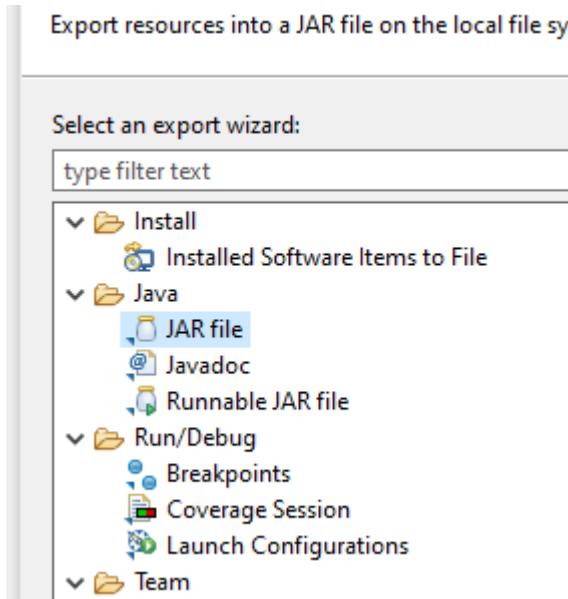
you can enter same configurations from project properties



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



Select JAR file



we configured output as

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

JAR Export

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

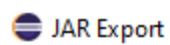


< Back

Next >

Finish

Cancel



JAR Packaging Options

Define the options for the JAR export.



Select options for handling problems:

- Export class files with compile errors
- Export class files with compile warnings

Create source folder structure

Build projects if not built automatically

Save the description of this JAR in the workspace

Description file:

[Browse...](#)

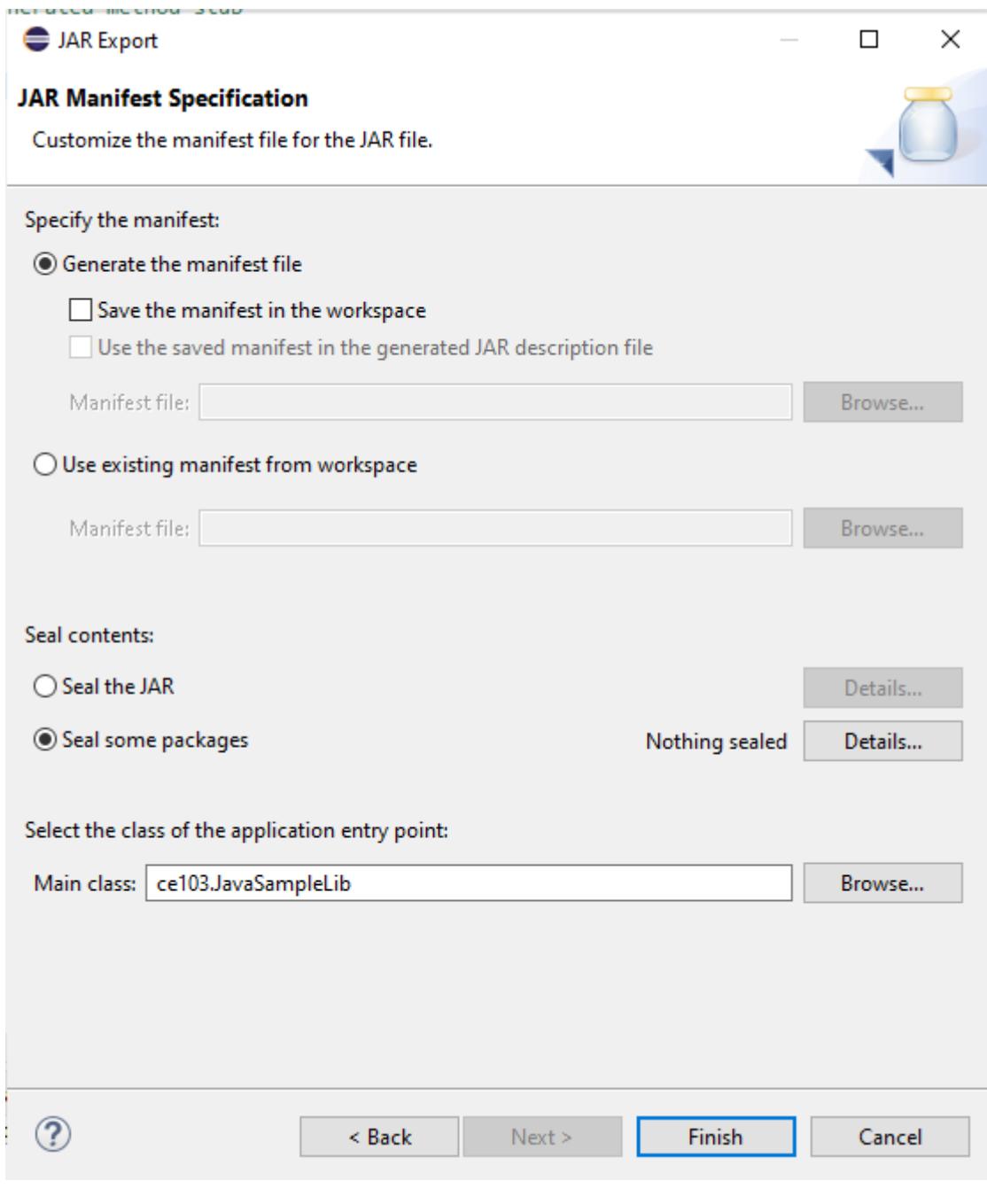


< Back

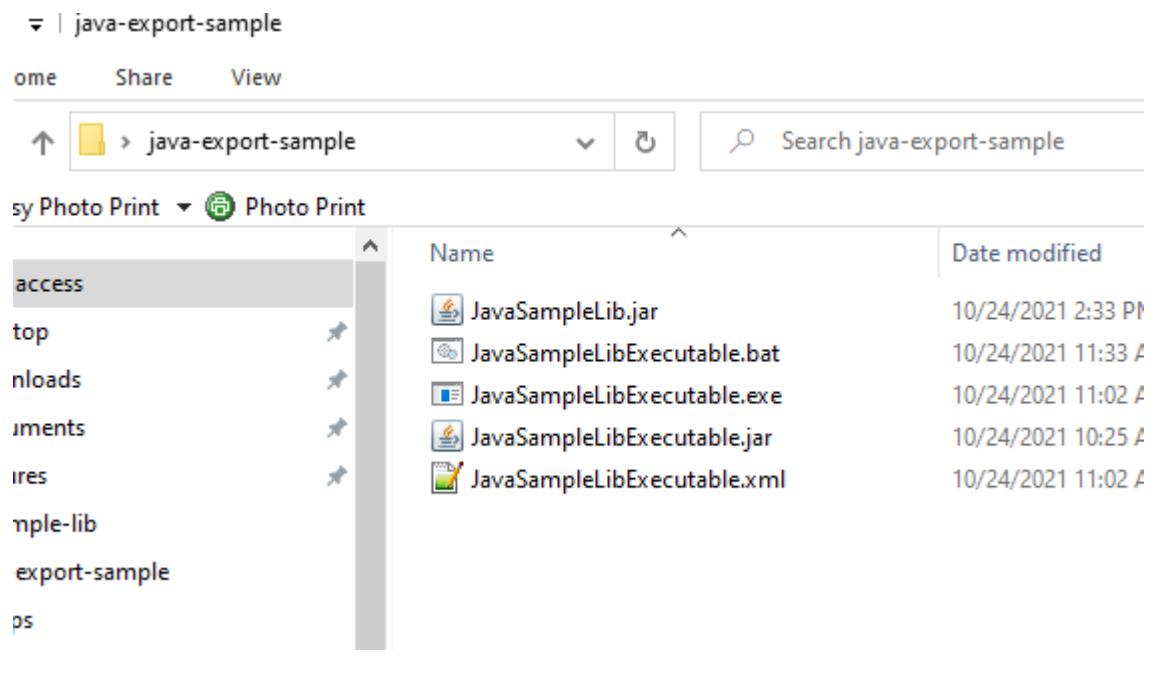
Next >

[Finish](#)

