

CE103 Algorithms and Programming I

Week-4

Introduction to Code Reusability and Automated Testing

Download [DOC](#), [SLIDE](#), [PPTX](#)



Outline

- Introduction to Code Reusability and Automated Testing
- Shared Library Development
 - C
 - C++
 - C#
 - Java
- Unit Testing
 - C
 - C++
 - C#
 - Java
- Continuous Integration Platforms

Introduction to Code Reusability and Automated Testing

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

Selected Development Environment

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

Example Content

- Each example will include two function
- "Hello <name>" 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 the result to the sum variable.
- We will locate them in the 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)

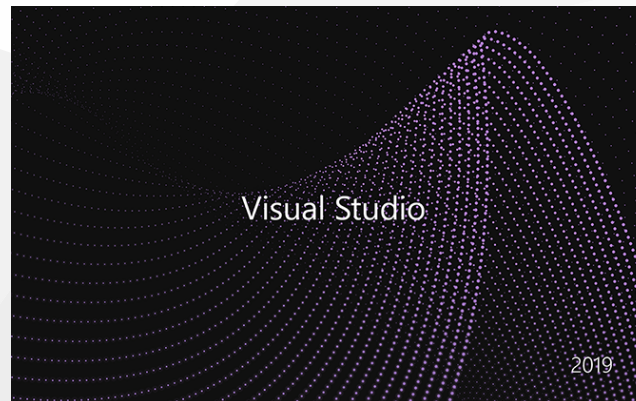
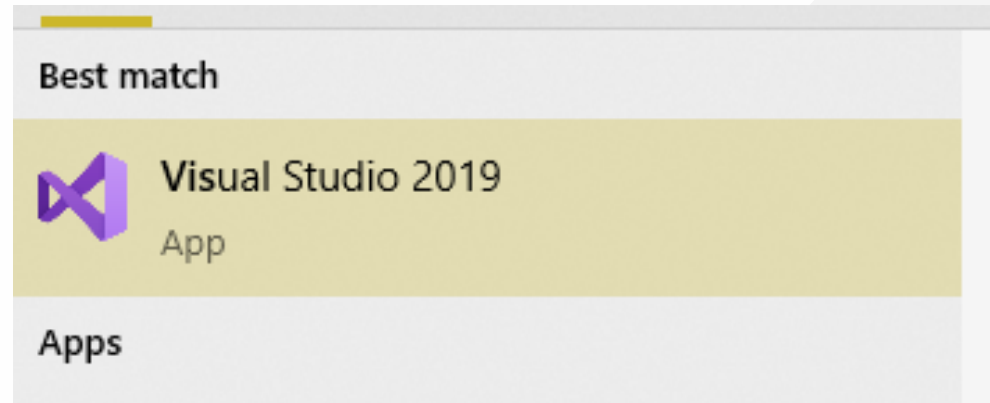
Visual Studio Community Edition

Shared Library Development - (VS C Static Library)-1

- In this sample, we will create a **c-lib-sample** project that contains a library, executable, unit tests and unit test runners.
- First of all, you install Visual Studio Community Edition from the website
 - [Visual Studio 2019 Community Edition - Son Ücretsiz Sürümü İndir](#)

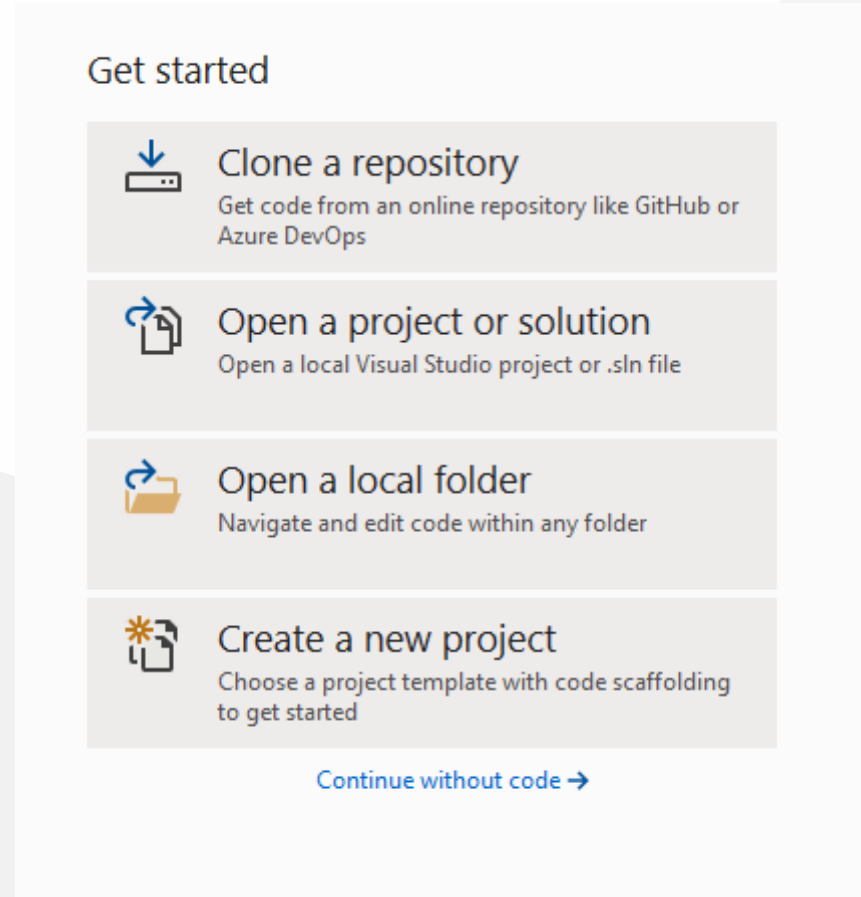
Shared Library Development - (VS C Static Library)-2

- Open visual studio community edition and select create a new project



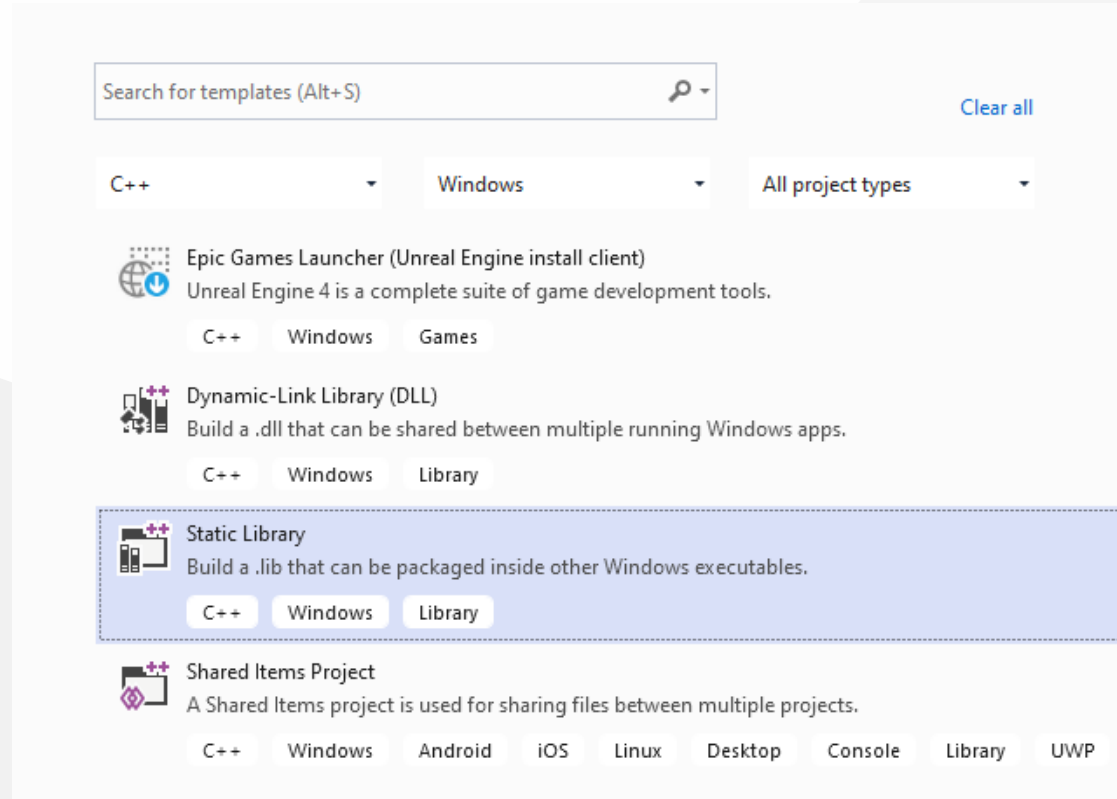
Shared Library Development - (VS C Static Library)-3

- Select create a new project



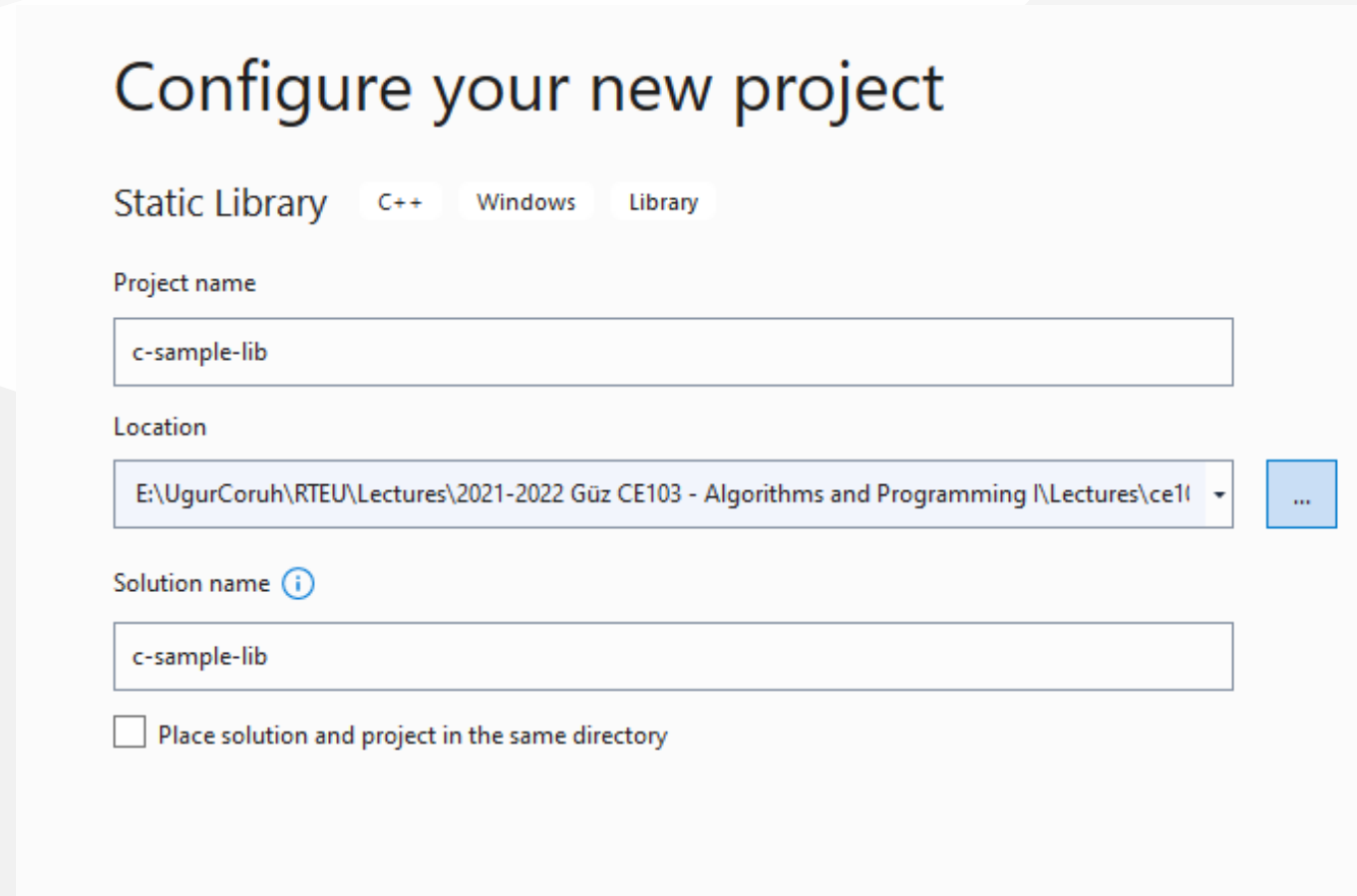
Shared Library Development - (VS C Static Library)-4

- Select C++ static library from the project list



Shared Library Development - (VS C Static Library)-5

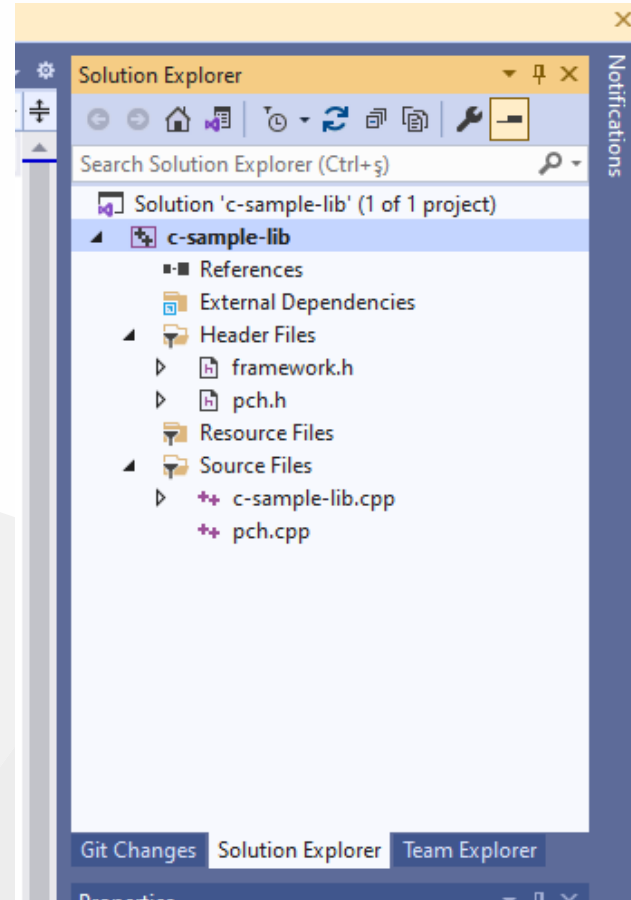
- Give static library project name



The screenshot shows the 'Configure your new project' dialog in Visual Studio. The project type is 'Static Library' under the 'C++' category. The 'Project name' field contains 'c-sample-lib'. The 'Location' field shows the path 'E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Programming I\Lectures\ce103' with a dropdown arrow and a blue ellipsis button to its right. The 'Solution name' field, which has an information icon (i) to its left, also contains 'c-sample-lib'. At the bottom, there is an unchecked checkbox labeled 'Place solution and project in the same directory'.

Shared Library Development - (VS C Static Library)-6

- Default configuration come with C++ project types and setting



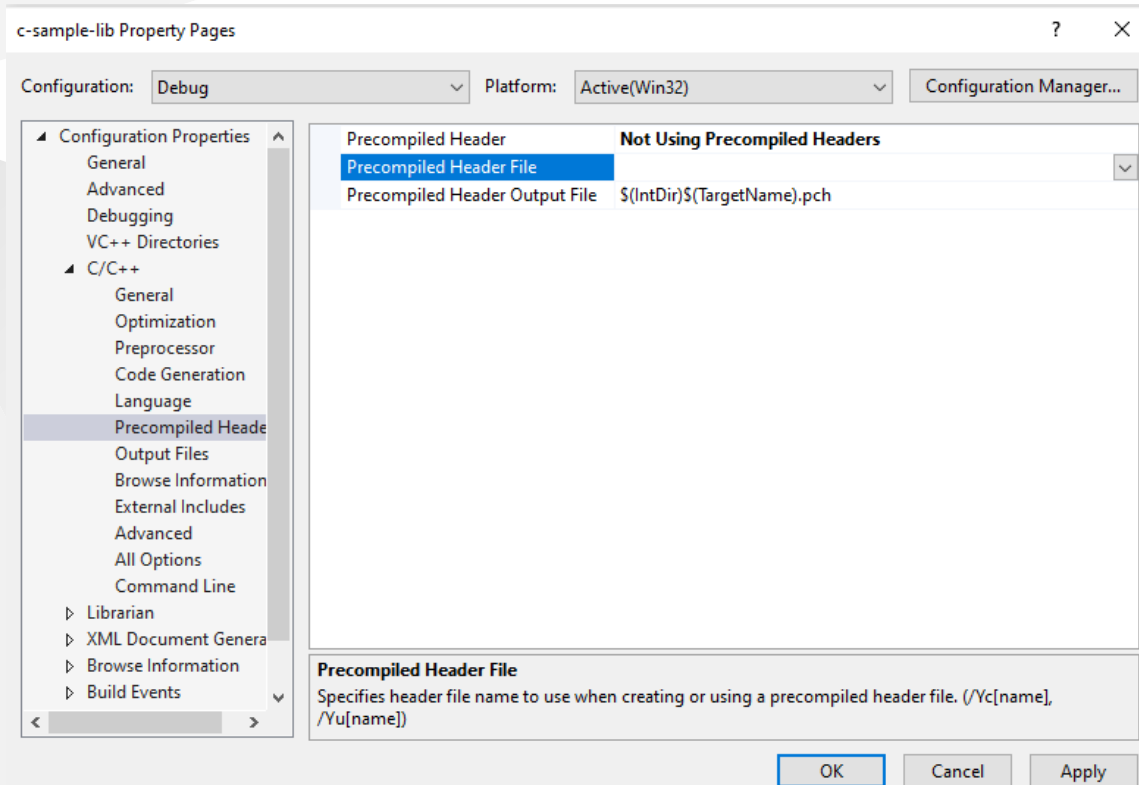
Shared Library Development - (VS C Static Library)-7

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

```
void fncsamplelib(){  
}
```

Shared Library Development - (VS C Static Library)-8

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.



Shared Library Development - (VS C Static Library)-9

- 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;
}
```

Shared Library Development - (VS C Static Library)-10

- Also, update `samplelib.h` as follows.

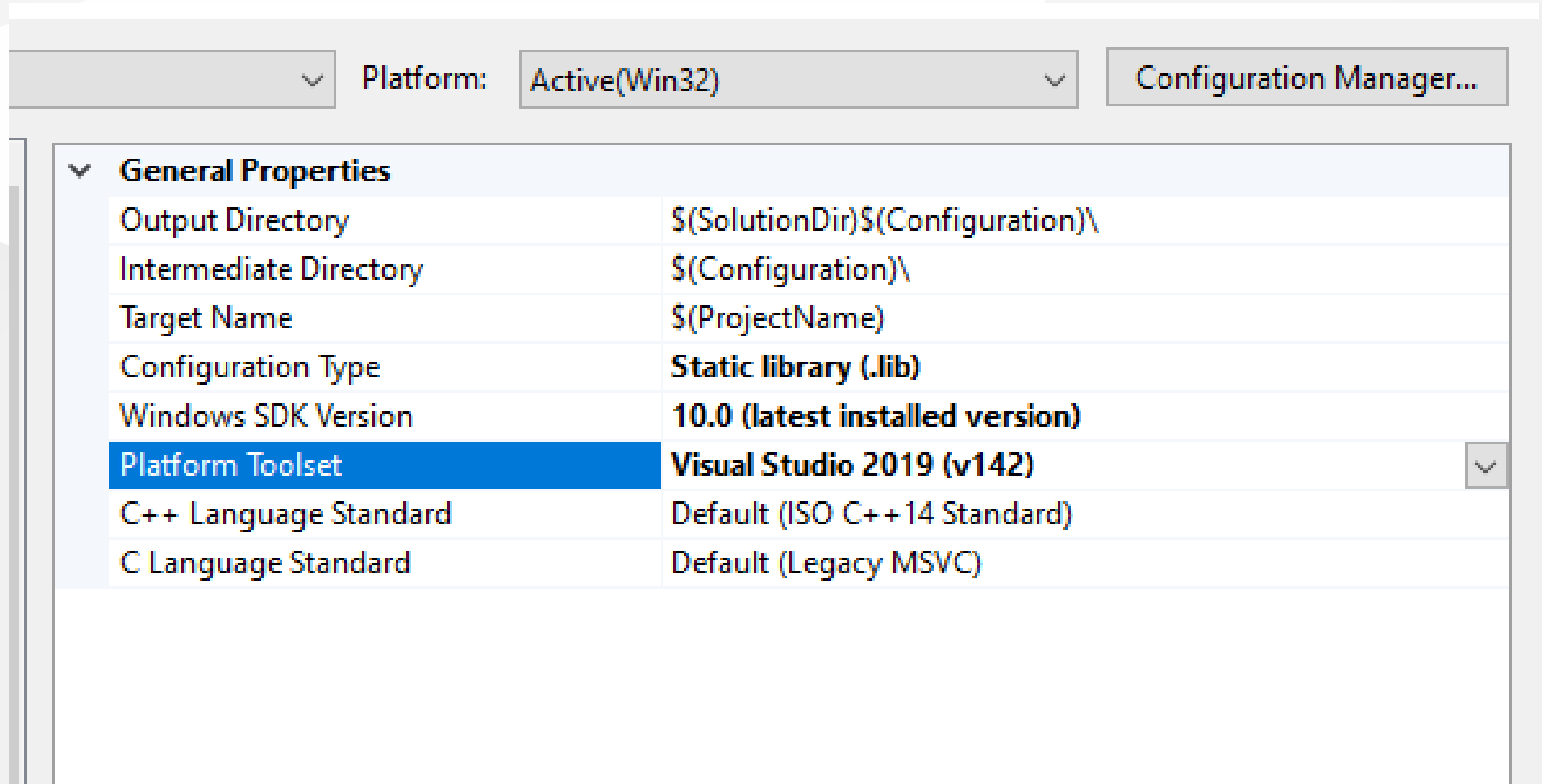
```
#pragma once

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

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

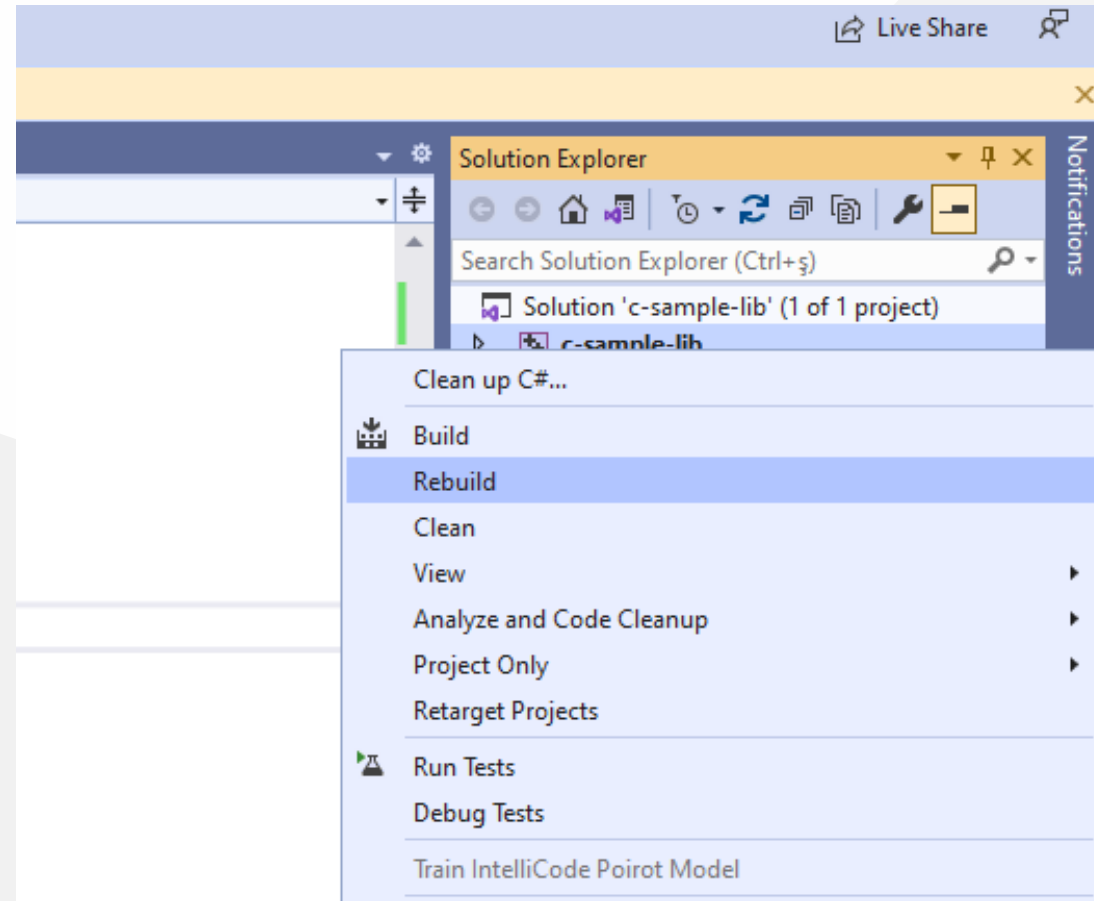

Shared Library Development - (VS C Static Library)-11

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



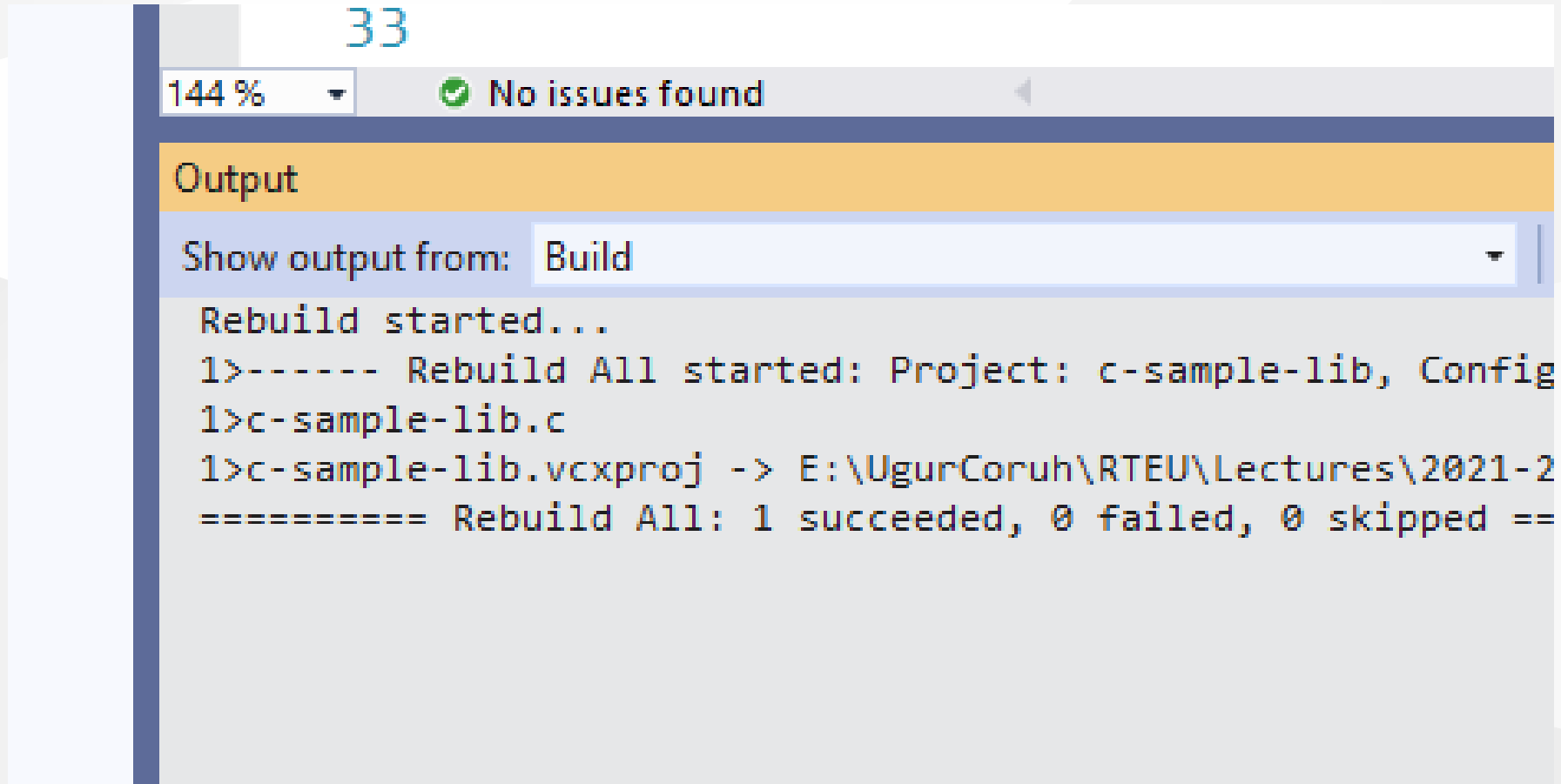
Shared Library Development - (VS C Static Library)-12

- Now we can compile our library



Shared Library Development - (VS C Static Library)-13

- You can follow operation from the output window

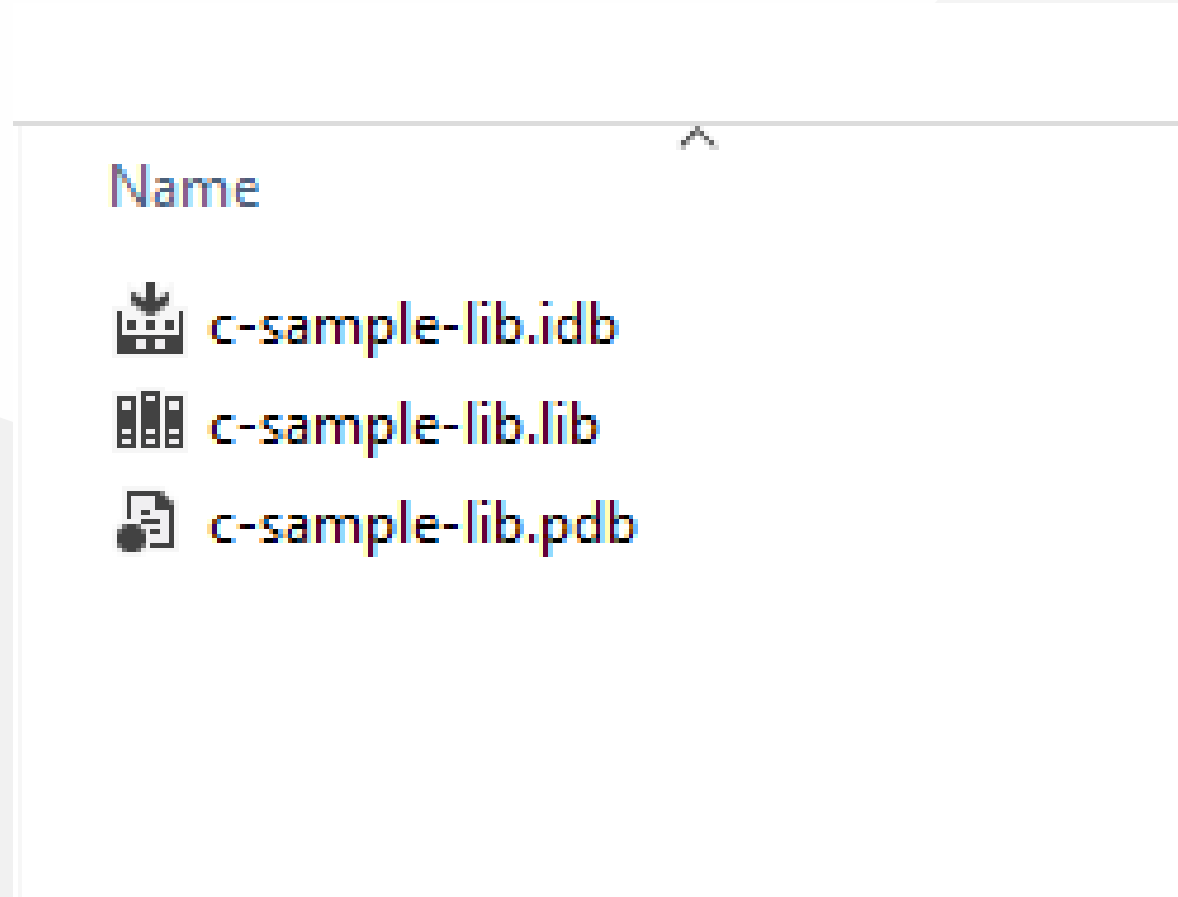


The screenshot shows the Visual Studio Output window. At the top, there is a zoom level of 144% and a status bar indicating 'No issues found'. The 'Output' window is titled 'Output' and has a dropdown menu set to 'Build'. The output text is as follows:

```
Rebuild started...  
1>----- Rebuild All started: Project: c-sample-lib, Config  
1>c-sample-lib.c  
1>c-sample-lib.vcxproj -> E:\UgurCoruh\RTEU\Lectures\2021-2  
===== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==
```

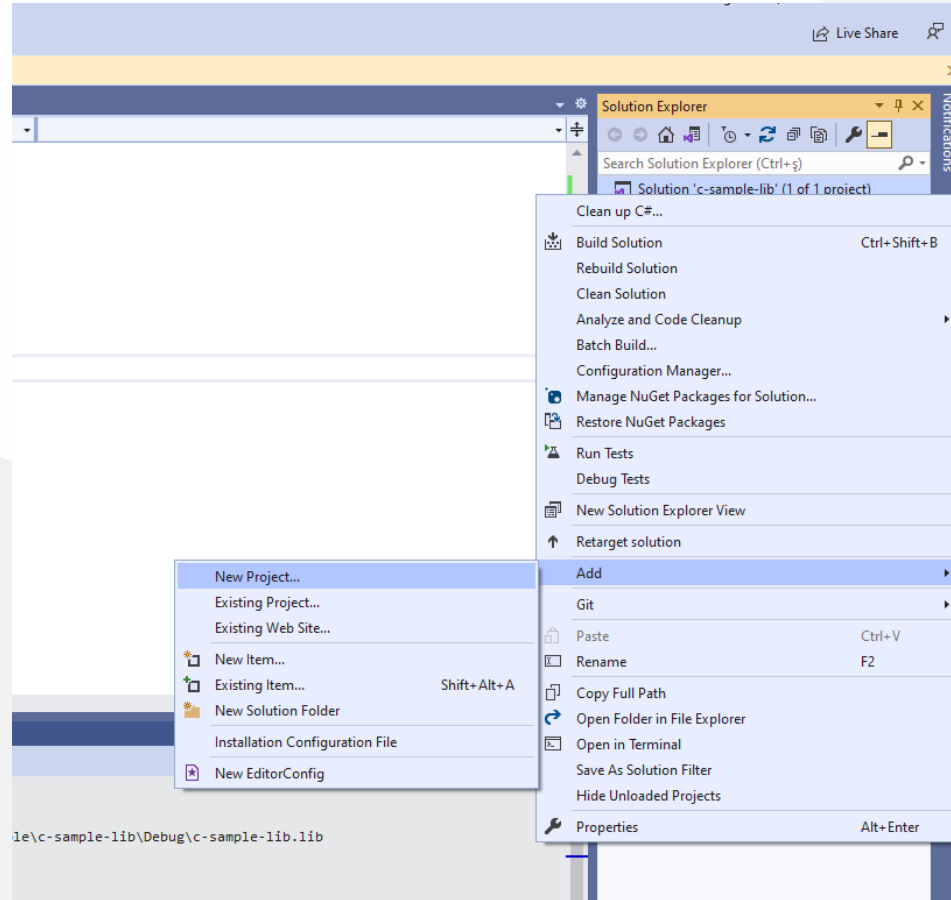
Shared Library Development - (VS C Static Library)-14