Cancel

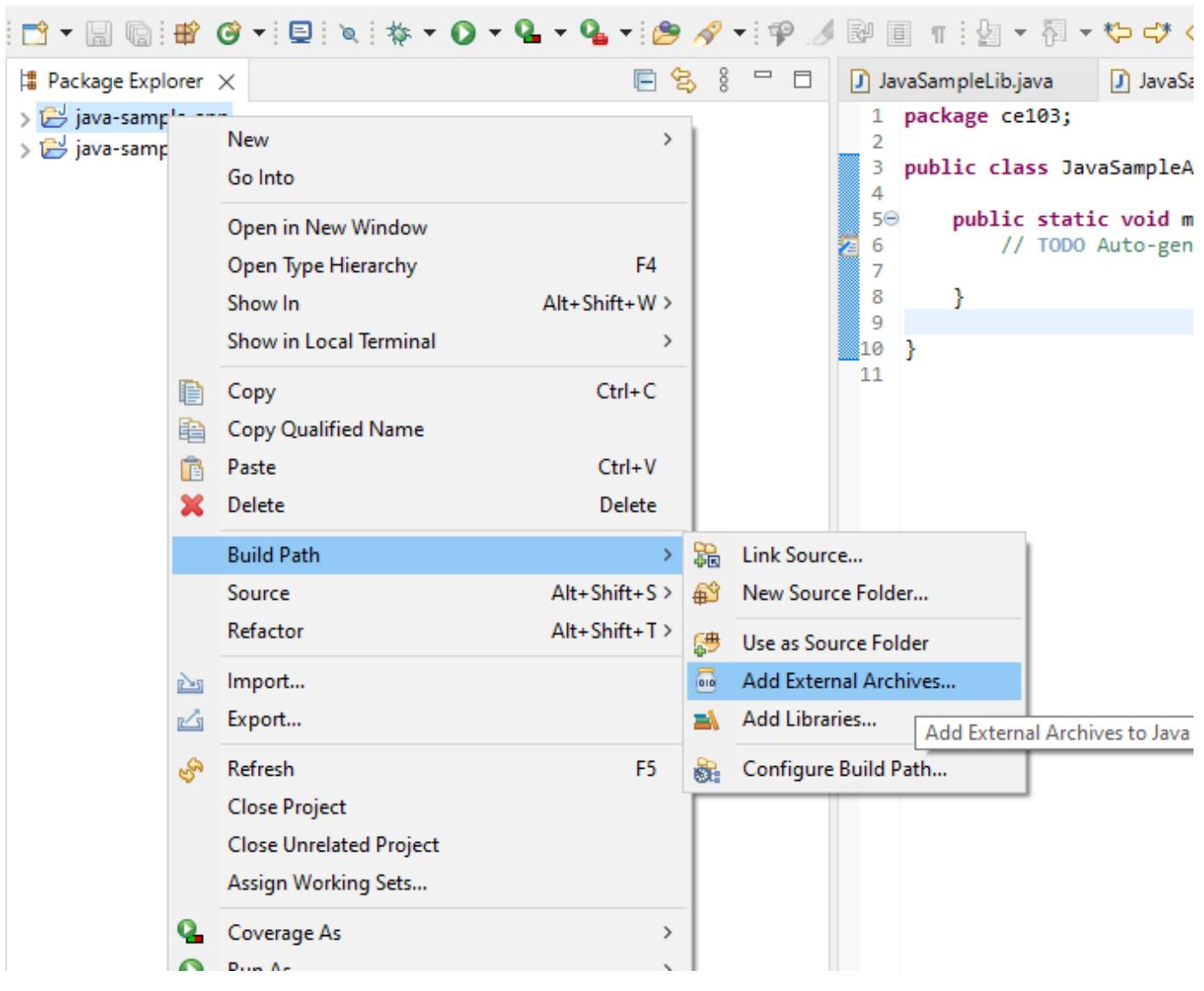


In the same export folder now we have JavaSampleLib.jar

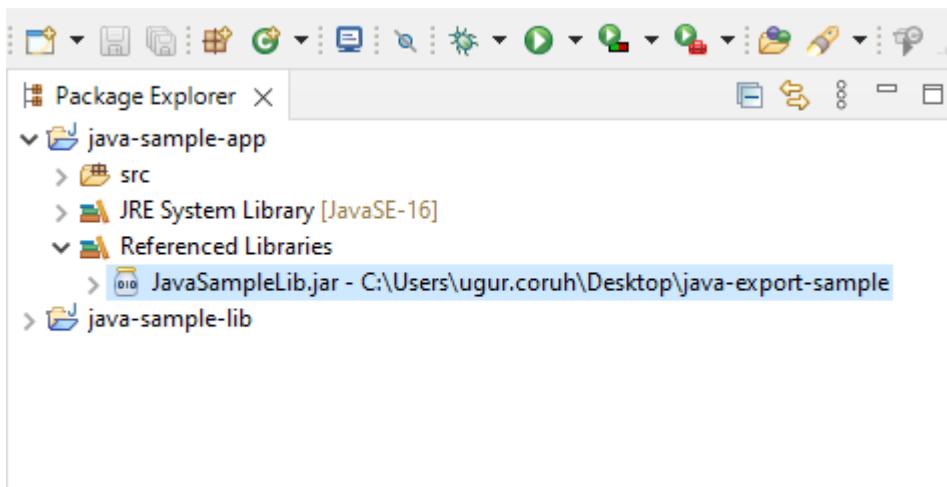


return back to java-sample-app and then add this jar file to our project

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 {

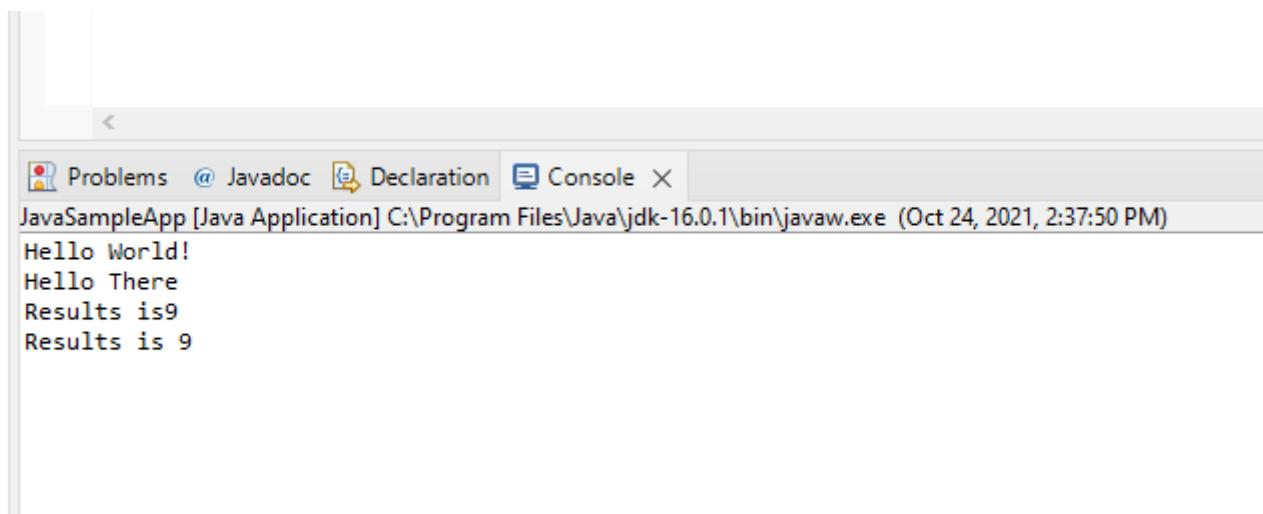
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        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();
        }
    }
}
```

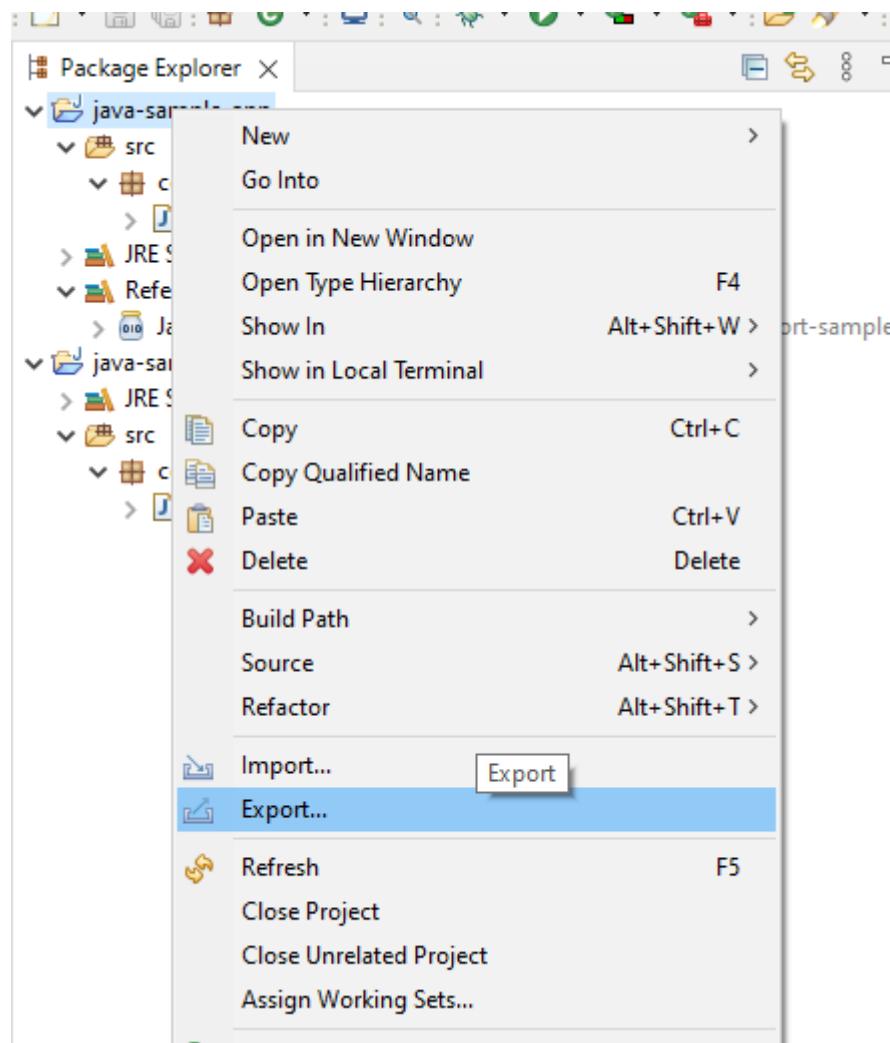
When we run application we will see similar output



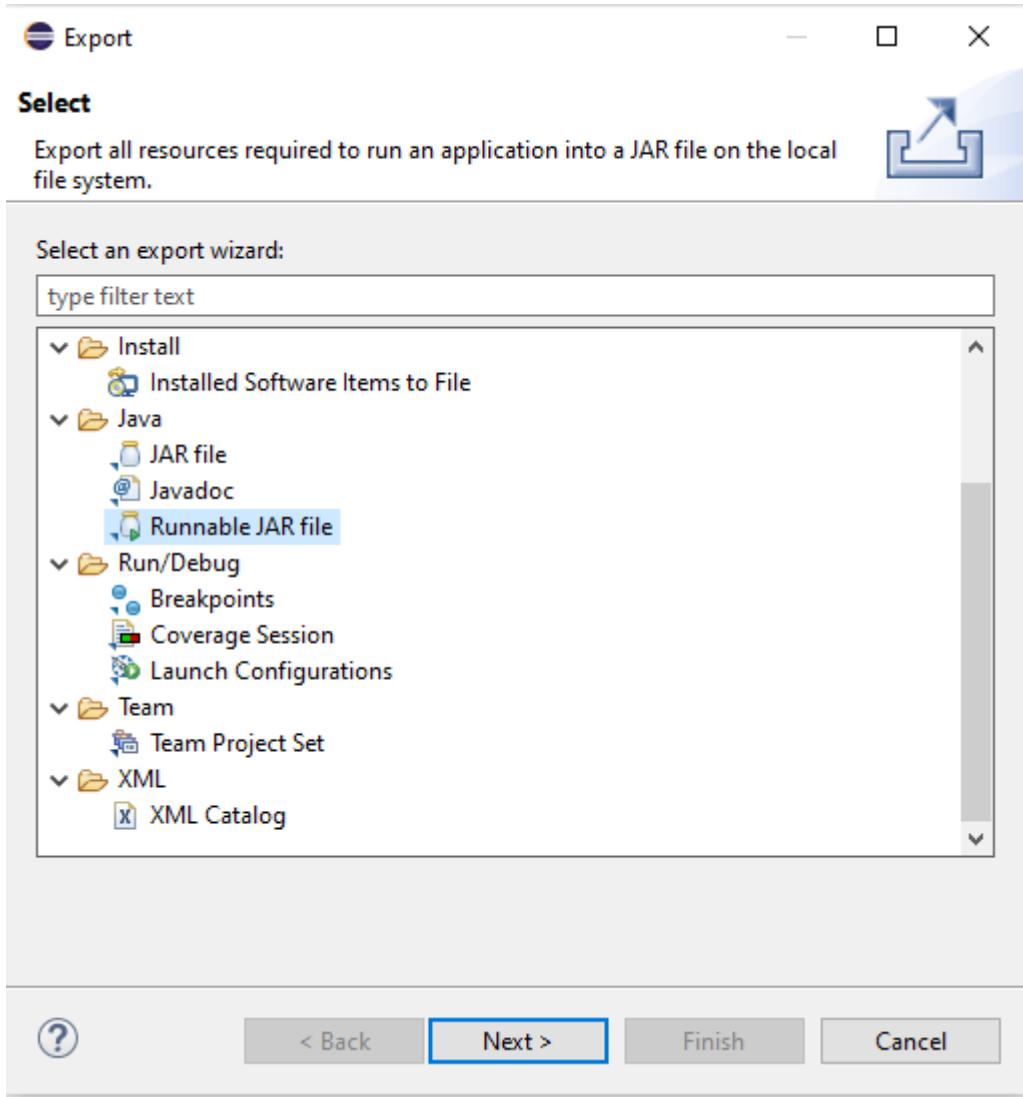
The screenshot shows the Eclipse IDE interface with the Java Sample Application running. The Console tab is active, displaying the following output:

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

Lets export this application with its dependent library

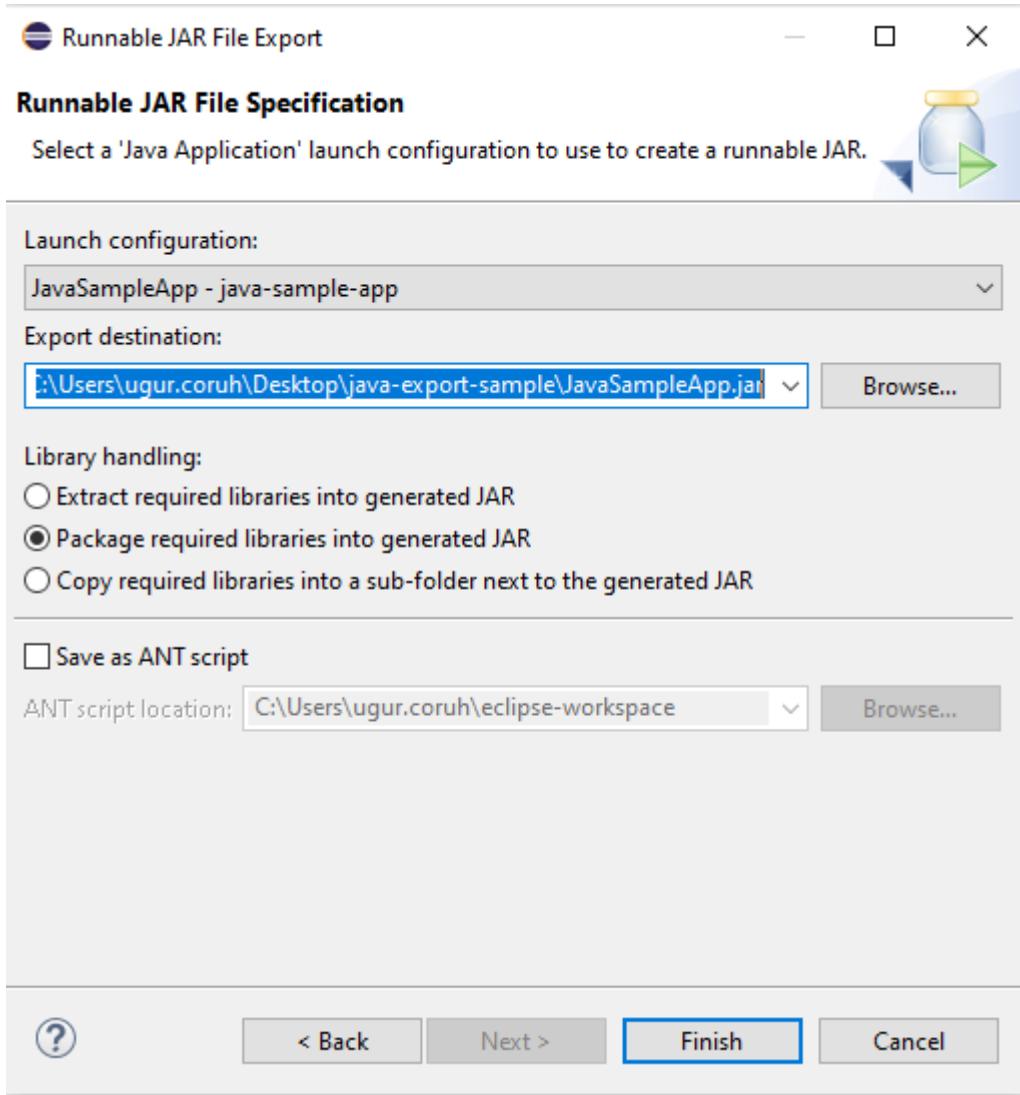


Select runnable jar



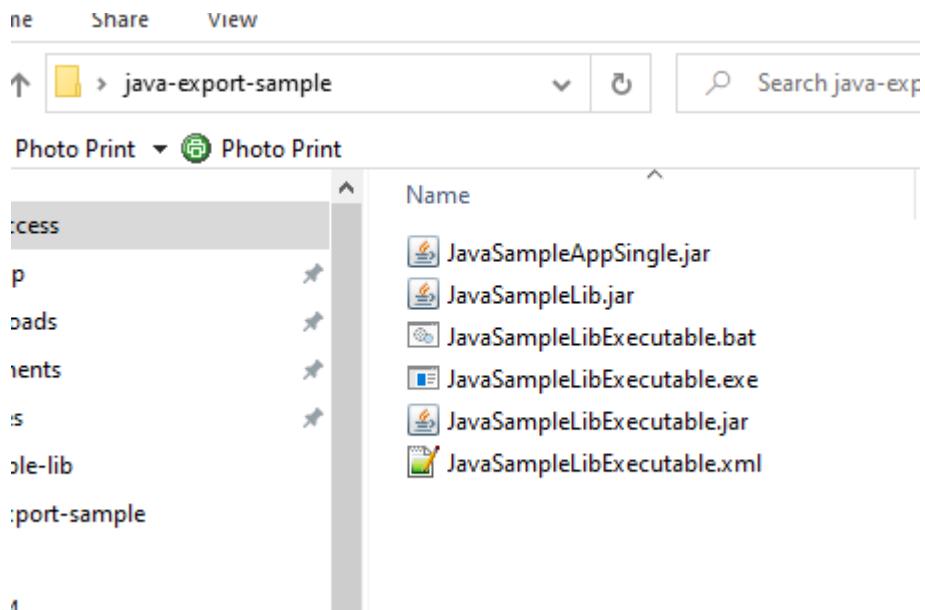
Set Launch configuration and Export destination

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

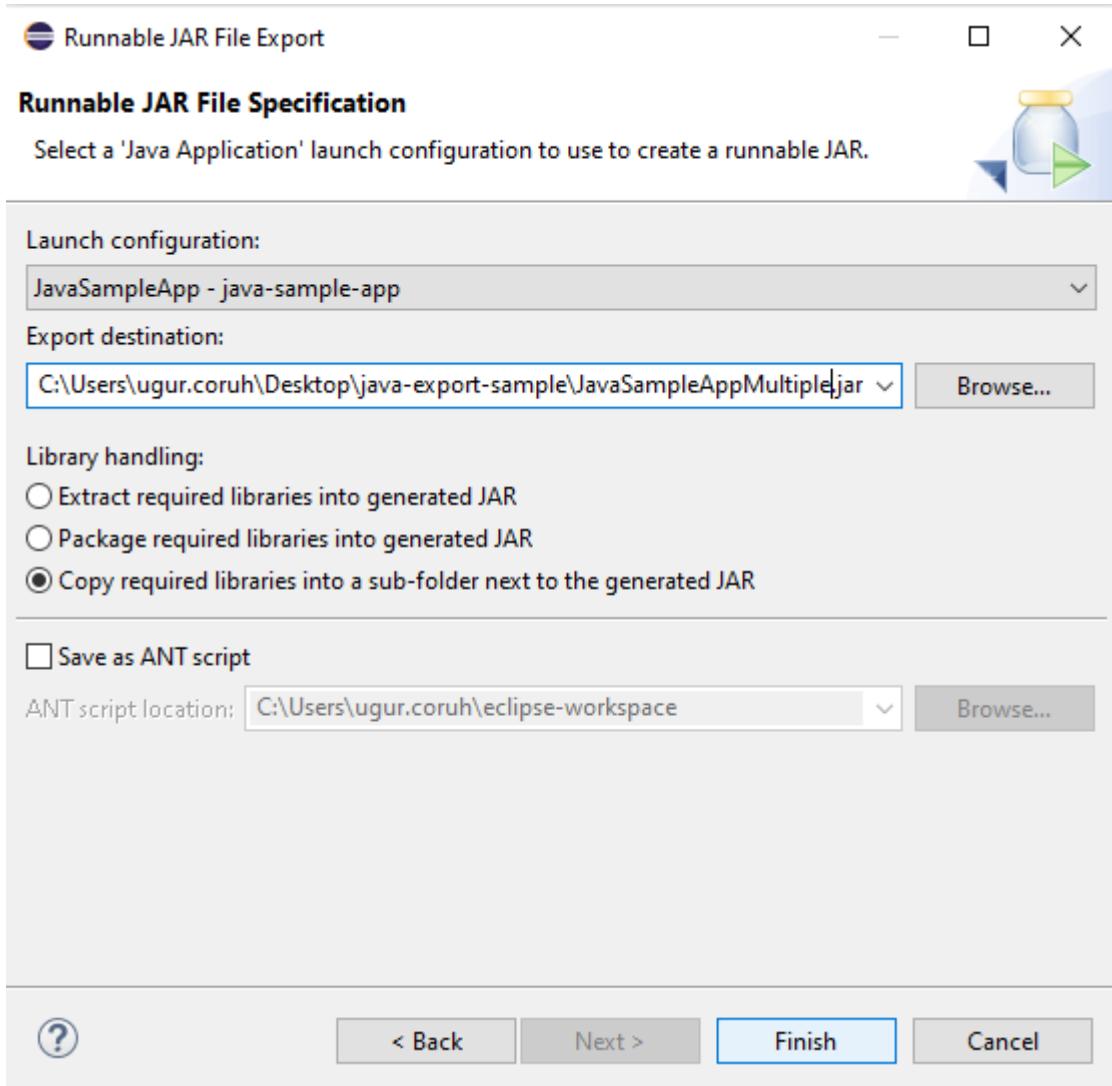


and we can run with command line

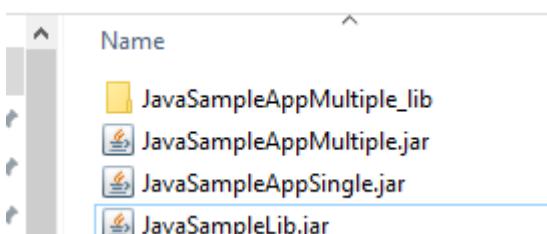
```
C:\Users\ugur.coruh\Desktop\java-export-sample>java -jar JavaSampleAppSingle.jar
Hello World!
Hello There
Results is9
Results is 9
```

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

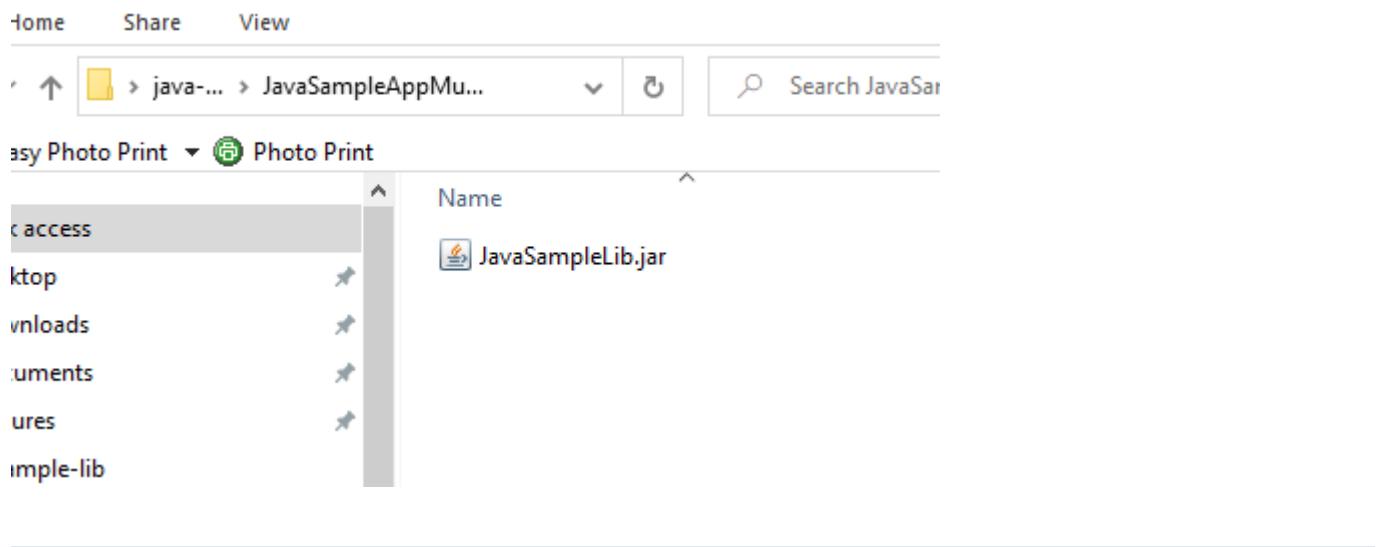
```
C:\Users\ugur.coruh\Desktop\java-export-sample>java -jar JavaSampleAppMultiple.jar
```



now we have a folder that contains our libraries referenced



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>
```

Program Testing

Unit Test Development

C Unit Tests

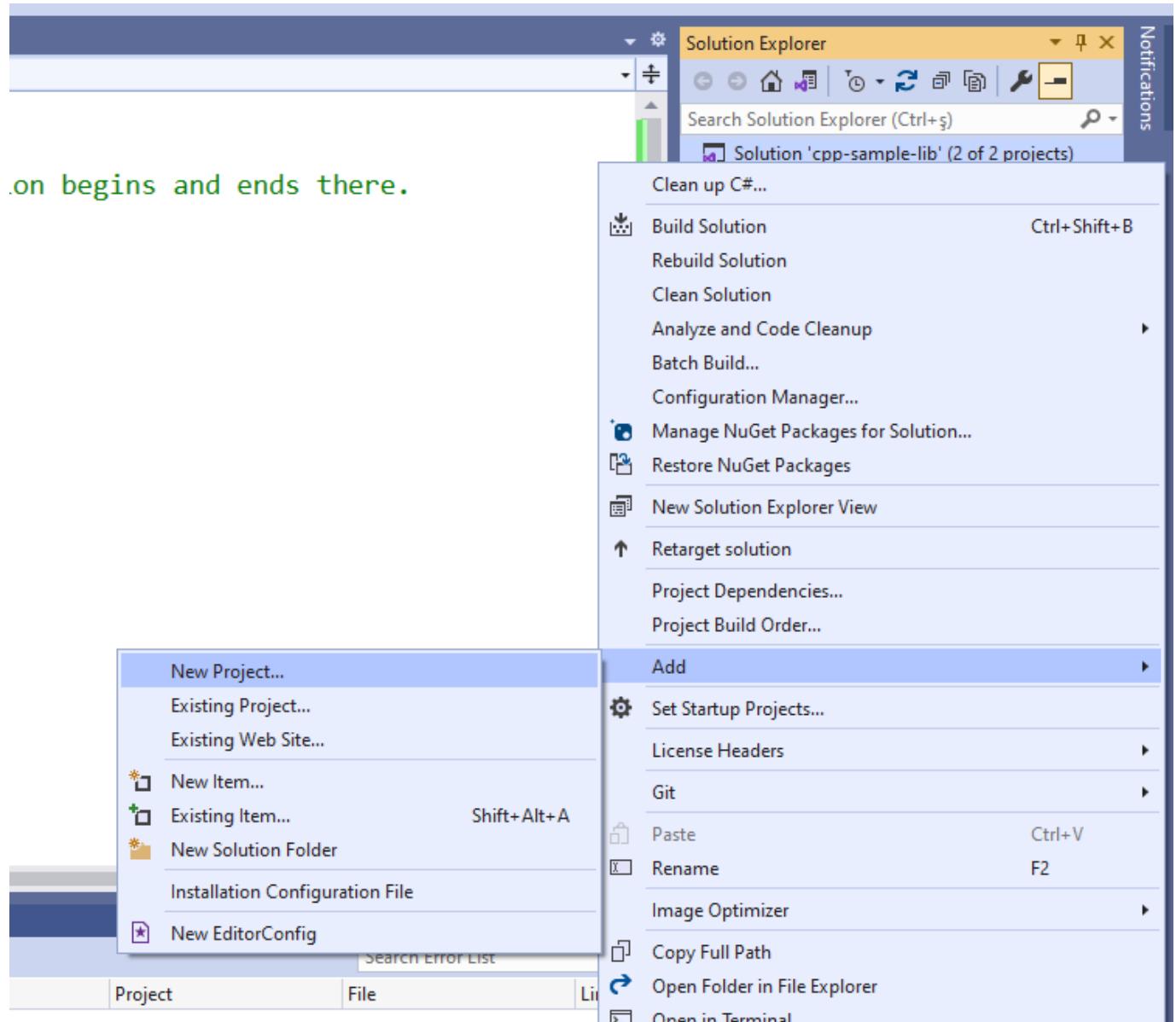
Visual Studio Community Edition

C++ Unit Tests

Visual Studio Community Edition

[C/C++ için birim testleri yazma - Visual Studio \(Windows\) | Microsoft Docs](#)

Use cpp-sample-lib project and add



select Native Unit Test

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

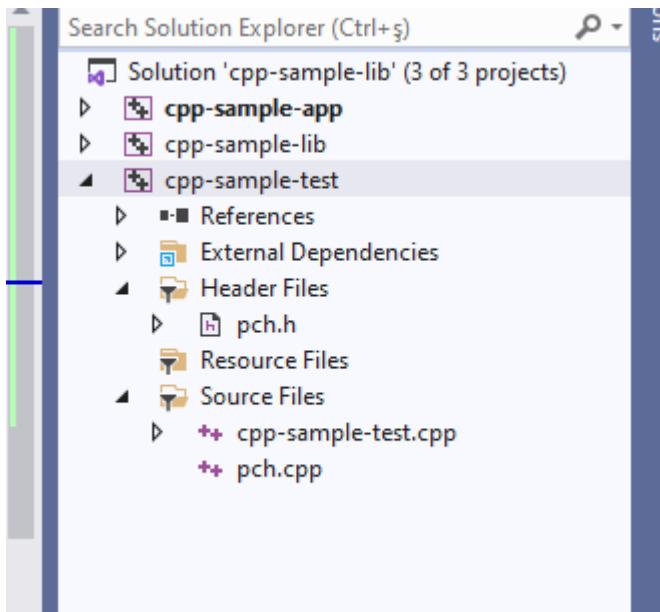
Configure your new project

Native Unit Test Project [C++](#) [Windows](#) [Test](#)

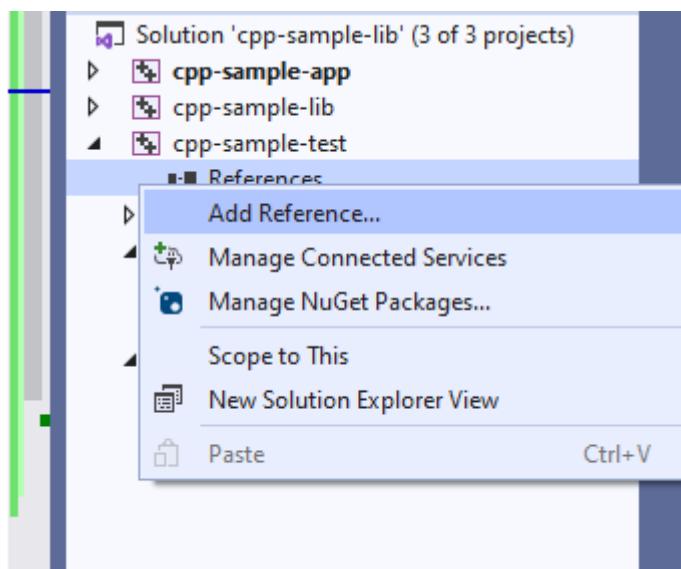
Project name

Location
 [...](#)

you will have cpp-sample-test project

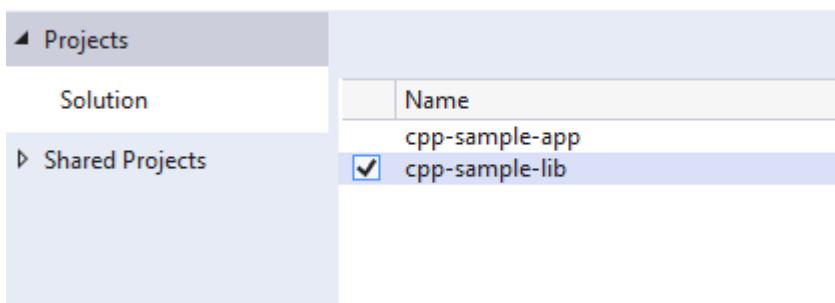


add library project from references



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. The top navigation bar includes tabs for 'cpp-sample-test.cpp', 'cpp-sample-app.cpp', 'samplelib.h', 'pch.h', 'pch.cpp', and 'cpp-sample-lib.cpp'. The current file is 'cpp-sample-test.cpp'. The code editor displays the following C++ code:

```
8 namespace cppsampletest
9 {
10     TEST_CLASS(cppsampletest)
11     {
12         public:
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 'Test Explorer' window at the bottom shows the following 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>

C# Unit Tests

Visual Studio Community Edition (MSTestV2+.Net)

Install extension fine code coverage

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

Create a .Net Framework Library

The screenshot shows the Visual Studio template browser interface. At the top, there is a search bar labeled "Search for templates (Alt+S)" and a "Clear all" button. Below the search bar, there are three dropdown menus: "C#", "Windows", and "Library". The "Library" menu is currently selected, indicated by a blue border around its dropdown arrow. Under the "Library" heading, there are four project templates listed:

- 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

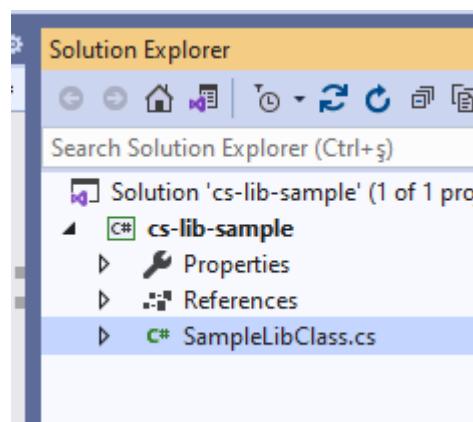
cs-lib-sample

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

0 references

```
public class SampleLibClass
{
    0 references
    public static string
    {
        string result ...

        if (!String.I
        {
            result =
        }
        else
        {
            result =
    
```

AttackFlow Sanitization

Clean up C#...

Quick Actions and Refactorings... Ctrl+.

Rename... Ctrl+R, Ctrl+R

Remove and Sort Usings Ctrl+R, Ctrl+G

View Code F7

Peek Definition

Go To Definition

Go To Base Alt+Home

Go To Implementation

Find All References

View Call Hierarchy Ctrl+K, Ctrl+T

Create Unit Tests

Open code in ILSpy

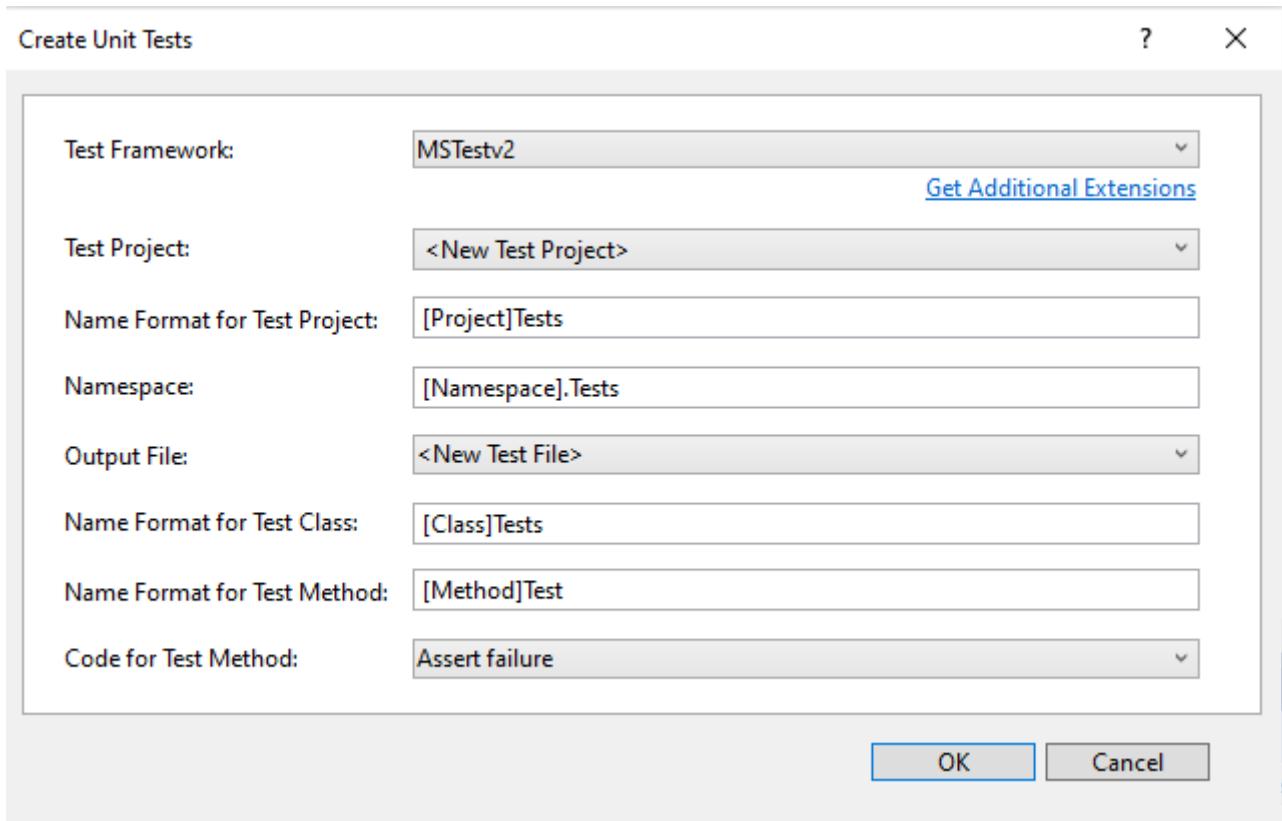
Breakpoint

Run To Cursor Ctrl+F10

Execute in Interactive

Snippet

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 should work");
        }

        [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();

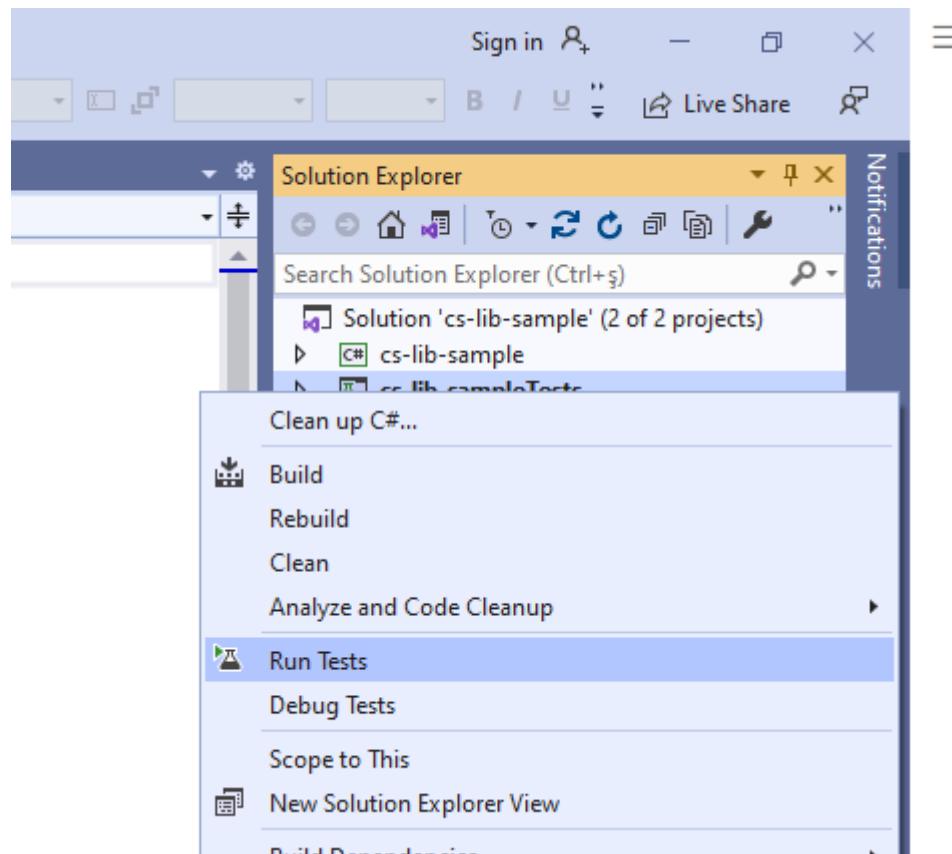
            Assert.AreEqual(20, sampleLib.multiply(4, 5), "Regular multiplication
should work");
        }
    }
}
```

```
    }

}

}
```

Run tests



you will code coverage and entered or passed branches

The screenshot shows a Visual Studio code editor window with a C# file named `SampleLibClass.cs`. The code defines a static method `sayHelloTo` that returns a greeting based on the input name. The code editor has syntax highlighting and a vertical ruler on the left.

```
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    }

```

Below the code editor is a 'Fine Code Coverage' tool window. It displays a table of coverage statistics for different projects and files. The table includes columns for Name, Covered, Uncovered, Coverable, Total, and Line coverage percentage, along with a corresponding progress bar.

Name	Covered	Uncovered	Coverable	Total	Line coverage
- 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%

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).

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).

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 I\Lectures\ce11

...

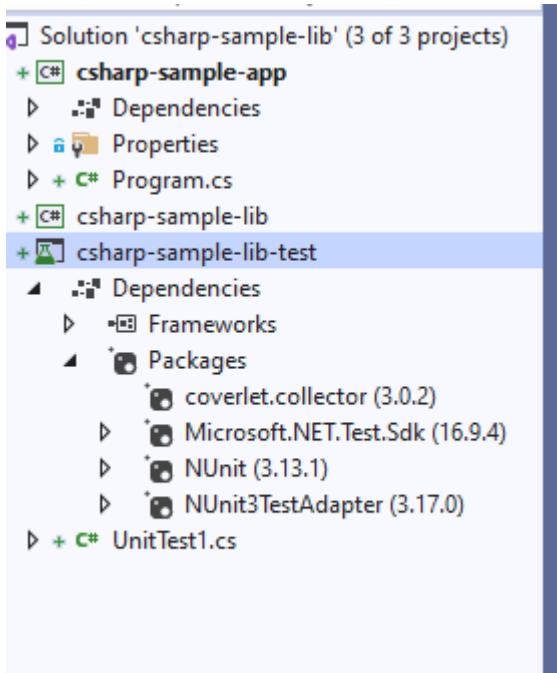
Additional information

NUnit Test Project C# Linux macOS Windows Desktop Test Web

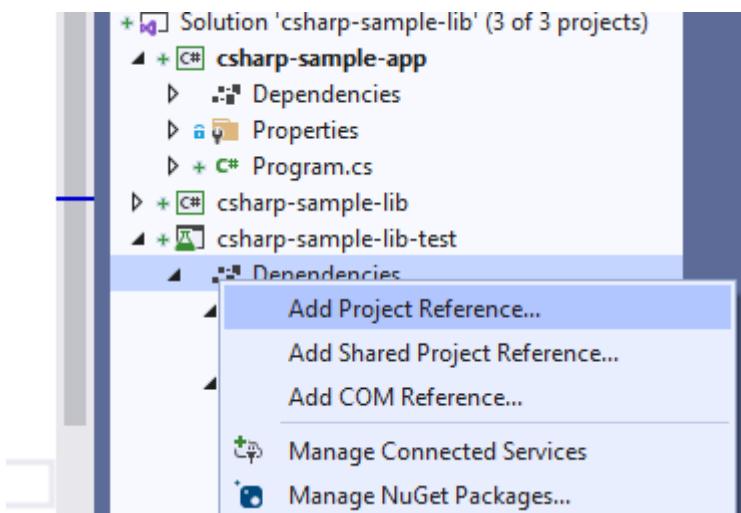
Target Framework 

.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 (
	Name	Path	
Solution	csharp-sample-app	E:\UgurCoruh\RTEU\L...	
Shared Projects	<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 should work");
        }

        [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");
        }

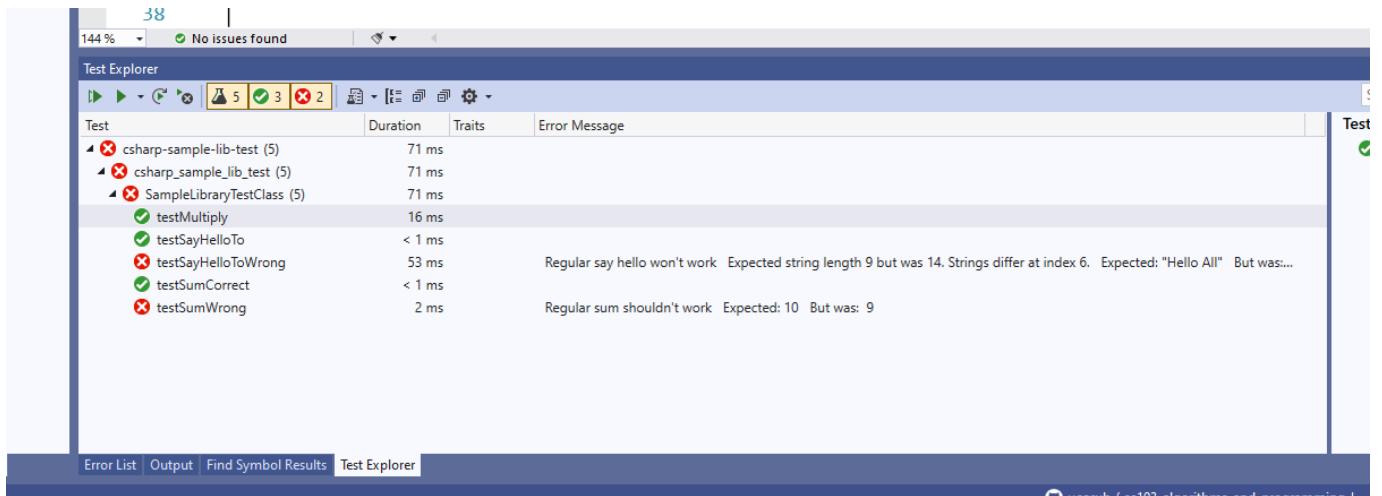
        [Test]
        public void testMultiply()
        {
            Assert.AreEqual(20, sampleLib.multiply(4, 5), "Regular multiplication
should work");
        }
    }
}
```