- In the debug folder, we will see our output



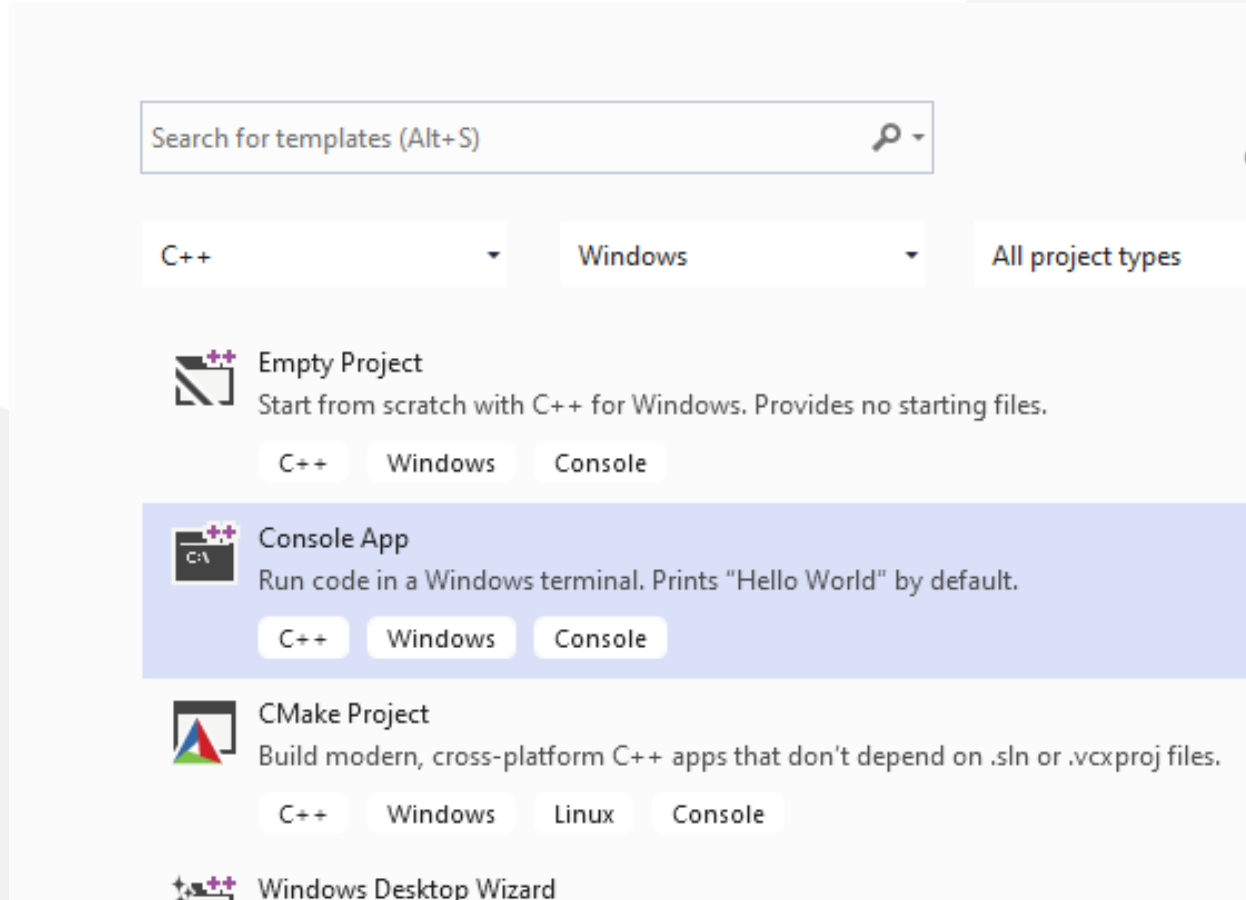
Shared Library Development - (VS C Static Library)-15

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



Shared Library Development - (VS C Static Library)-16

select C++ Windows Console Application from list



Shared Library Development - (VS C Static Library)-17

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

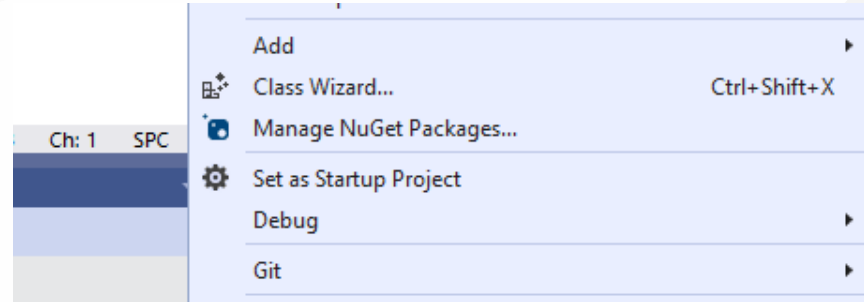
we will convert `c-sample-app.c` to following code

```
#include <stdio.h>

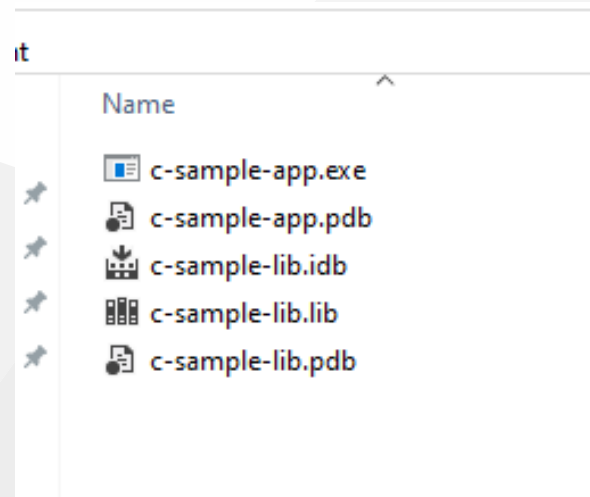
/// <summary>
///
/// </summary>
/// <returns></returns>
int main()
{
    printf("Hello World!\n");
}
```

Shared Library Development - (VS C Static Library)-18

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 the application we will see only `"Hello World"`

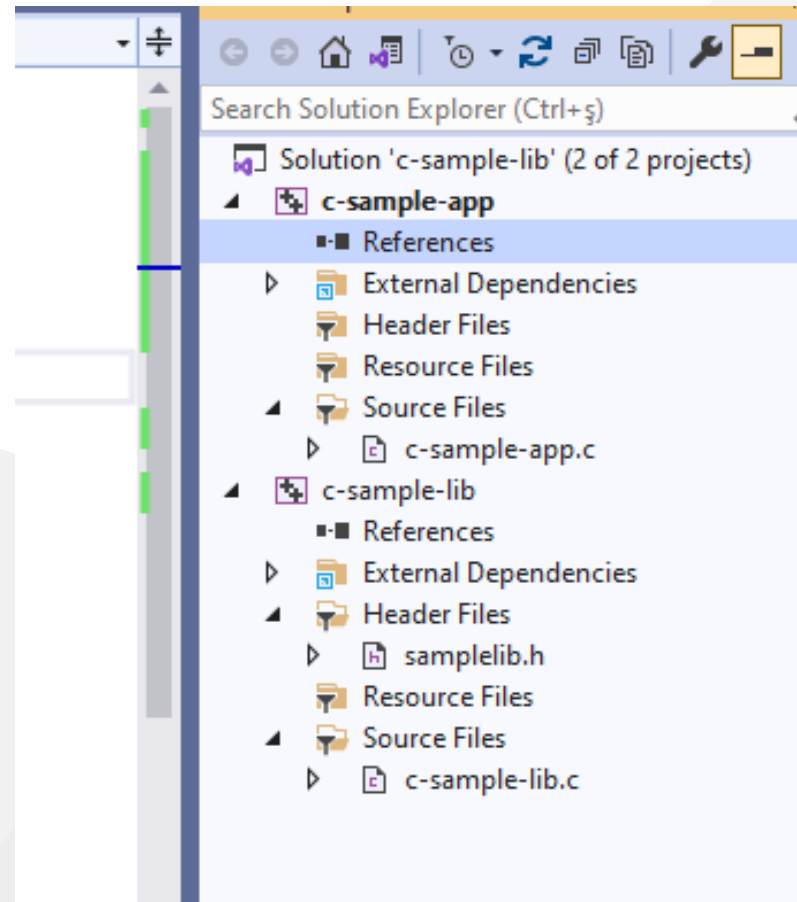
Shared Library Development - (VS C Static Library)-19

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

Shared Library Development - (VS C Static Library)-20

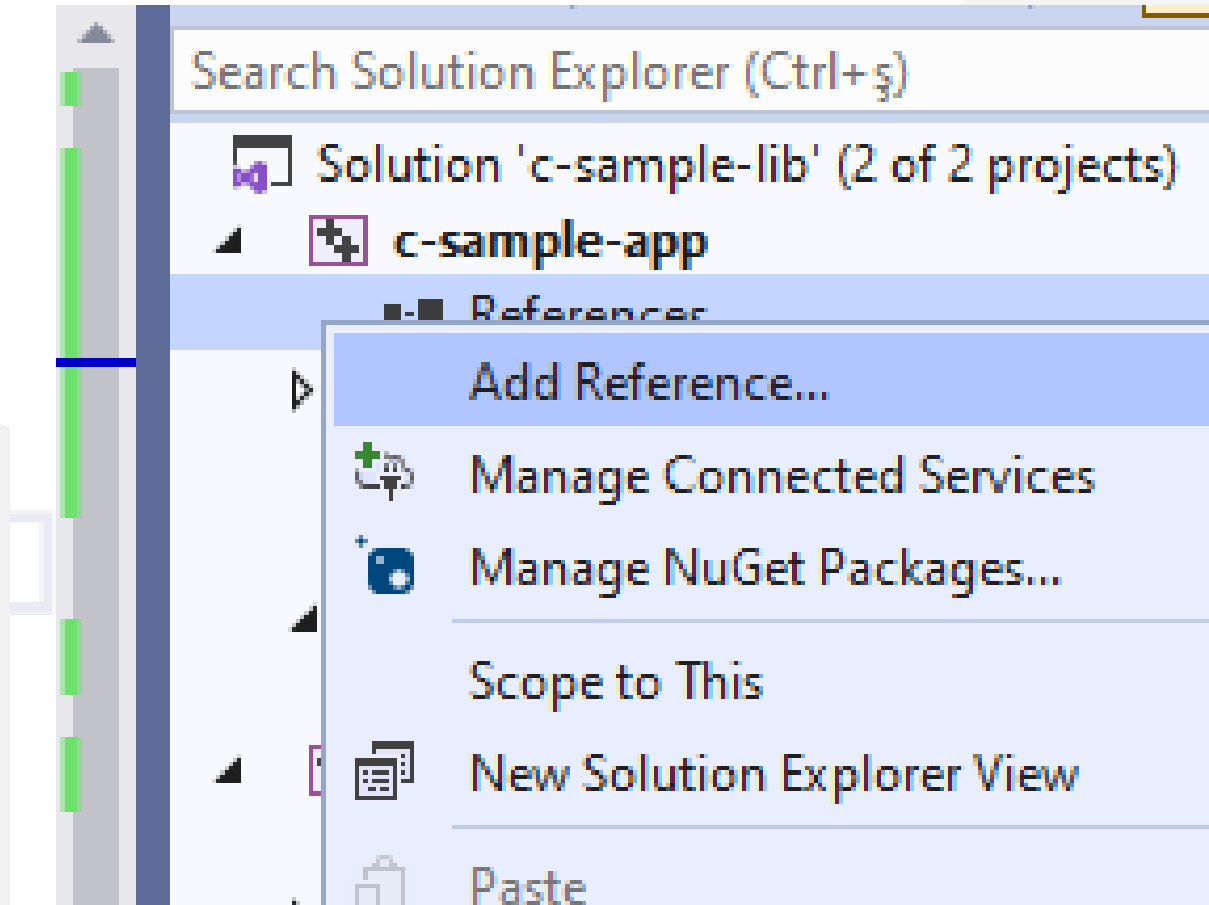
First option

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



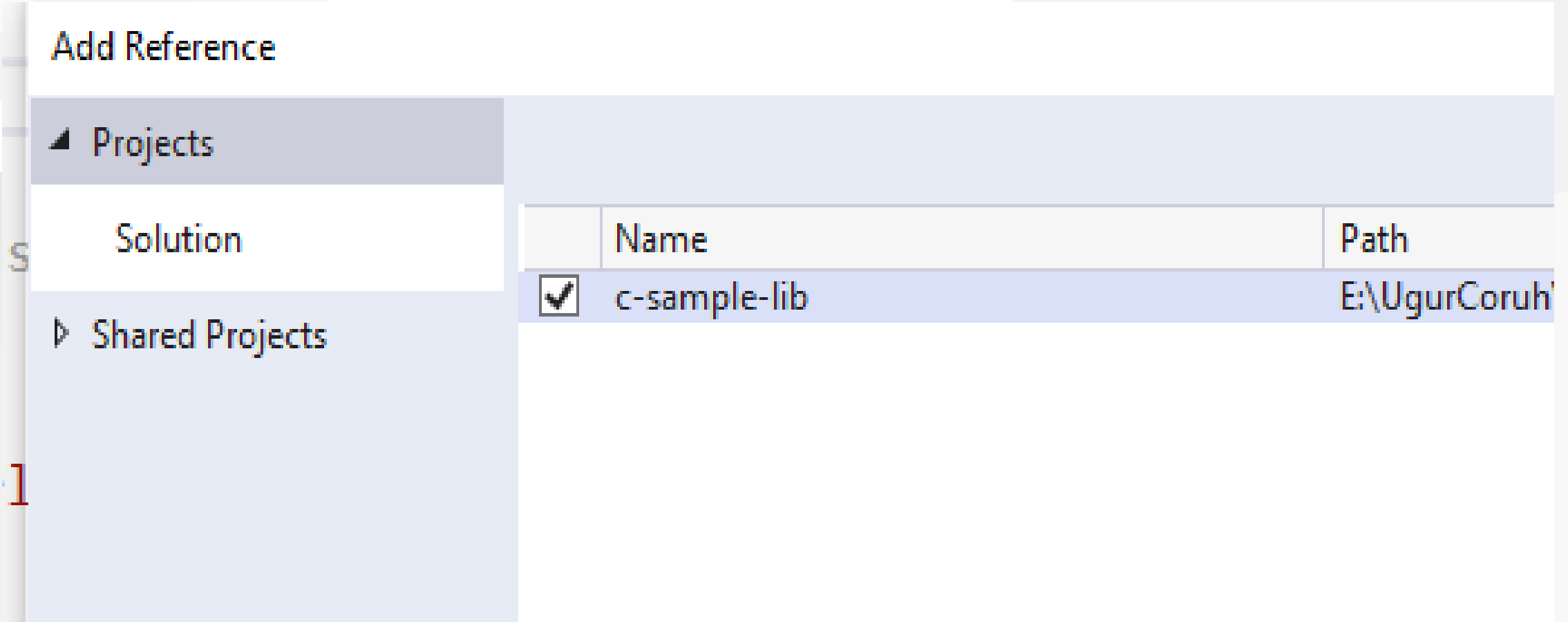
Shared Library Development - (VS C Static Library)-21

- Select Add Reference



Shared Library Development - (VS C Static Library)-22

- Browse for solution and select `c-sample-lib`

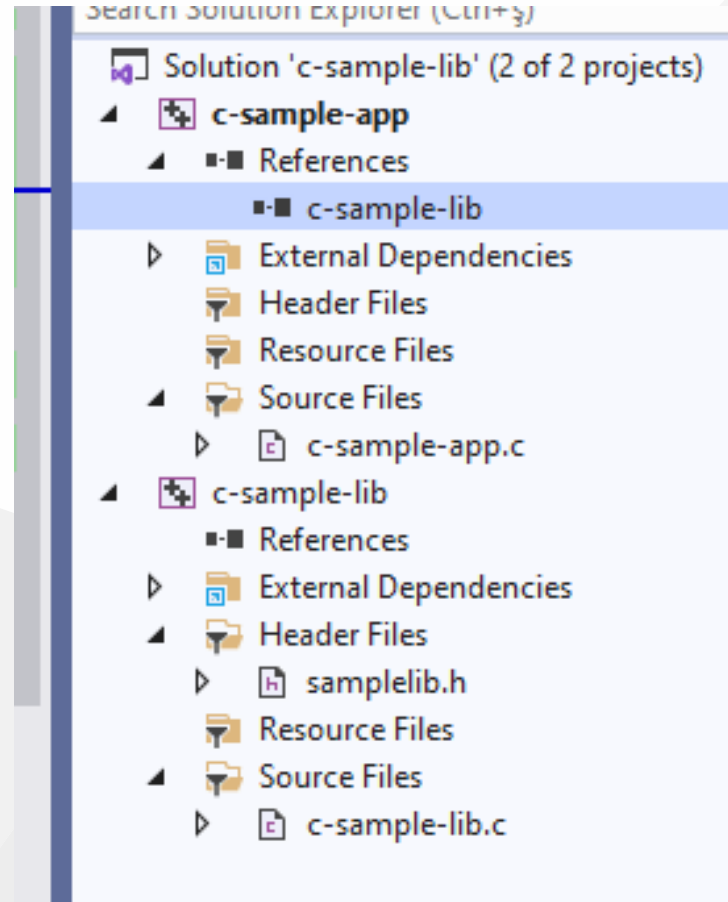


The screenshot shows the 'Add Reference' dialog box in Visual Studio. The 'Projects' section is expanded to show 'Shared Projects' containing 'c-sample-lib', which is selected with a checkmark. The 'Path' column shows 'E:\UgurCoruh'.

	Name	Path
<input checked="" type="checkbox"/>	c-sample-lib	E:\UgurCoruh

Shared Library Development - (VS C Static Library)-23

You can check added reference from references section



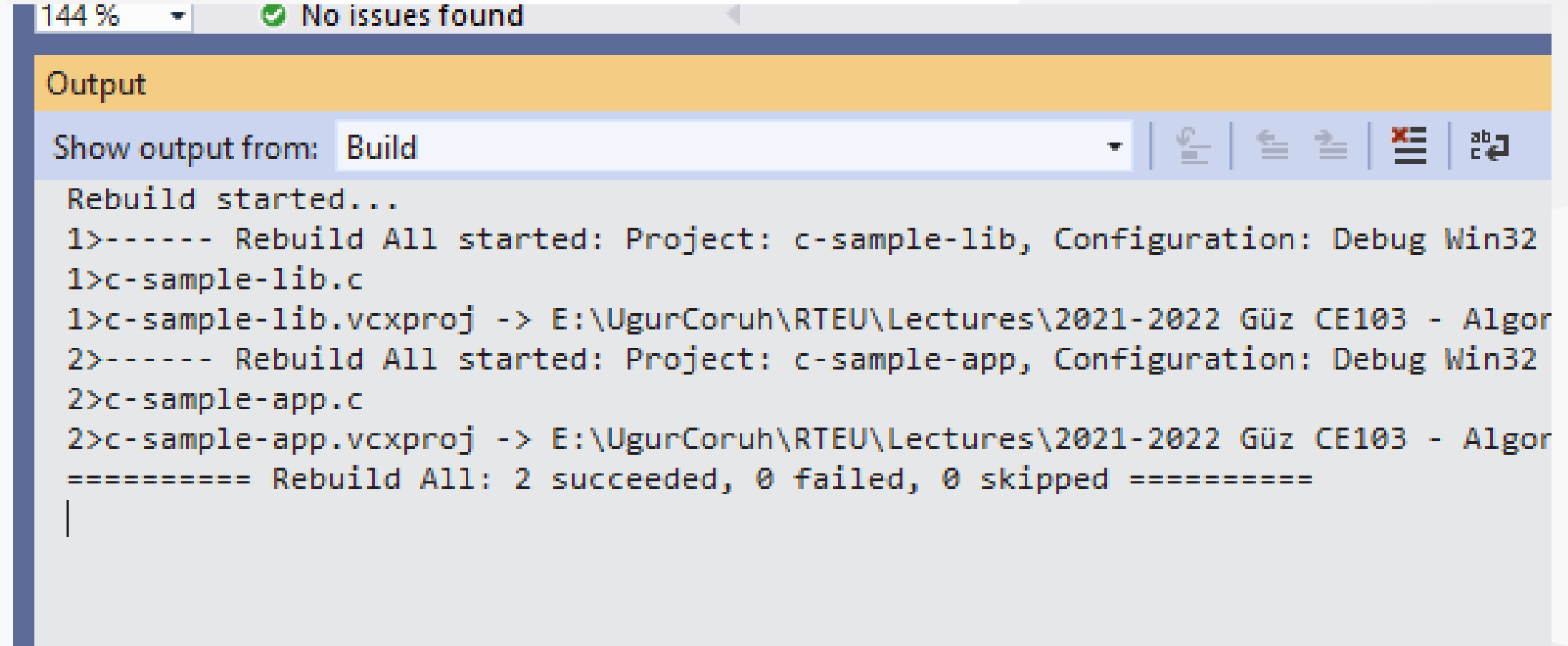
Shared Library Development - (VS C Static Library)-24

- 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");
}
```

Shared Library Development - (VS C Static Library)-25

- we can build our `c-sample-app`



The screenshot shows the Visual Studio Output window with a zoom level of 144% and a status bar indicating 'No issues found'. The output text is as follows:

```
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 =====
|
```

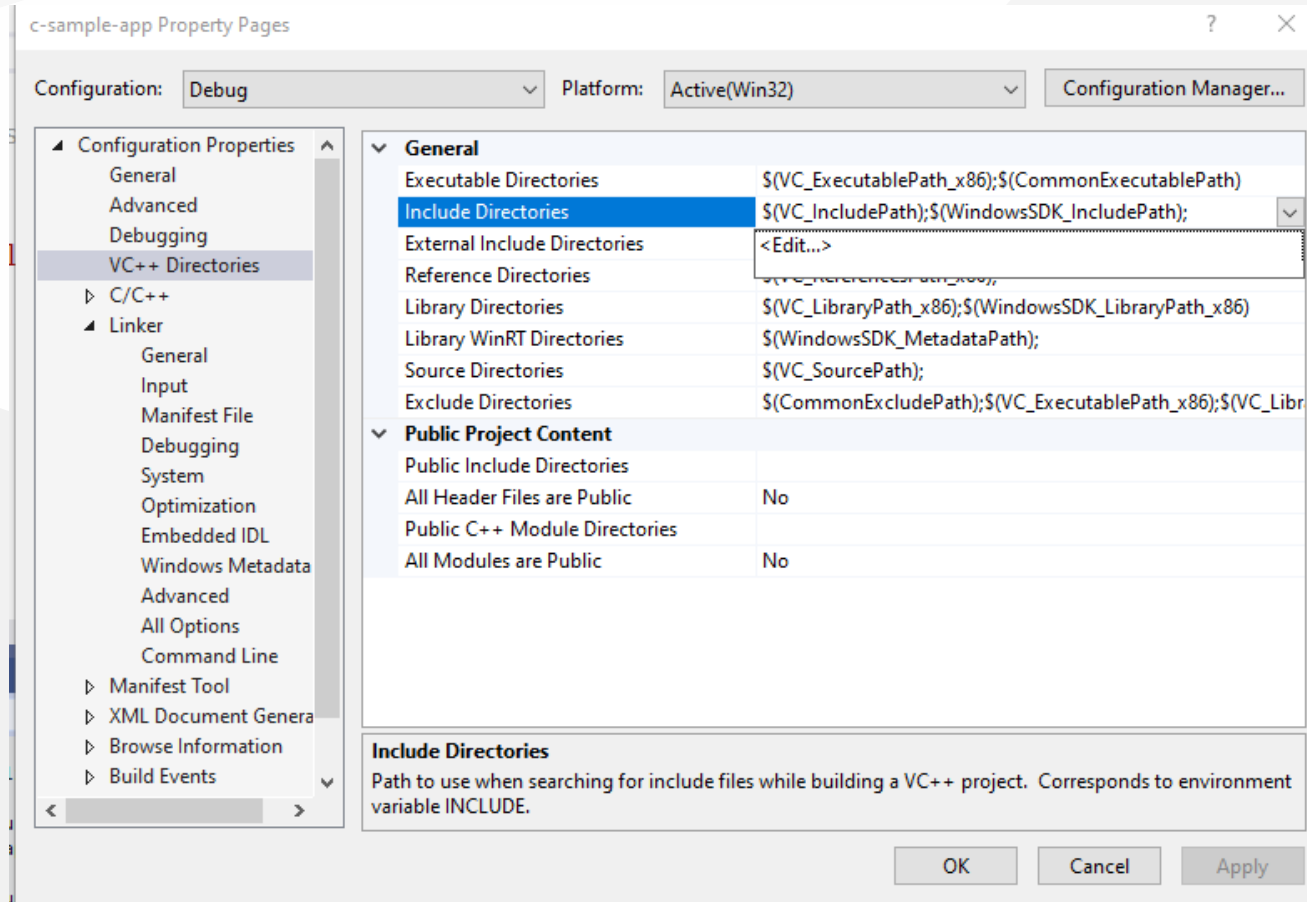
Shared Library Development - (VS C Static Library)-26

- Also we can only write header name

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

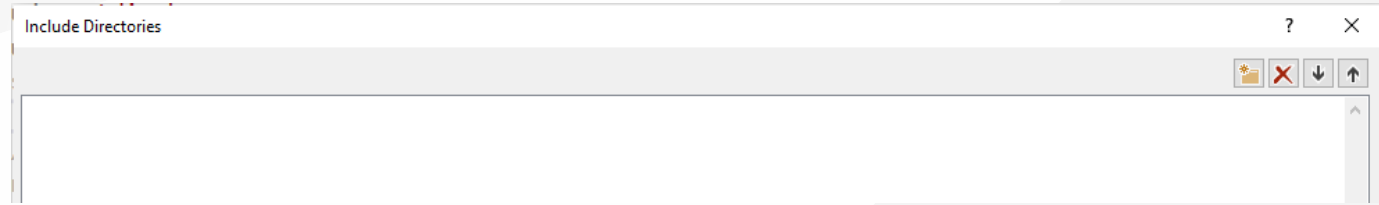

Shared Library Development - (VS C Static Library)-27

- For this option, we need to configure include directories

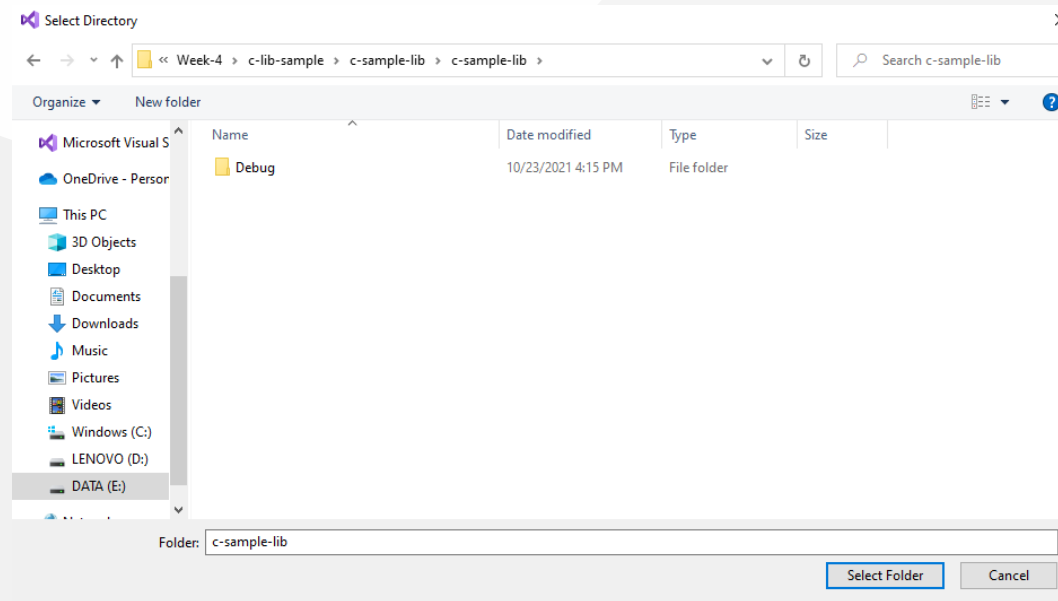


Shared Library Development - (VS C Static Library)-28

select c-sample-lib header file location

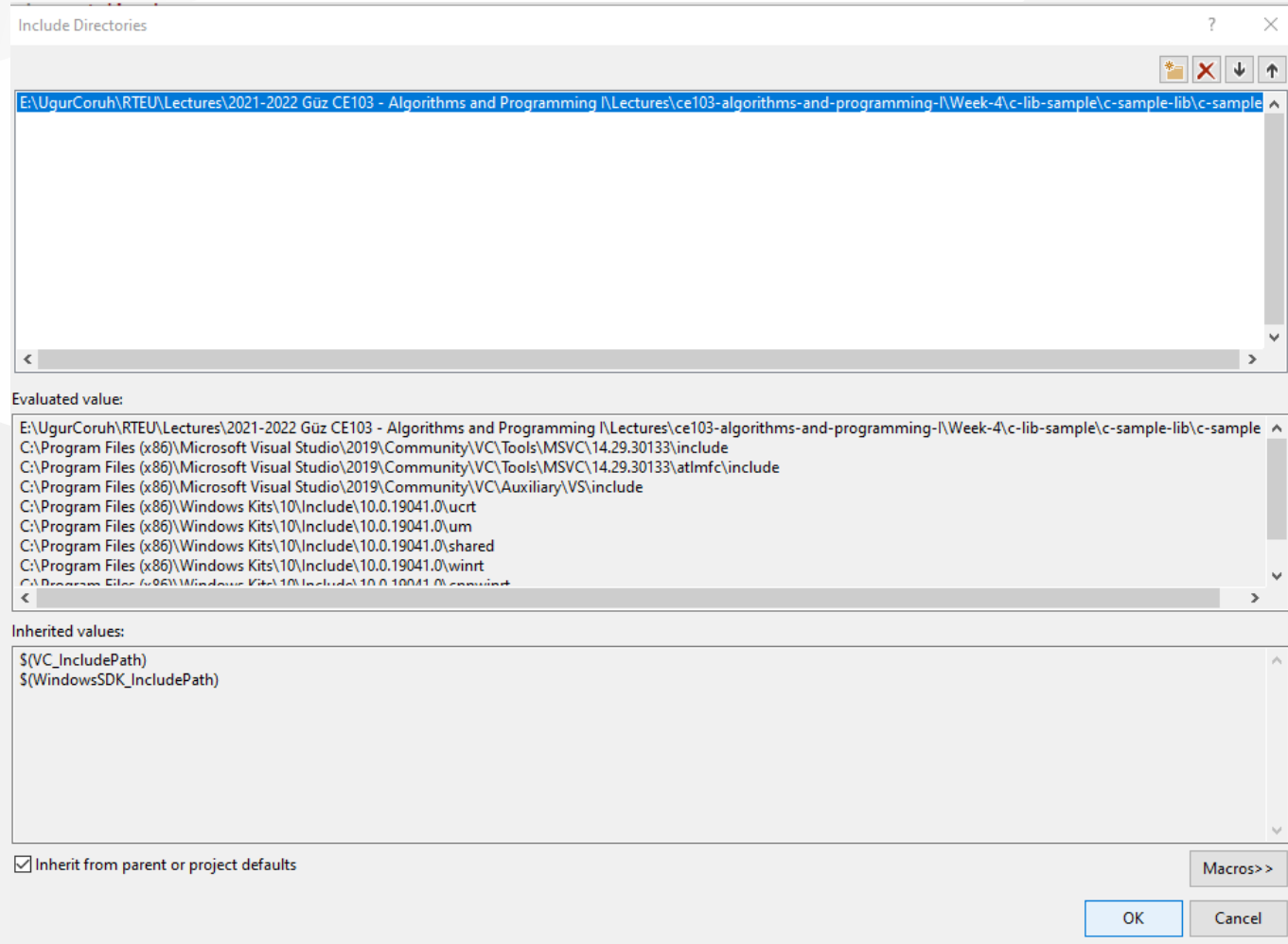


browse for folder



Shared Library Development - (VS C Static Library)-29

your full path will be added to your configuration



Shared Library Development - (VS C Static Library)-30

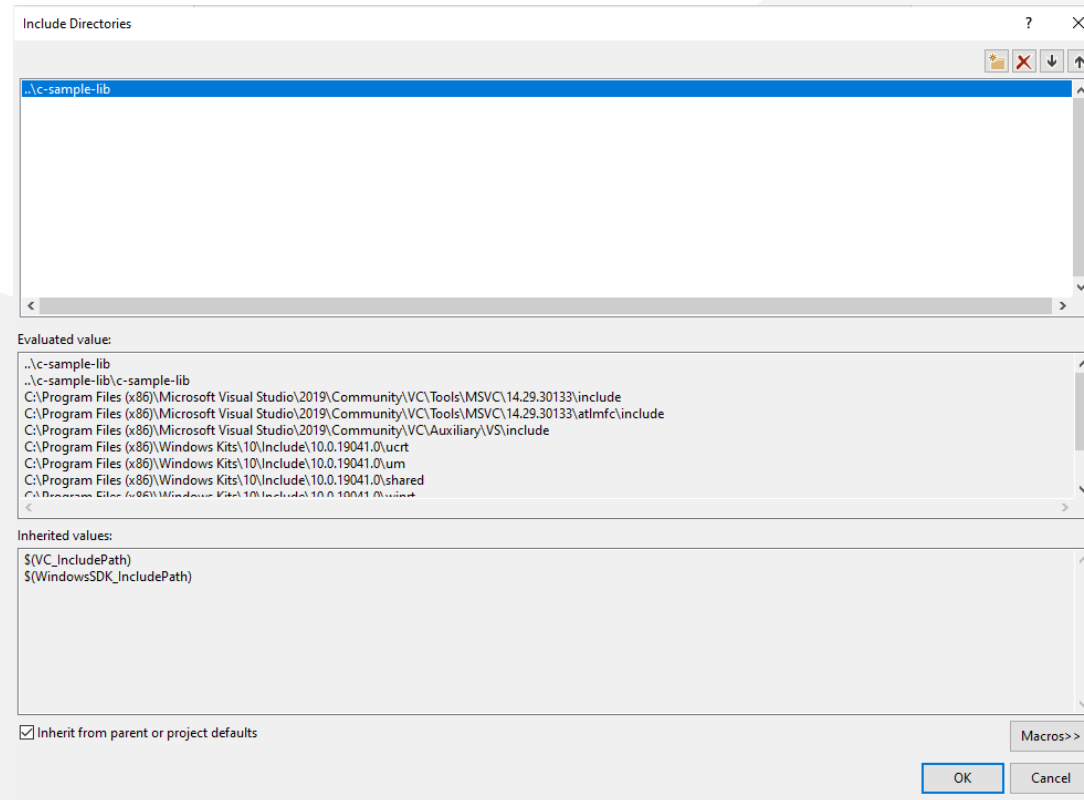
if you add header file paths to your configuration you can use header files by name in your source code

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

Shared Library Development - (VS C Static Library)-31

- 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
```