```
    }  
}
```

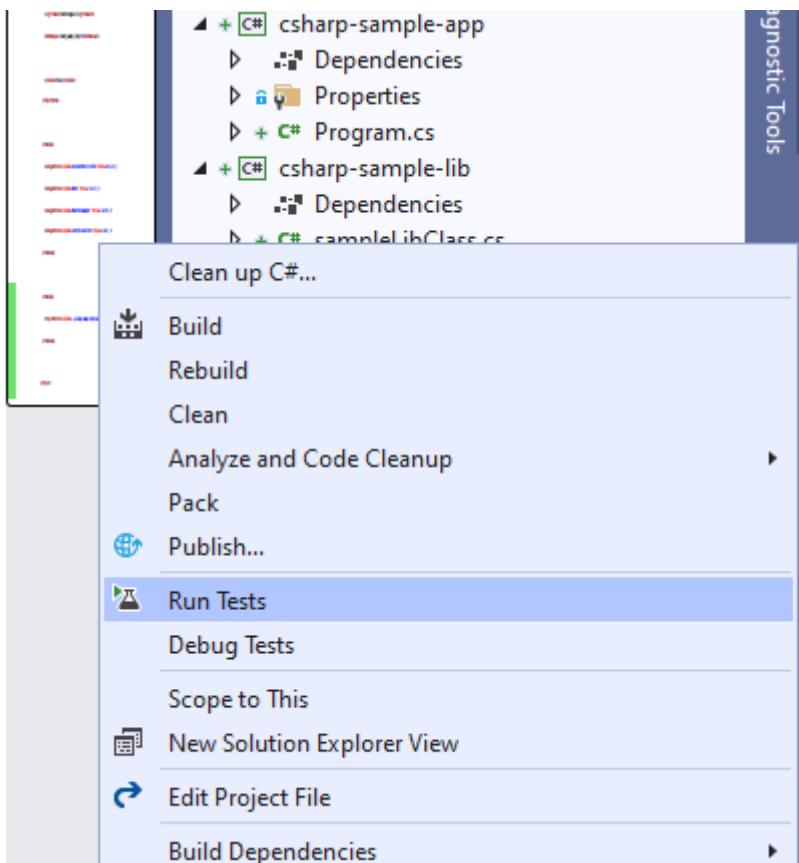
sample class library

```
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

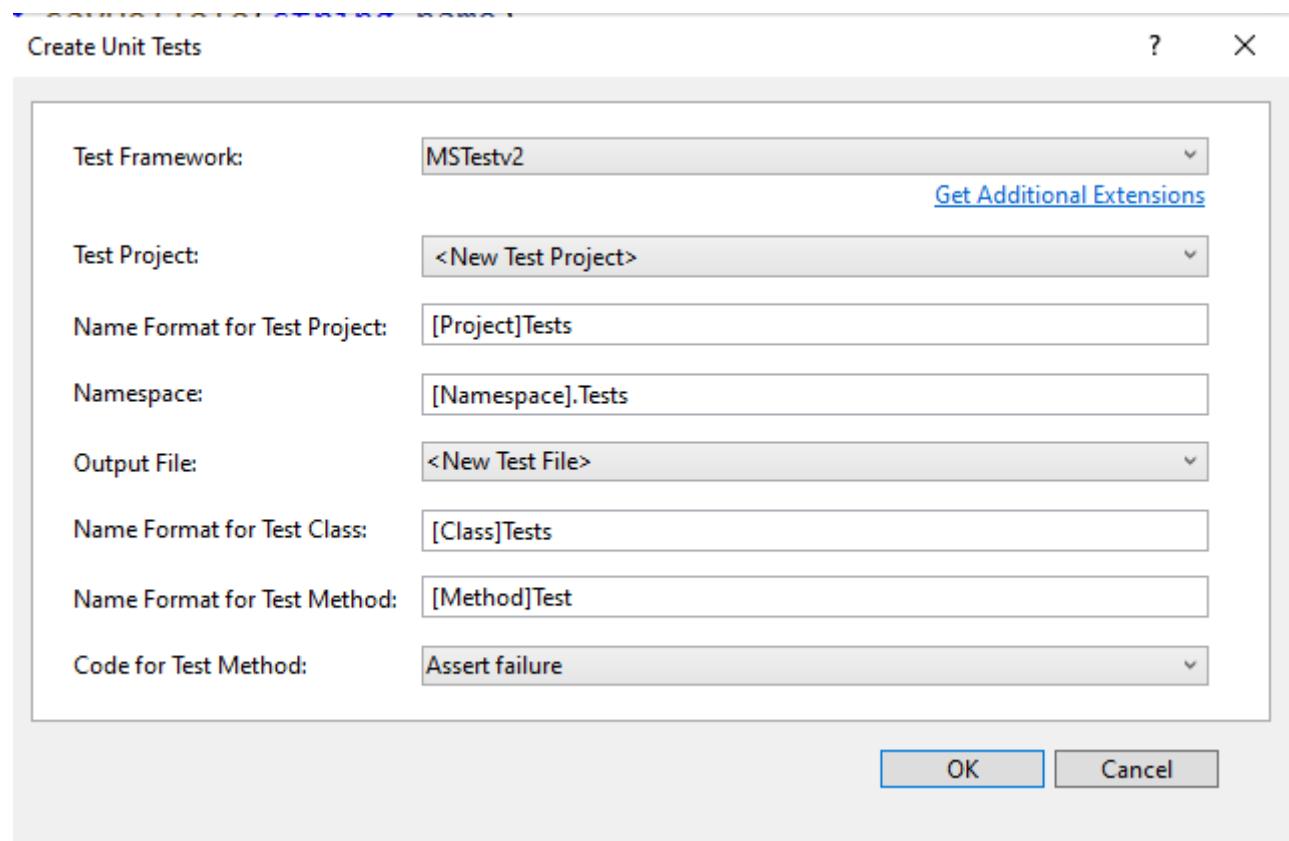
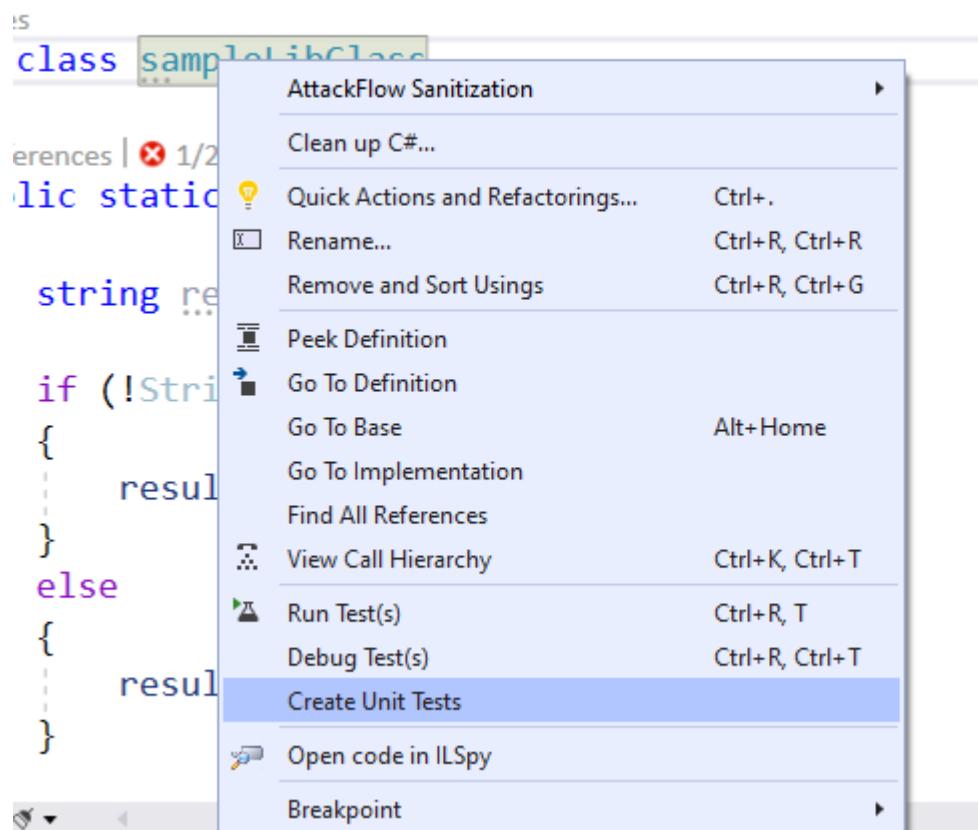


or you can run from project



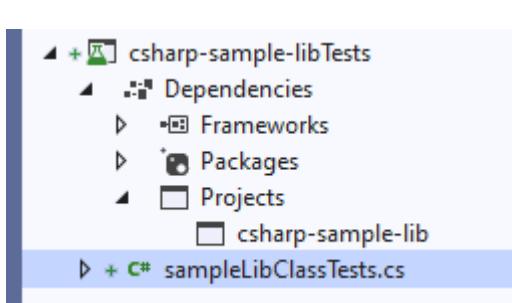
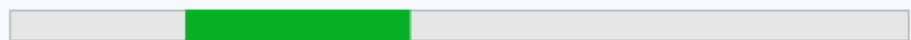
Also we can create unit test from library class,

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

The screenshot shows the Visual Studio IDE interface. The top menu bar includes Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and Search. The title bar says "csharp-sample-lib". The main window displays the code for "sampleLibClass.cs" with syntax highlighting and code navigation features. To the right is the Solution Explorer pane, which lists the solution structure: "csharp-sample-lib" (3 of 3 projects), "csharp-sample-app", "csharp-sample-lib", and "csharp-sample-lib-test". Below the Solution Explorer is the Properties pane. A status bar at the bottom shows "144% No issues found".

Fine Code Coverage

Coverage [Summary | Risk Hotspots] Rate & Review Log Issue/Suggestion Buy me a coffee Filter: []

Collapse all | Expand all

Name	Covered	Uncovered	Coverable	Total	Line coverage
csharp-sample-lib	17	3	20	37	85%
sampleLibClass	17	3	20	37	85%
csharp-sample-lib-test	16	2	18	47	88.8%
SampleLibraryTestClass	16	2	18	47	88.8%

Fine Code Coverage | Error List | Output | Find Symbol Results | Test Explorer

The screenshot shows the Test Explorer window in Visual Studio. The title bar says "csharp-sample-lib-test" and "197 ms". The main area displays a list of tests under the group "csharp-sample-lib-test". The tests include:

- csharp-sample-lib-test (5) Duration: 197 ms
- SampleLibraryTestClass (5)
 - testMultiply (54 ms)
 - testSayHelloTo (< 1 ms)
 - testSayHelloToWrong (136 ms) Description: Regular say hello won't work. Expected string length 9 but was 14. Strings differ at index 6. Expected: "Hello All" But was:...
 - testSumCorrect (< 1 ms)
 - testSumWrong (7 ms) Description: Regular sum shouldn't work. Expected: 10 But was: 9

Group Summary

csharp-sample-lib-test
Tests in group: 5
Total Duration: 197 ms

Outcomes

3 Passed
2 Failed

Fine Code Coverage | Error List | Output | Find Symbol Results | Test Explorer

Inline code highlight is part of enterprise visual studio edition

Analyzing code coverage in Visual Studio - DEV Community

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

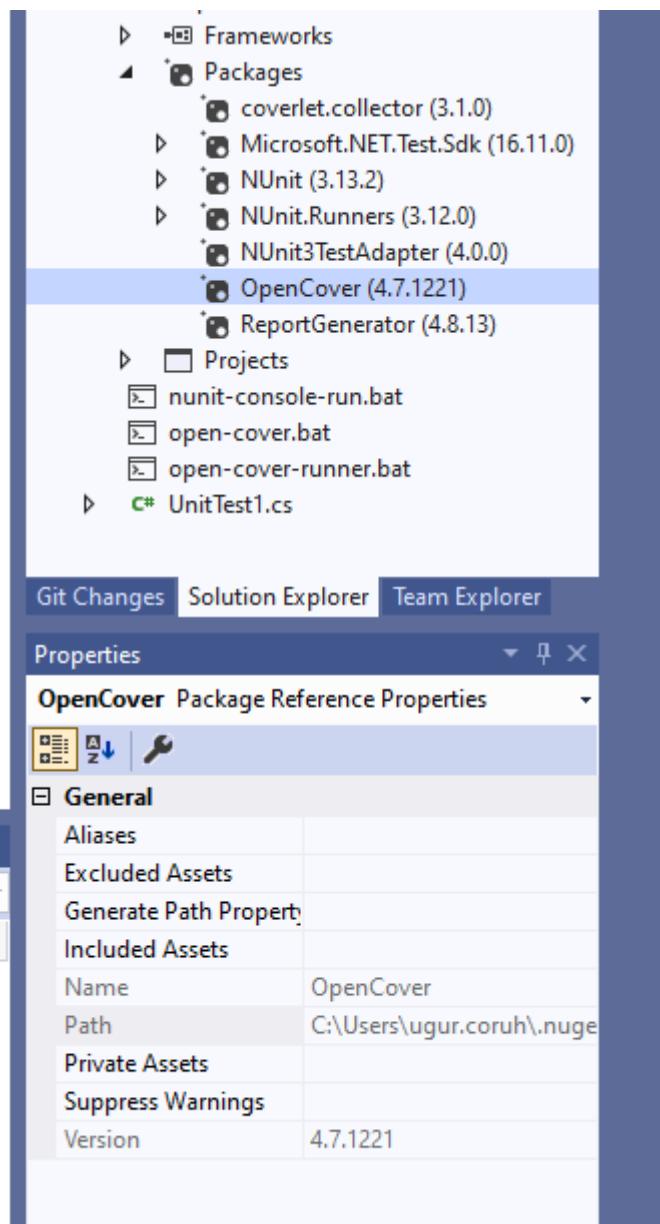
c# - The NUnit 3 driver encountered an error while executing reflected code
(NUnit.Engine.NUnitEngineException) - Stack Overflow

Follow the instructions on the link

[CMD OpenCover · sukhoi1/Useful-Notes Wiki · GitHub](#)

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



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" -targetdir:""%dllpath%\coverage"
pause
```

but nunit3-console.exe gives error

```
C:\Users\ugur.coruh\Desktop\csharp-sample-lib>C:\Users\ugur.coruh\.nuget\packages\opencover\4.7.1221\tools\OpenCover.Console.exe" -targetargs:"C:\Users\ugur.coruh\Desktop\csharp-sample-lib\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\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\coverReport.xml" -targetdir:"C:\Users\ugur.coruh\Desktop\csharp-sample-lib\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\coverage"
Executing: C:\Users\ugur.coruh\.nuget\packages\nunit.consolerunner\3.12.0\tools\nunit3-console.exe
NUnit Console Runner 3.12.0 (.NET 2.0)
Copyright (c) 2018 Charlie Poole. All rights reserved.
Monday, October 25, 2021 2:01:06 AM

Routine Environment
OS Version: Microsoft Windows NT 6.2.9200.0
Runtime: .NET Framework CLR v4.0.30319.42000
Test File: C:\Users\ugur.coruh\Desktop\csharp-sample-lib\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\csharp-sample-lib-test.dll

Errors, Failures and Warnings
() Error :
NUnit.Engine.NUnitEngineException : The NUnit 3 driver encountered an error while executing reflected code.
->NUnitEngineException : Unable to cast transparent proxy to type 'System.Web.UI.CallbackEventHandler'.
The NUnit 3 driver encountered an error while executing reflected code.

Server stack trace:
at NUnit.Engine.Drivers.NUnit3FrameworkDriver.CreateObject(String typeName, Object[] args)
at NUnit.Engine.Drivers.NUnit3FrameworkDriver.LoadString(testAssemblyPath, IDictionary`2 settings)
at NUnit.Engine.Runners.DirectTestRunner.LoadDriver(IFrameworkDriver driver, String testFile, TestPackage subPackage)
at NUnit.Engine.Runners.TestDomain.Runner.LoadPackage()
at NUnit.Engine.Runners.DirectTestRunner.EnsurePackage(IHandle<Object>)
at NUnit.Engine.Runners.AbstractTestRunner.Run(IFilteredEventListener listener, TestFilter filter)
at NUnit.Engine.Runners.AbstractTestRunner.Run(IFilteredEventListener listener, TestFilter filter)
at System.Runtime.Remoting.Messaging.StackBuilderSink.PrivateProcessMessage(IntPtr md, Object[] args, Object server, Object[]& outArgs)
at System.Runtime.Remoting.Messaging.StackBuilderSink.PrivateProcessMessage(HuntimeMethodHandle md, Object[] args, Object server, Object[]& outArgs)
at System.Runtime.Remoting.Messaging.StackBuilderSink.OnProcessMessage(Message msg)

Exception thrown: 'System.Runtime.Remoting.Proxies.RealProxy.HandleReturnMessage(Message reqMsg, IMessage retMsg)' occurred in System.Runtime.Remoting.dll
at System.Runtime.Remoting.Proxies.RealProxy.PrivateInvoke(MessageData& msgData, Int32 type)
at System.Runtime.Remoting.IUnknownInternal.FastInvoke(IntPtr hUnknown, Listener listener, Filter filter)
at NUnit.Engine.Runners.ProcessRunner.Handle(IFilteredEventListener listener, TestFilter filter)
->InvalidOperationException : Unable to cast transparent proxy to type 'System.Web.UI.CallbackEventHandler'.
at NUnit.Framework.Api.FrameworkController.LoadTestsAction..ctor(FrameworkController controller, Object handler)

Test Run Summary
Test Count: 0 Passed: 0 Failed: 0. Warnings: 0. Inconclusive: 0. Skipped: 0
Start time: 2021-10-24 22:01:12
End time: 2021-10-24 22:01:12
Durations: 2.015 seconds

Results (nunit3) saved as TestResult.xml
Committing...
Most likely this could be for a number of reasons. The most common reasons are:
1> missing PDBs for the assemblies that match the filter please review the assembly references in your project and make sure they are valid
2> the profiler needs not be registered correctly, please refer to the Usage guide and the --register switch
3> the --filter switch is misspelled or not included, refer to the --help message and check your filters/code
4> you are targeting .NET Core and your assemblies under test were built for .NET Framework, please add the --platform switch
5> you are targeting 4.8 or .NET Core and your assemblies under test were built for .NET Standard, please add the --platform switch
2021-10-25T02:01:12: Arguments
2021-10-25T02:01:12:   -reports:C:\Users\ugur.coruh\Desktop\csharp-sample-lib\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\coverReport.xml
2021-10-25T02:01:12:   -filter:+[csharp-sample-lib]* -[*test]*
2021-10-25T02:01:13: Writing report file 'C:\Users\ugur.coruh\Desktop\csharp-sample-lib\csharp-sample-lib-test\bin\Debug\netcoreapp3.1\coverage\index.html'
2021-10-25T02:01:13: Report generation took 0.3 seconds

C:\Users\ugur.coruh\Desktop\csharp-sample-lib>pause
Press any key to continue . . .
```