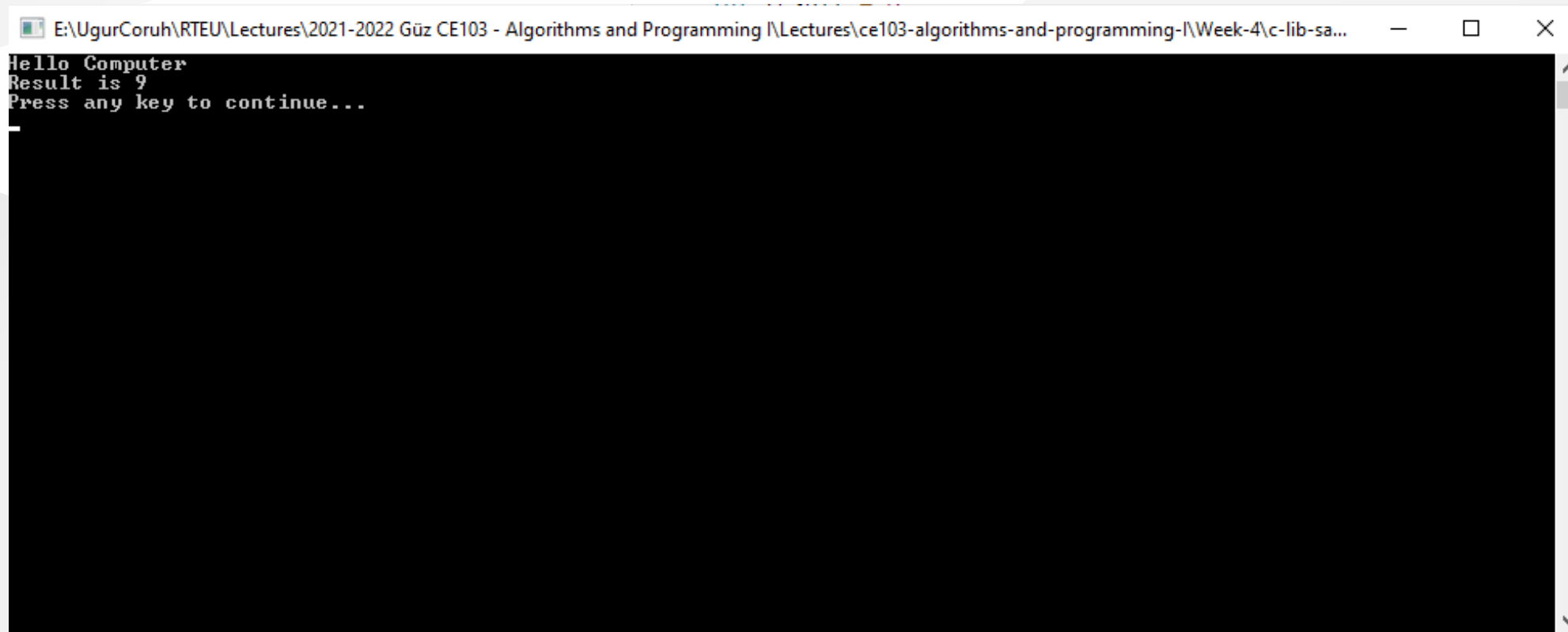
Shared Library Development - (VS C Static Library)-32

- 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;
}
```

Shared Library Development - (VS C Static Library)-33

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



The screenshot shows a Windows command prompt window with the following text:

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

Shared Library Development - (VS C Static Library)-34

- A 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. In another case you can use a precompiled static library with you or this library can be part of any installation then if there is an installed app and static libraries are placed on the system folder or any different location then you can use configuration files to set library path and included header paths

Shared Library Development - (VS C Static Library)-35

- Now we can remove the 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
```

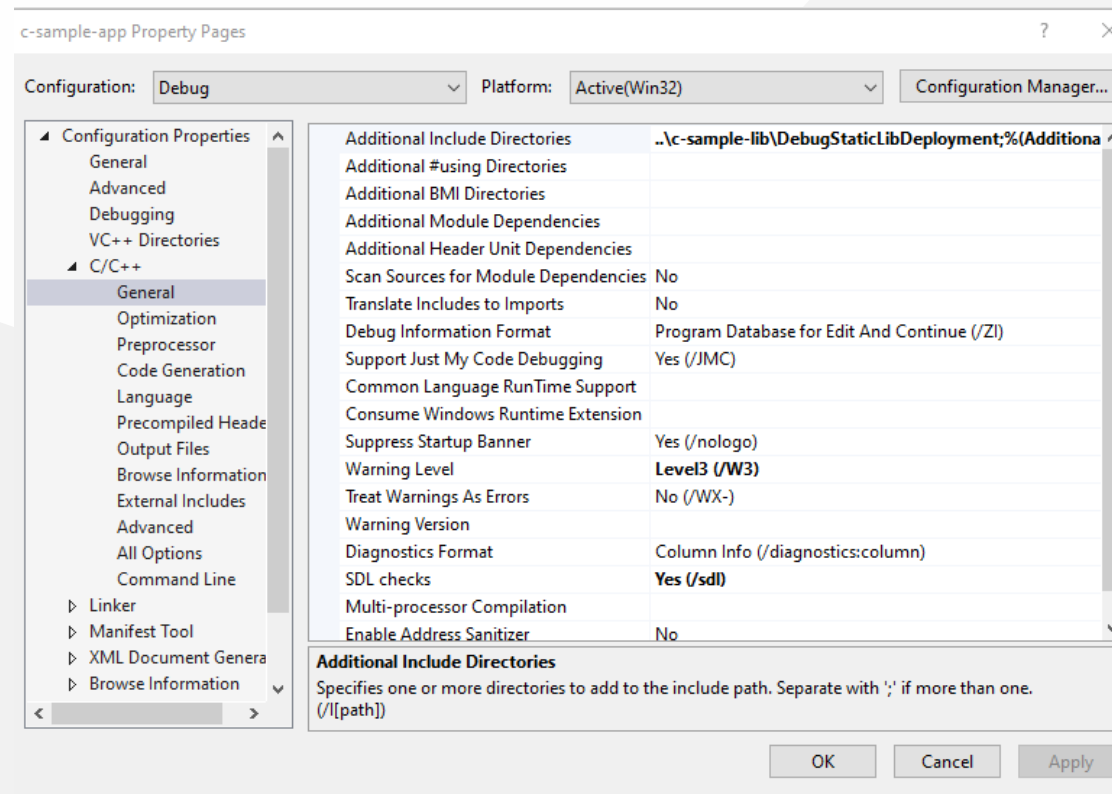
- Set C/C++ -> General -> Additional Include Directories

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

Shared Library Development - (VS C Static Library)-36

Not Working Solution

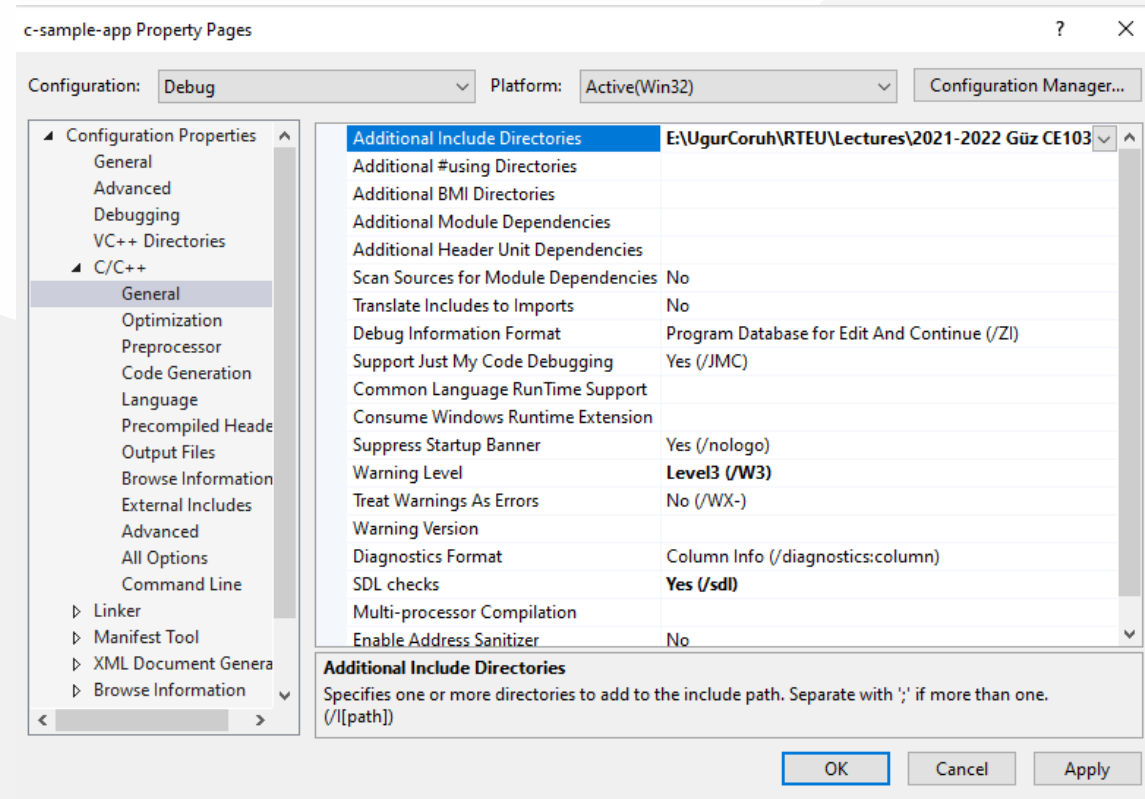
```
..\c-sample-lib\DebugStaticLibDeployment
```



Shared Library Development - (VS C Static Library)-37

Working Solution

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



Shared Library Development - (VS C Static Library)-38

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 Solution

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

The screenshot shows the 'c-sample-app Property Pages' dialog box. The 'Configuration' is set to 'Debug' and the 'Platform' is 'Active(Win32)'. The 'VC++ Directories' tab is selected, and the 'Library Directories' field is highlighted with the path 'E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - A'. The 'General' section includes fields for Executable Directories, Include Directories, External Include Directories, Reference Directories, Library Directories, Library WinRT Directories, Source Directories, and Exclude Directories. The 'Public Project Content' section includes Public Include Directories, All Header Files are Public, Public C++ Module Directories, and All Modules are Public. The 'Library Directories' field is currently empty, and the path is shown in a dropdown menu.

Property	Value
Executable Directories	\$(VC_ExecutablePath_x86);\$(CommonExecutablePath)
Include Directories	\$(IncludePath)
External Include Directories	\$(ExternalIncludePath)
Reference Directories	\$(VC_ReferencesPath_x86);
Library Directories	E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - A
Library WinRT Directories	\$(WindowsSDK_MetadataPath);
Source Directories	\$(VC_SourcePath);
Exclude Directories	\$(CommonExcludePath);\$(VC_ExecutablePath_x86);\$(VC_Libr

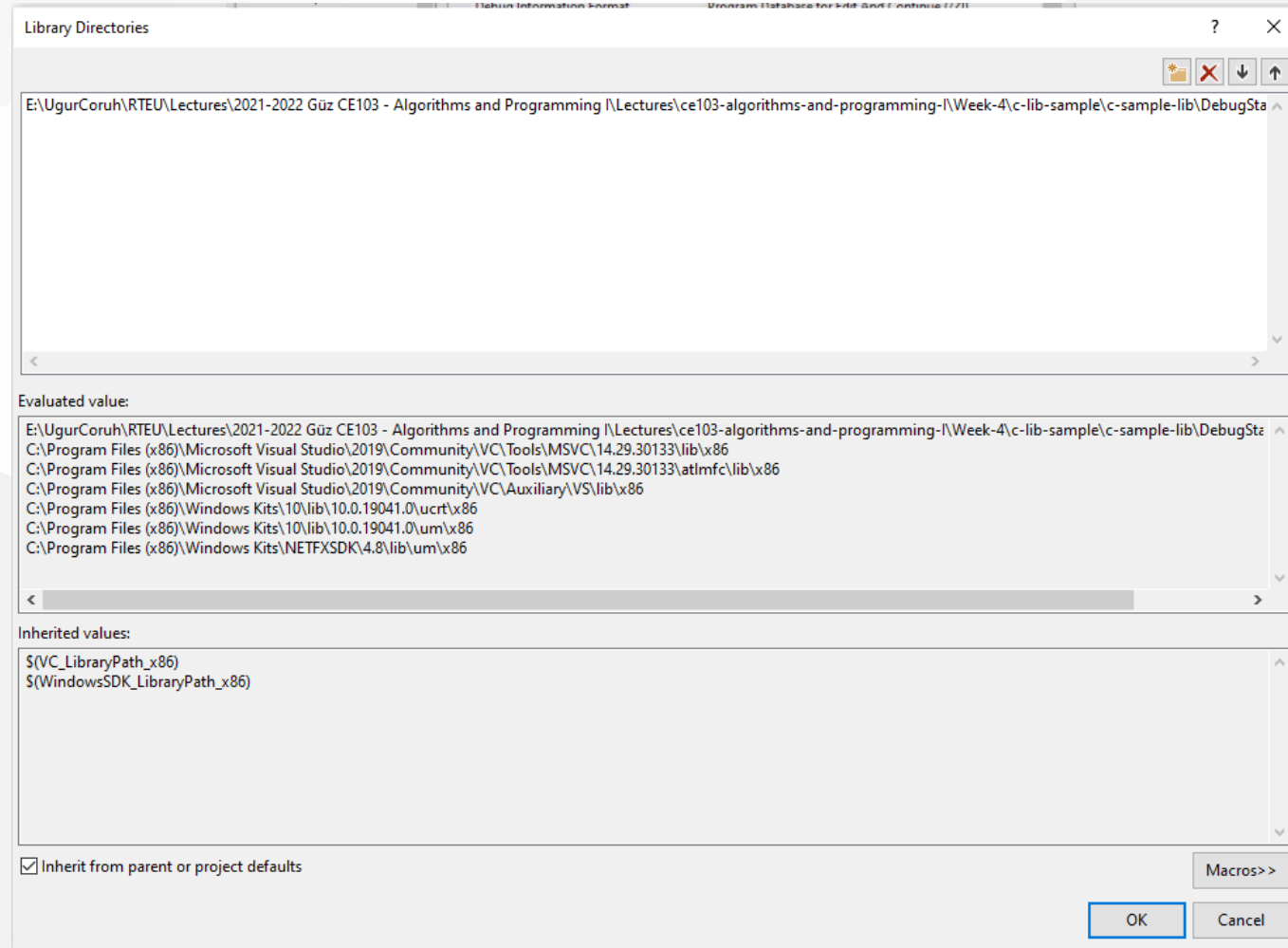
Public Project Content

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

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

OK Cancel Apply

Shared Library Development - (VS C Static Library)-40



Not Working

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

c-sample-app Property Pages

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

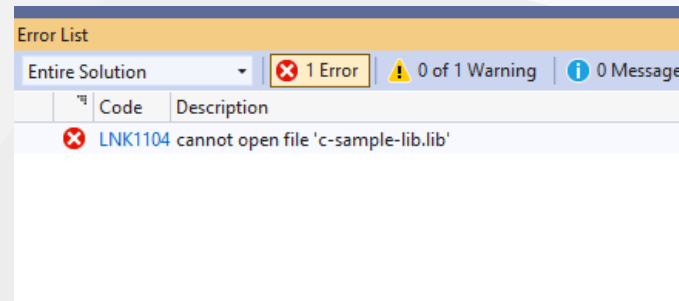
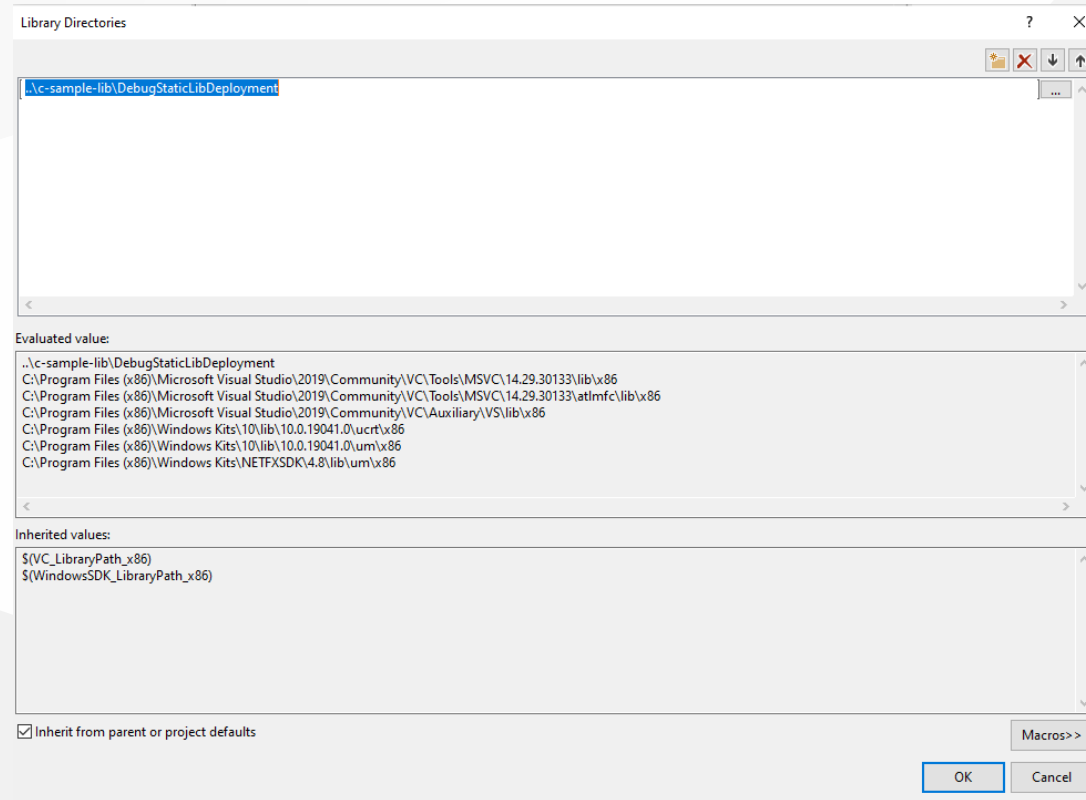
- Configuration Properties
 - General
 - Advanced
 - Debugging
 - VC++ Directories
 - C/C++
 - Linker
 - Manifest Tool
 - XML Document Generator
 - Browse Information
 - Build Events
 - Custom Build Step
 - Code Analysis

General	
Executable Directories	\$(VC_ExecutablePath_x86);\$(CommonExecutablePath)
Include Directories	\$(IncludePath)
External Include Directories	\$(ExternalIncludePath)
Reference Directories	\$(VC_ReferencesPath_x86);
Library Directories	..\c-sample-lib\DebugStaticLibDeployment;\$(LibraryPath)
Library WinRT Directories	\$(WindowsSDK_MetadataPath);
Source Directories	\$(VC_SourcePath);
Exclude Directories	\$(CommonExcludePath);\$(VC_ExecutablePath_x86);\$(VC_Libr
Public Project Content	
Public Include Directories	
All Header Files are Public	No
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

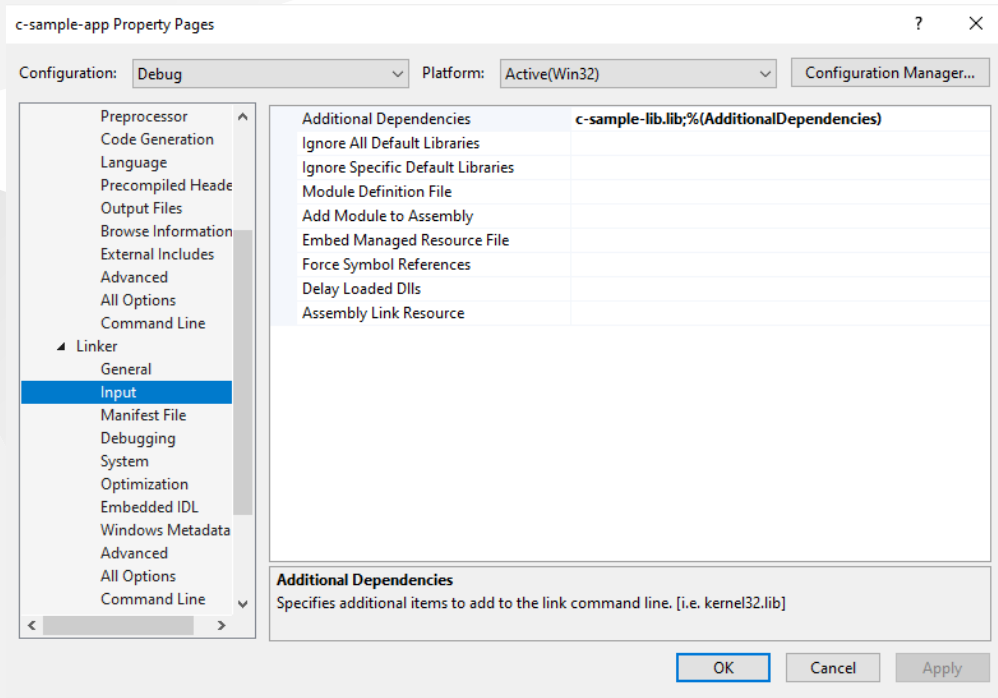
Shared Library Development - (VS C Static Library)-42



Shared Library Development - (VS C Static Library)-43

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 a different location and they are ready to use.

Shared Library Development - (VS C Static Library)-44

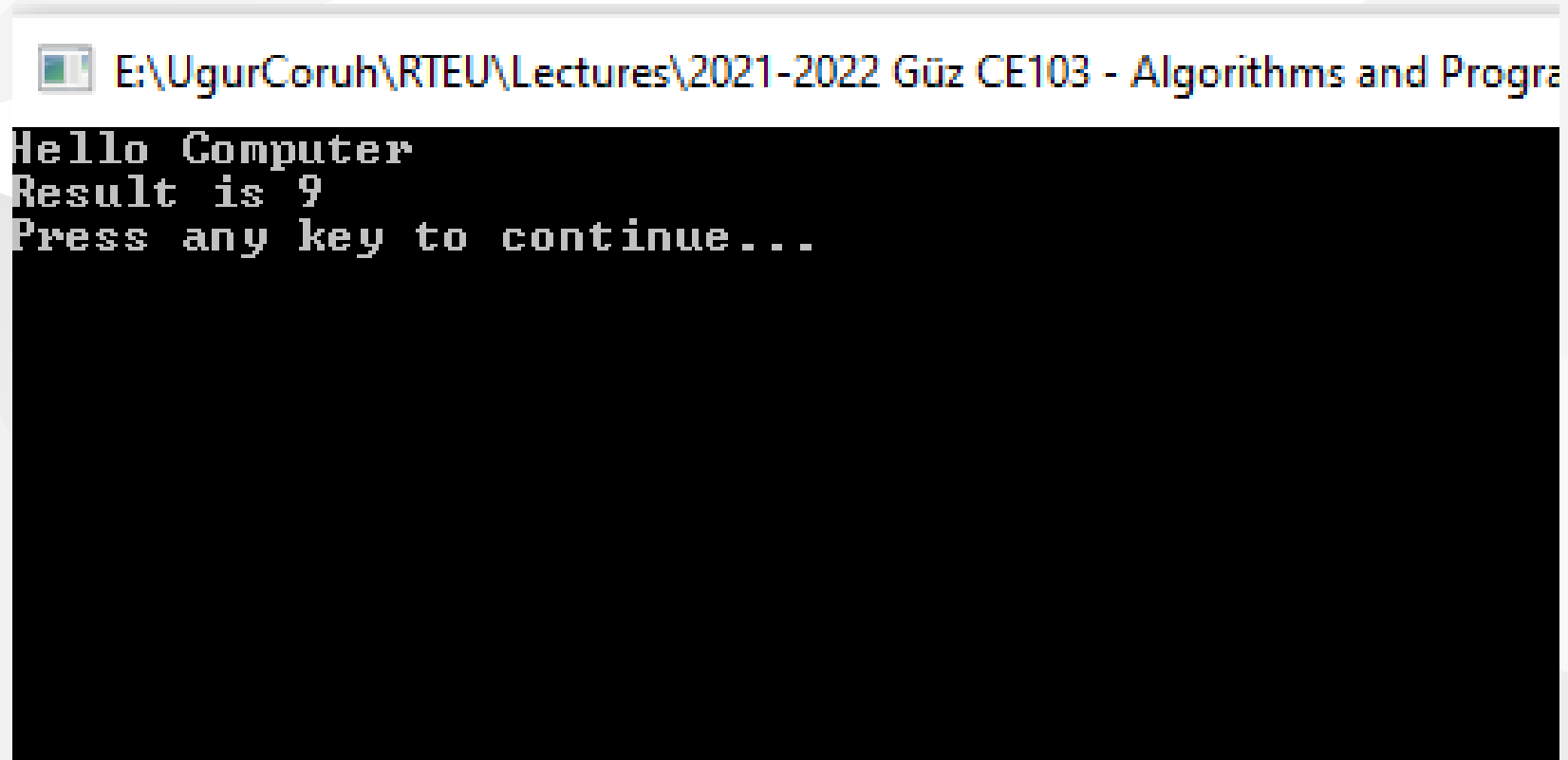
current source code will be like that nothing changed

```
#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;
}
```

Shared Library Development - (VS C Static Library)-45

- and output will be as follow



The screenshot shows a Windows command prompt window with the following text:

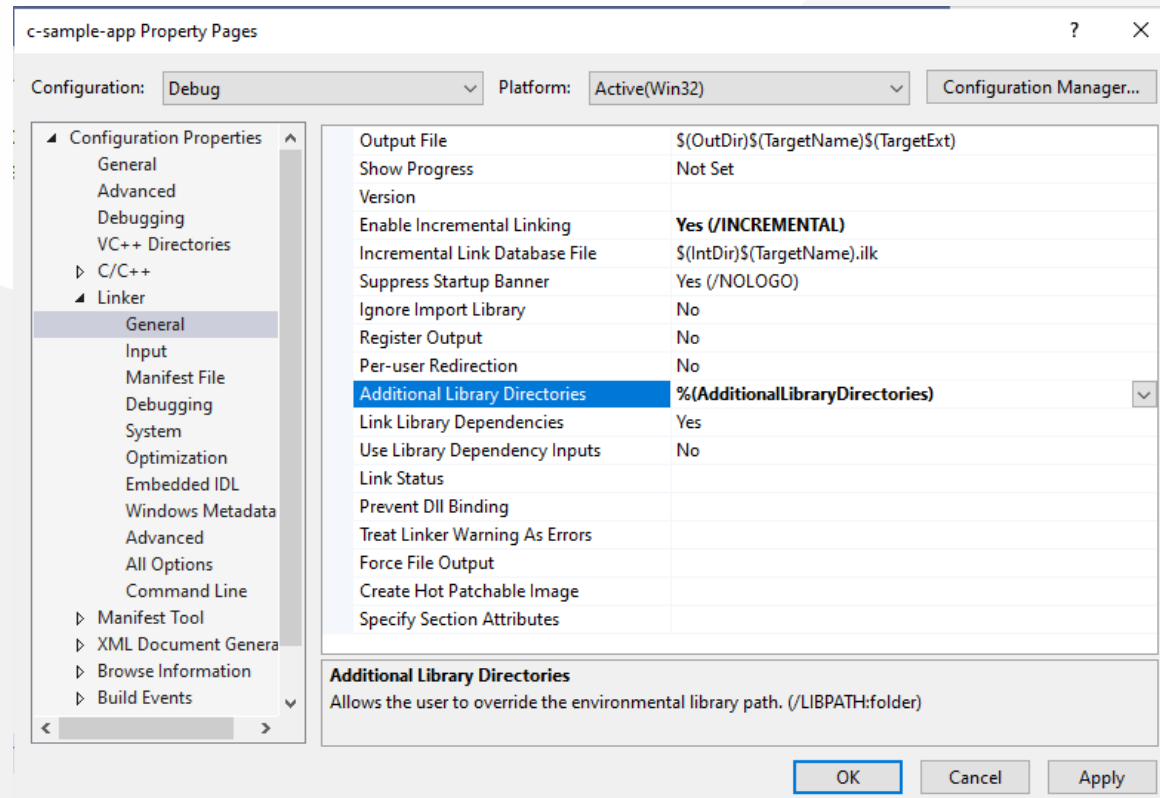
```
E:\UgurCoruh\RTEU\Lectures\2021-2022 Güz CE103 - Algorithms and Progra  
Hello Computer  
Result is 9  
Press any key to continue...
```

Shared Library Development - (VS C Static Library)-46

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

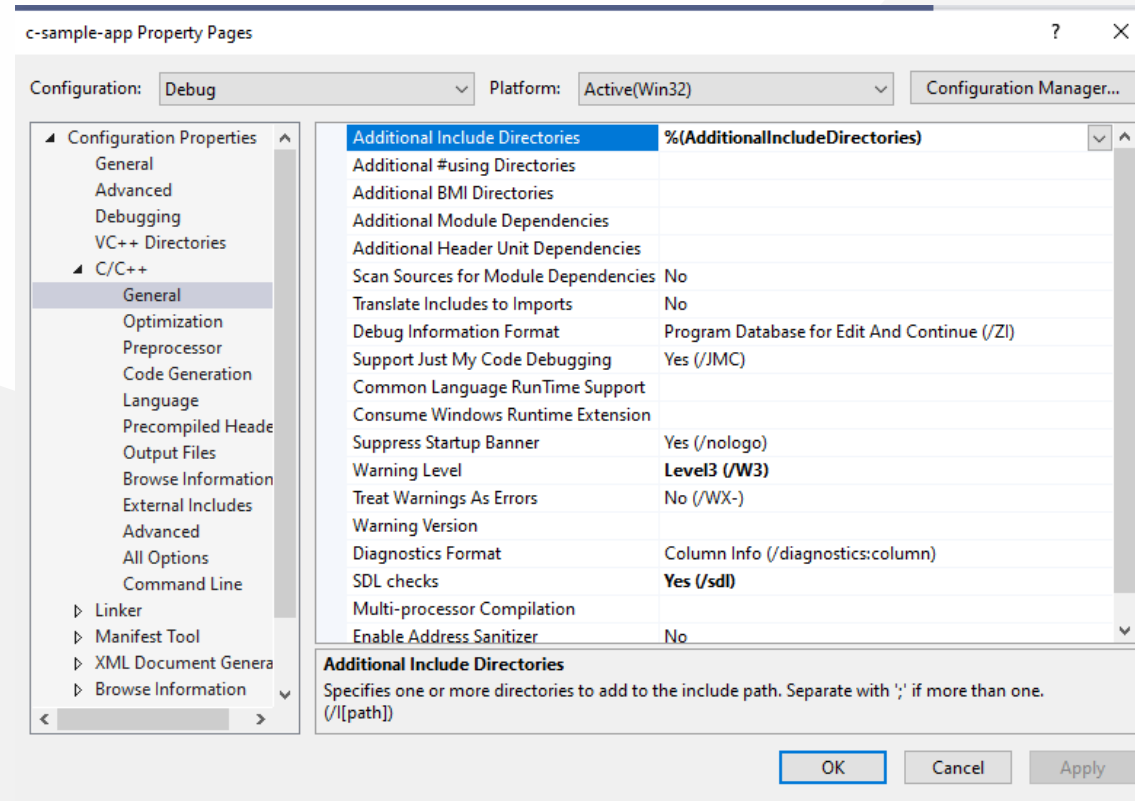
We will remove all library related settings from configurations and we will write them in source code

Clear linker->general->additional library directories



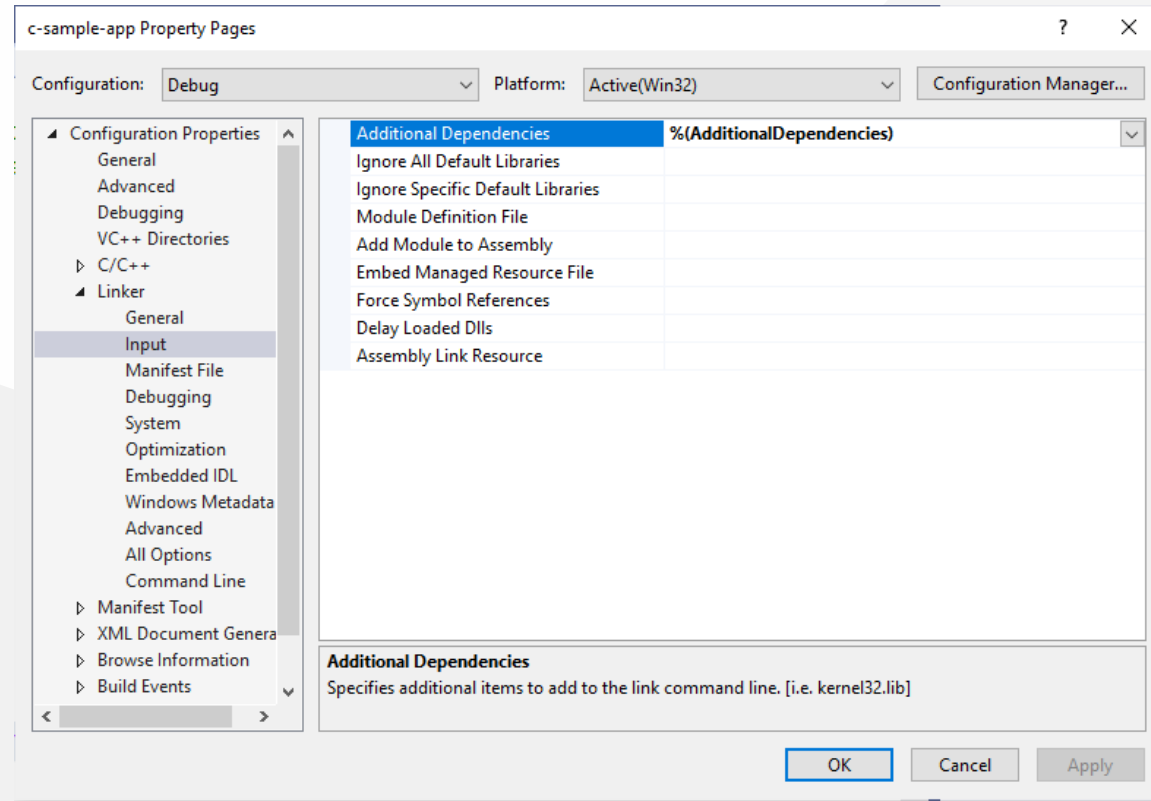
Shared Library Development - (VS C Static Library)-47

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



Shared Library Development - (VS C Static Library)-48

Clear Linker->Input->Additional Dependencies



Shared Library Development - (VS C Static Library)-49

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.

Shared Library Development

C++ Programming (Static Library)

Visual Studio Community Edition

Shared Library Development - (VS Cpp Static Library)-1

- 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 compiled version.

Shared Library Development - (VS Cpp Static Library)-2

- 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;
}
```

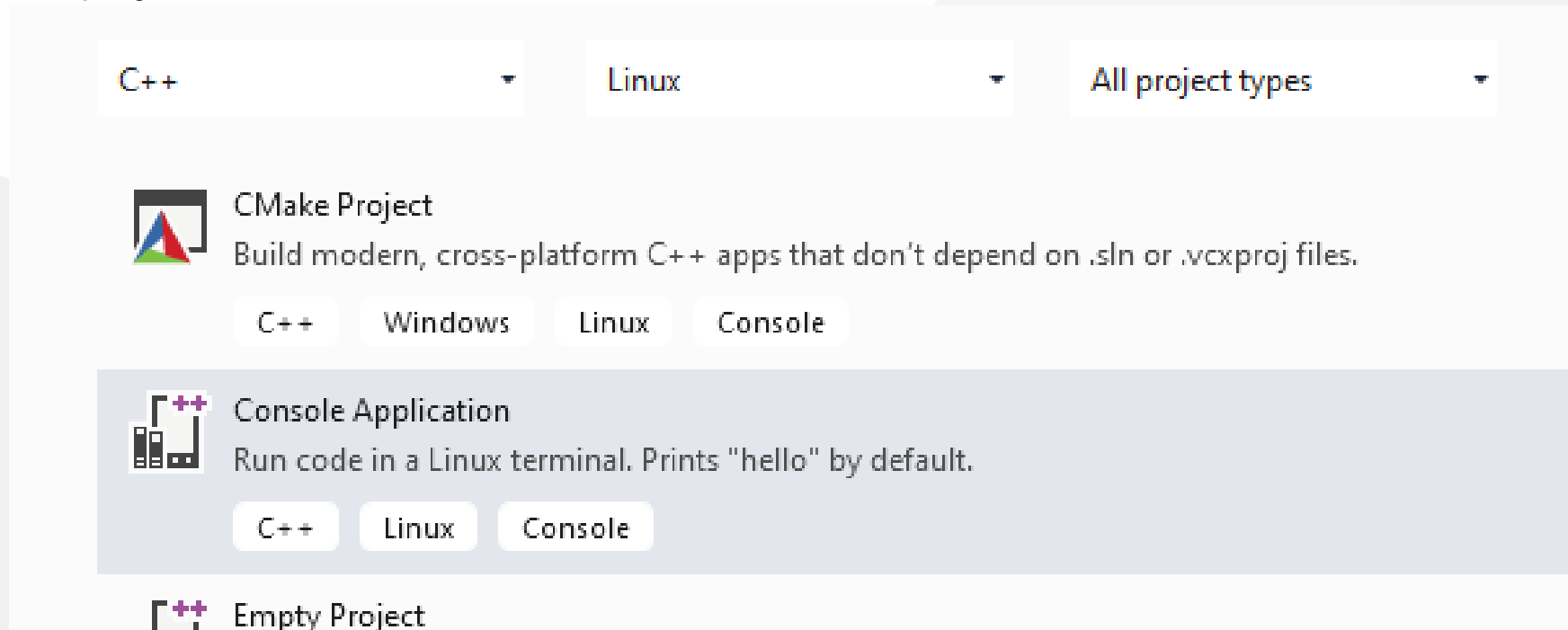
Shared Library Development

C++ Programming (Static Library)

Visual Studio Community Edition WSL Option

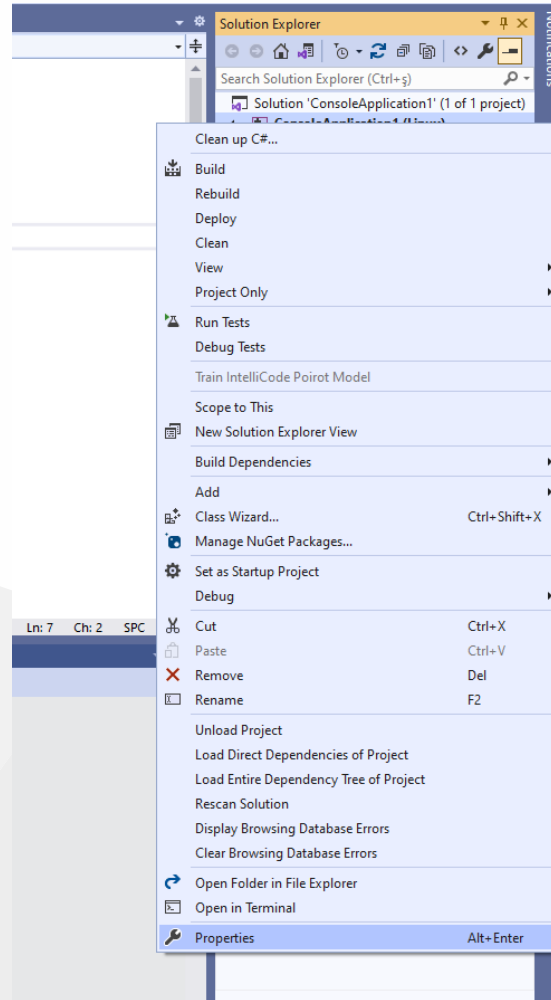
Shared Library Development - (VS Cpp WSL Static Library)-1

- Install WSL2
 - [GitHub - ucoruh/ns3-wsl-win10-setup: ns3 windows 10 WSL2 setup and usage](#)
- Create a Linux project



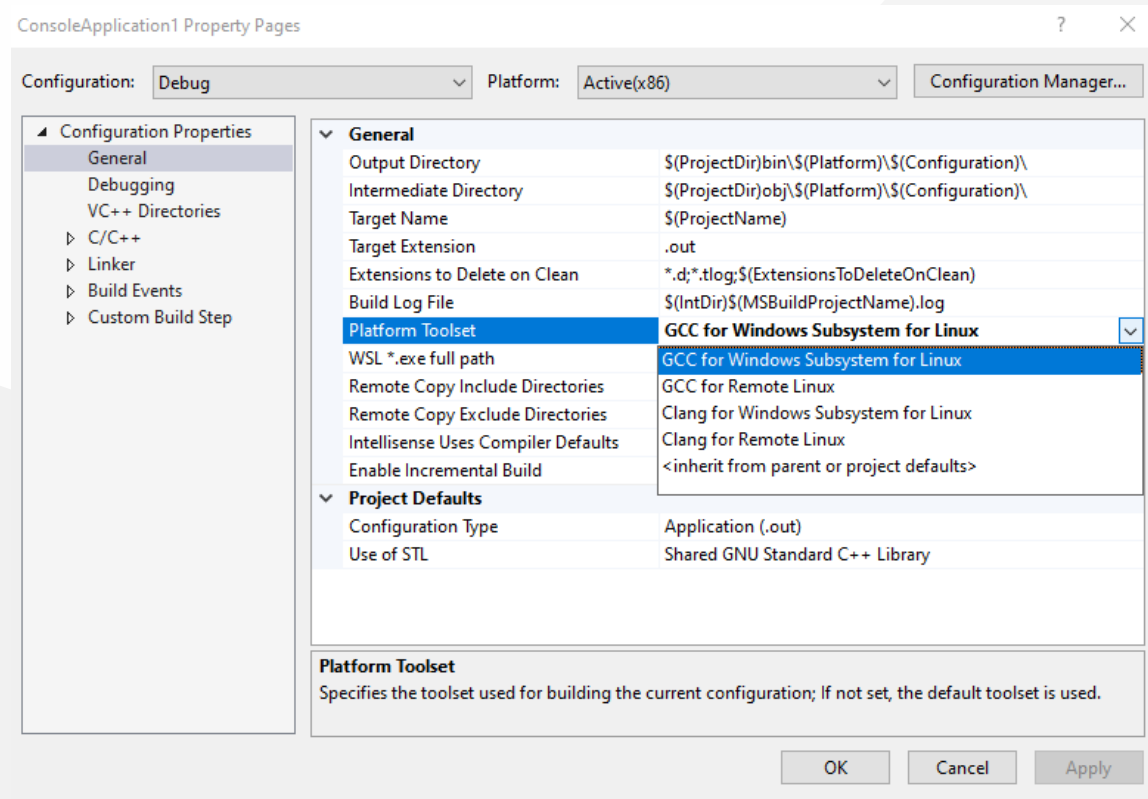
Shared Library Development - (VS Cpp WSL Static Library)-2

- Configure Platform Toolset to WSL



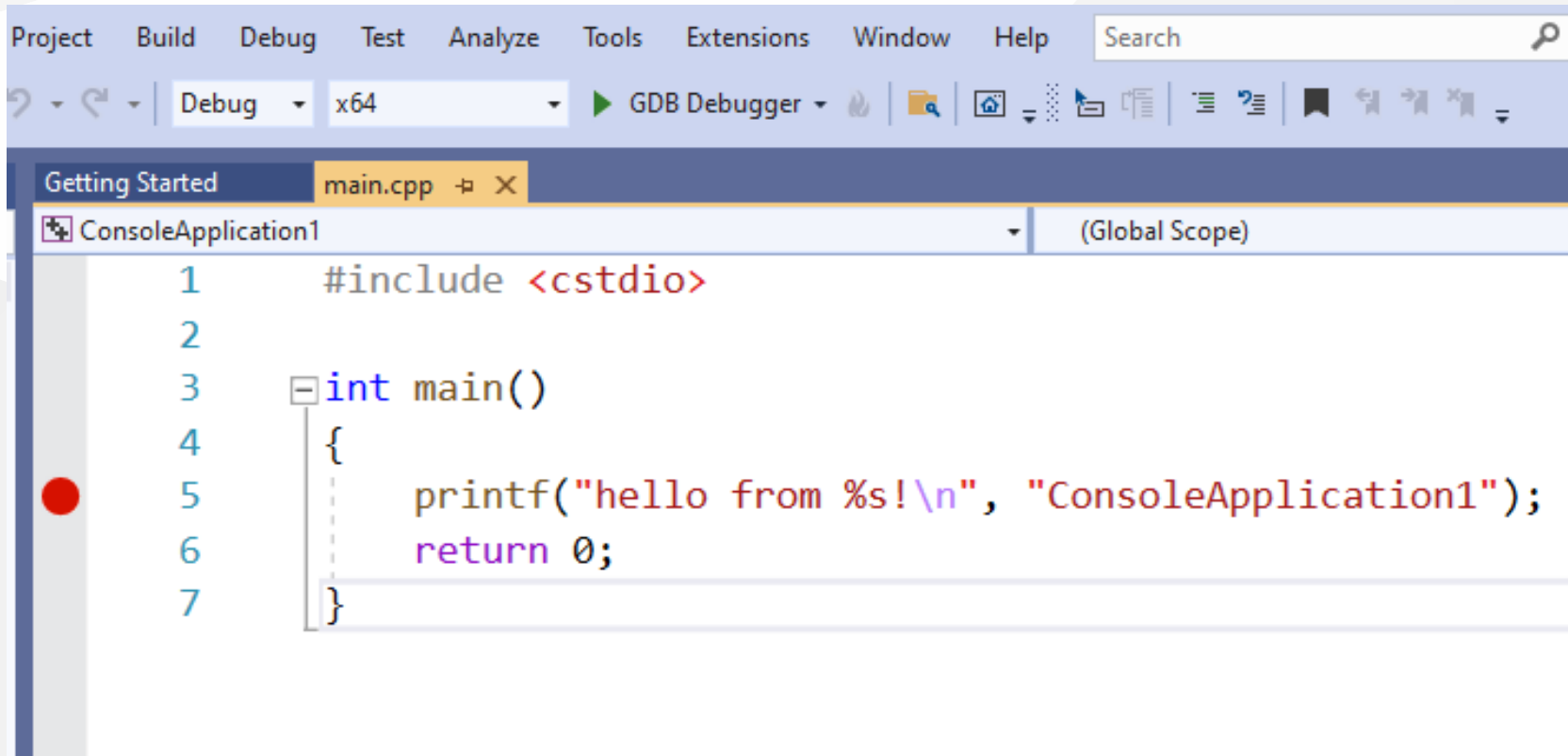
Shared Library Development - (VS Cpp WSL Static Library)-3

- Select GCC for Windows Subsystem for Linux



Shared Library Development - (VS Cpp WSL Static Library)-4

Put a breakpoint and run debugger



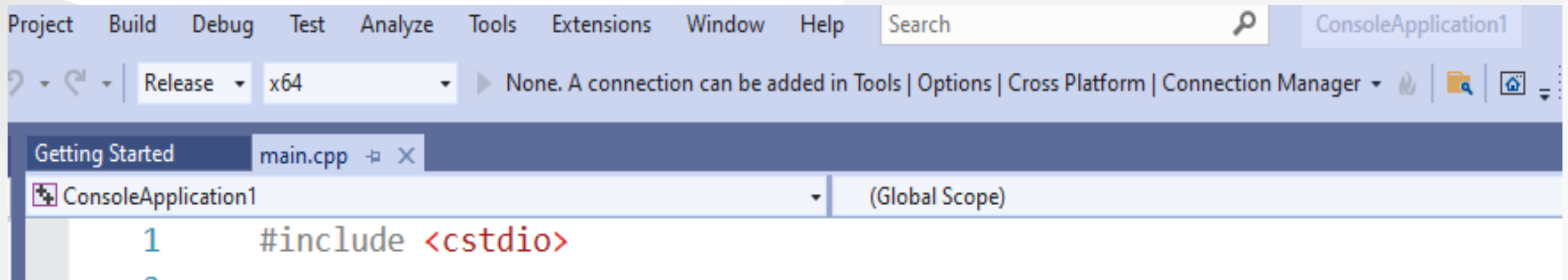
The screenshot shows the Visual Studio Code interface with a C++ project named 'ConsoleApplication1'. The code in 'main.cpp' is as follows:

```
1  #include <cstdio>
2
3  int main()
4  {
5      printf("hello from %s!\n", "ConsoleApplication1");
6      return 0;
7  }
```

A red circular breakpoint is set on line 5. The interface also shows the 'Debug' menu, 'x64' architecture, and 'GDB Debugger' selected.

Shared Library Development - (VS Cpp WSL Static Library)-5

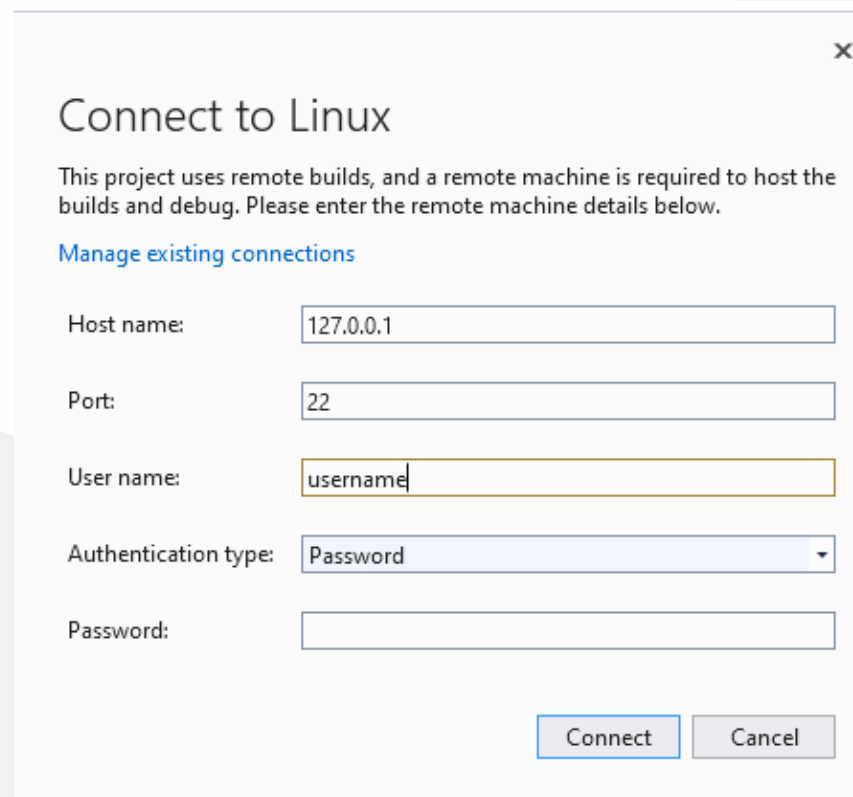
In the debugger for WSL you can use local WSL installation but if you want to run it on Release setting it require a SSH connection.



```
Project Build Debug Test Analyze Tools Extensions Window Help Search ConsoleApplication1
Release x64 None. A connection can be added in Tools | Options | Cross Platform | Connection Manager
Getting Started main.cpp
ConsoleApplication1 (Global Scope)
1 #include <cstdio>
```


Shared Library Development - (VS Cpp WSL Static Library)-6

- Configure SSH parameters



Connect to Linux

This project uses remote builds, and a remote machine is required to host the builds and debug. Please enter the remote machine details below.

[Manage existing connections](#)

Host name:

Port:

User name:

Authentication type:

Password:

Shared Library Development - (VS Cpp WSL Static Library)-7

- 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
 - [Bağlan hedef Linux sistemimize Visual Studio | Microsoft Docs](#)

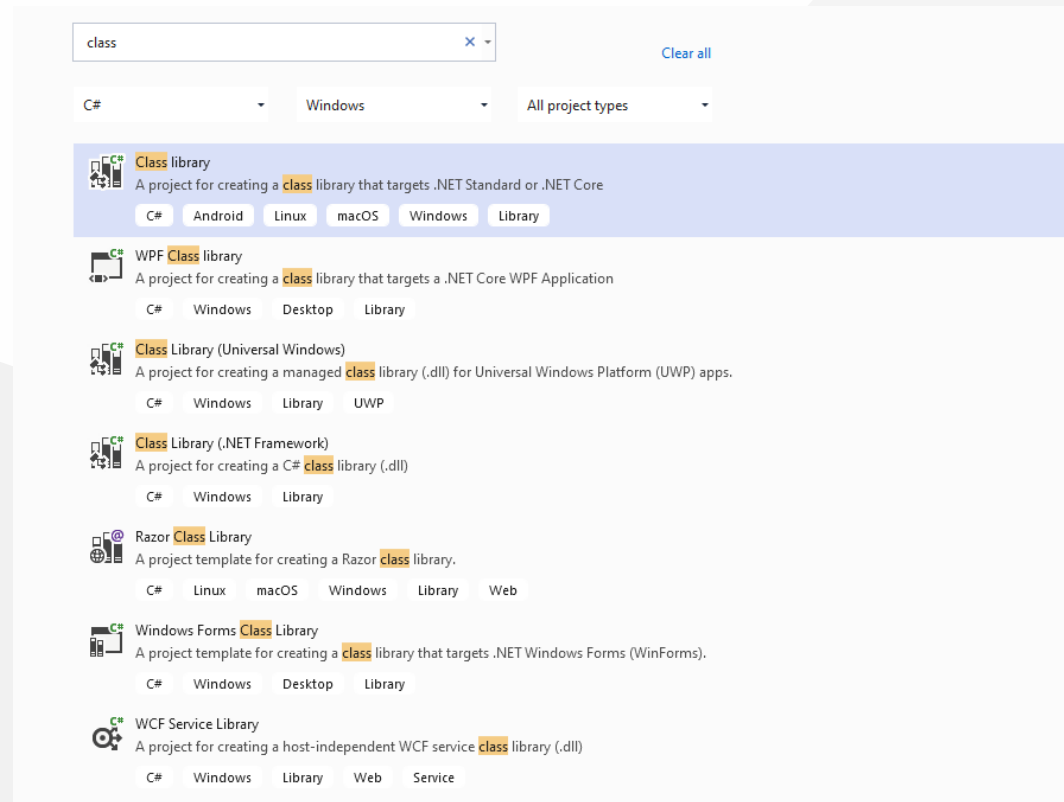
Shared Library Development

C# Programming (Dinamik Library)

Visual Studio Community Edition

Shared Library Development - (VS Csharp Dynamic Library)-1

- 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



Shared Library Development - (VS Csharp Dynamic Library)-2

- 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\ce103

...

Solution name ⓘ

csharp-sample-lib

Place solution and project in the same directory

Shared Library Development - (VS Csharp Dynamic Library)-3

- We will select .Net Core 3.1

Additional information

Class library

C#

Android

Linux

macOS

Windows

Library

Target Framework 

.NET Core 3.1 (Long-term support)

.NET Standard 2.0

.NET Standard 2.1

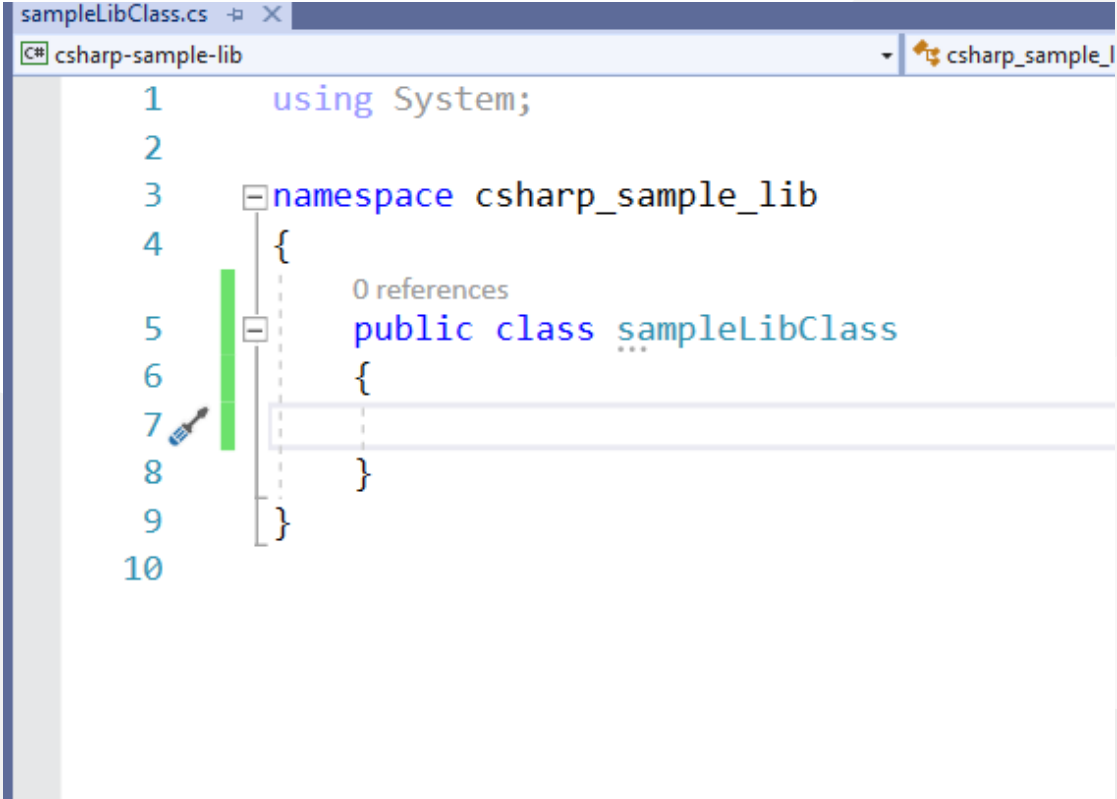
.NET Core 2.1 (Long-term support)

.NET Core 3.1 (Long-term support)

.NET 5.0 (Current)

Shared Library Development - (VS Csharp Dynamic Library)-4

- You will have default empty class library file

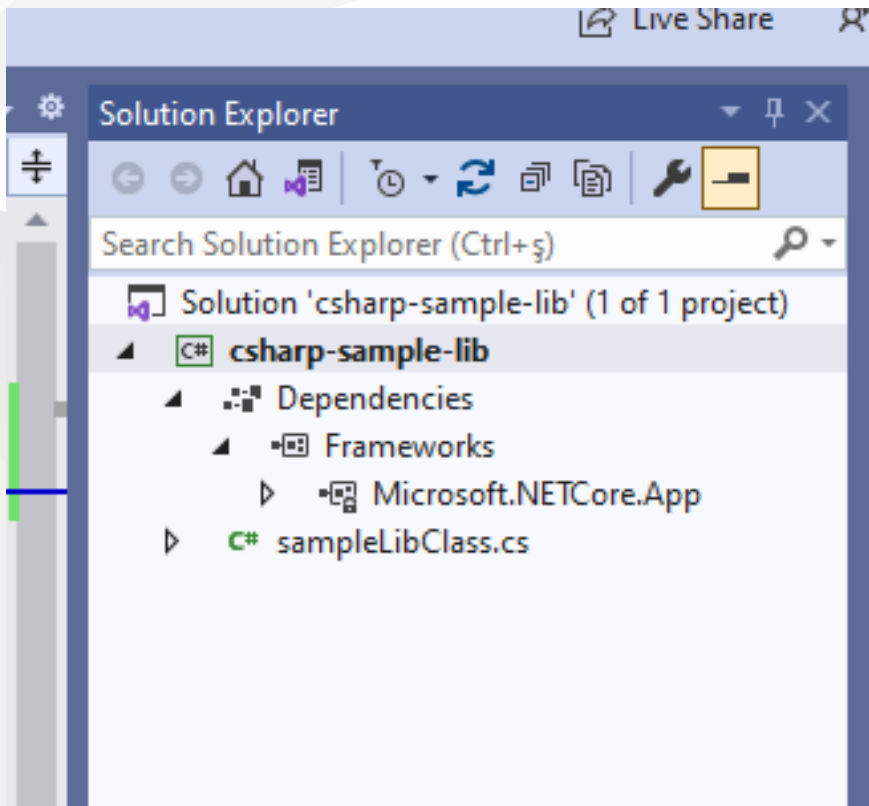


```
sampleLibClass.cs  X
C# csharp-sample-lib  csharp_sample_l

1  using System;
2
3  namespace csharp_sample_lib
4  {
5      0 references
6      public class sampleLibClass
7      {
8      }
9  }
10
```

Shared Library Development - (VS Csharp Dynamic Library)-5

- In the project you can see .NETcore reference



Shared Library Development - (VS Csharp Dynamic Library)-6

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

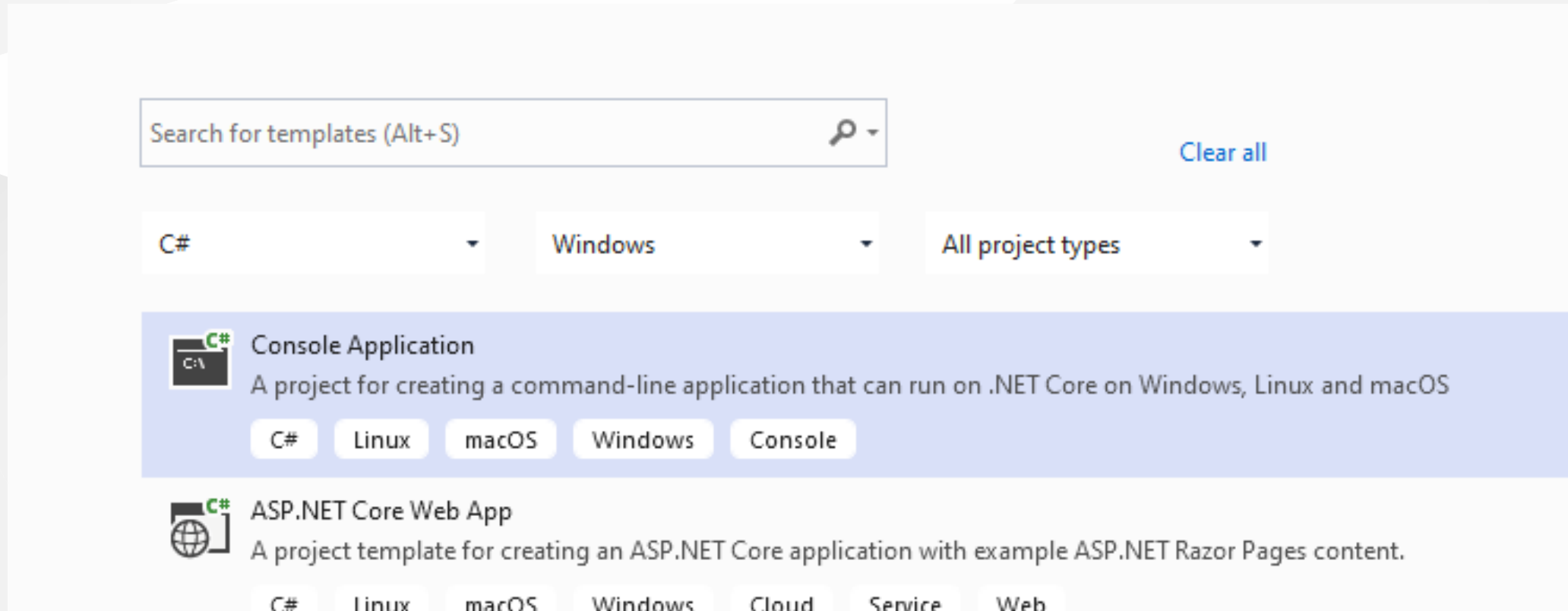
```
Ln: 7  Cr  
e-lib\csharp-sample-lib\csharp-sample-lib.csproj (in 3 ms).  
e\csharp-sample-lib\csharp-sample-lib\bin\Debug\netcoreapp3.1\csharp-sample-lib.dll
```

Shared Library Development - (VS Csharp Dynamic Library)-7

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

Shared Library Development - (VS Csharp Dynamic Library)-8

- Select New Project



Shared Library Development - (VS Csharp Dynamic Library)-9

- Set project name

Configure your new project

Console Application C# Linux macOS Windows Console

Project name

csharp-sample-app

Location

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

...

Shared Library Development - (VS Csharp Dynamic Library)-10

- 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)

Shared Library Development - (VS Csharp Dynamic Library)-11

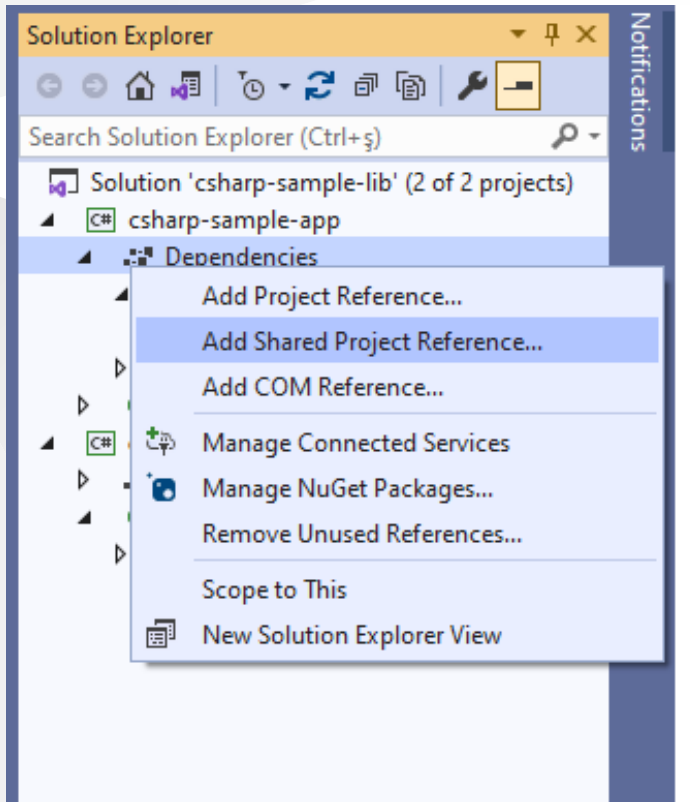
- You will have the following sample main.cs file

```
using System;

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

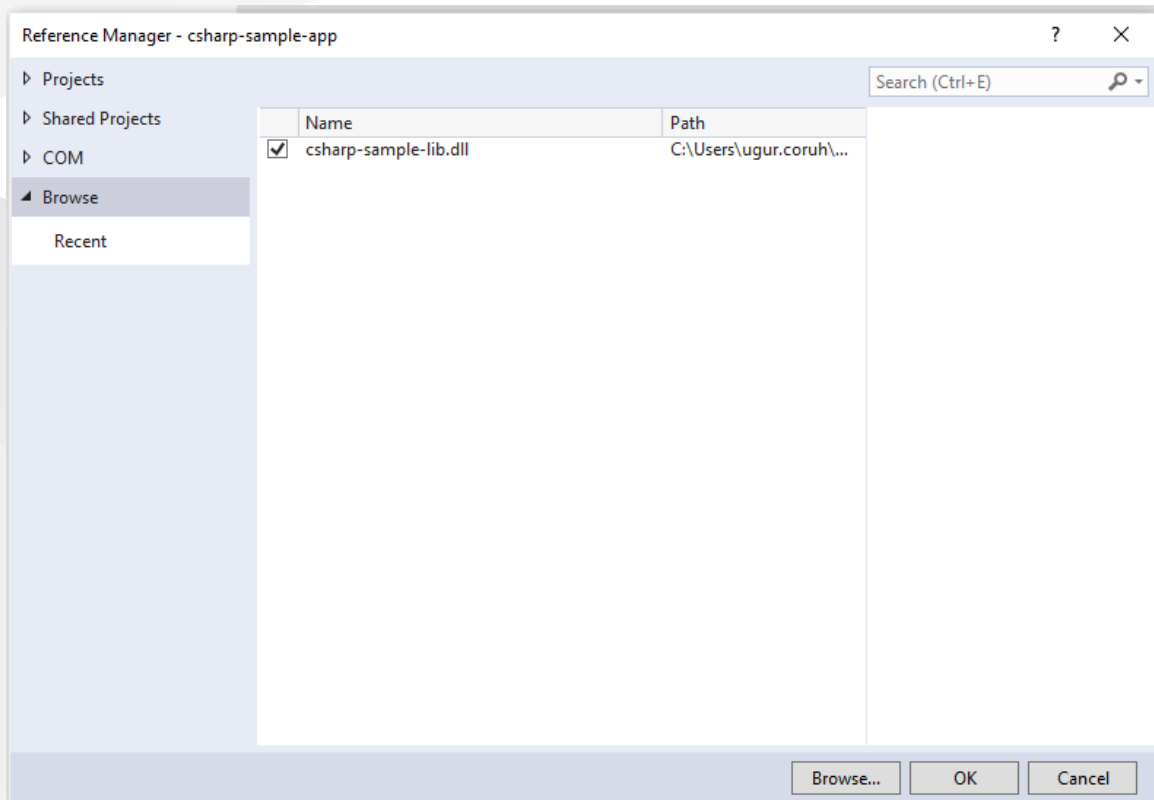
Shared Library Development - (VS Csharp Dynamic Library)-12

- Now we can link projects with adding references open reference section



Shared Library Development - (VS Csharp Dynamic Library)-13

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



Shared Library Development - (VS Csharp Dynamic Library)-14

- now we can update our library code and use it in console application
- copy following sample to sampleLibClass file in the library

Shared Library Development - (VS Csharp Dynamic Library)-15

```
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;
        }
    }
}
```

Shared Library Development - (VS Csharp Dynamic Library)-16

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

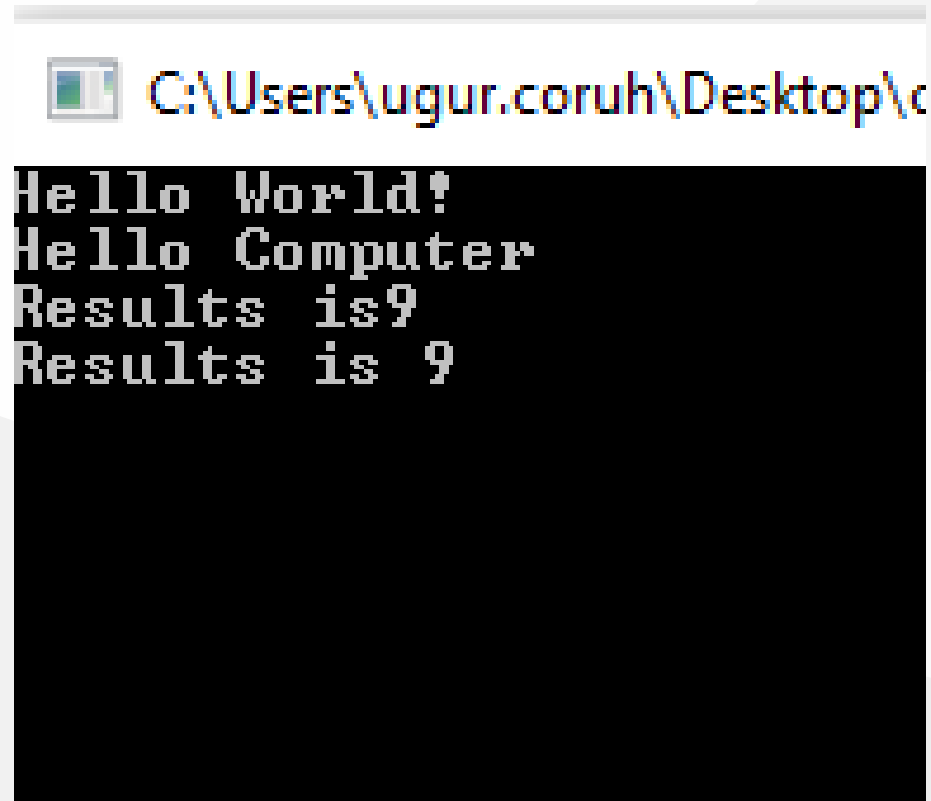
```
using csharp_sample_lib;
using System;

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();
        }
    }
}
```

Shared Library Development - (VS Csharp Dynamic Library)-17

- You will see following output that mean we called DLL functions



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

Shared Library Development - (VS Csharp Dynamic Library)-18

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

Shared Library Development - (VS Csharp Dynamic Library)-19

- 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
```

Shared Library Development - (VS Csharp Dynamic Library)-20

Step 2 – Installing Dotnet Core SDK

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

Shared Library Development - (VS Csharp Dynamic Library)-21

Step 3 – Install Dotnet Core Runtime Only

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