```
    at NUnit.Engine.Runners.ProcessRunner.RunTests<ITestEventListener listener, TestFilter filter>
    InvalidCastException
    Unable to cast transparent proxy to type 'System.Web.UI.ICallbackEventHandler'.
    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
```

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 Rhapsody](#)

[Code coverage of .NET Core unit tests with OpenCover – Automation Rhapsody](#)

Sample OpenCover report

[Summary - Coverage Report](#)

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,

OpenCover

[Releases · OpenCover/opencover · GitHub](#)

OpenCover (Release) 4.7.1221

Latest

Merge pull request #1040 from sawilde/big/1029_chocolatey_dependency

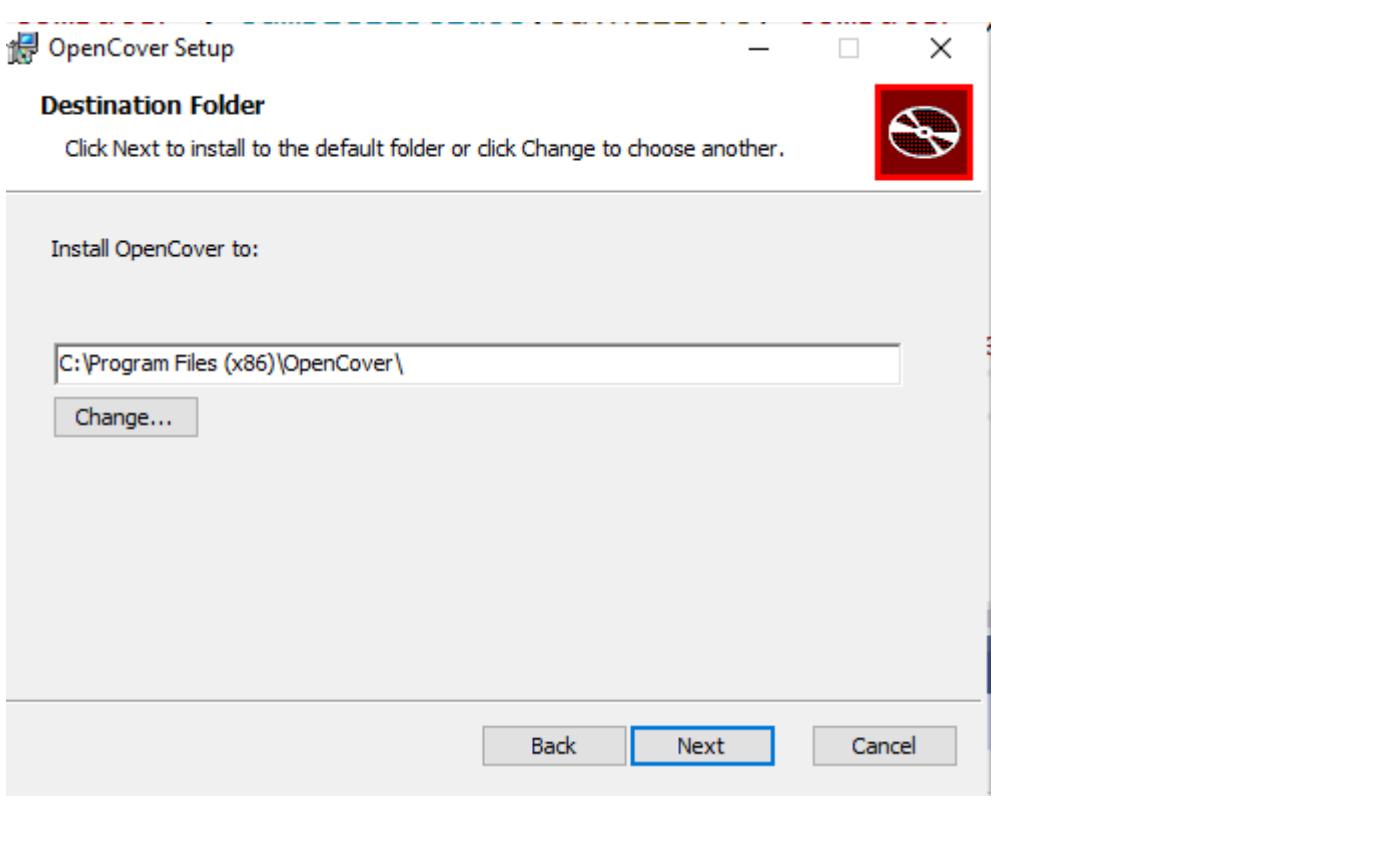
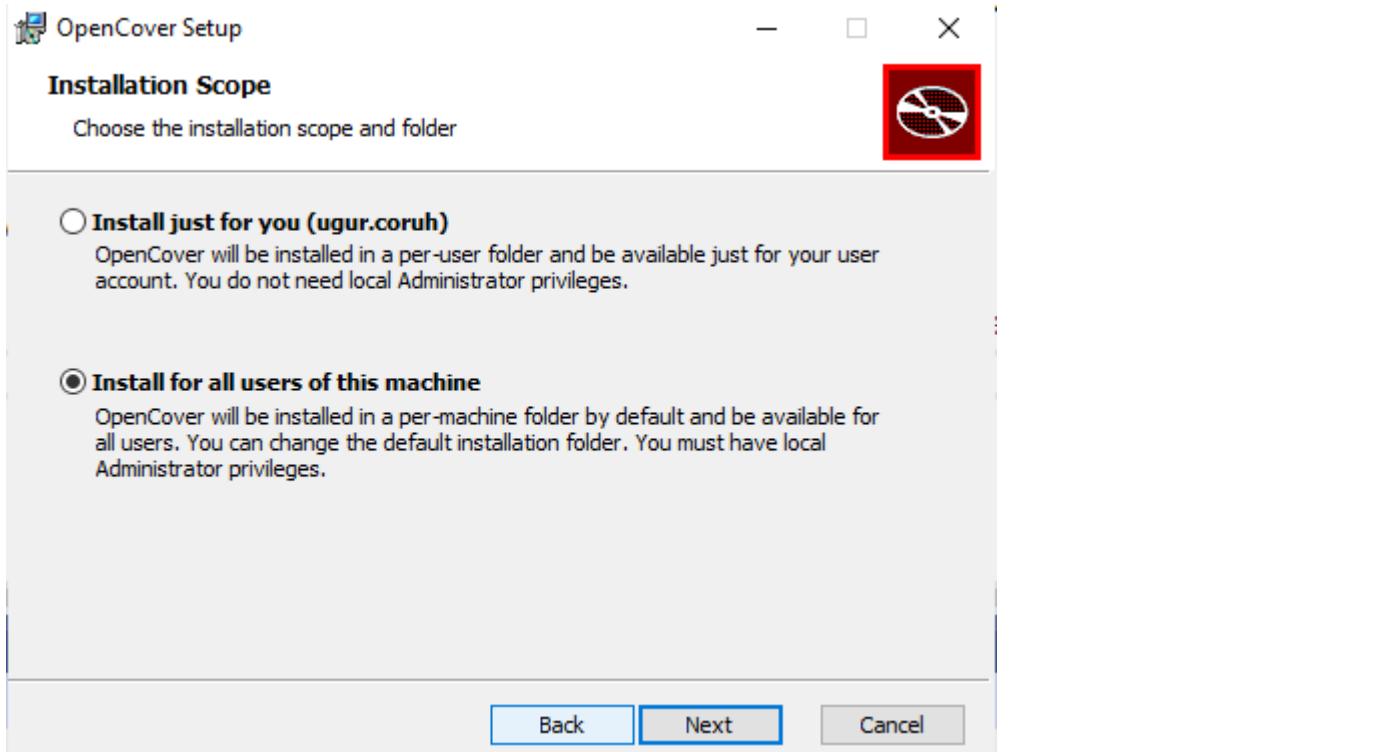
add dependency to dotnet 4.7.2 to chocolatey packages

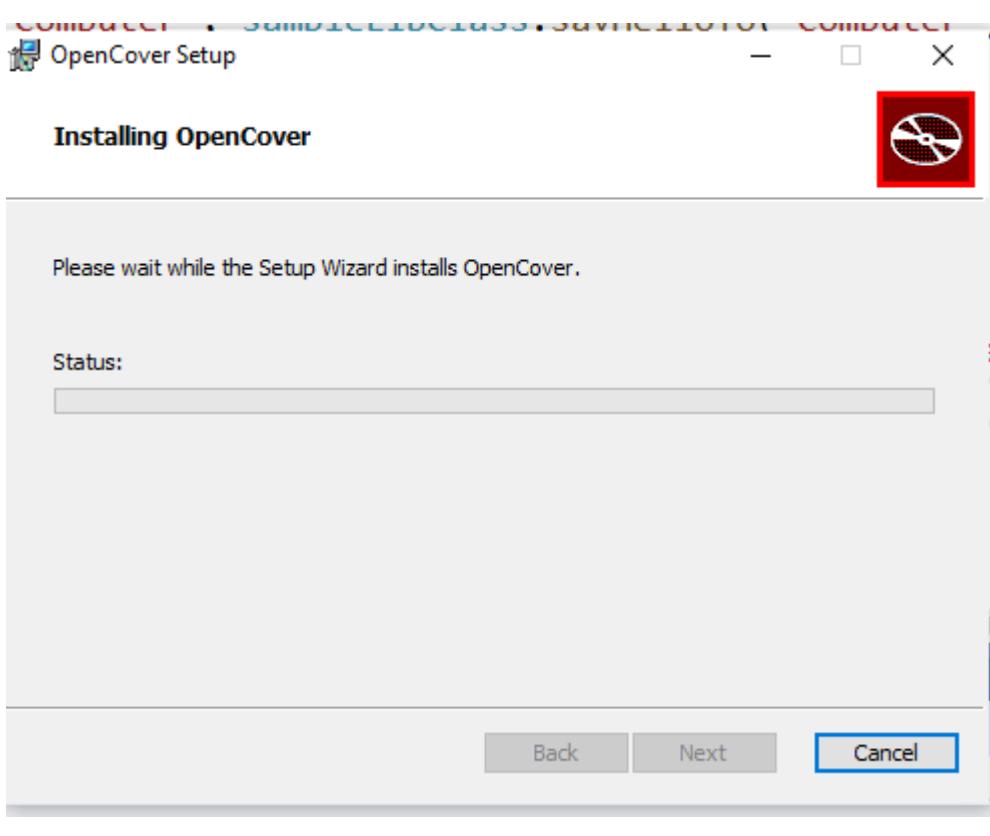
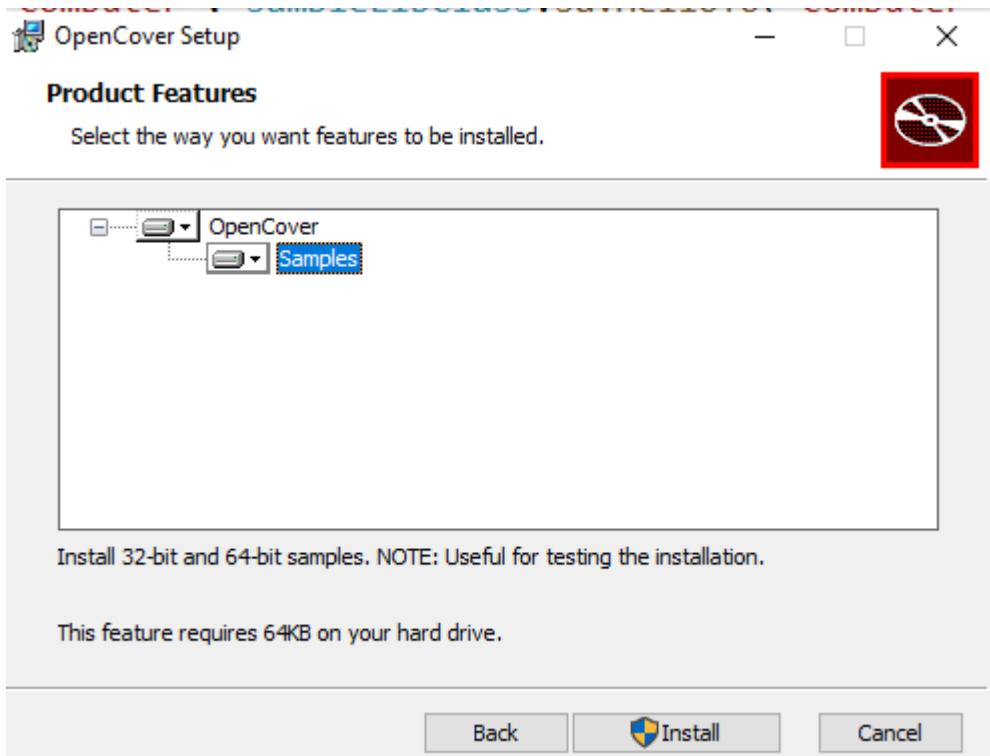
▼ Assets 6

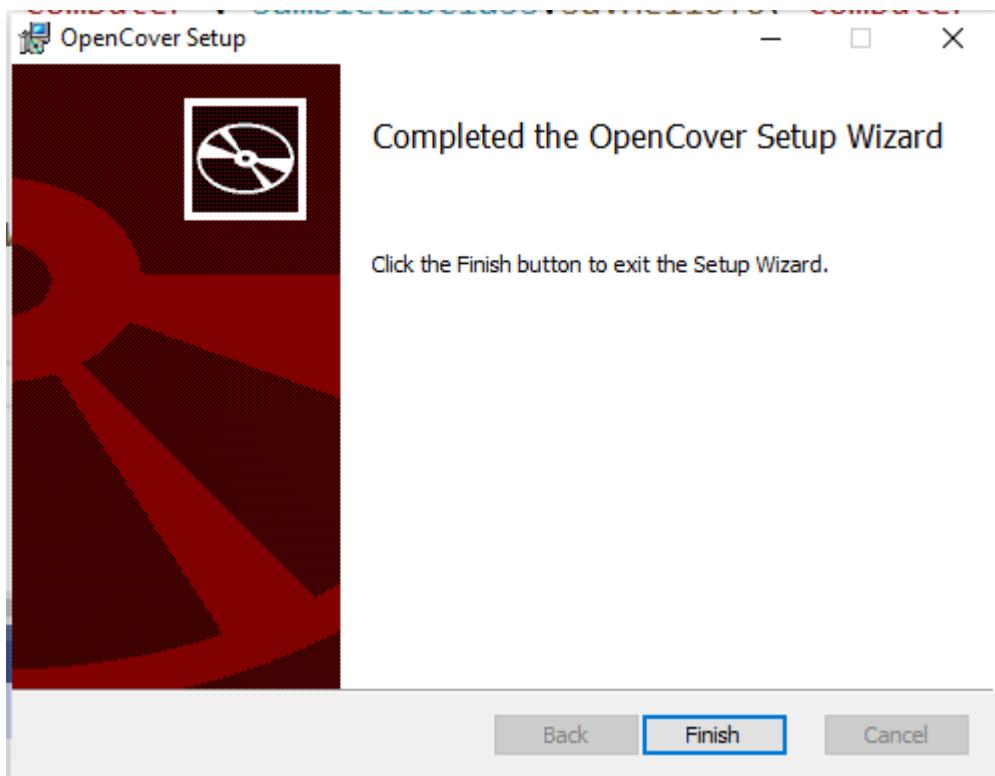
 checksum.installer.txt	66 Bytes
 checksum.zip.txt	66 Bytes
 opencover.4.7.1221.msi	8.09 MB
 opencover.4.7.1221.zip	7.76 MB
 Source code (zip)	
 Source code (tar.gz)	



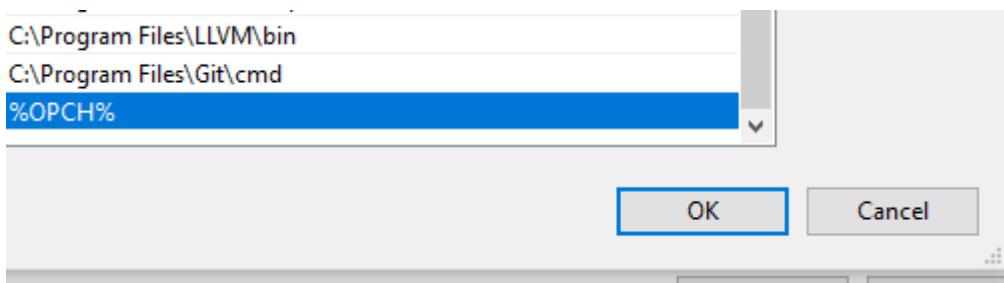
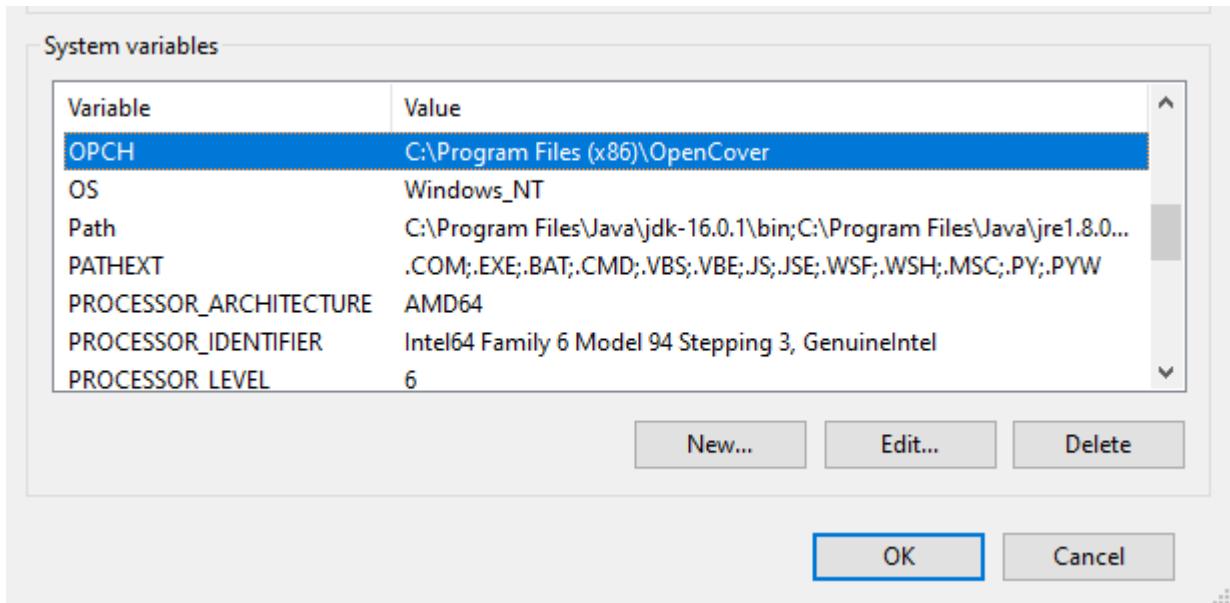
Select advanced and then install for all users







Mono.Cecil.dll	9/19/202
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.pdb	6/10/202