```
sudo apt update
```


Shared Library Development - (VS Csharp Dynamic Library)-22

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.

Shared Library Development - (VS Csharp Dynamic Library)-23

Step 4 – (Optional) Check .NET Core Version

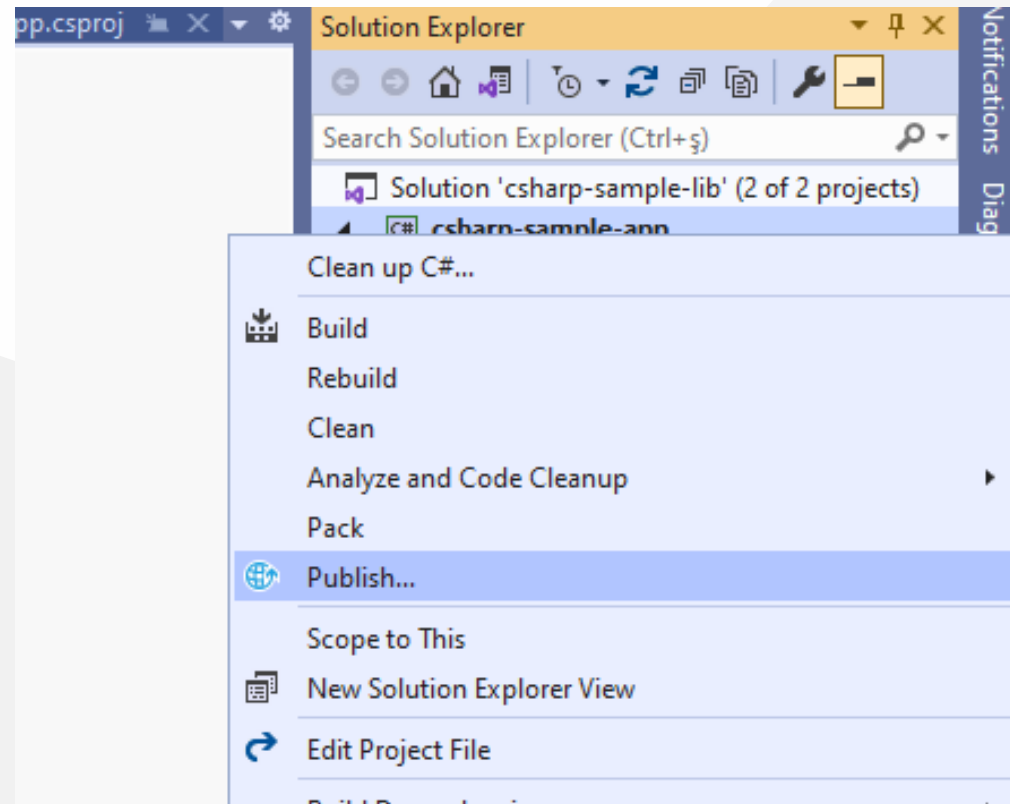
You can use dotnet command line utility to check installed version of .NET Core on your system. To check dotnet version, type:

```
dotnet --version
```

```
ucoruh@LAPTOP-RQNNS  
3.1.414  
ucoruh@LAPTOP-RQNNS
```

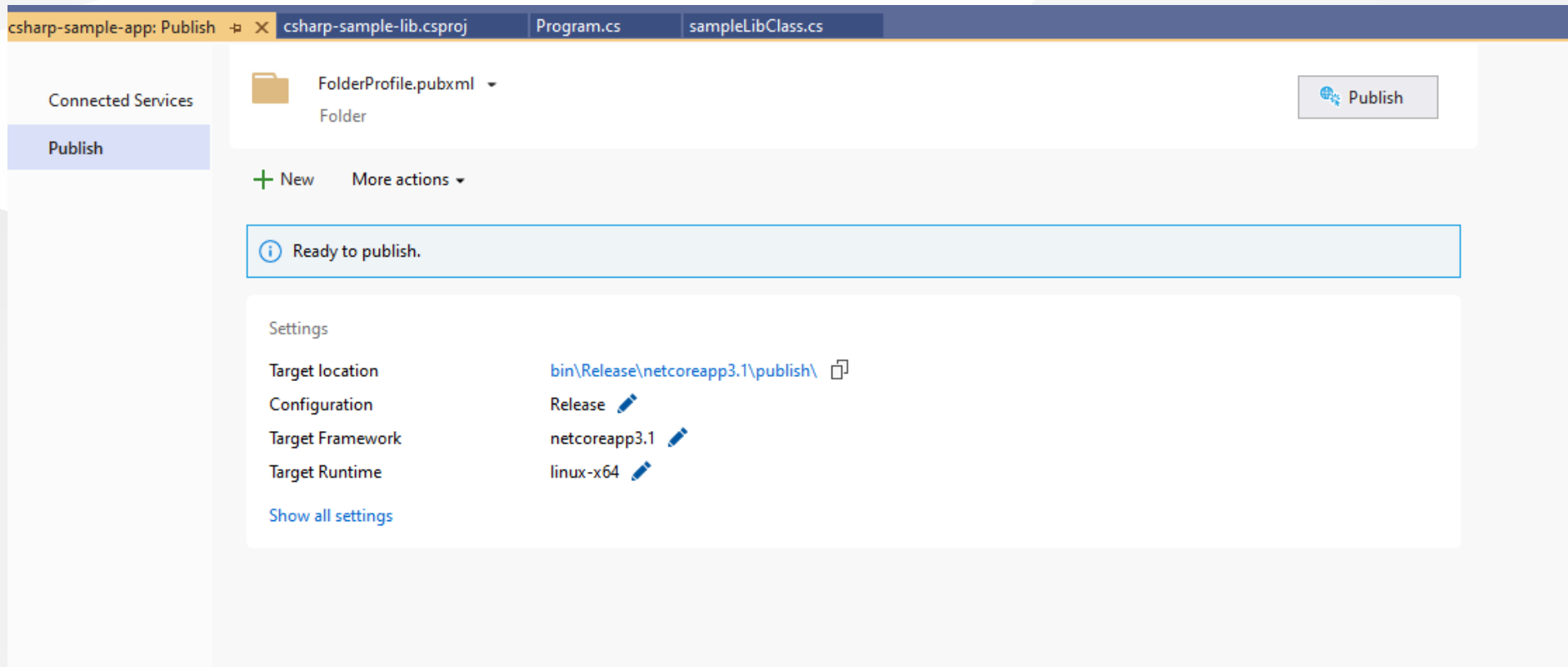
Shared Library Development - (VS Csharp Dynamic Library)-24

- Now we will publish our application as single executable
- Open publish menu



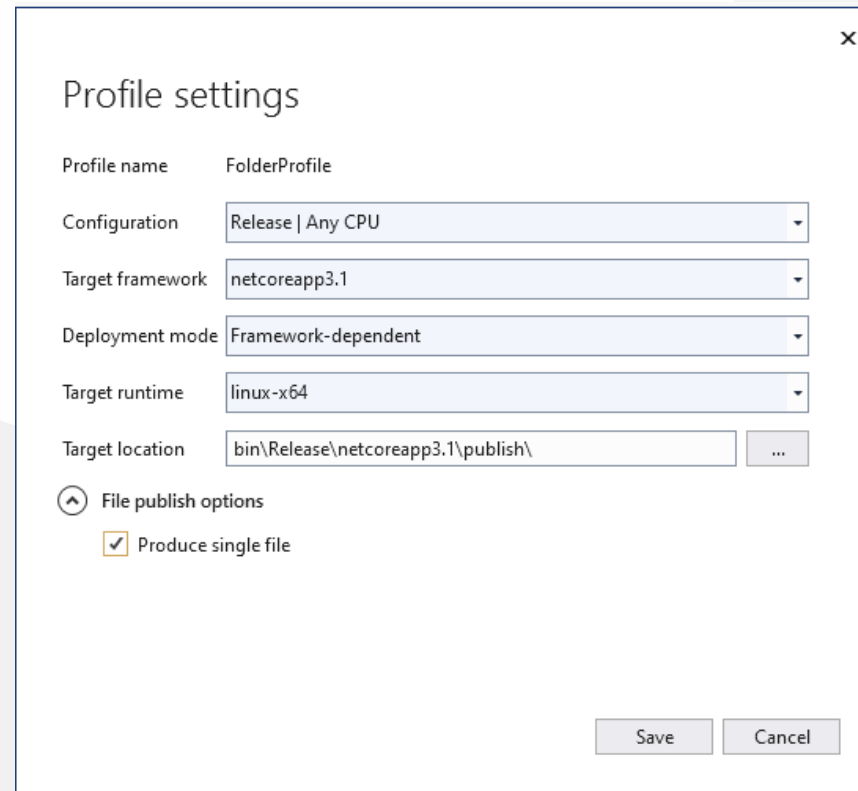
Shared Library Development - (VS Csharp Dynamic Library)-25

- Select netcoreapp3.1 and Release for linux-x64



Shared Library Development - (VS Csharp Dynamic Library)-26

- Select produce single file







Shared Library Development - (VS Csharp Dynamic Library)-27

- After succesfull 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
```

it

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

Shared Library Development - (VS Csharp Dynamic Library)-28

- 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-RQMN59IG:/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-RQMN59IG:/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.pdb      csharp-sample-lib.pdb      packages-microsoft-prod.deb
ucoruh@LAPTOP-RQMN59IG:/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
```

Shared Library Development - (VS Csharp Dynamic Library)-29

check dotnet --version and then run application

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

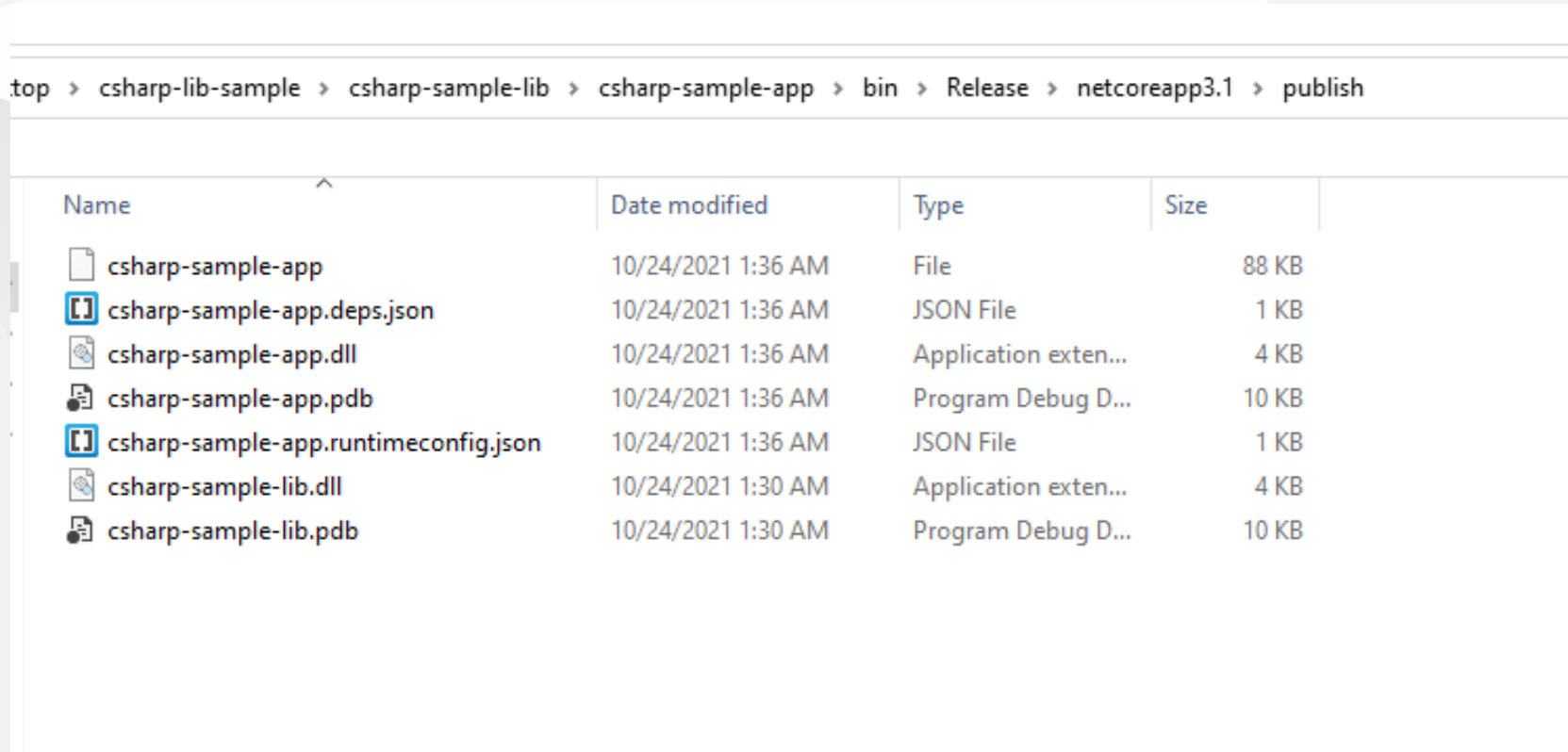
- you will see similar output

```
ucoruh@LAPTOP-RQNS9IG:/mnt/c/  
csharp-sample-app  
ucoruh@LAPTOP-RQNS9IG:/mnt/c/  
Hello World!  
Hello Computer  
Results is9  
Results is 9
```


Shared Library Development - (VS Csharp Dynamic Library)-30

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

Shared Library Development - (VS Csharp Dynamic Library)-31

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

```
ucoruh@LAPTOP-RQNNS  
Hello World!  
Hello Computer  
Results is9  
Results is 9
```

Shared Library Development

Java Programming

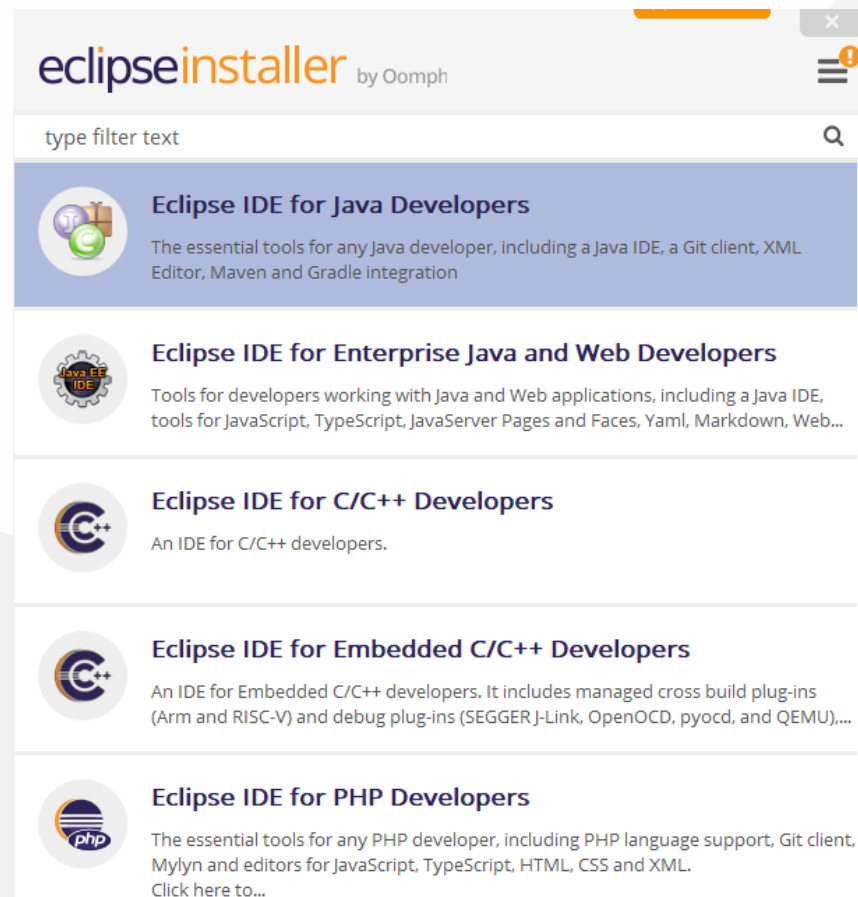
Eclipse IDE

Shared Library Development - (Eclipse Java Jar Library)-1

- You should download and install eclipse installer and then you should select Eclipse IDE for Java Developers
 - [Eclipse Installer 2021-09 R | Eclipse Packages](#)



Shared Library Development - (Eclipse Java Jar Library)-2



Shared Library Development - (Eclipse Java Jar Library)-3

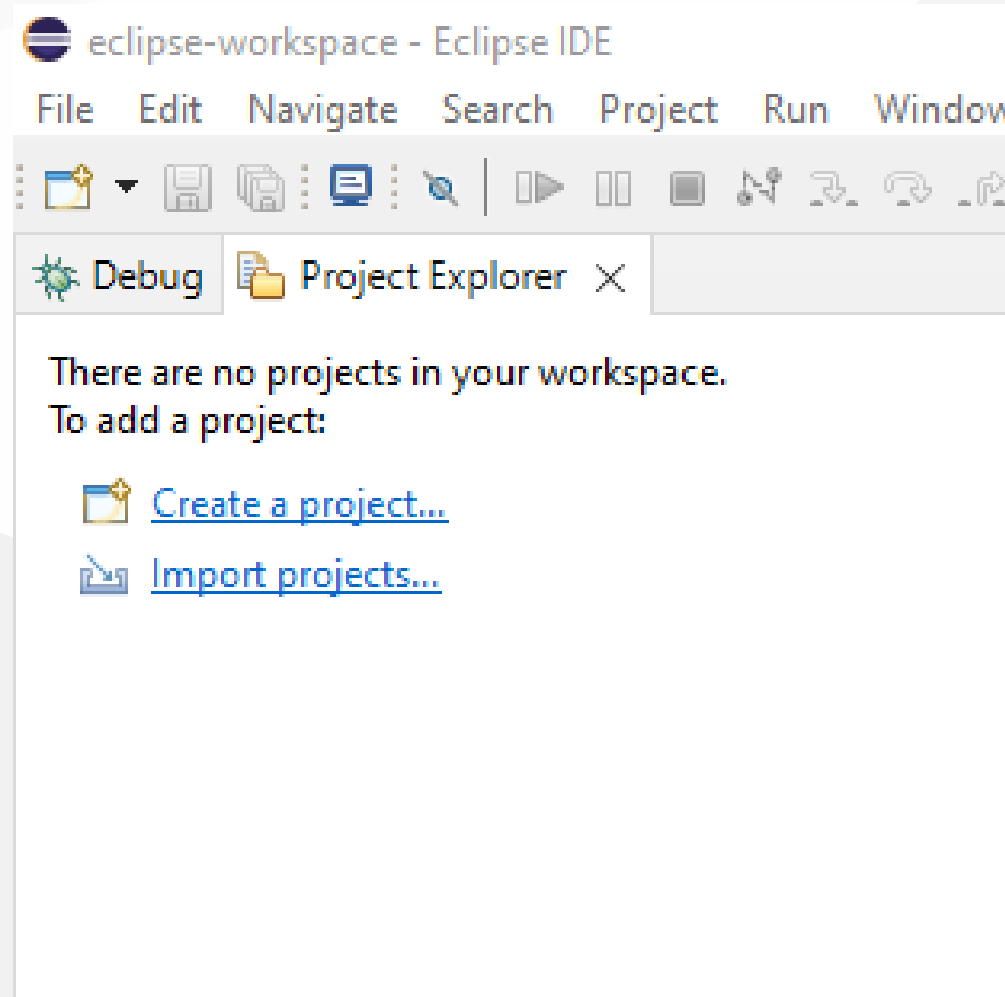


Shared Library Development - (Eclipse Java Jar Library)-4



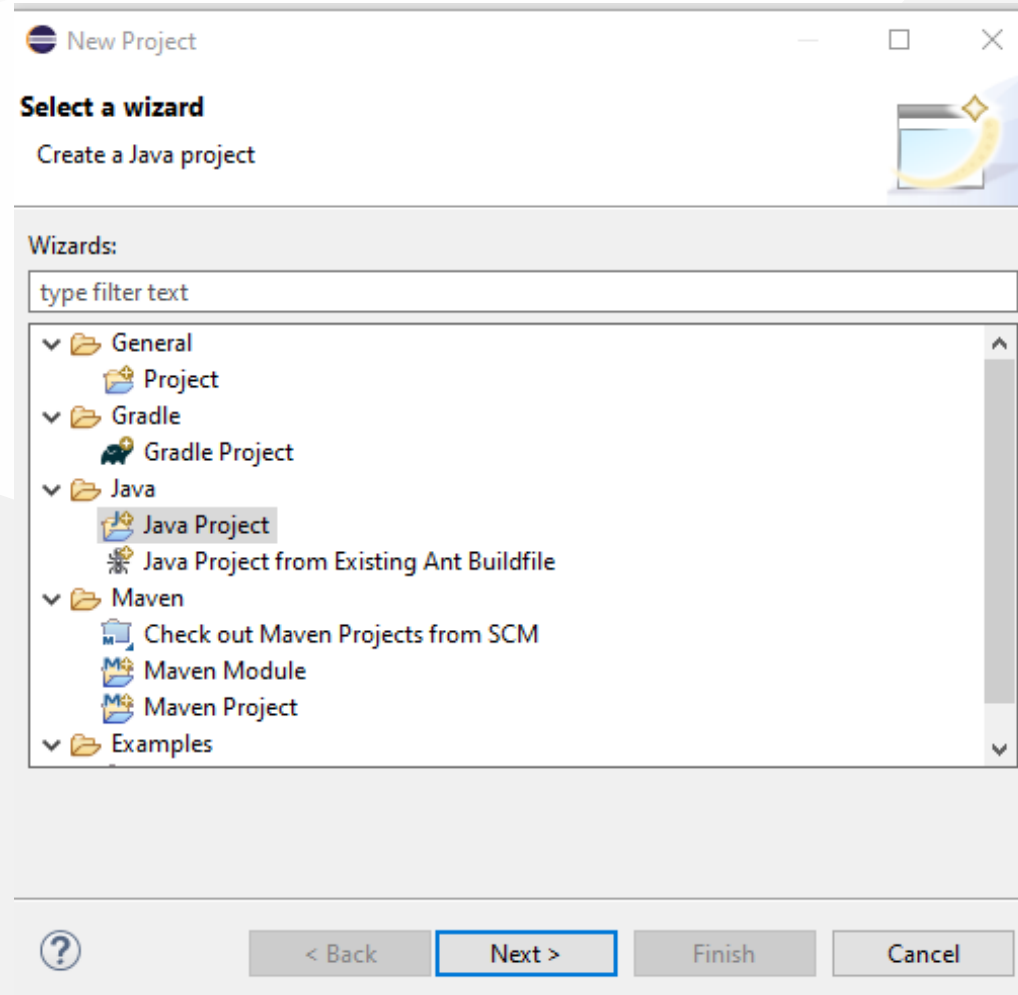
Shared Library Development - (Eclipse Java Jar Library)-5

- select create a project



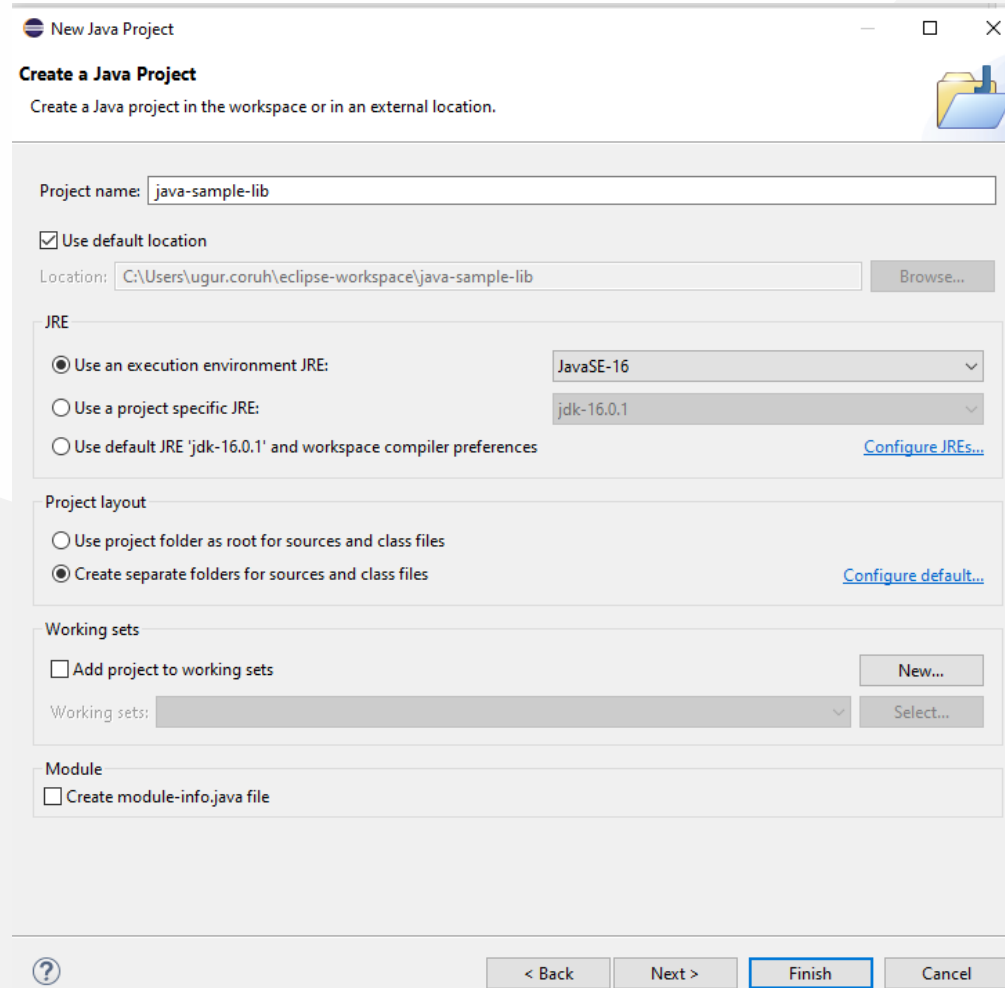
Shared Library Development - (Eclipse Java Jar Library)-6

select java project



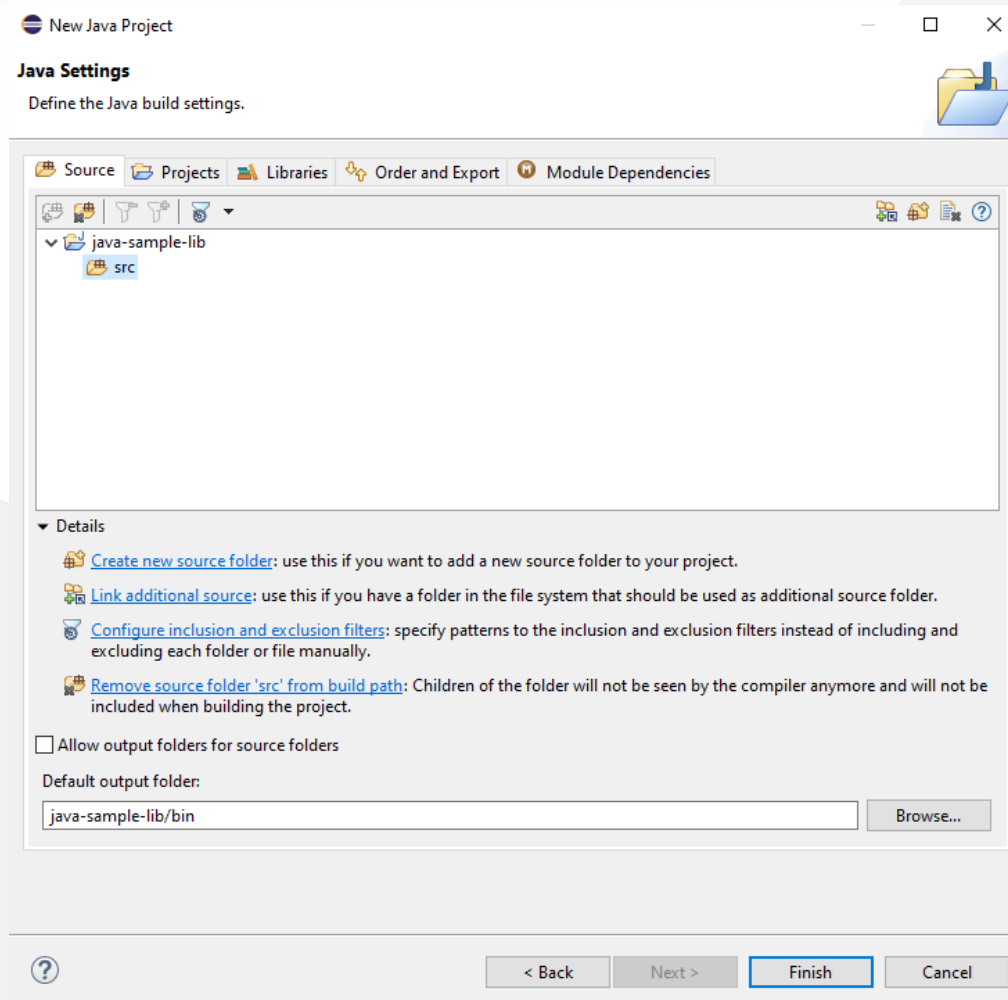
Shared Library Development - (Eclipse Java Jar Library)-7

- give project name



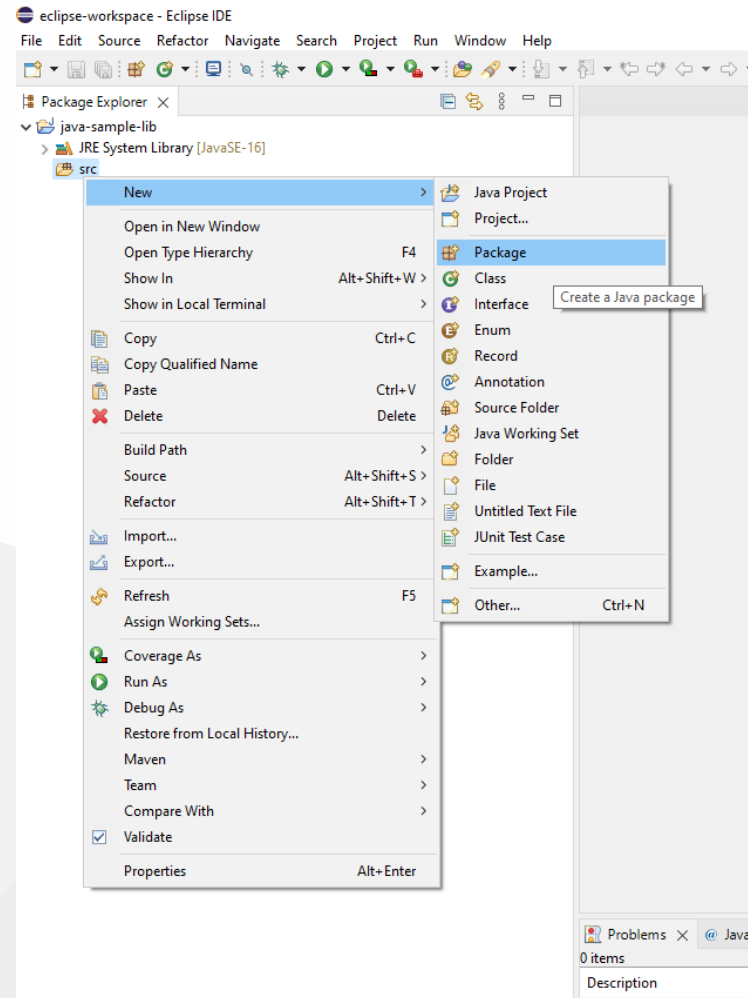
Shared Library Development - (Eclipse Java Jar Library)-8

- select finish



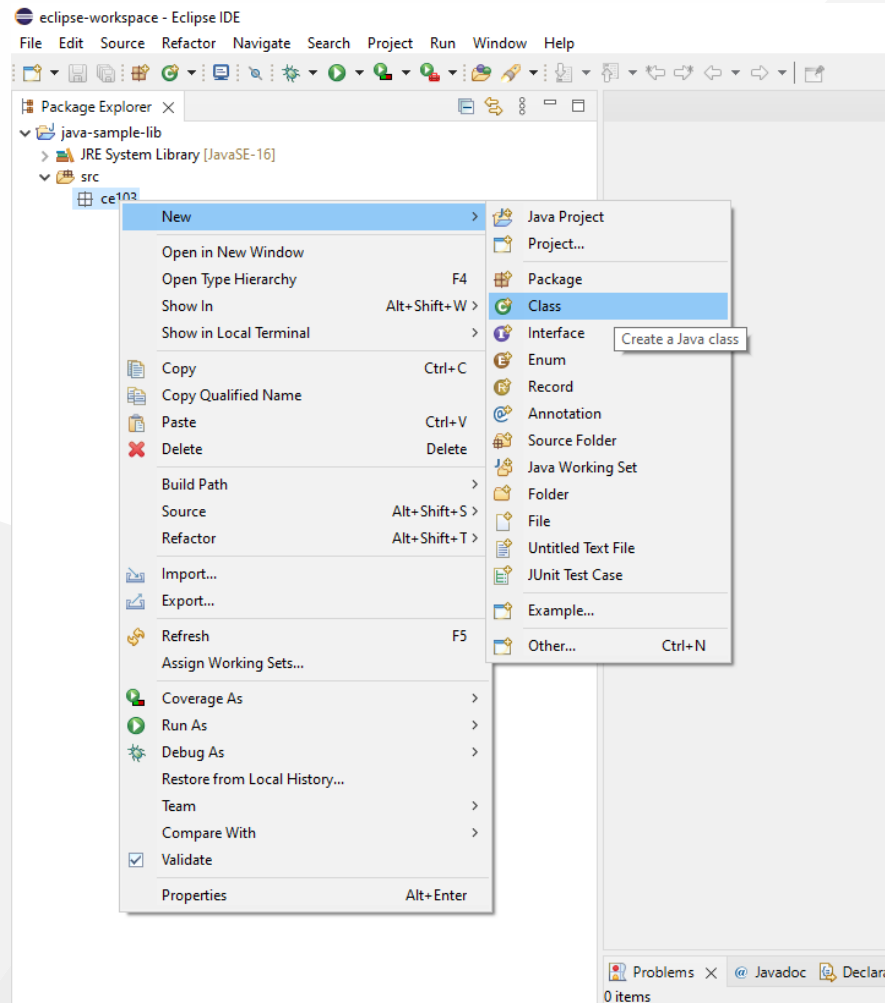
Shared Library Development - (Eclipse Java Jar Library)-9

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



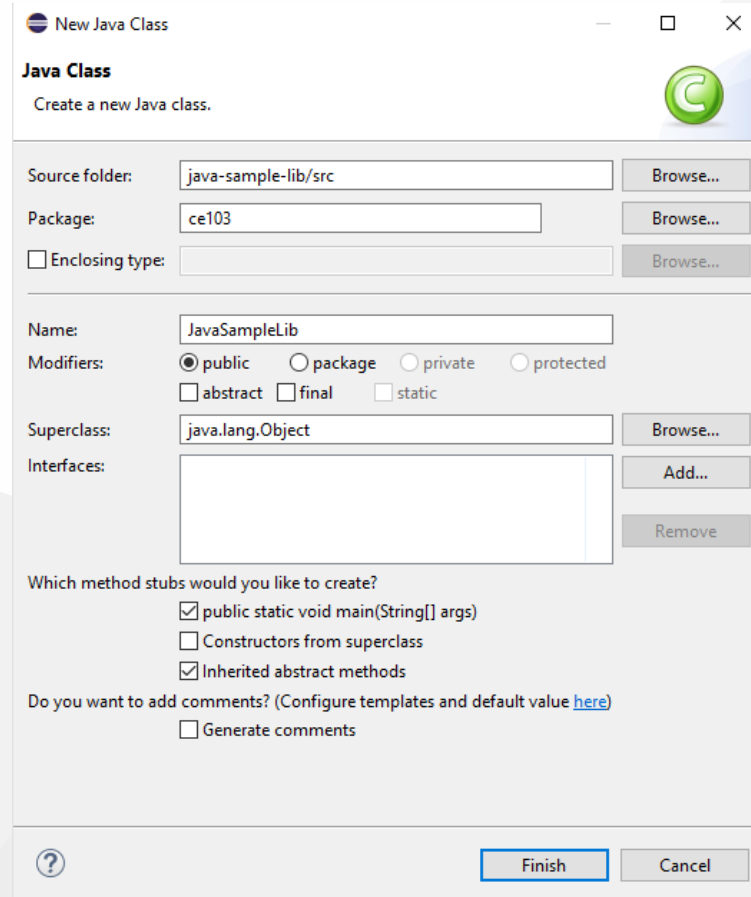
Shared Library Development - (Eclipse Java Jar Library)-10

- then we can create our class that includes our functions



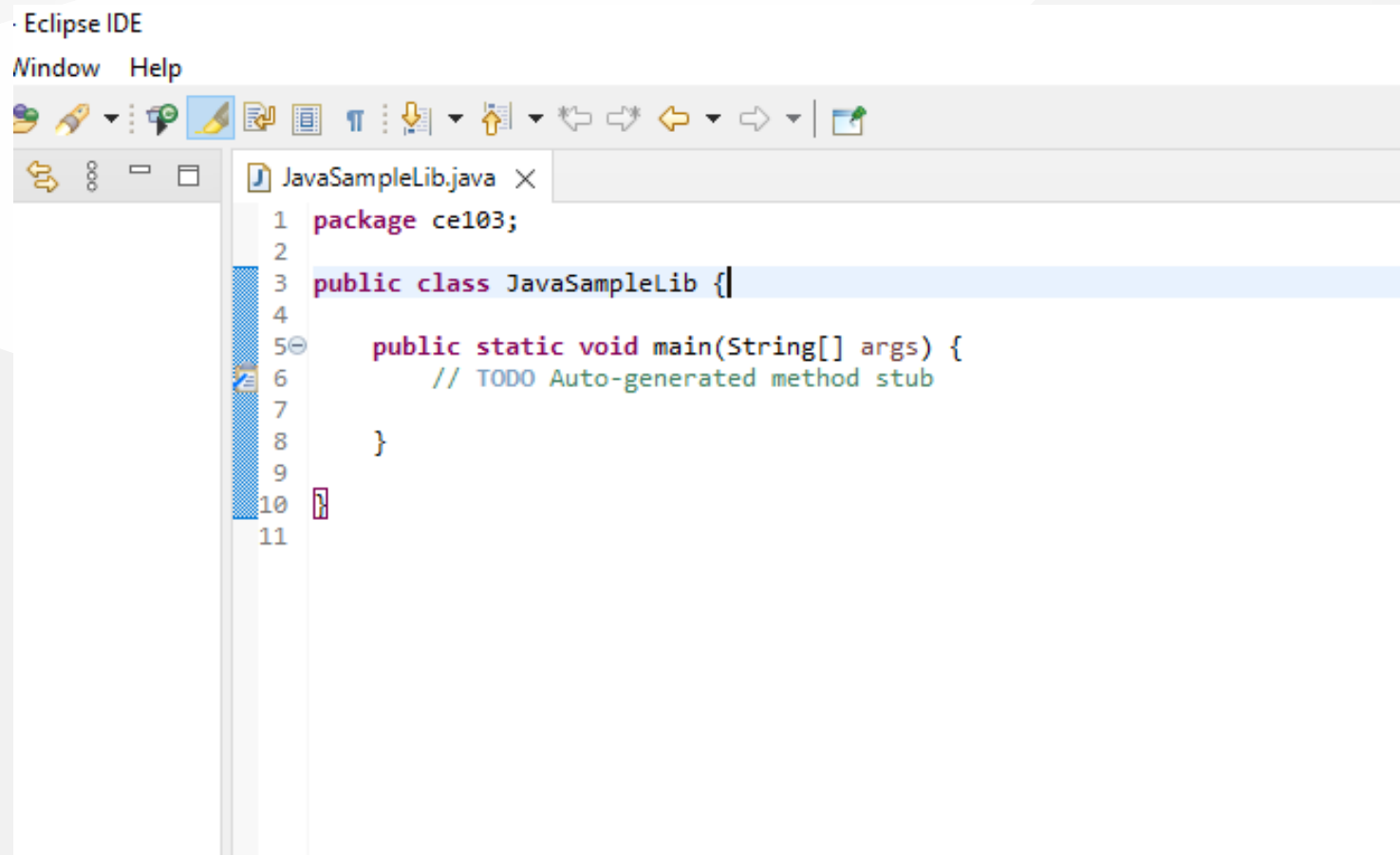
Shared Library Development - (Eclipse Java Jar Library)-11

- give class a name



Shared Library Development - (Eclipse Java Jar Library)-12

- you will have following class with main



```
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
```

Shared Library Development - (Eclipse Java Jar Library)-13

- 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;
    }

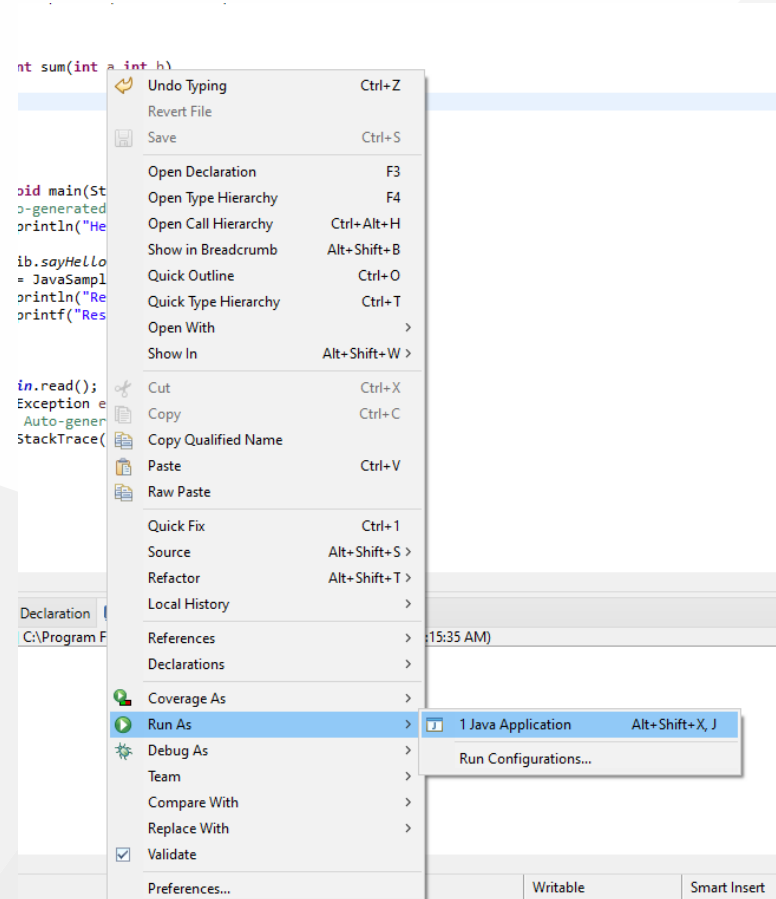
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Hello World!");

        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();
        }
    }
}
```

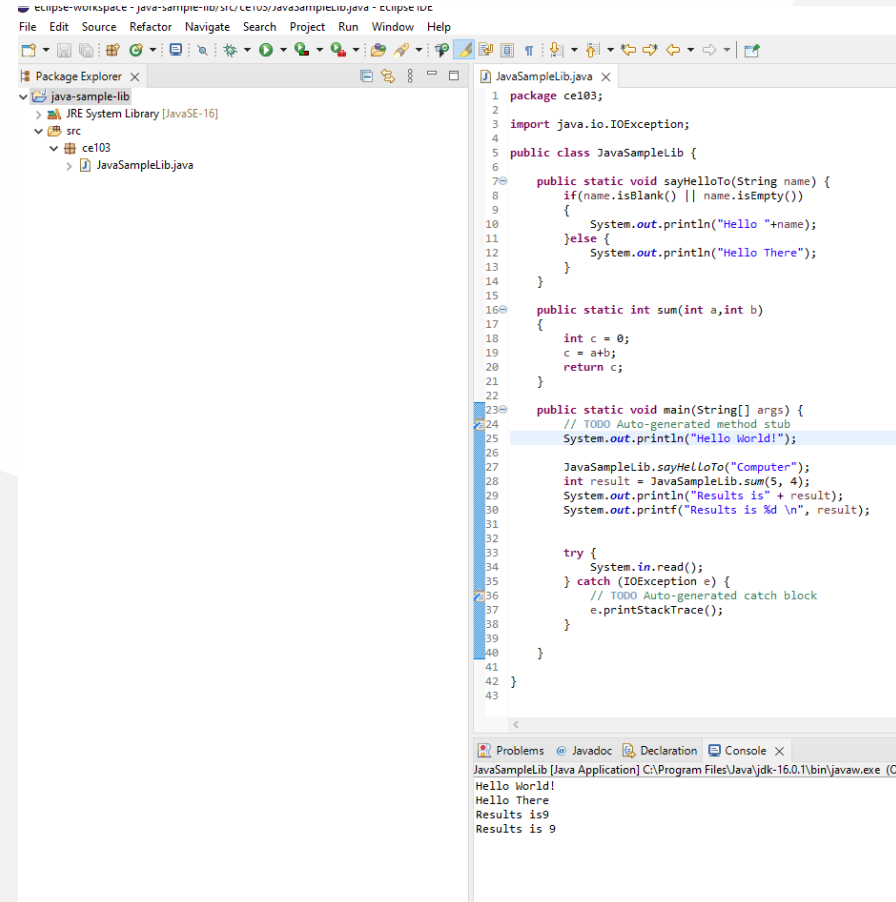

Shared Library Development - (Eclipse Java Jar Library)-14

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



Shared Library Development - (Eclipse Java Jar Library)-15

- we can see output from console as below



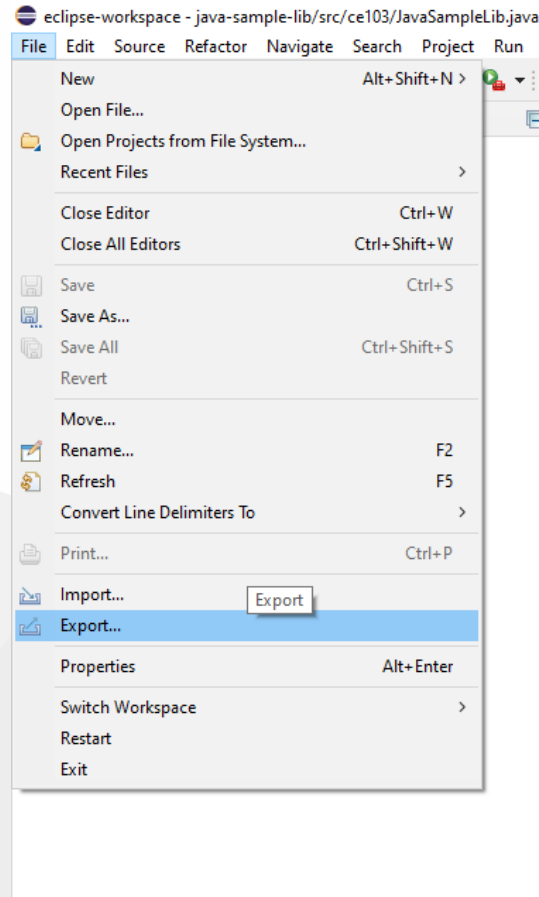
The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure: java-sample-lib > JRE System Library [JavaSE-16] > src > ce103 > JavaSampleLib.java. The main editor window shows the source code for JavaSampleLib.java. The code includes a package declaration, an import for java.io.IOException, and a public class JavaSampleLib with three methods: sayHelloTo, sum, and main. The main method calls sayHelloTo, sum, and prints the results. The Console window at the bottom shows the output of the program: Hello World!, Hello There, Results is 9, and Results is 9.

```
1 package ce103;
2
3 import java.io.IOException;
4
5 public class JavaSampleLib {
6
7     public static void sayHelloTo(String name) {
8         if(name.isBlank() || name.isEmpty())
9         {
10            System.out.println("Hello "+name);
11        }else {
12            System.out.println("Hello There");
13        }
14    }
15
16    public static int sum(int a,int b)
17    {
18        int c = 0;
19        c = a+b;
20        return c;
21    }
22
23    public static void main(String[] args) {
24        // TODO Auto-generated method stub
25        System.out.println("Hello World!");
26
27        JavaSampleLib.sayHelloTo("Computer");
28        int result = JavaSampleLib.sum(5, 4);
29        System.out.println("Results is" + result);
30        System.out.printf("Results is %d \n", result);
31
32
33        try {
34            System.in.read();
35        } catch (IOException e) {
36            // TODO Auto-generated catch block
37            e.printStackTrace();
38        }
39
40    }
41
42 }
43
```

JavaSampleLib [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (O
Hello World!
Hello There
Results is 9
Results is 9

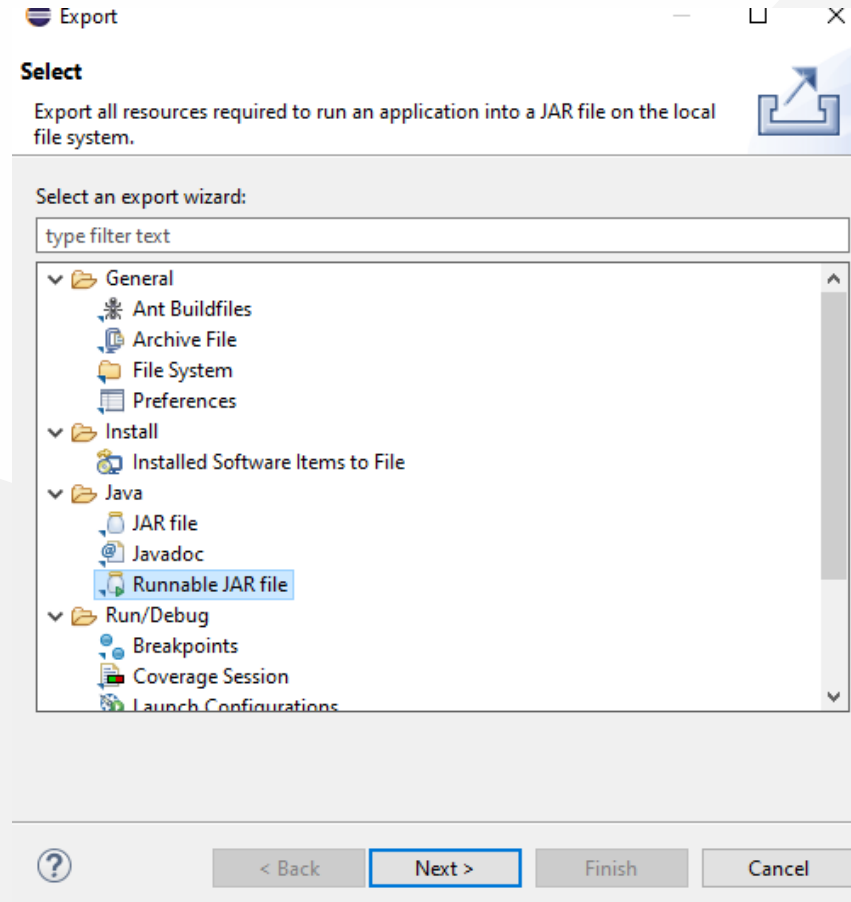
Shared Library Development - (Eclipse Java Jar Library)-16

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



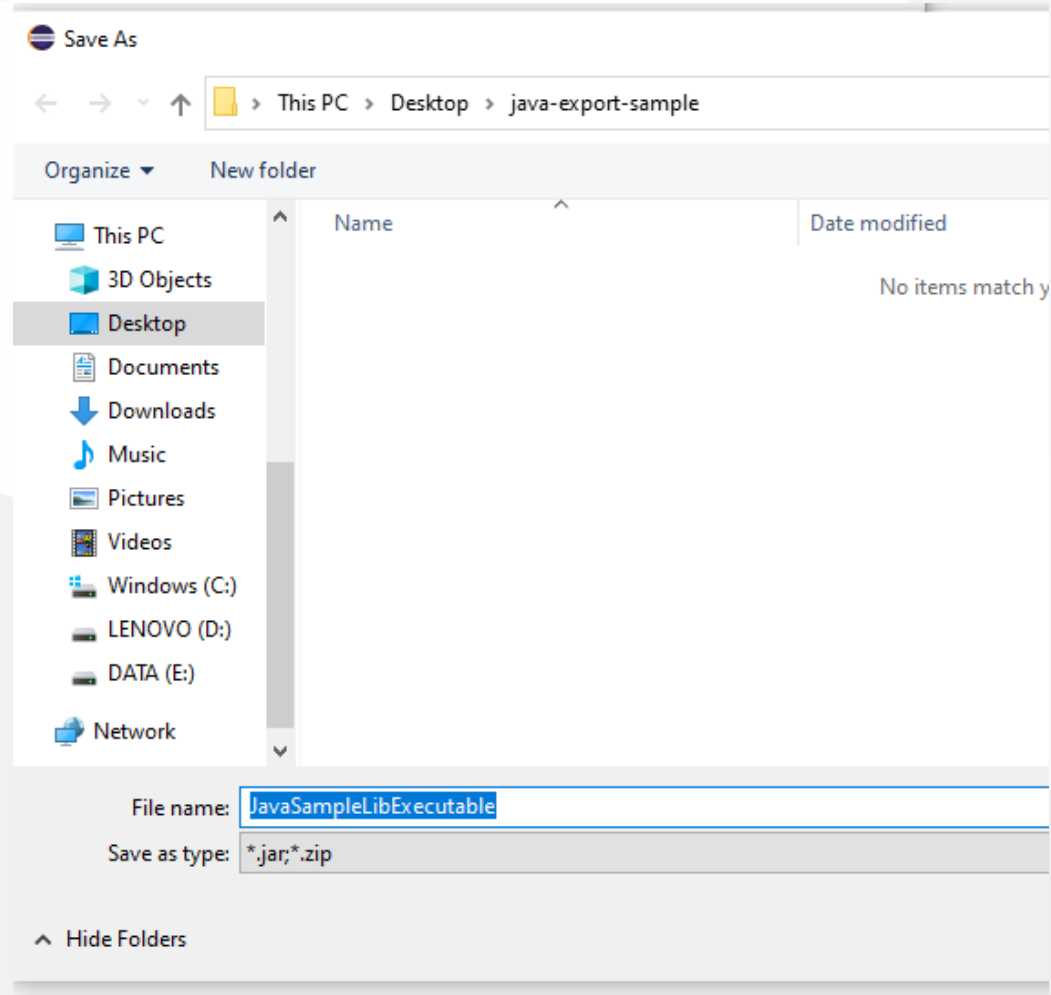
Shared Library Development - (Eclipse Java Jar Library)-17

- Select Java->Runnable JAR File



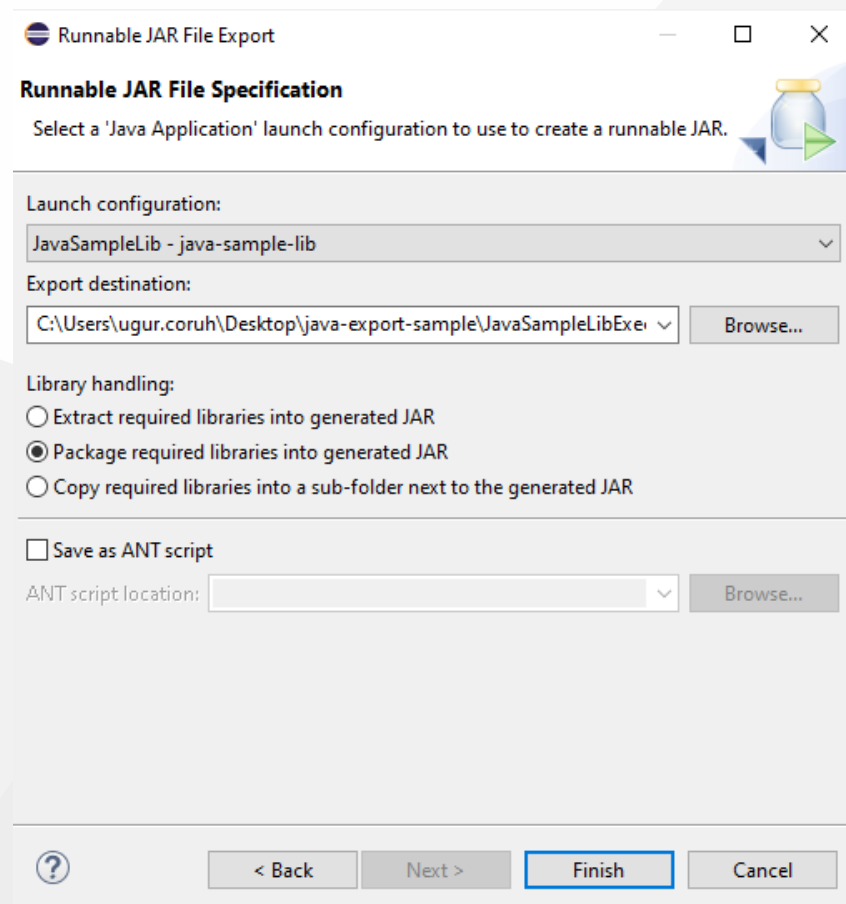
Shared Library Development - (Eclipse Java Jar Library)-18

click next and set output path for jar file



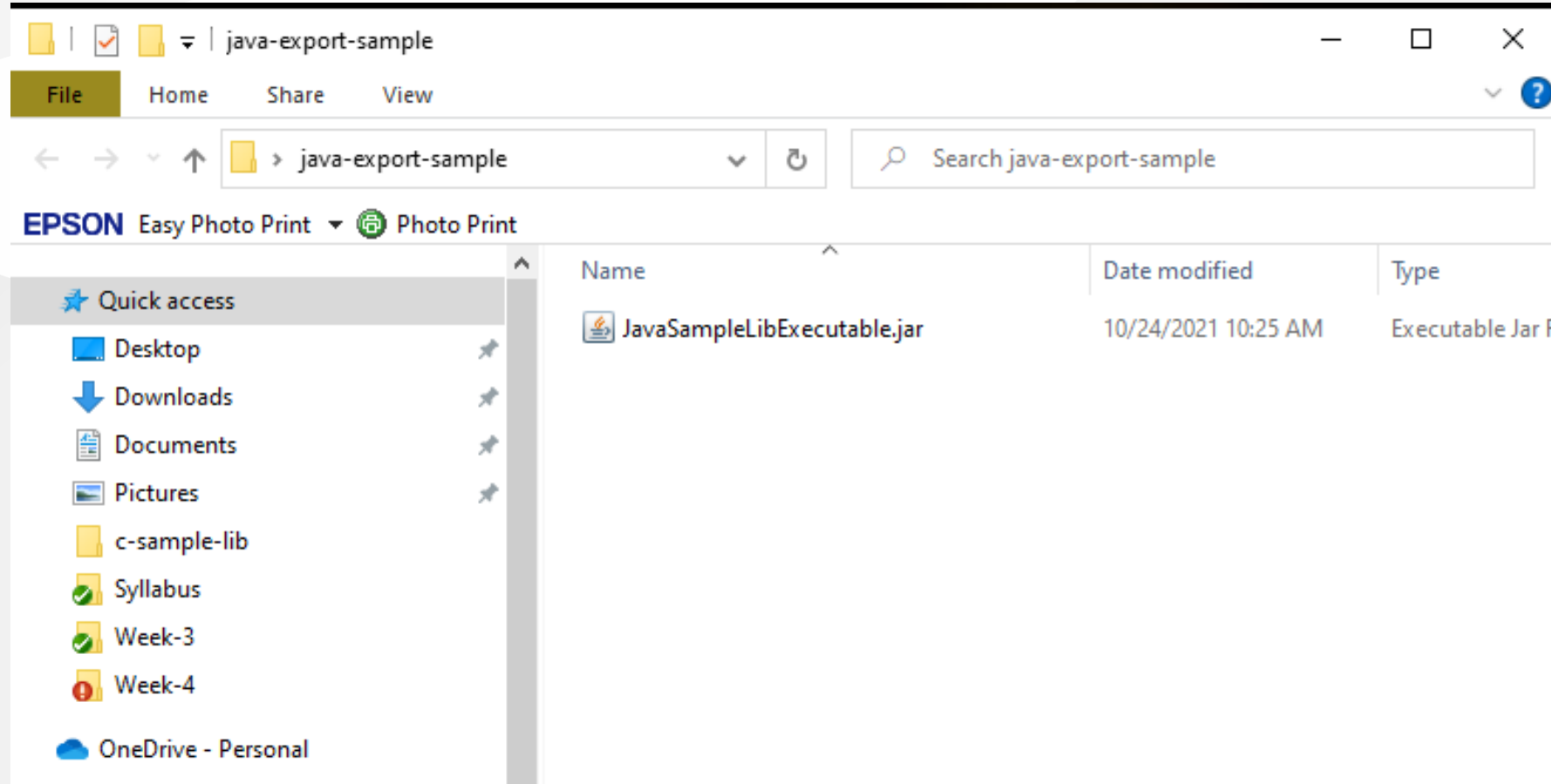
Shared Library Development - (Eclipse Java Jar Library)-19

- 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



Shared Library Development - (Eclipse Java Jar Library)-20

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



Shared Library Development - (Eclipse Java Jar Library)-21

- 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
t version of the Java Runtime (class file version 60.0), this version of the Java Runtime only recognizes class file ve
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

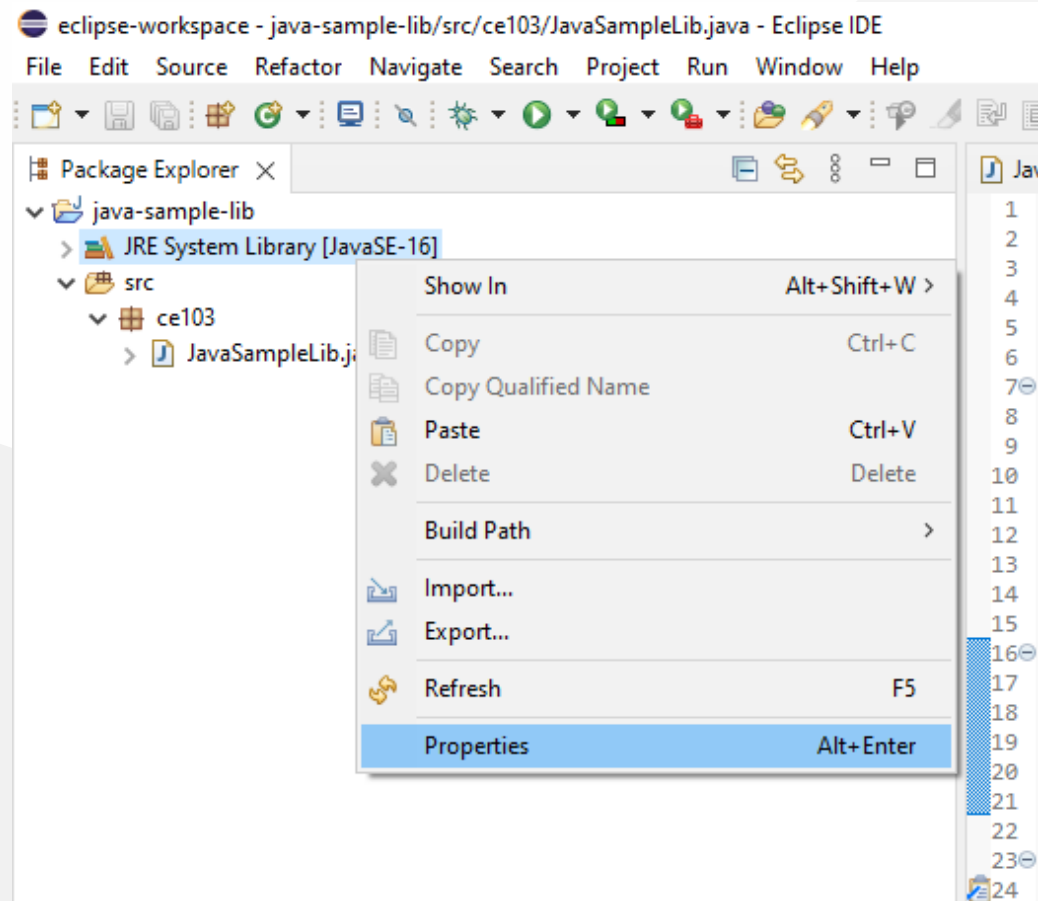
If we check version we can see problem Java version 1.8.0_231

```
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...]
```

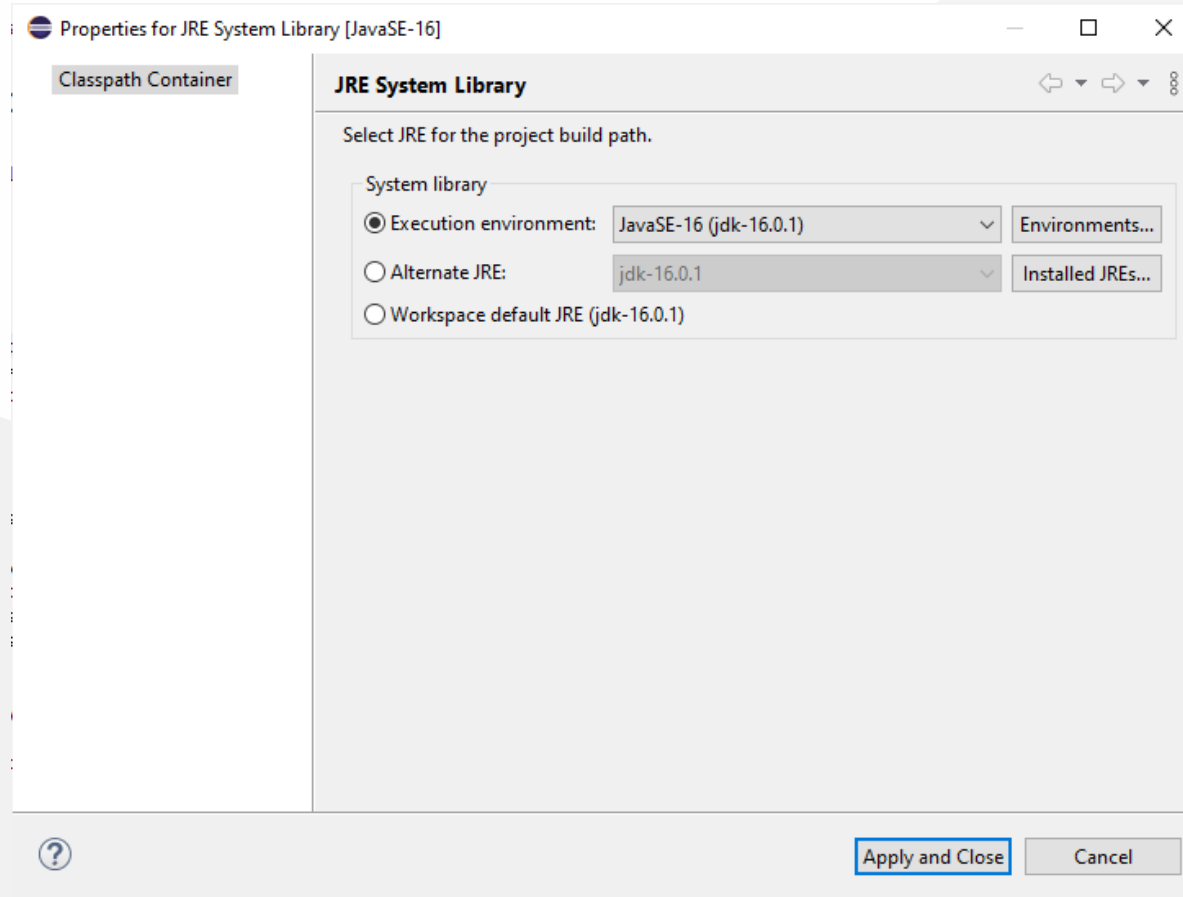

Shared Library Development - (Eclipse Java Jar Library)-22

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



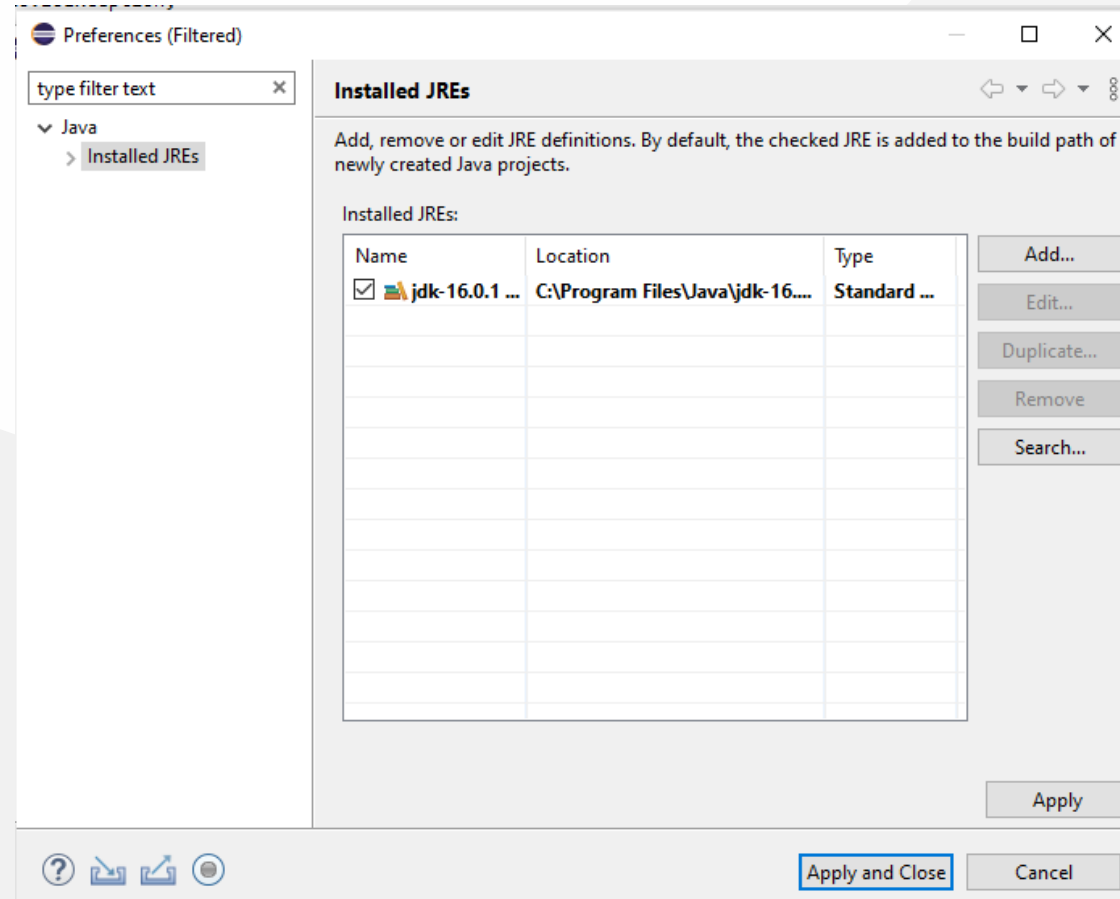
Shared Library Development - (Eclipse Java Jar Library)-23