```
C:\> C:\WINDOWS\SYSTEM32\CMD.EXE
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
  [[""]-targetargs:<arguments for the target process>[""]]
```

ReportGenerator

[Release ReportGenerator_4.8.13 · danielpalme/ReportGenerator · GitHub](#)

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 >

Photo Print Photo Print

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

NUnit Console

[Downloads](#)

hunit.org/download/

CERN Open Data P... sentinelcustomer.sa... Paletton - The Colo... C# - DataGridView... Modamob Akıllı Mo... LED series parallel a... Inline Digital Hydro... SESSİZ JENERATÖR... Android - Frida • A... Eric Lau - Everyt... News Download Documentation Contact Slack GitHub

Downloads

Download Types

The preferred way to download NUnit is through the [NuGet](#) package manager.

The latest releases can always be found on the relevant GitHub releases pages.

Latest NUnit 3 Releases	
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
NUnit 3 Template for dotnet new CLI	

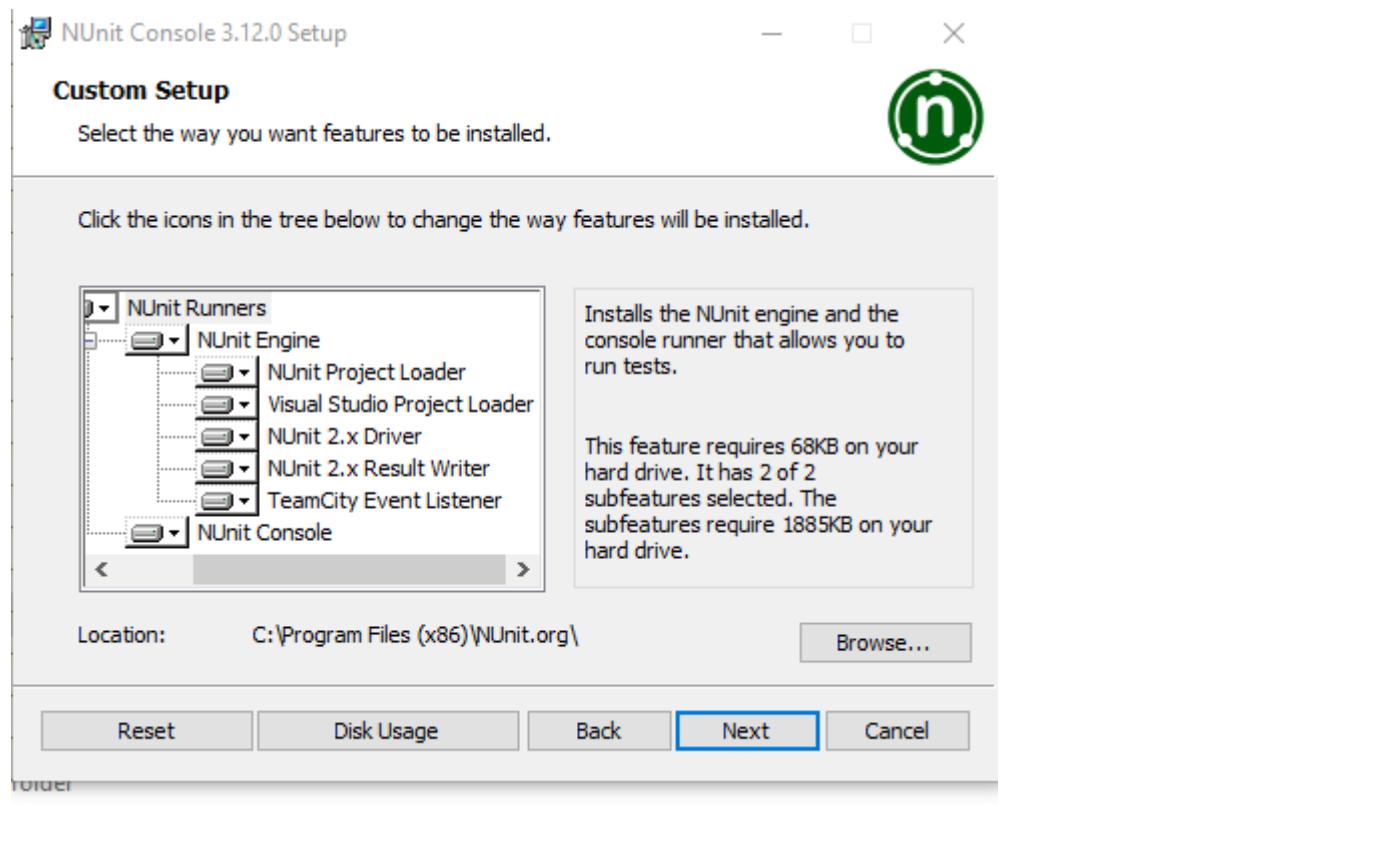
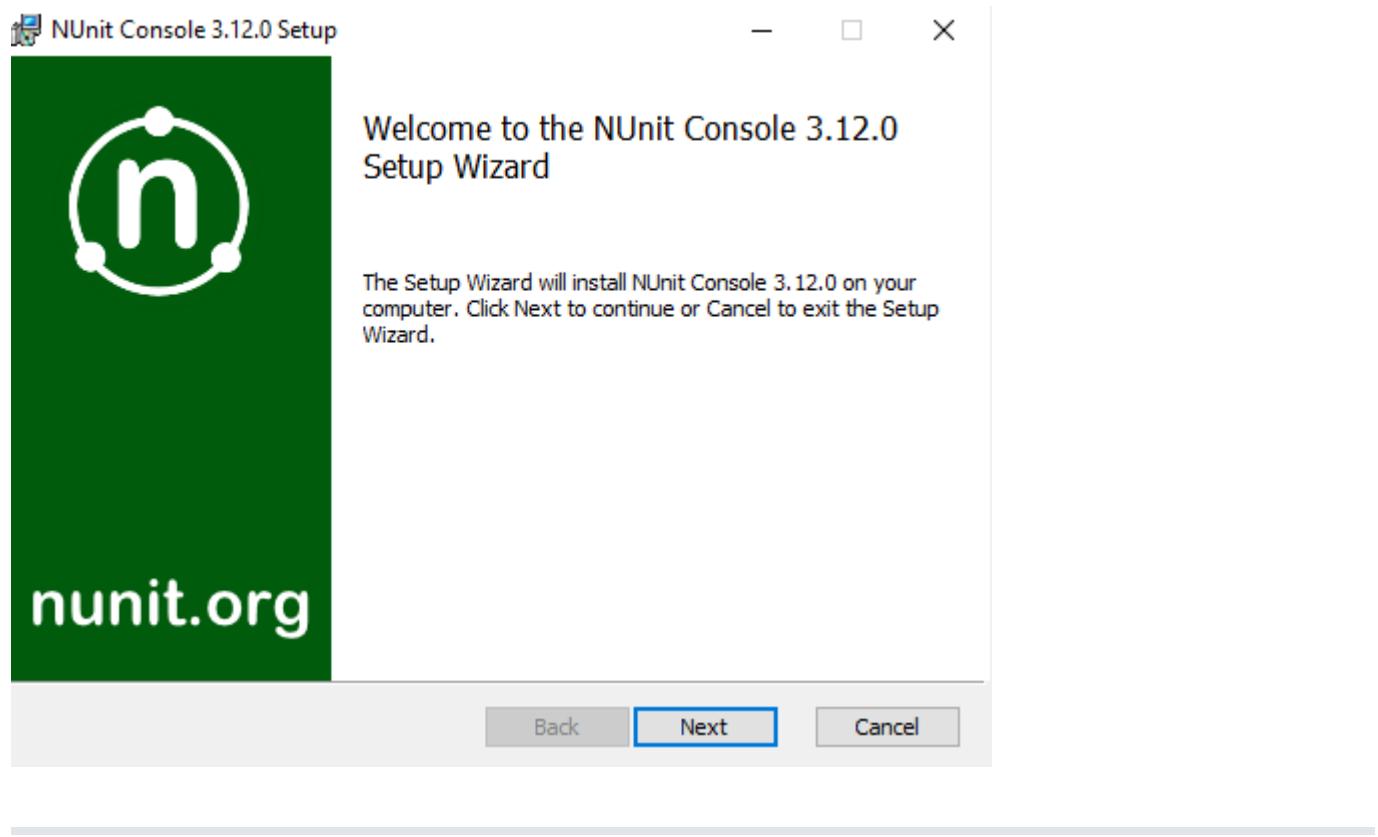
Latest NUnit 2 Release	
NUnit 2.7.1	August 19, 2019
NUnit Test Adapter 2.2	June 5, 2019

Older Releases

These releases are needed by many people for legacy work, so we keep them around for download. Bugs are accepted on older releases only if they can be reproduced on a current release.

▼ Assets 10

nunit-console-runner.3.12.0.nupkg	733 KB
NUnit.Console-3.12.0.msi	1.04 MB
NUnitConsole-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)	



Share View

This PC > Windows (C:) > Program Files (x86) > NUnit.org > nunit-console >

Print Photo Print

	Name	Date modified	Type	Size
ple-lib	addons	10/24/2021 11:30 PM	File folder	
-sample	agents	10/24/2021 11:30 PM	File folder	
ersonal	nunit.bundle.addins	4/2/2018 2:18 PM	ADDINS File	1 KB
	nunit.engine.api.dll	1/23/2021 3:03 PM	Application exten...	18 KB
	nunit.engine.api.xml	1/23/2021 3:03 PM	XML File	55 KB
	nunit.engine.core.dll	1/23/2021 3:03 PM	Application exten...	91 KB
	nunit.engine.dll	1/23/2021 3:03 PM	Application exten...	54 KB
	nunit3-console.exe	1/23/2021 3:04 PM	Application	163 KB
	nunit3-console.exe.config	12/27/2020 3:39 PM	Configuration Sou...	2 KB
	testcentric.engine.metadata.dll	9/3/2020 6:49 PM	Application exten...	173 KB

NUnit + MSTest Batch Report Generation (Not Tested)

[OpenCover and ReportGenerator Unit Test Coverage in Visual Studio 2013 and 2015 – CodeHelper.Net](#)

[OpenCover and ReportGenerator Unit Test Coverage in Visual Studio 2013 and 2015 - CodeProject](#)

Java Unit Tests

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 tests](#)

[JUnit 5](#)

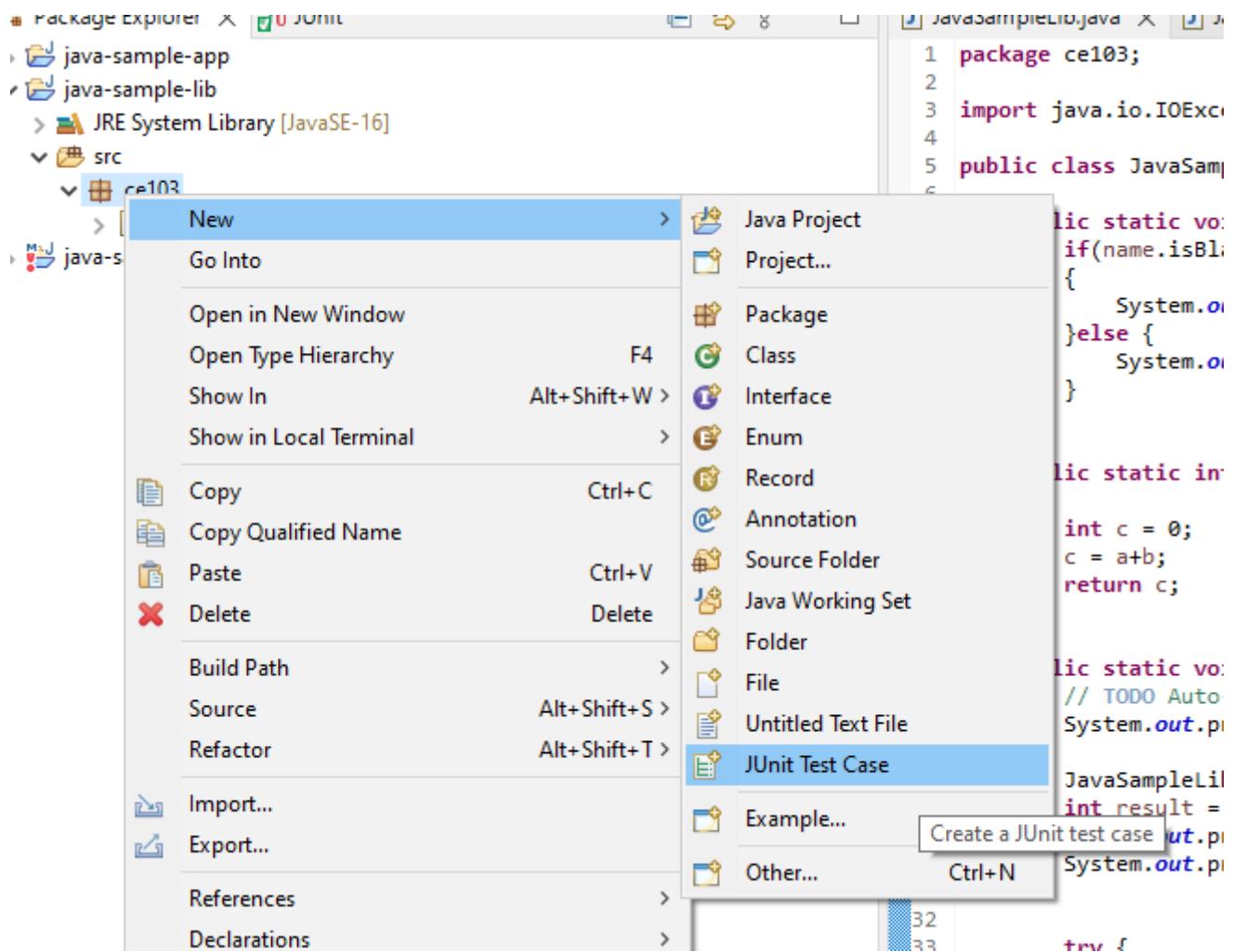
[JUnit 5 User Guide](#)

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

[JUnit Hello World Example - Examples Java Code Geeks - 2021](#)

Java Application + JUnit

In normal java application we can right click the project java-sample-lib and add 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:

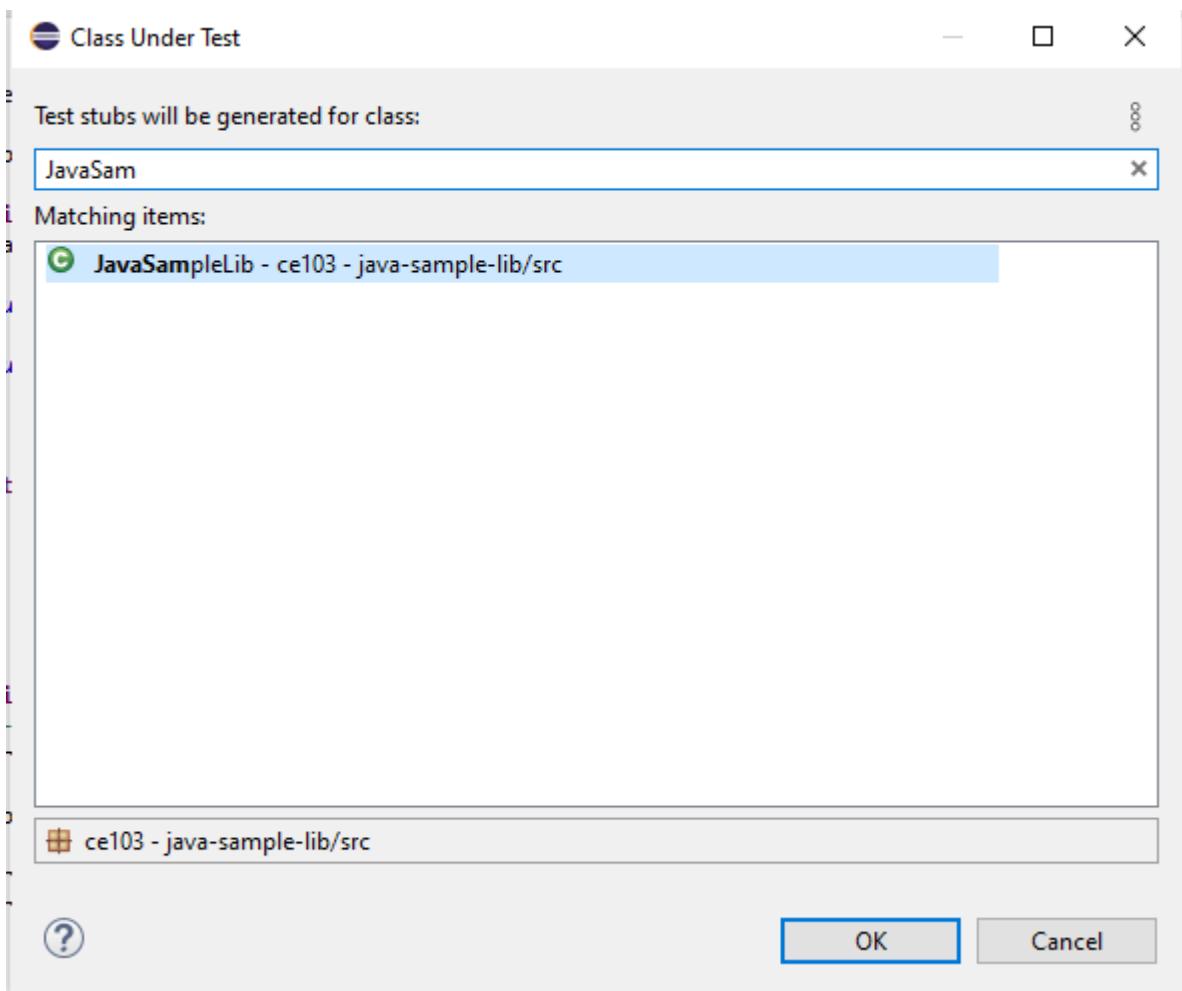


< Back

Next >

Finish

Cancel



New JUnit Test Case

Test Methods

Select methods for which test method stubs should be created.



Available methods:

- ▼ **C** JavaSampleLib
 - S** sayHelloTo(String)
 - S** sum(int, int)
 - S** main(String[])
- ▼ **C** Object
 - c** Object()
 - NF** getClass()
 - N** hashCode()
 - c** equals(Object)
 - N** clone()
 - c** toString()
 - NF** notify()
 - NF** notifyAll()
 - F** wait()

Select All

Deselect All

2 methods selected.

- Create final method stubs
 Create tasks for generated test methods



< Back

Next >

Finish

Cancel

New JUnit Test Case



JUnit 5 is not on the build path. Do you want to add it?

- Not now
 Open the build path property page
 Perform the following action:

Add JUnit 5 library to the build path

OK

Cancel

and you will have the following test class

The screenshot shows the Eclipse IDE interface with the 'Package Explorer' view on the left and the 'Java Editor' view on the right. The 'Package Explorer' shows a project named 'java-sample-app' containing a 'src' folder with a package named 'ce103' containing two files: 'JavaSampleLib.java' and 'JavaSampleLibTest.java'. It also shows a 'JUnit 5' entry and a 'java-sample-lib-test' entry. The 'Java Editor' view displays the code for 'JavaSampleLibTest.java'. The code is a JUnit 5 test class with the following structure:

```
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));
}
}

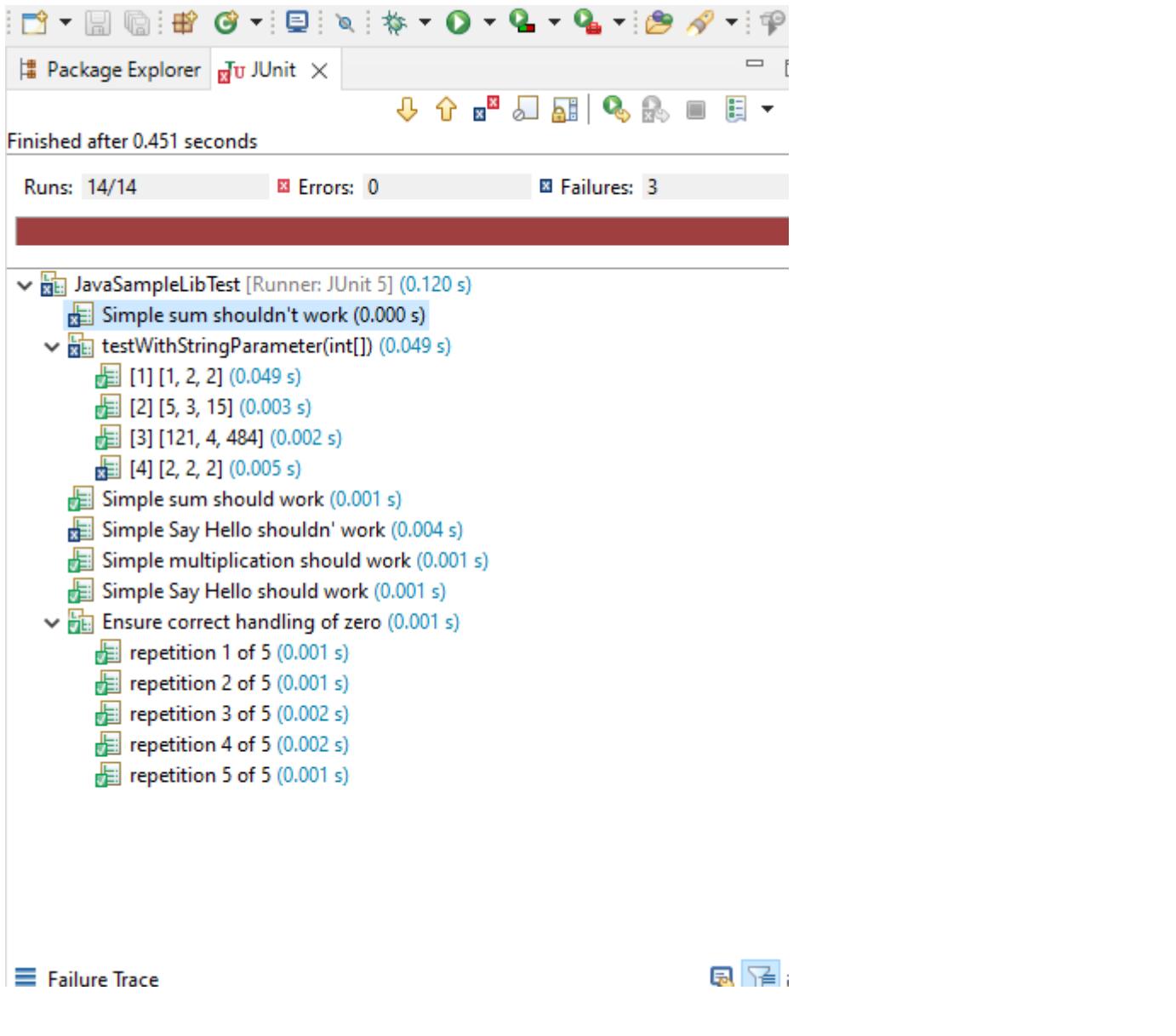
```

if we run tests

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

- Project Explorer (left):** Shows the project structure with `java-sample-app`, `java-sample-lib`, `JRE System Library [JavaSE-16]`, `src` (containing `ce103`), and `JUnit 5`.
- Java Sample Lib Test.java (selected):** The code for the test class is displayed, including imports for JUnit Jupiter and assertions.
- Context Menu (open on JavaSampleLibTest.java):**
 - Run As > JUnit Test** is selected, highlighted in blue.
 - Other options include New, Open, Copy, Delete, Import..., Export..., and Coverage As.
- Run Configuration Dialog (center):**
 - Shows "1 JUnit Test" selected.
 - Buttons: Run Configurations... and Run.
- Console (bottom right):**
 - Shows the output of the test: "Hello Computer" and "Hello Computer".
 - Status: `<terminated> JavaSampleLibTest (1) [JUnit] C:\Program`

we will see all results there



also we can see the code coverage of tests

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