- select environments



Shared Library Development - (Eclipse Java Jar Library)-24

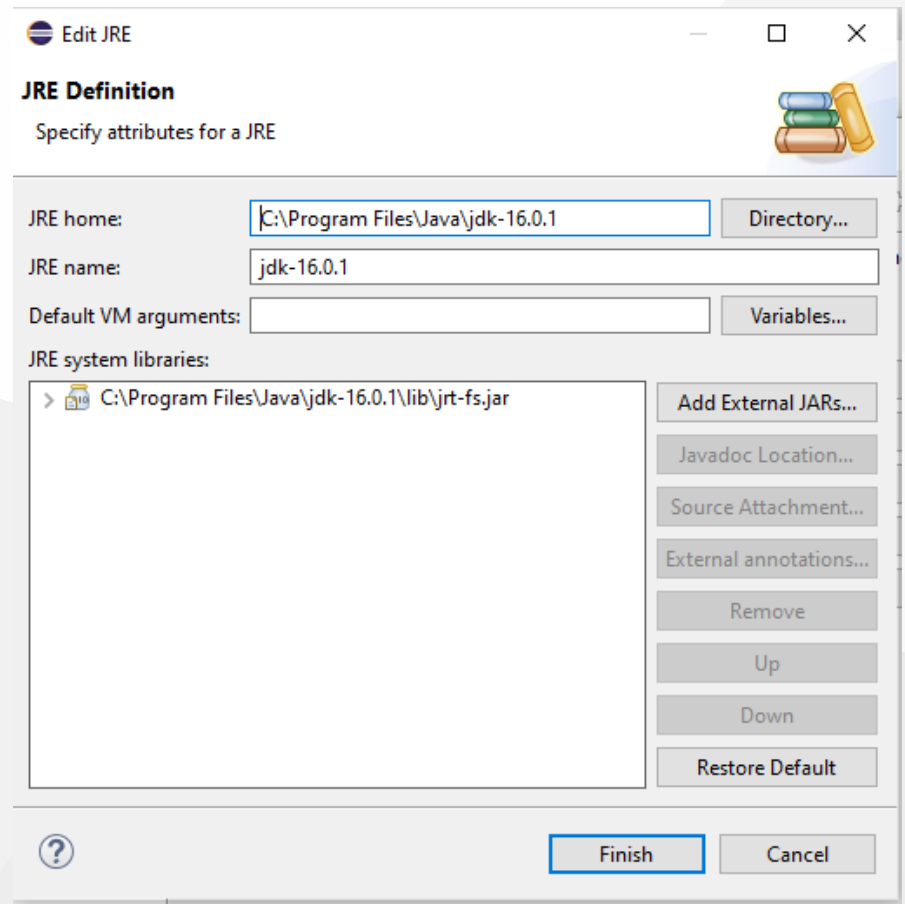
- select installed JRE or JDK



Shared Library Development - (Eclipse Java Jar Library)-25

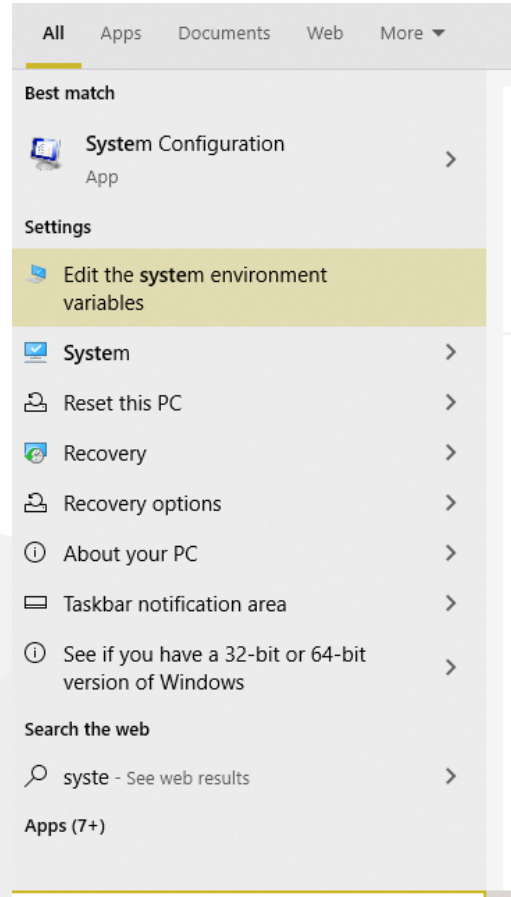
- you can see installed JRE or JDK home

C:\Program Files\Java\jdk-16.0.1

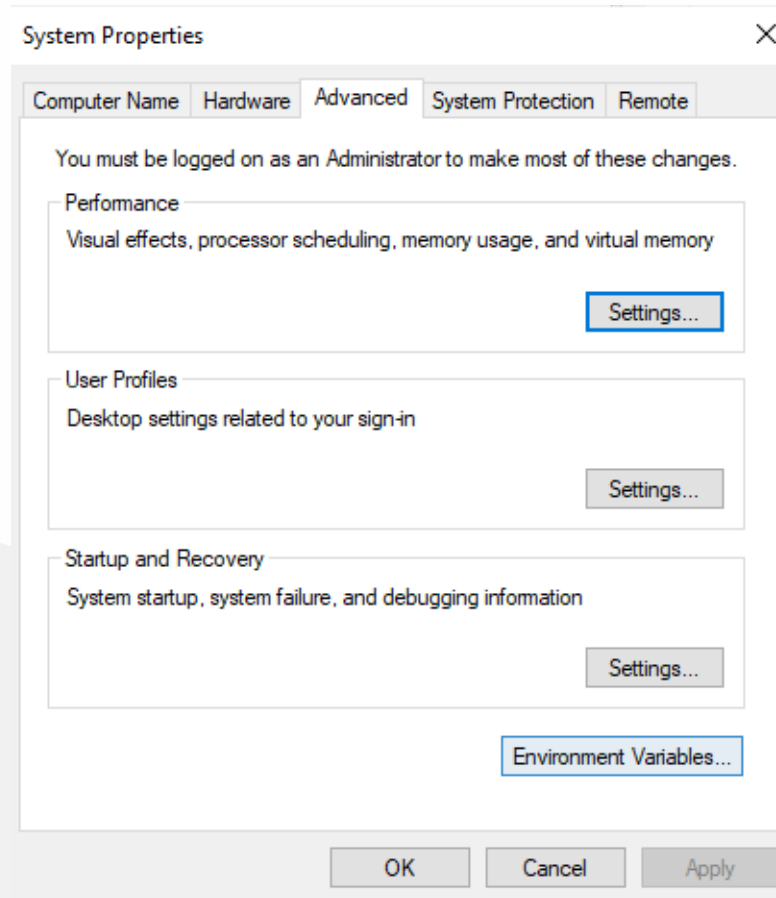


Shared Library Development - (Eclipse Java Jar Library)-26

- Open system environment to fix this problem

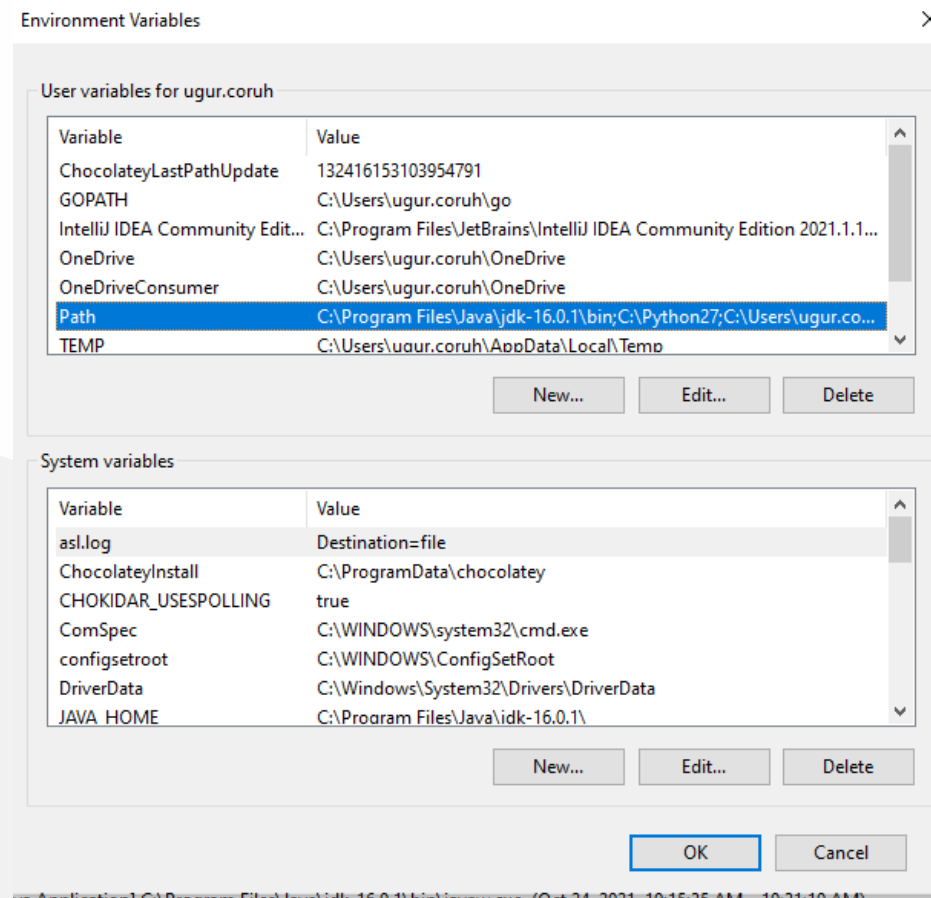


Shared Library Development - (Eclipse Java Jar Library)-27

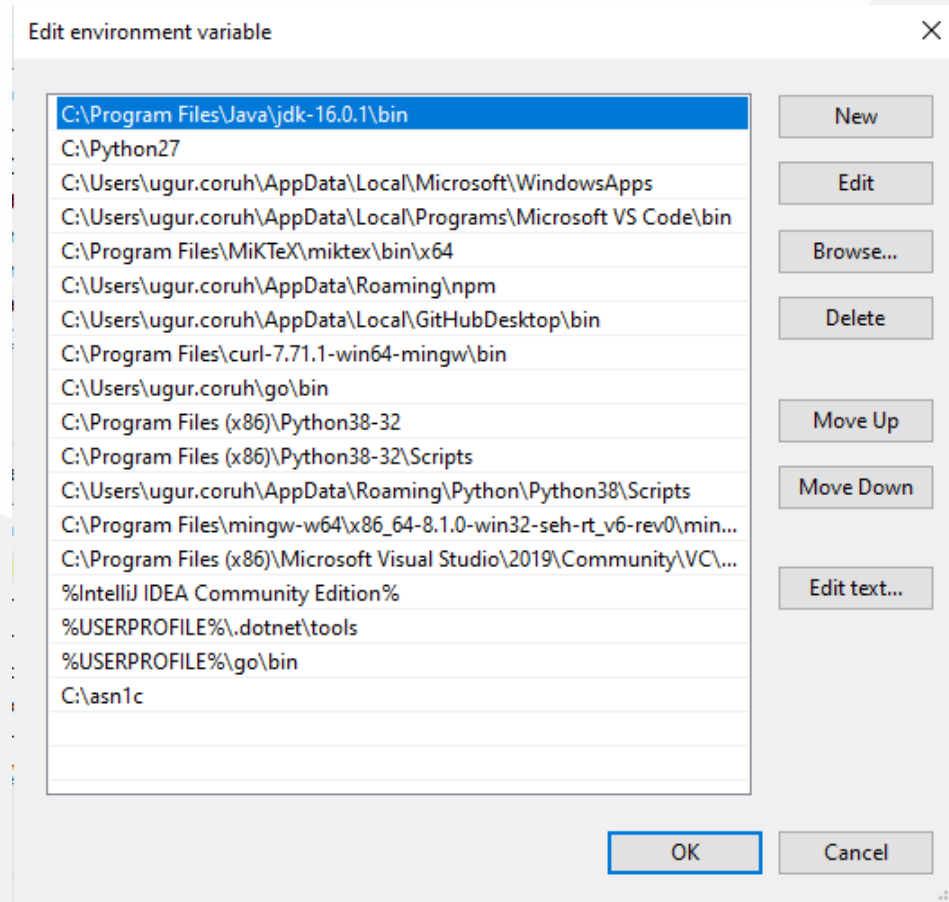


Shared Library Development - (Eclipse Java Jar Library)-28

- Check user settings first

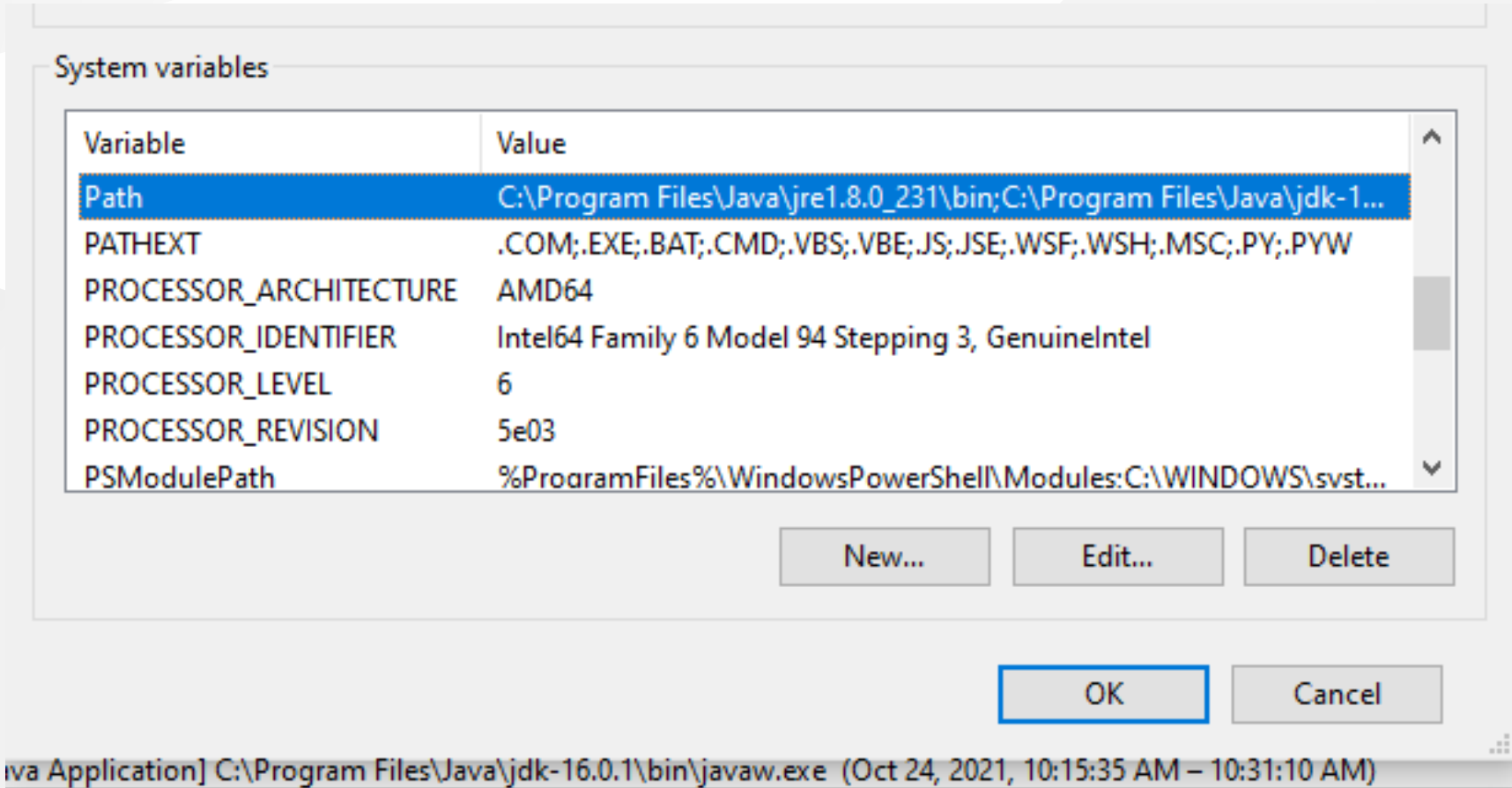


Shared Library Development - (Eclipse Java Jar Library)-29

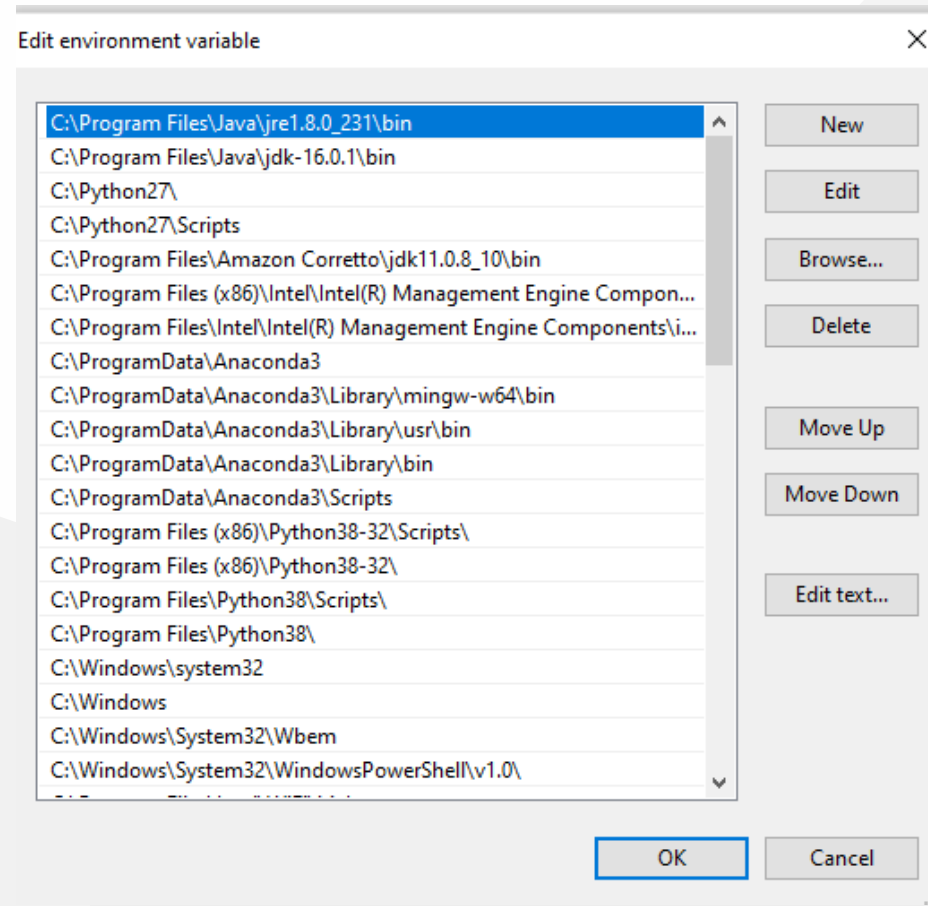


Shared Library Development - (Eclipse Java Jar Library)-30

- Check system settings

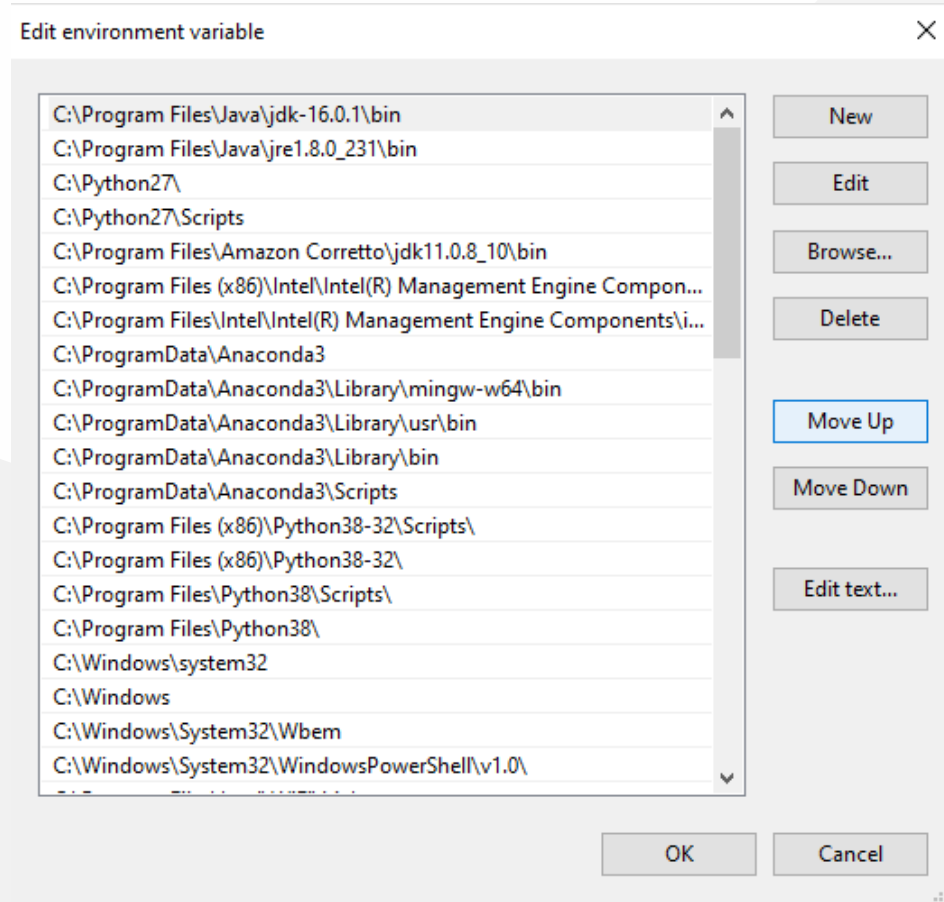


Shared Library Development - (Eclipse Java Jar Library)-31



Shared Library Development - (Eclipse Java Jar Library)-32

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



Shared Library Development - (Eclipse Java Jar Library)-33

- 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

Shared Library Development - (Eclipse Java Jar Library)-34

- 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:\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>
```

Shared Library Development - (Eclipse Java Jar Library)-35

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 is 9
Results is 9
```

Shared Library Development - (Eclipse Java Jar Library)-36

- 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](#)

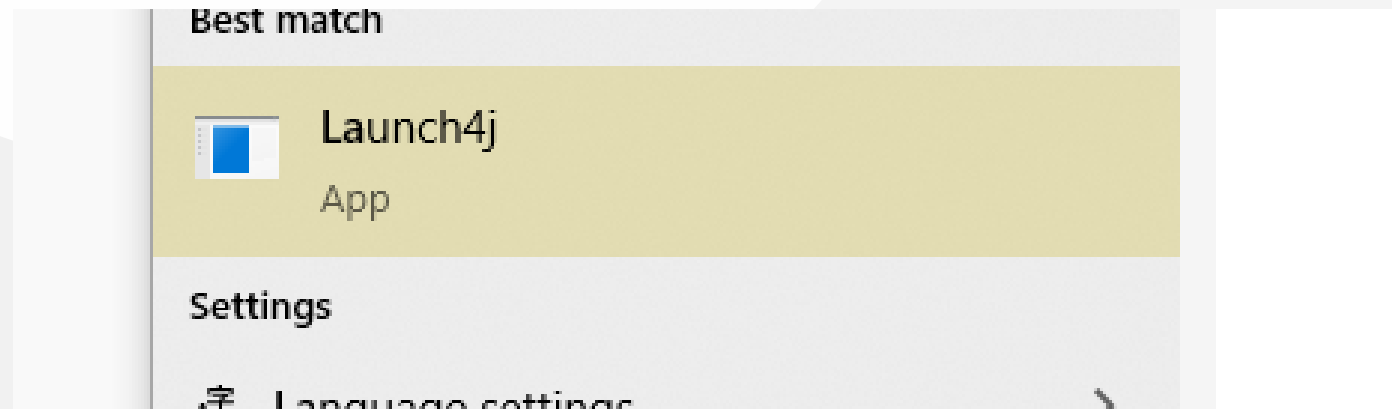


Shared Library Development - (Eclipse Java Jar Library)-37

- you can watch this tutorial also
 - [How to convert jar to exe using Launch4J Full explanation - YouTube](#)

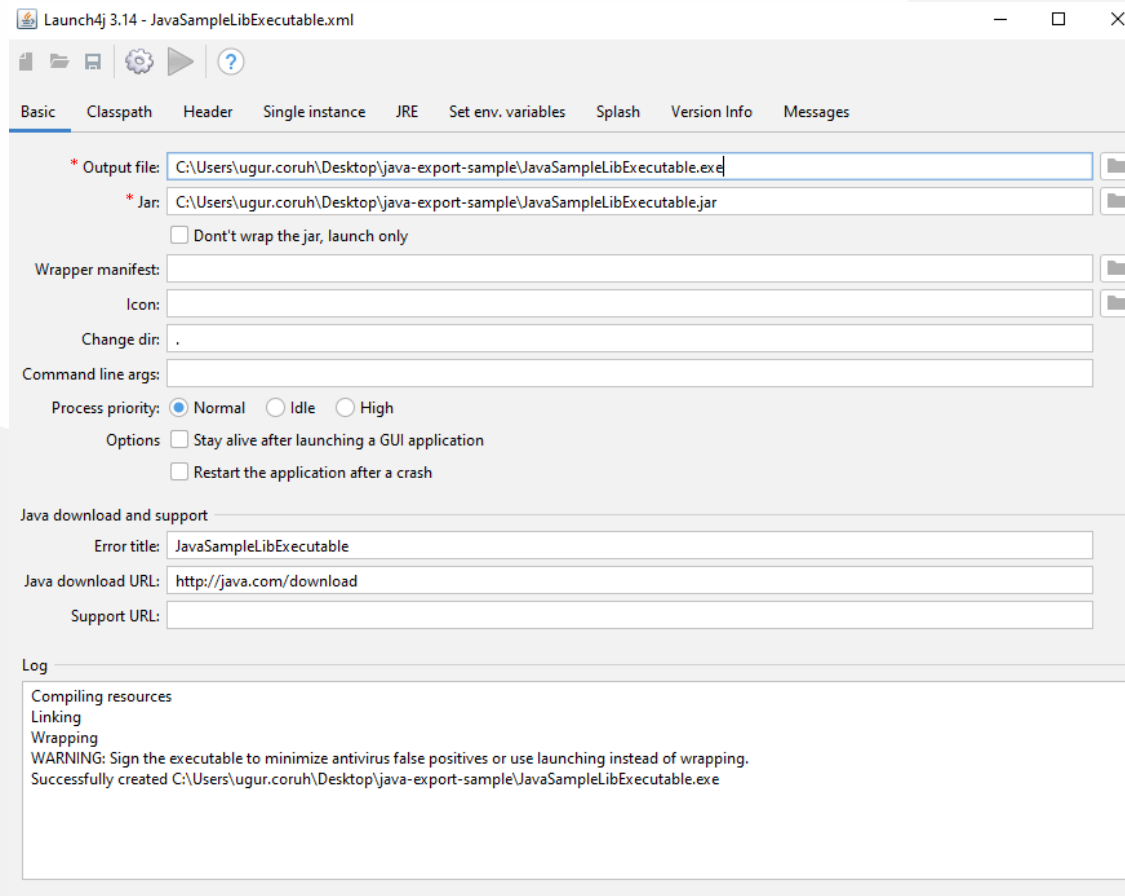
Shared Library Development - (Eclipse Java Jar Library)-38

- Download and install launch4j and open application



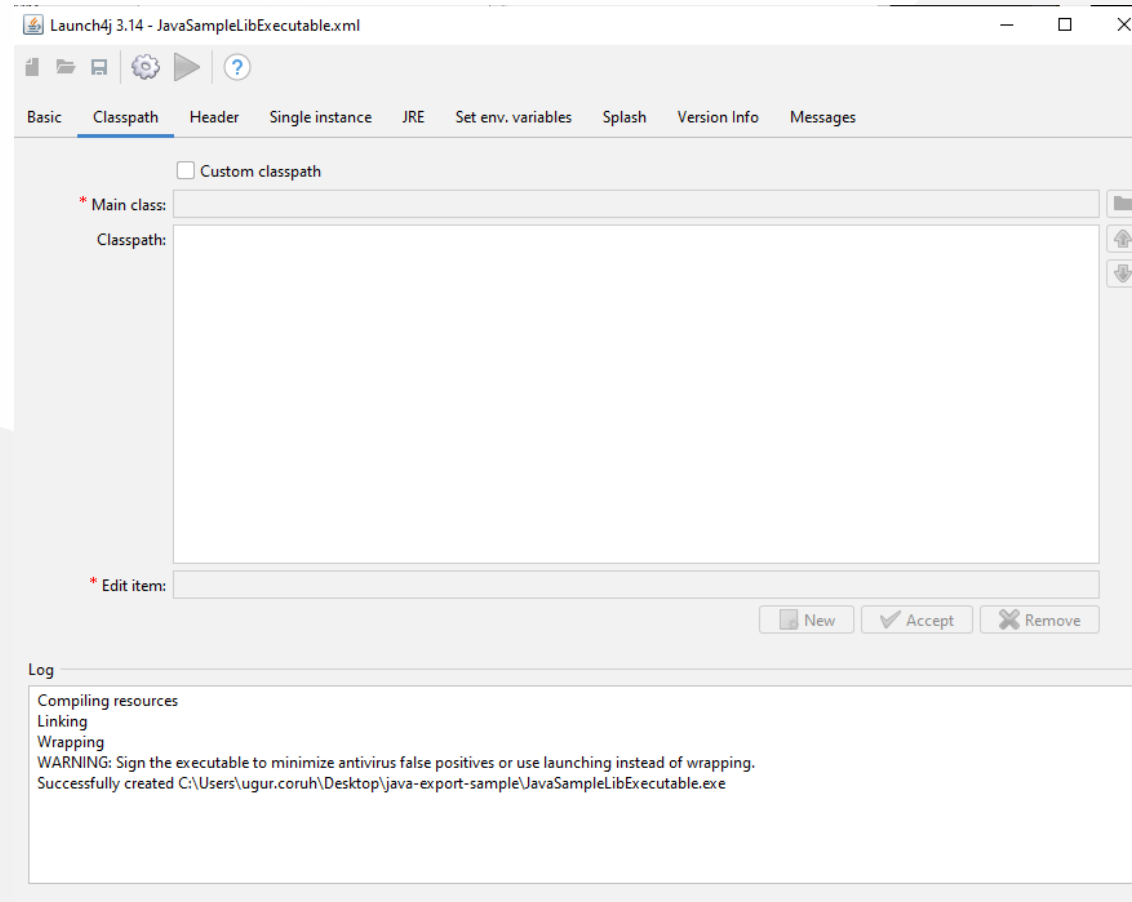
Shared Library Development - (Eclipse Java Jar Library)-39

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



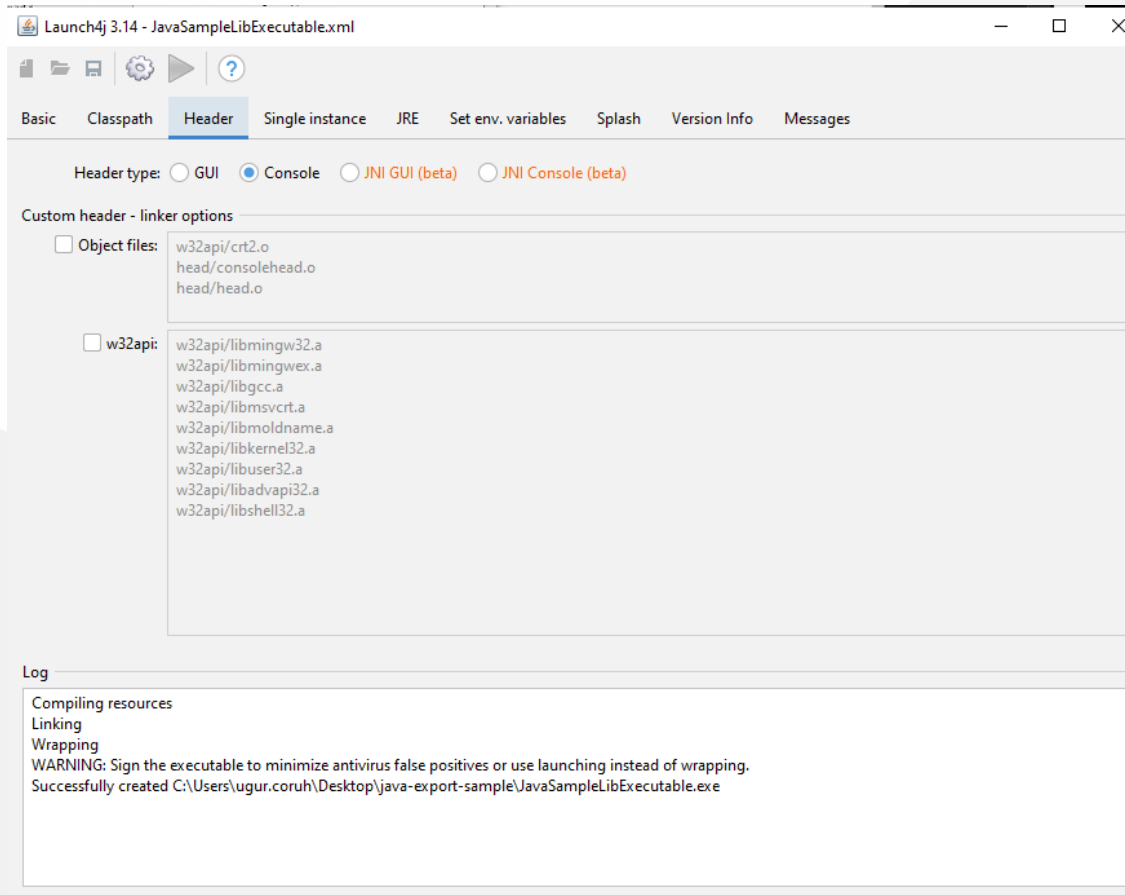
Shared Library Development - (Eclipse Java Jar Library)-40