```

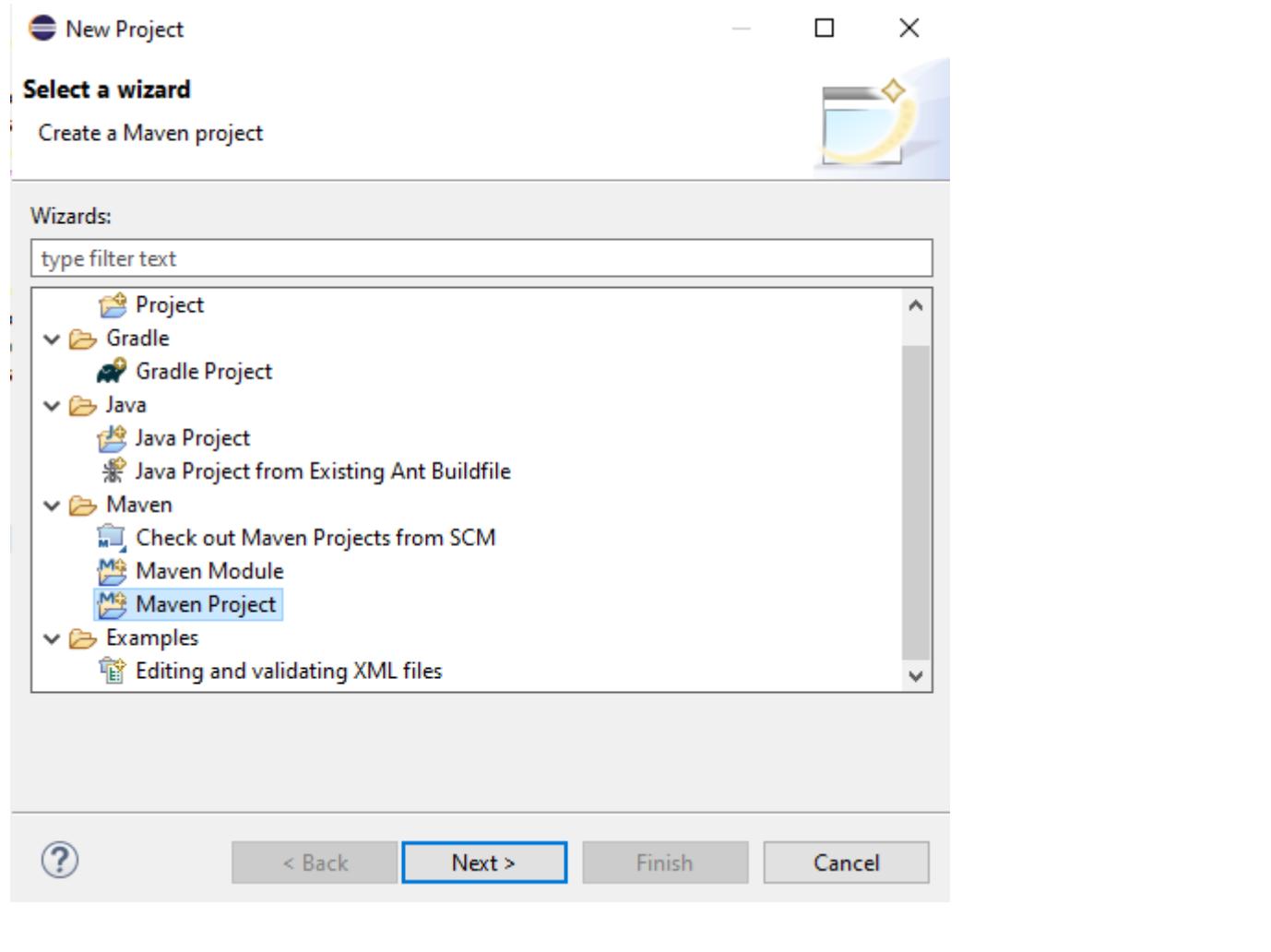
1 package ce103;
2
3 public class JavaSampleLib {
4
5     public static String sayHelloTo(String name) {
6
7         String output = "";
8
9         if(!name.isBlank() && !name.isEmpty()){
10             output = "Hello "+name;
11         }else {
12             output = "Hello There";
13         }
14
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

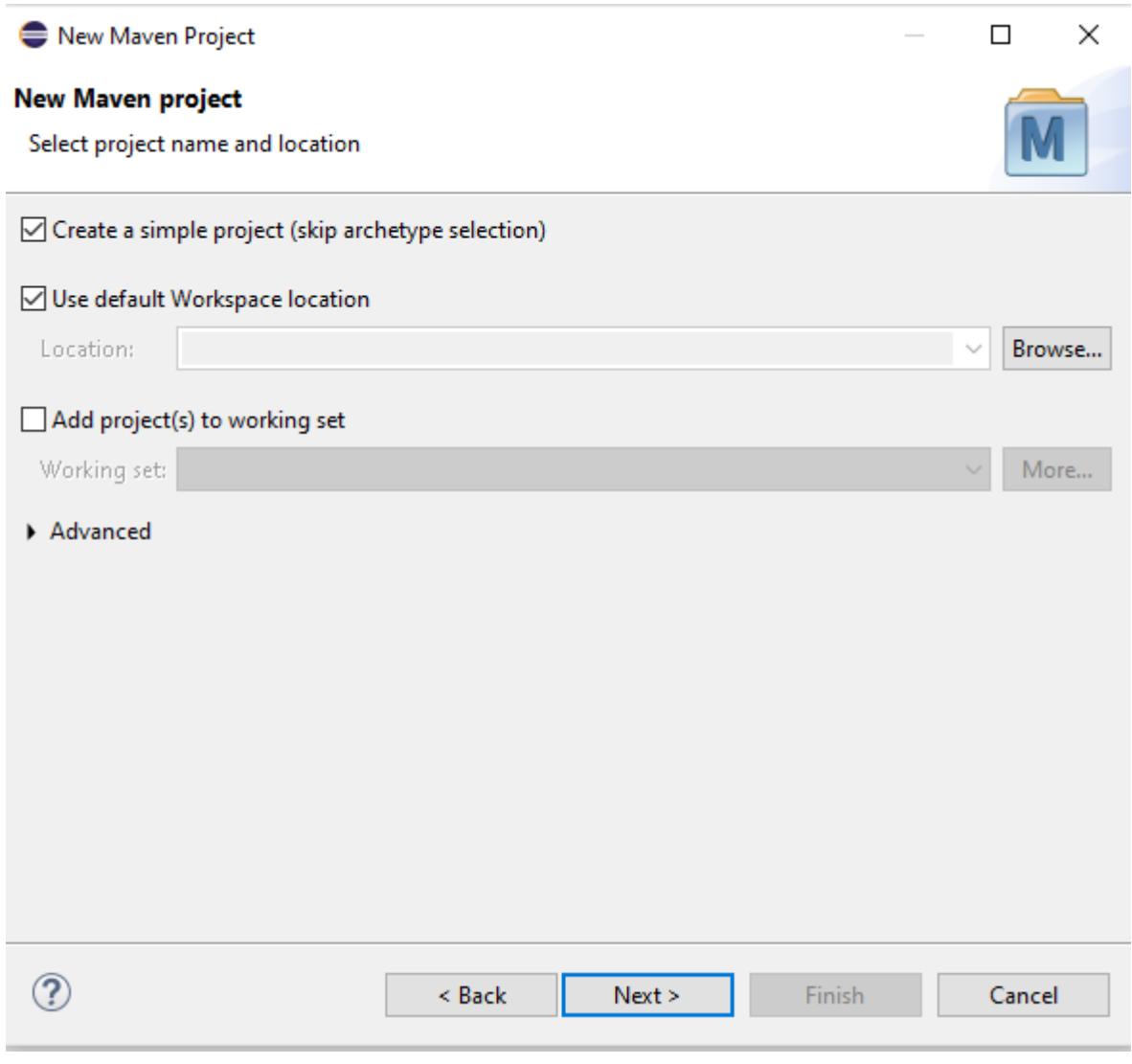
```

Lets create Maven project with tests

Create a maven project

File -> New -> Maven Project





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

[Maven – Introduction to the Standard Directory Layout](#)

Also for intro you can use this

[JUnit Hello World Example - Examples Java Code Geeks - 2021](#)

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

New Maven Project

New Maven project

Configure project



Artifact

Group Id: com.ce103

Artifact Id: java-sample-lib-ext

Version: 0.0.1-SNAPSHOT

Packaging: jar

Name: Java Sample Lib

Description: Java Sample with Unit Test

Parent Project

Group Id:

Artifact Id:

Version:

[Browse...](#)

[Clear](#)

[Advanced](#)



[Back](#)

[Next](#)

[Finish](#)

[Cancel](#)

java-sample-lib-ext

src/main/java

src/main/resources

src/test/java

src/test/resources

JRE System Library [J2SE-1.5]

src

target

pom.xml

pom.xml file

```
<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>
</project>
```

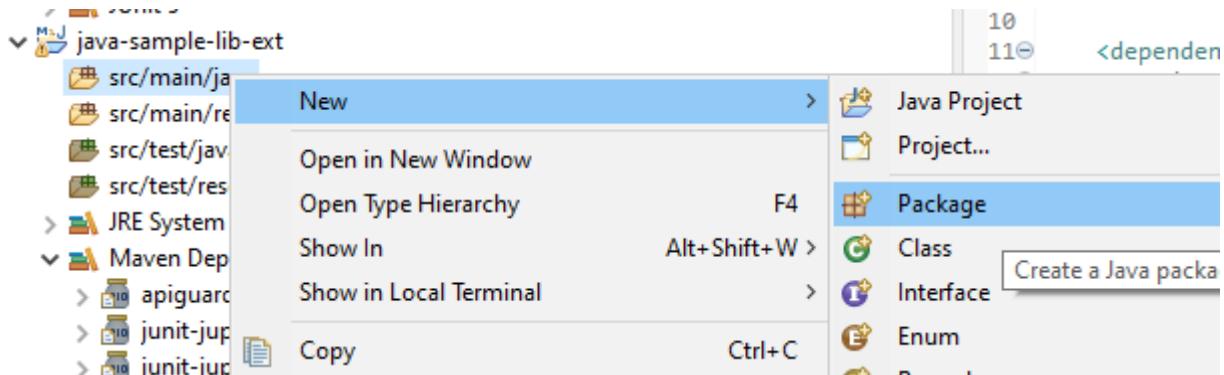
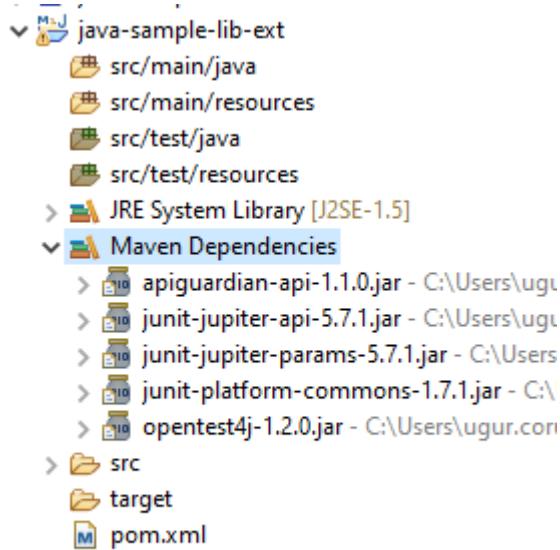
we will add JUnit 5 for our project

```
<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>

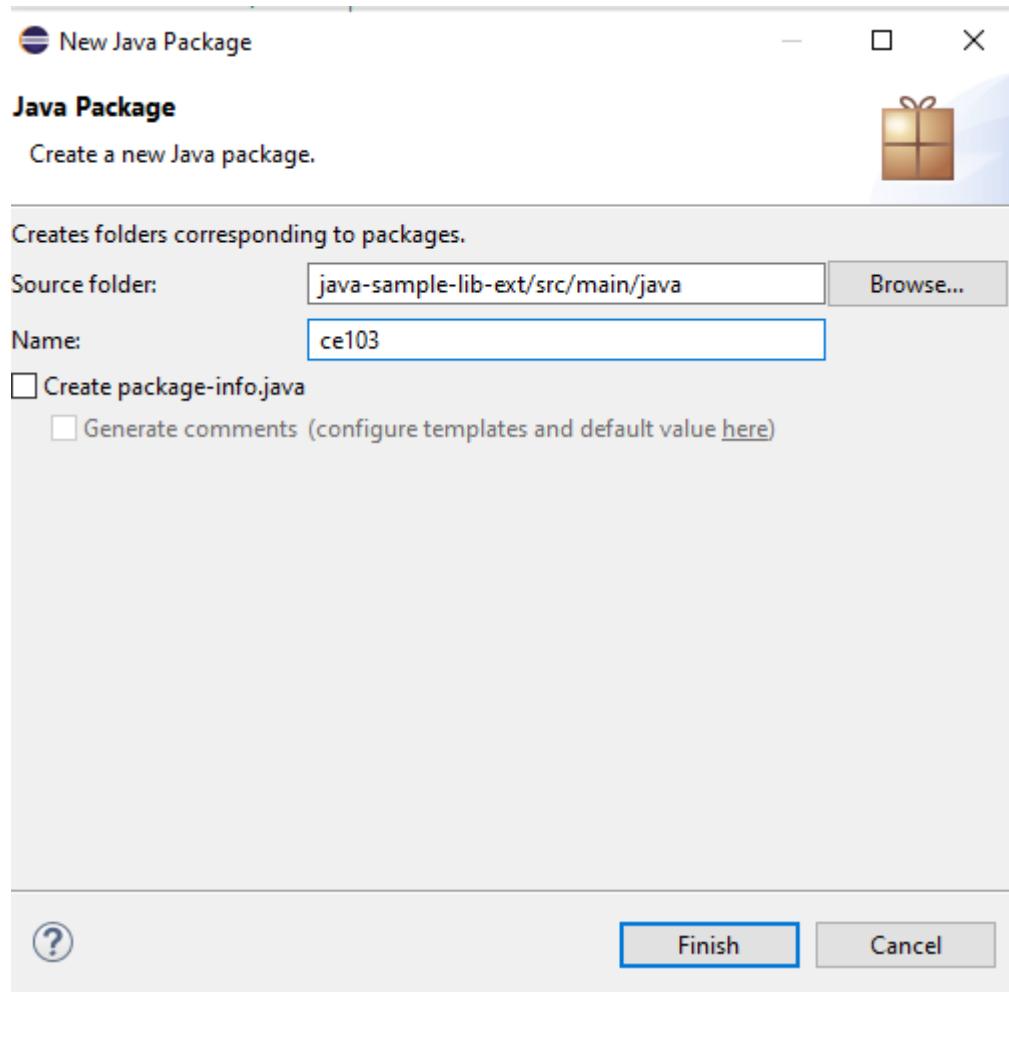
  <dependencies>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-params</artifactId>
      <version>5.7.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

</project>
```

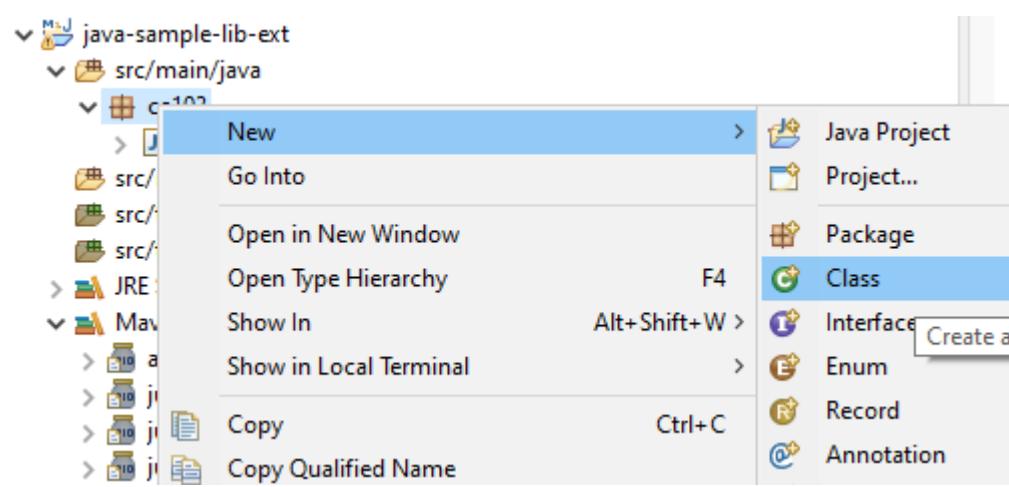
it will automatically download libraries

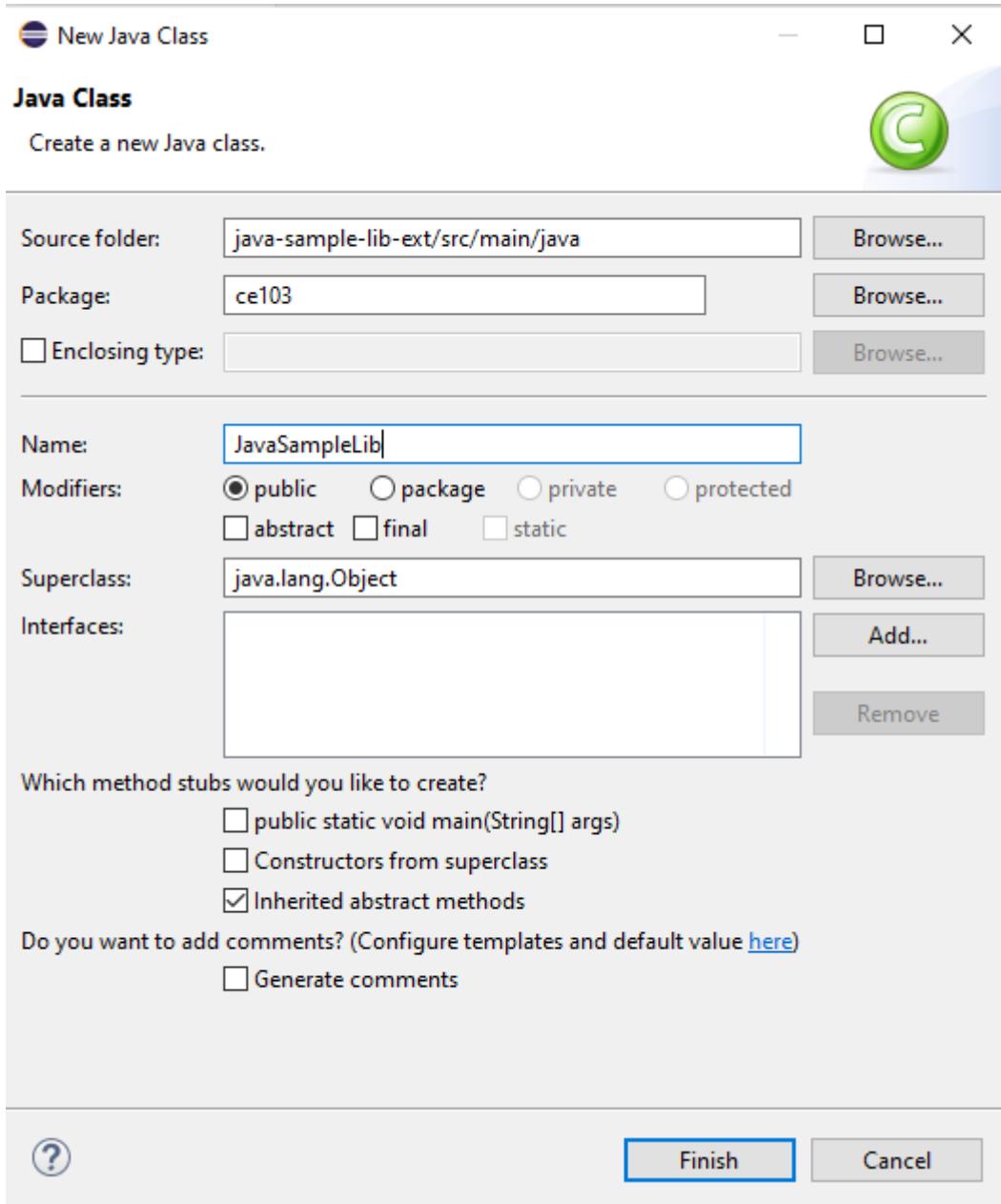


Create java sample library in ce103 package, first create java package



In this package create library class





copy content from other library

```
package ce103;

public class JavaSampleLib {

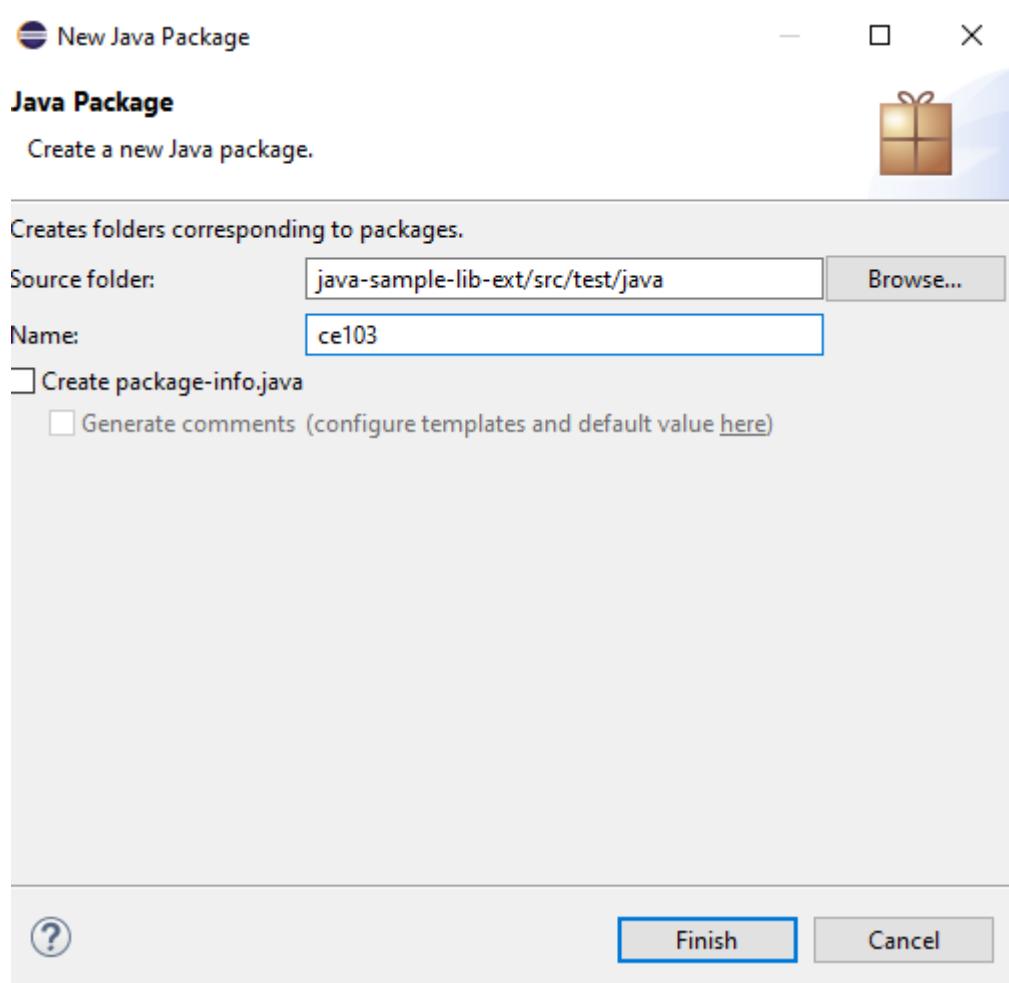
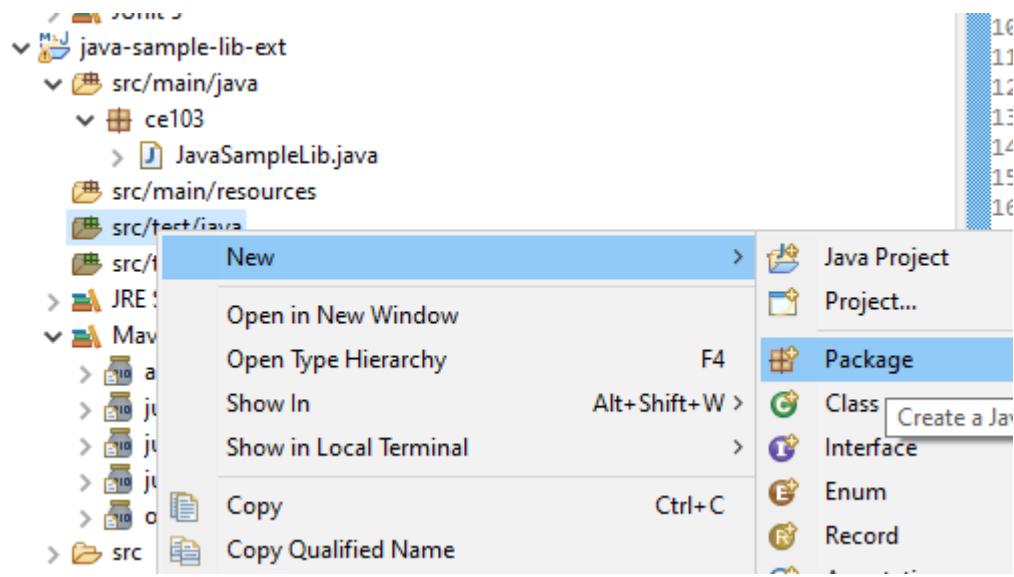
    public static String sayHelloTo(String name) {
        String output = "";
        if(!name.isBlank() && !name.isEmpty()){
            output = "Hello "+name;
        }else {
            output = "Hello There";
        }
        System.out.println(output);
        return output;
    }

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

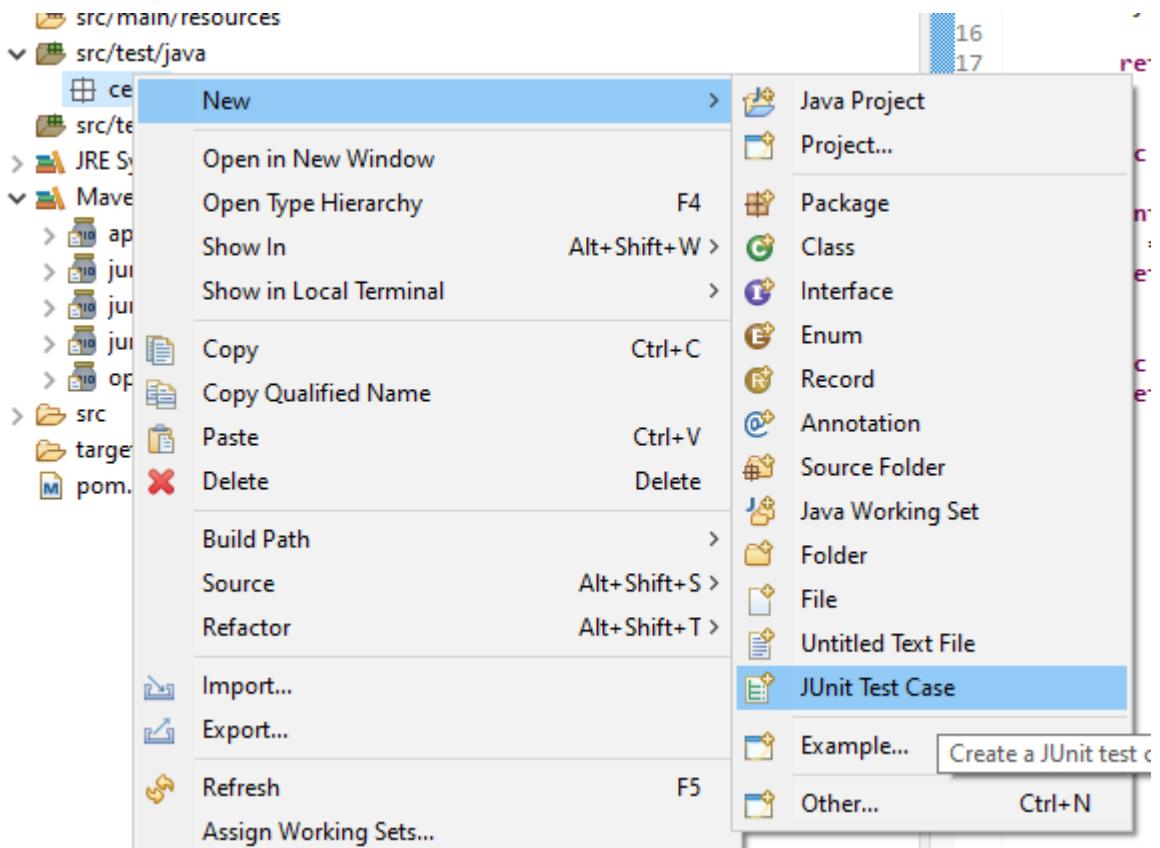
    public int multiply(int a, int b) {
        return a * b;
    }

}
```

Now lets create tests inf src/test/java



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.

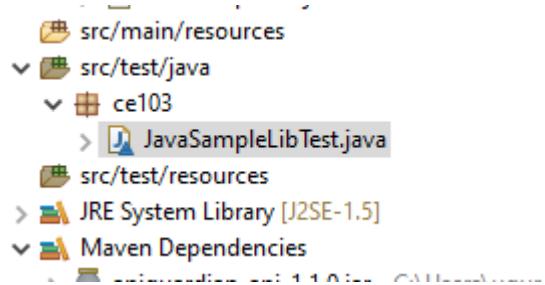
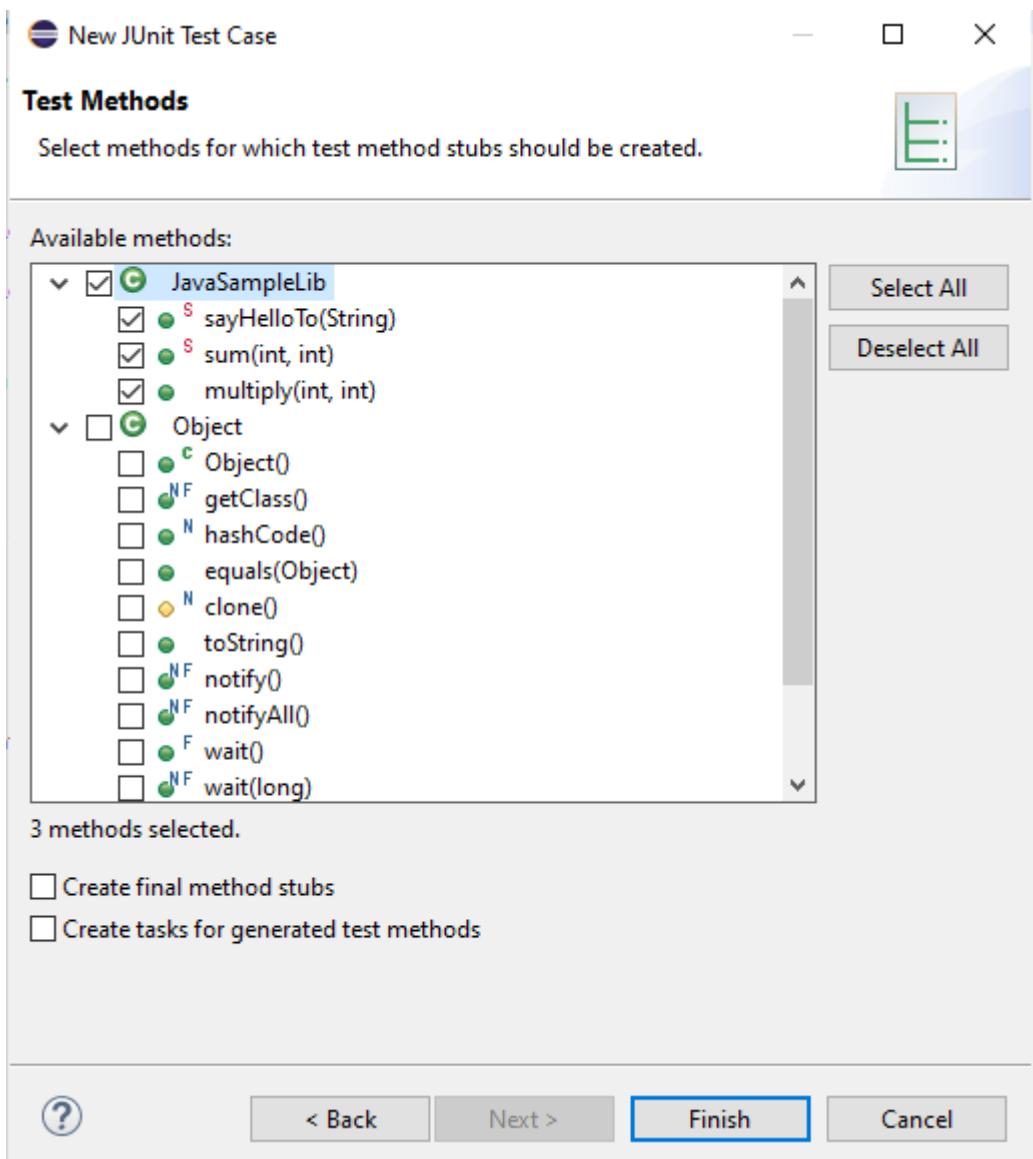


< Back

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

Java Sample Application Structure:

```

> JavaSampleLibTest.java
> JUnit 5
java-sample-lib-ext
  > src/main/java
    > ce103
      > JavaSampleLib.java
  > src/main/resources
  > src/test/java
    > ce103
      > JavaSampleLibTest.java
  > src/test/resources
  > JRE System Library
  > Maven Dependencies
    > apiguardian-api
    > junit-jupiter-api
    > junit-jupiter-parameterized
    > junit-platform-common
    > opentest4j-1.2.0
  > src
  > target
  pom.xml

```

Context menu for JavaSampleLibTest.java (Run As selected):

- New
- Open F3
- Open With
- Open Type Hierarchy F4
- Show In Alt+Shift+W
- Show in Local Terminal
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Build Path
- Source Alt+Shift+S
- Refactor Alt+Shift+T
- Import...
- Export...
- References
- Declarations
- Refresh F5
- Assign Working Sets...
- Coverage As
- Run As
- Debug As
- Restore from Local History...
- Team
- Compare With
- Replace With

JavaSampleLibTest.java Content:

```

18
19     @BeforeAll
20         static void setUpBefore() {
21     }
22
23     @AfterAll
24         static void tearDown() {
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 test")
38         void testSayHelloTo() {
39             assertEquals("Hello", sampleLib.sayHello("World"));
40         }
41
42     @Test
43         @DisplayName("Simple test")
44         void testSayHelloTo() {
45             assertEquals("Hello", sampleLib.sayHello("World"));
46         }
47
48
49     @Test
50         @DisplayName("Simple test")
51         void testSumCorrectly() {
52             assertEquals(9, sampleLib.sum(4, 5));
53         }
54
55     @Test
56         @DisplayName("Simple test")
57         void testSumCorrectly() {
58             assertEquals(9, sampleLib.sum(4, 5));
59         }

```

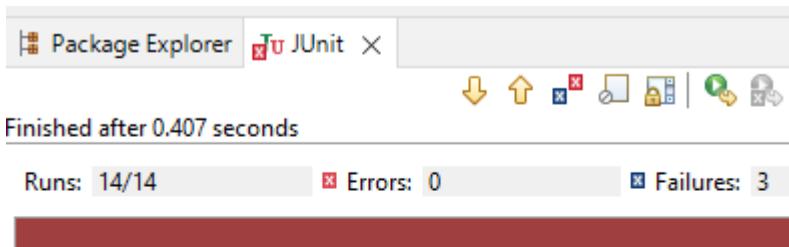
Run Configurations...

Java Sample Application Structure (right side):

```

java-sample-lib-ext
  > src
    > ce103
      > JavaSampleLibTest.java
        > JavaSampleLibTest
      > JavaSampleLib.java
        > JavaSampleLib

```



- 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)

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

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit X

finished after 0.697 seconds

Runs: 14/14 Errors: 0 Failures: 3

JavaSampleLibTest [Runner: JUnit 5] (0.278 s)

- Simple sum shouldn't work (0.017 s)
- testWithStringParameter(int[]) (0.084 s)
- Simple sum should work (0.002 s)
- Simple Say Hello shouldn't work (0.005 s)
- Simple multiplication should work (0.002 s)
- Simple Say Hello should work (0.002 s)
- Ensure correct handling of zero (0.002 s)

JavaSampleLib.java

```

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 }
33

```

Failure Trace

- org.opentest4j.AssertionFailedError: Regular sum shouldn't work ==> expected: <10> |
- at ce103.JavaSampleLibTest.testSumWrong(JavaSampleLibTest.java:58)
- at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
- at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)

Coverage

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
JavaSampleLibTest.java	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

Thats a part of java unit testing...

TDD (Test Driven Development)

Test and Deployment Automation Management

Travis-CI + C

Travis-CI + Cpp

Travis-CI + C#

Travis-CI + Java

References

[GitHub - MicrosoftDocs/cpp-docs: C++ Documentation](#)