- We can customize main class if have multiple main class



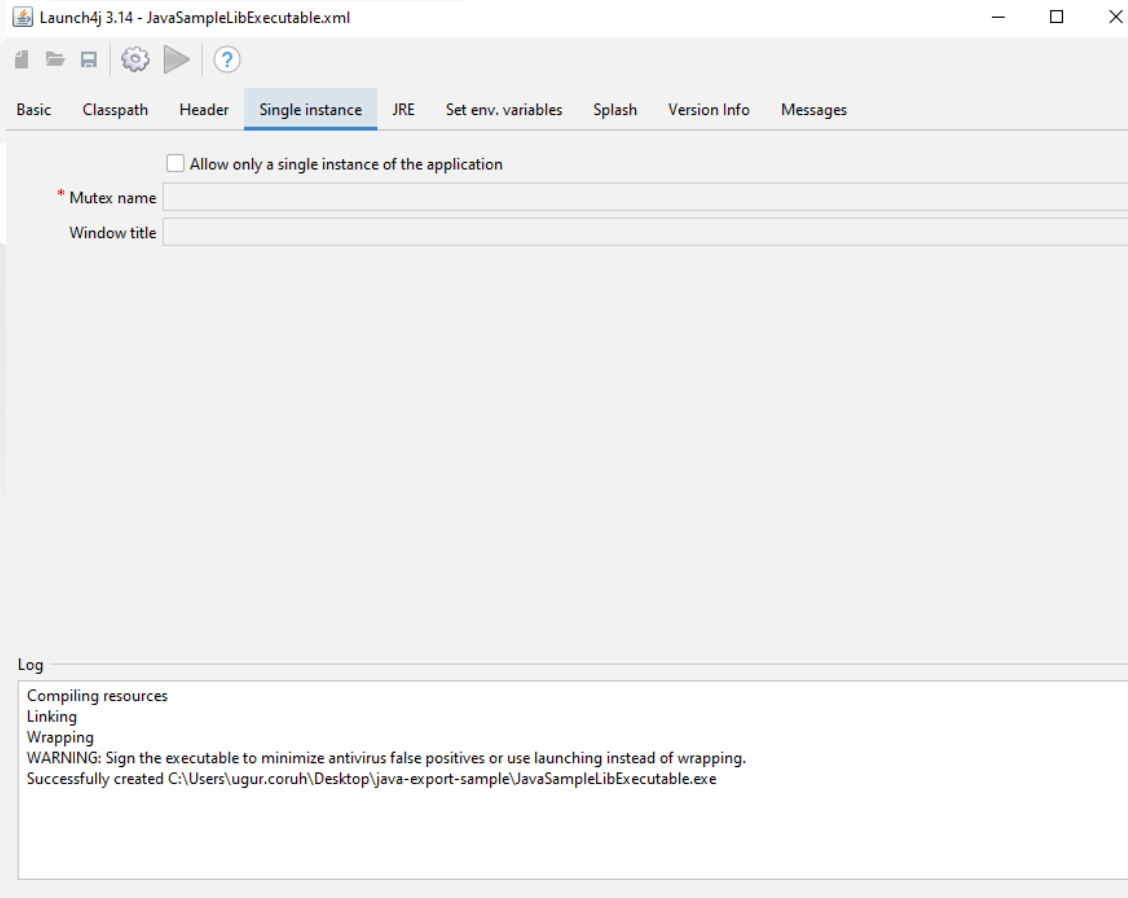
Shared Library Development - (Eclipse Java Jar Library)-41

select console from setting for this application



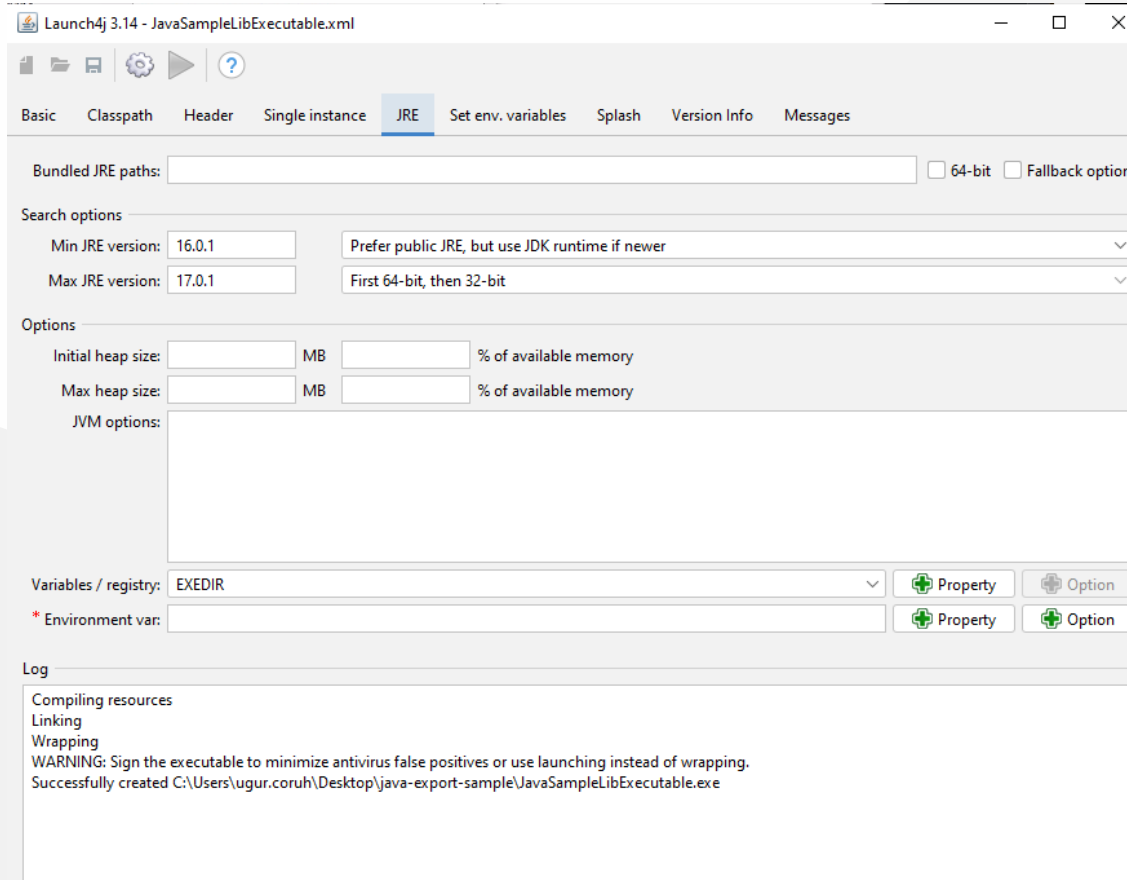
Shared Library Development - (Eclipse Java Jar Library)-42

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



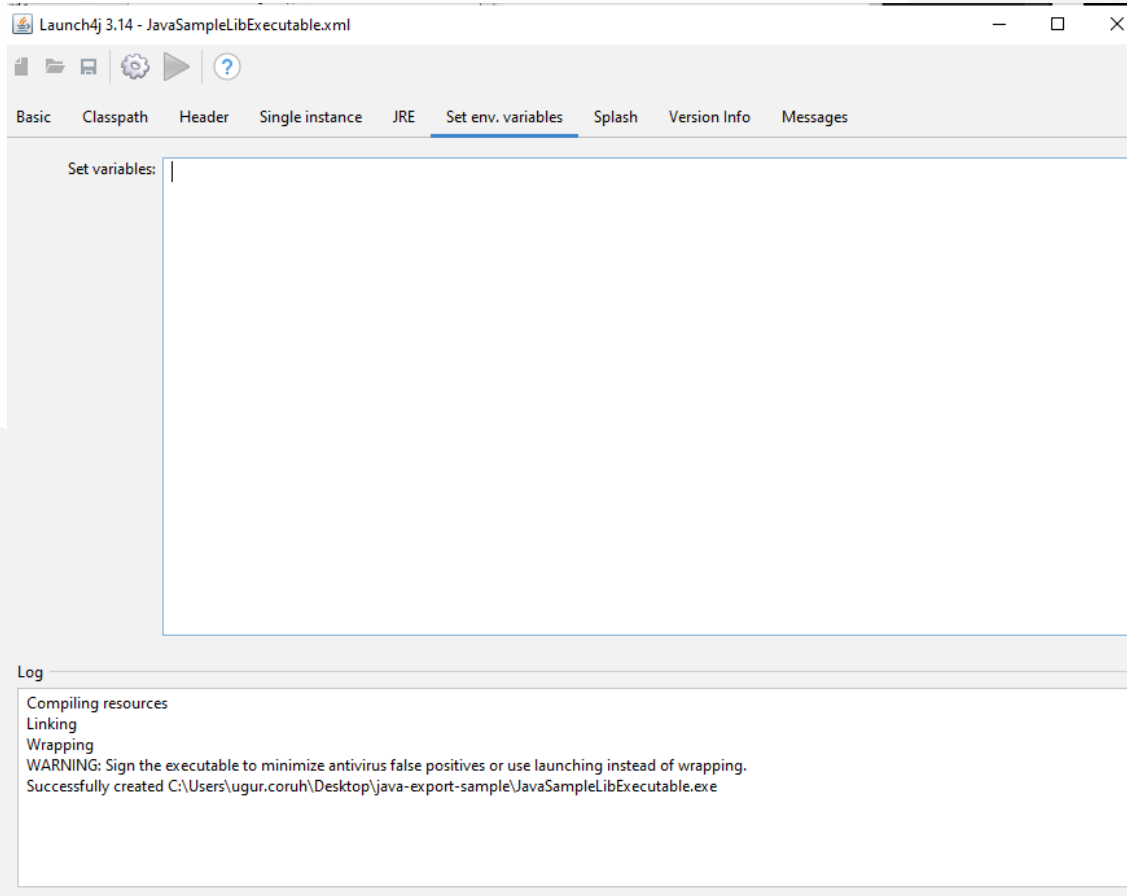
Shared Library Development - (Eclipse Java Jar Library)-43

- we need to set runtime environment versions



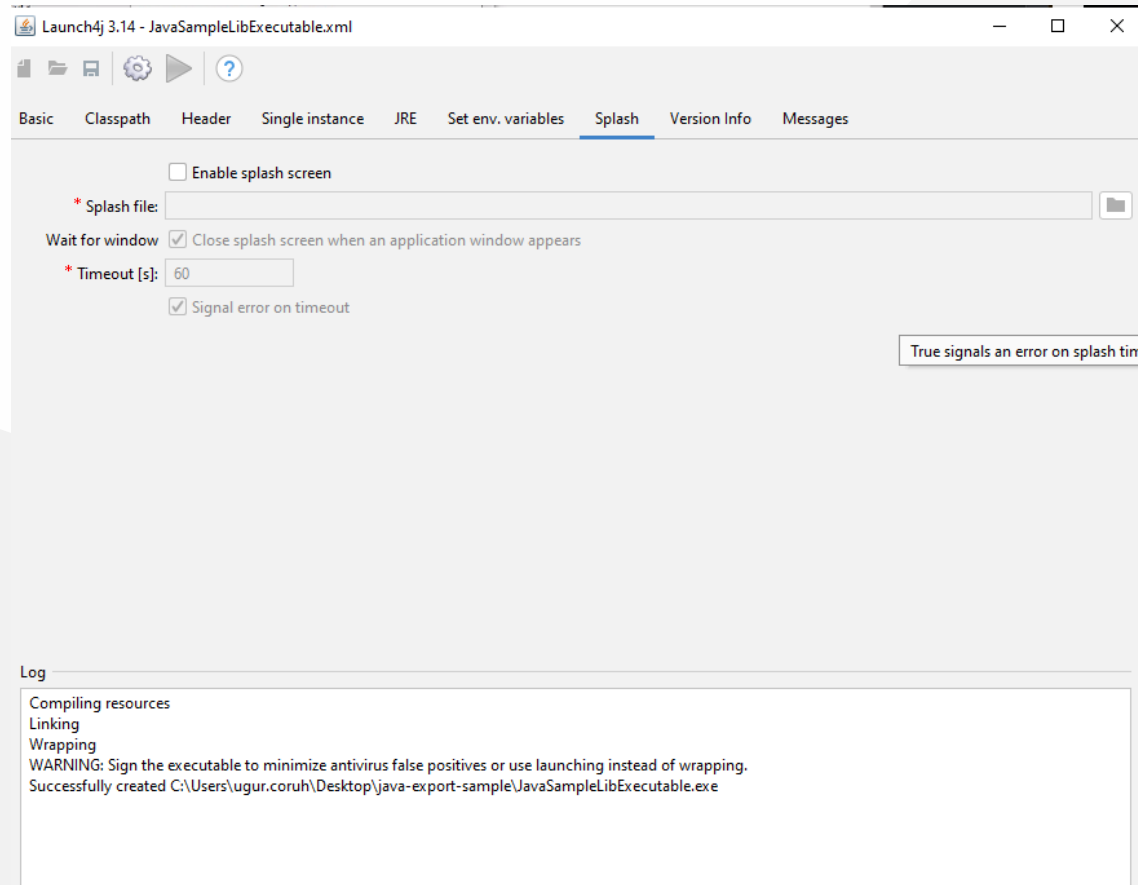
Shared Library Development - (Eclipse Java Jar Library)-44

you can set system parameters before running application



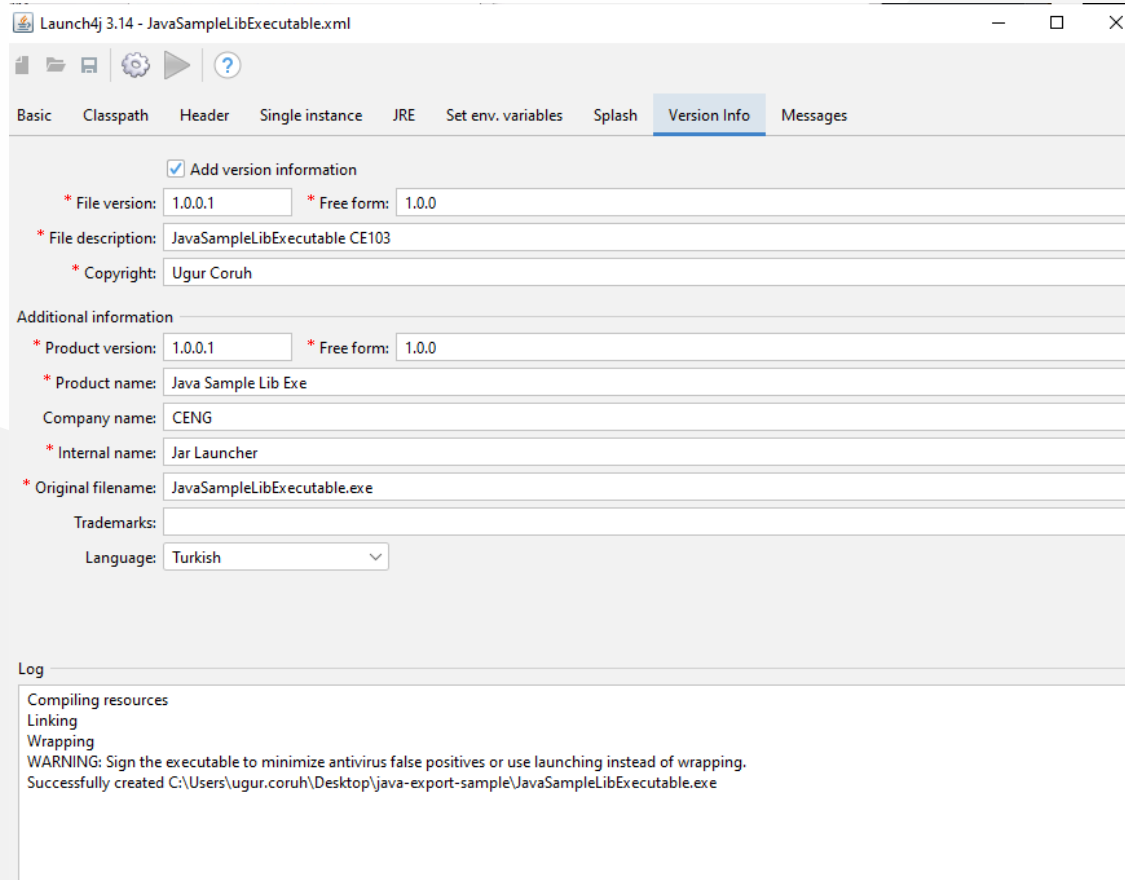
Shared Library Development - (Eclipse Java Jar Library)-45

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



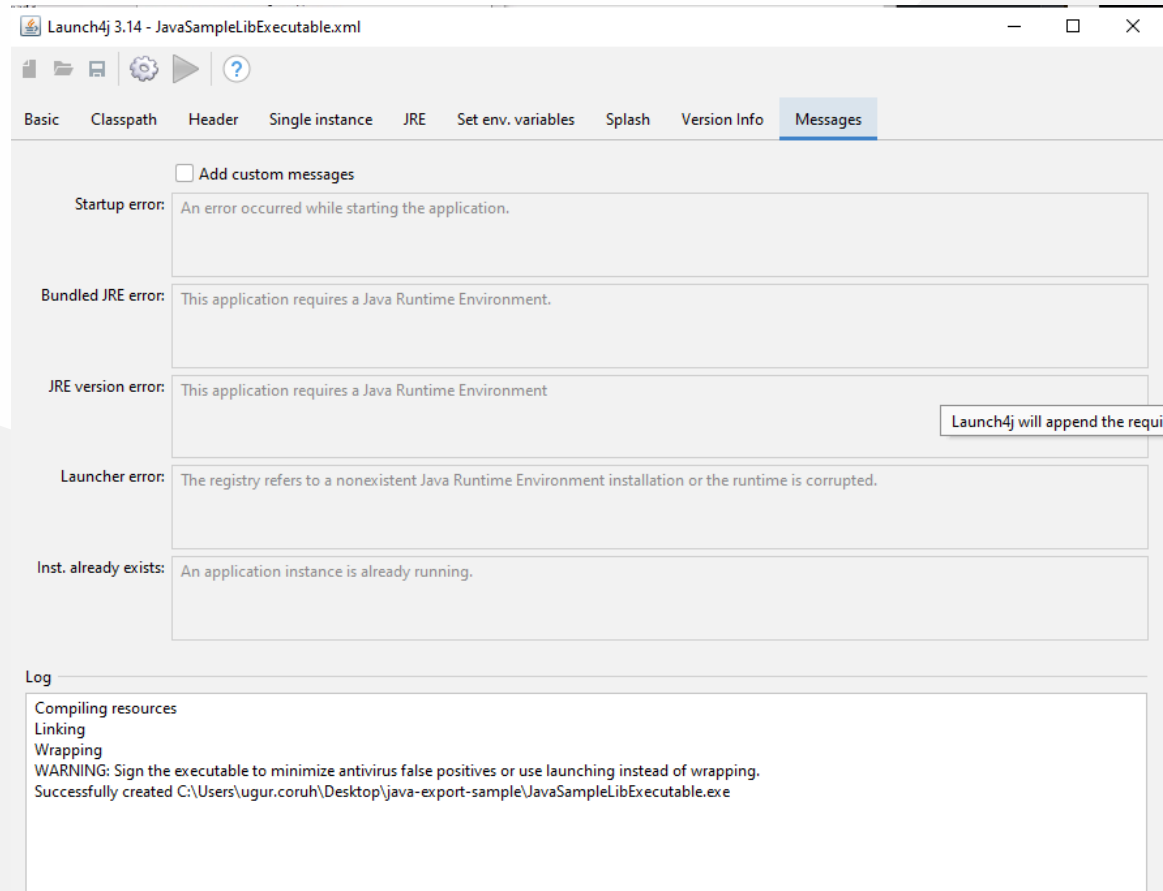
Shared Library Development - (Eclipse Java Jar Library)-46

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



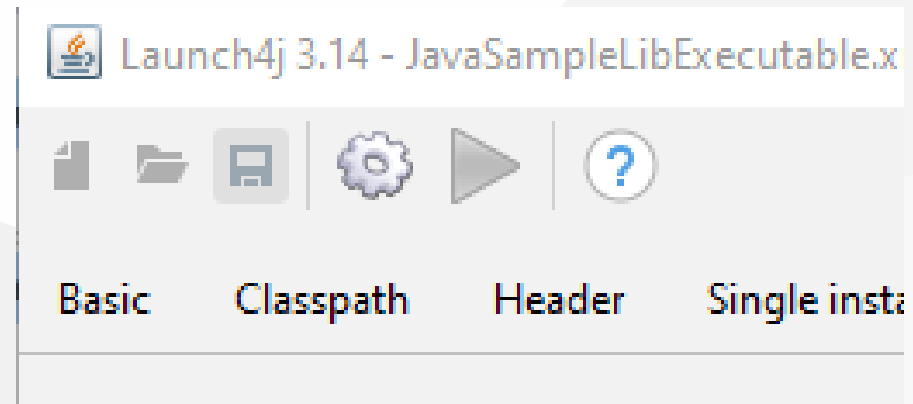
Shared Library Development - (Eclipse Java Jar Library)-47

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



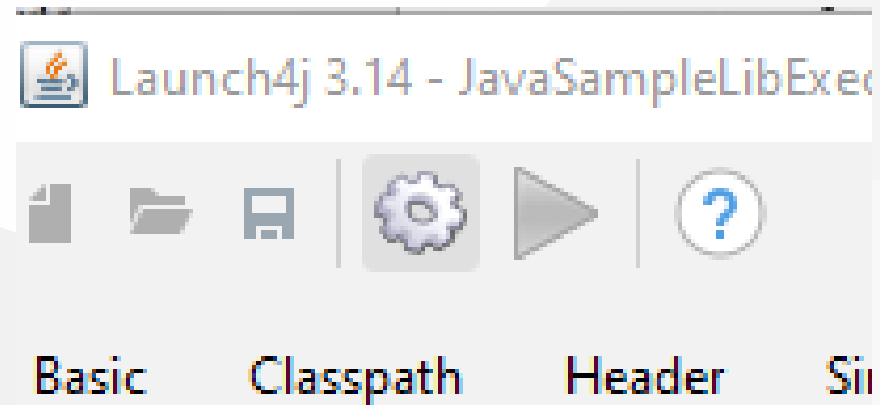
Shared Library Development - (Eclipse Java Jar Library)-48

- with this options save configuration file xml



Shared Library Development - (Eclipse Java Jar Library)-49

- and compile settings



Shared Library Development - (Eclipse Java Jar Library)-50

- You will see generated output file in log screen

```
Compiling resources
```

```
Linking
```

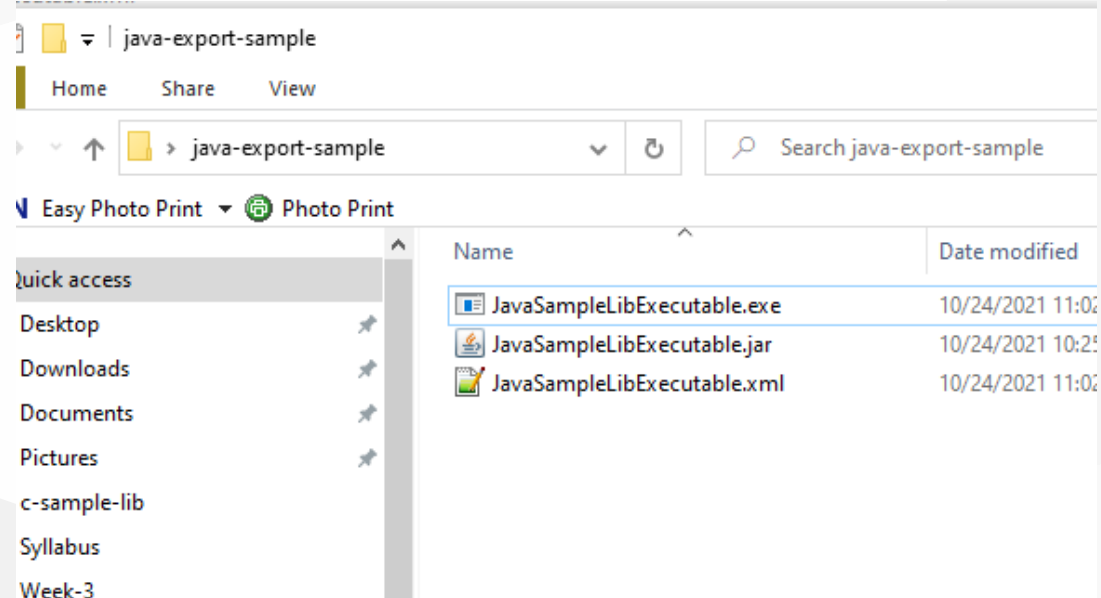
```
Wrapping
```

```
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
```

Shared Library Development - (Eclipse Java Jar Library)-51

- now we can run exe by click



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

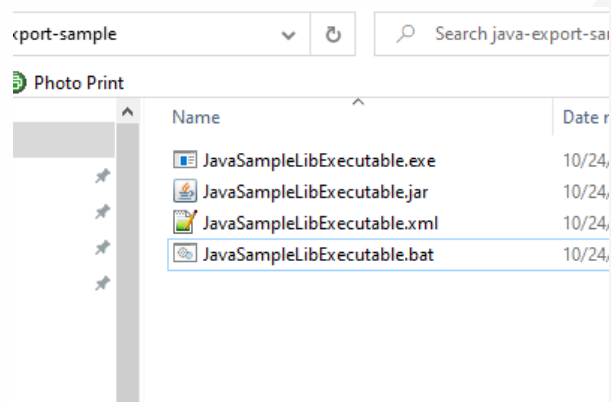
Shared Library Development - (Eclipse Java Jar Library)-52

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

Shared Library Development - (Eclipse Java Jar Library)-53

JavaSampleLibExecutable.bat

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



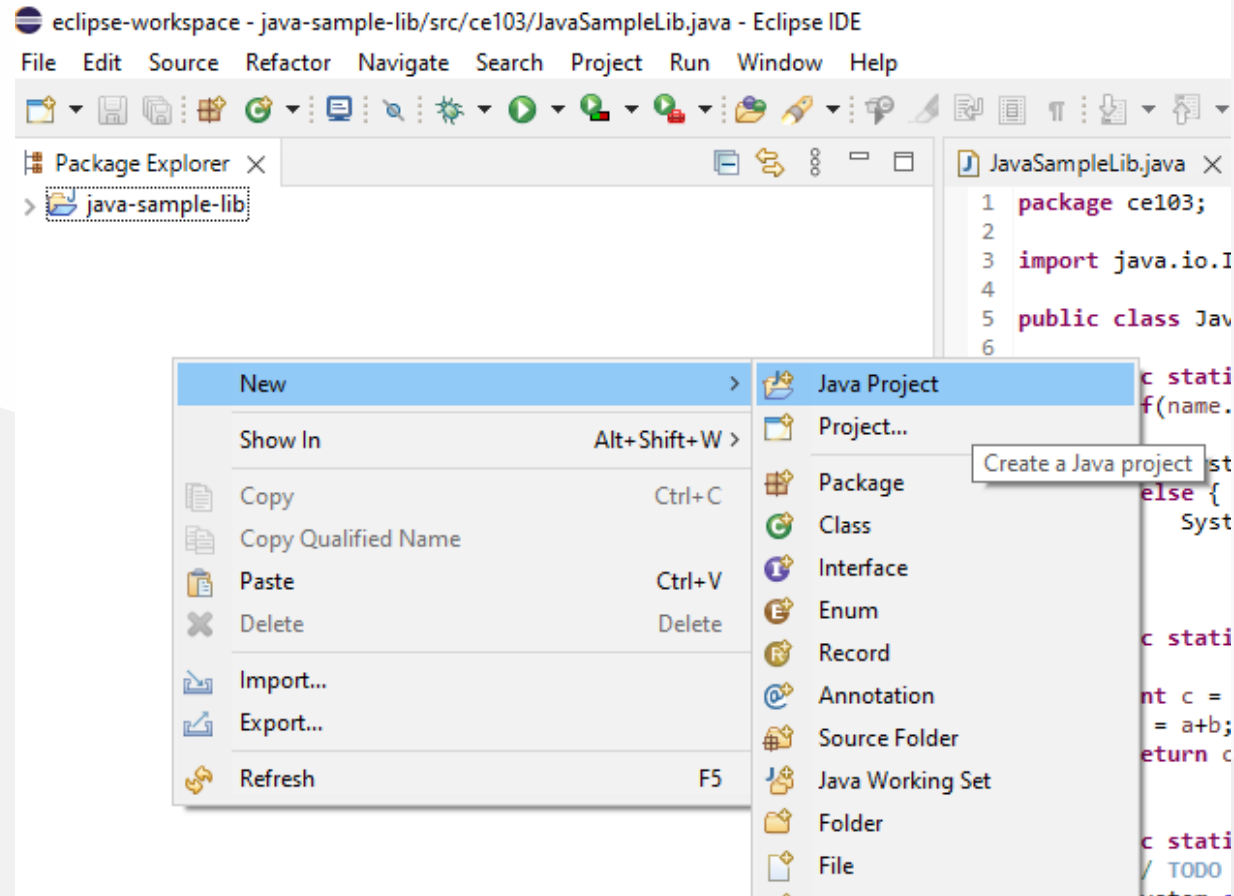
- if we click bat file then we will automate command line task for current jar file

```
C:\WINDOWS\system32\cmd.exe

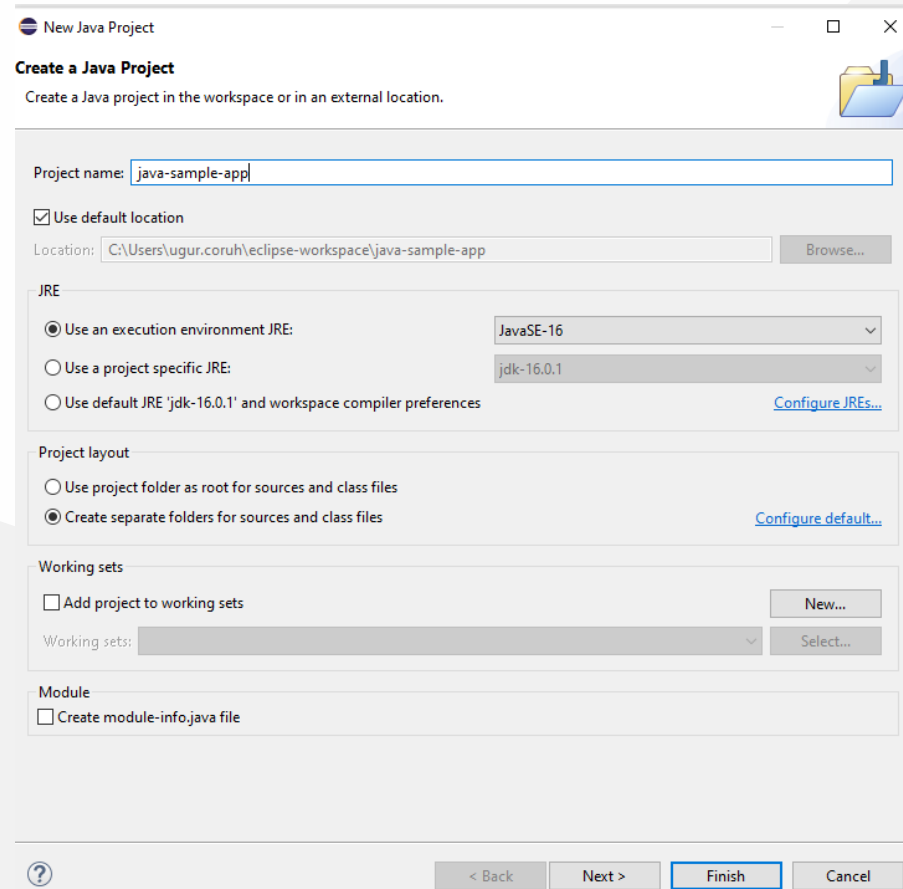
C:\Users\ugur.coruh\Desktop\java-export-sample>java -jar JavaSampleLibExecutable.jar
Hello World!
Hello There
Results is 9
Results is 9
```


Shared Library Development - (Eclipse Java Jar Library)-54

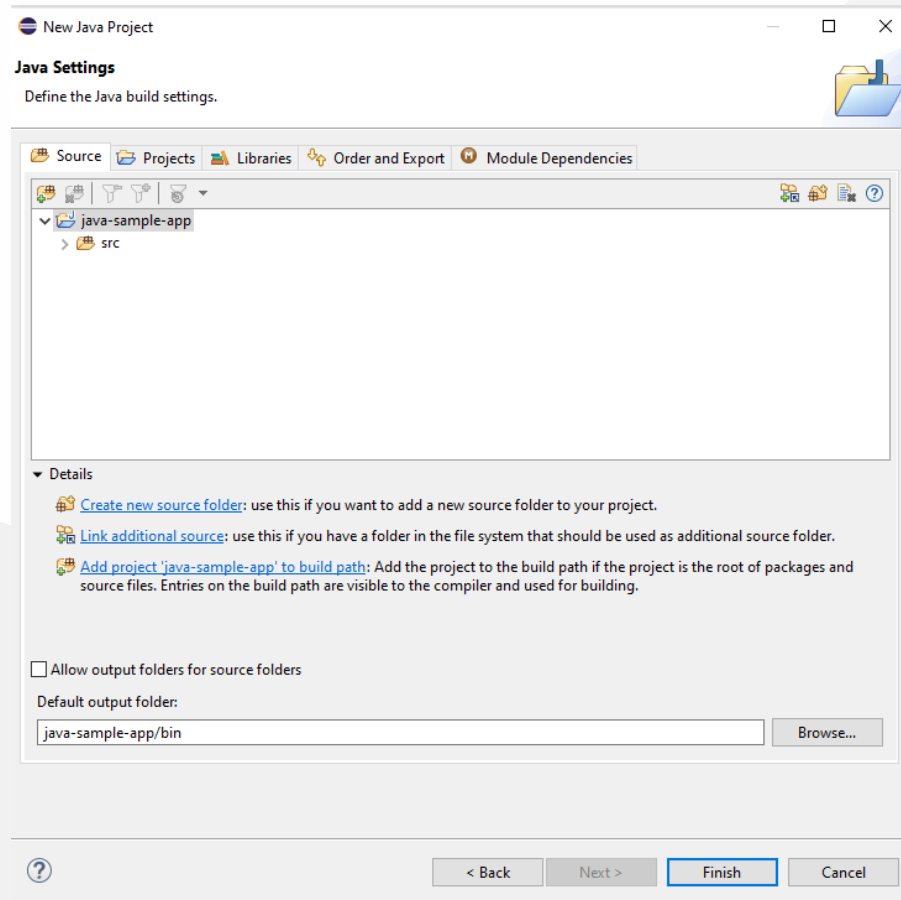
Now return back to our java library and create another console application that use library functions



Shared Library Development - (Eclipse Java Jar Library)-55

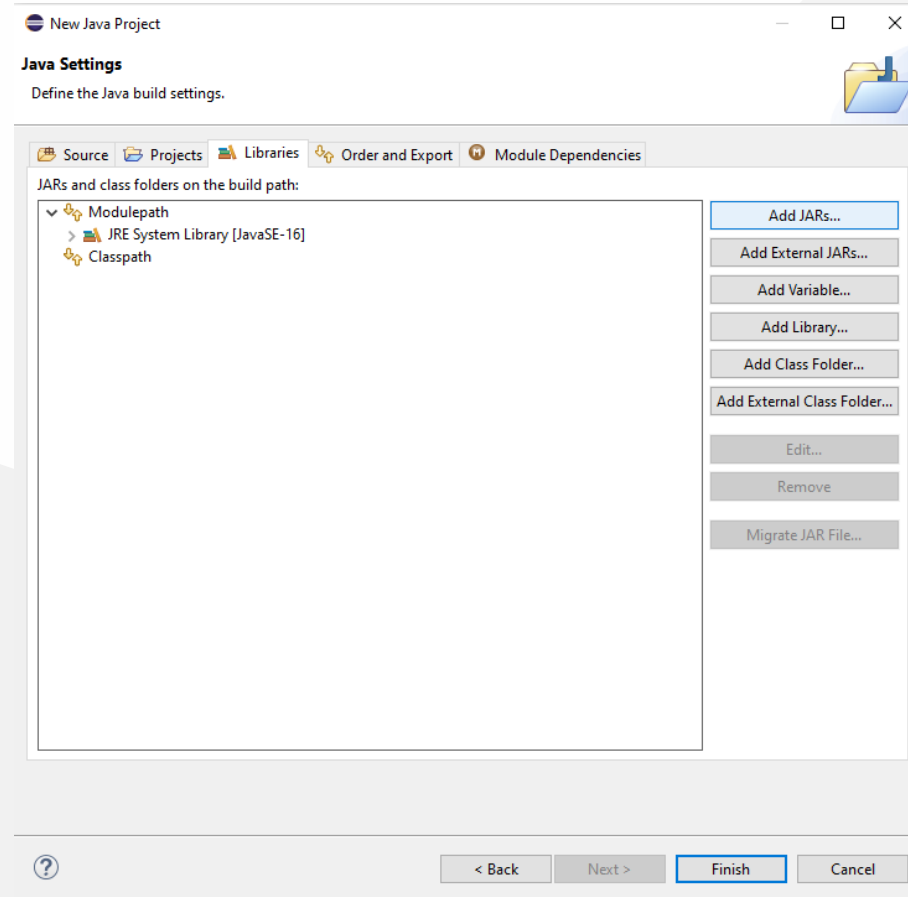


Shared Library Development - (Eclipse Java Jar Library)-56



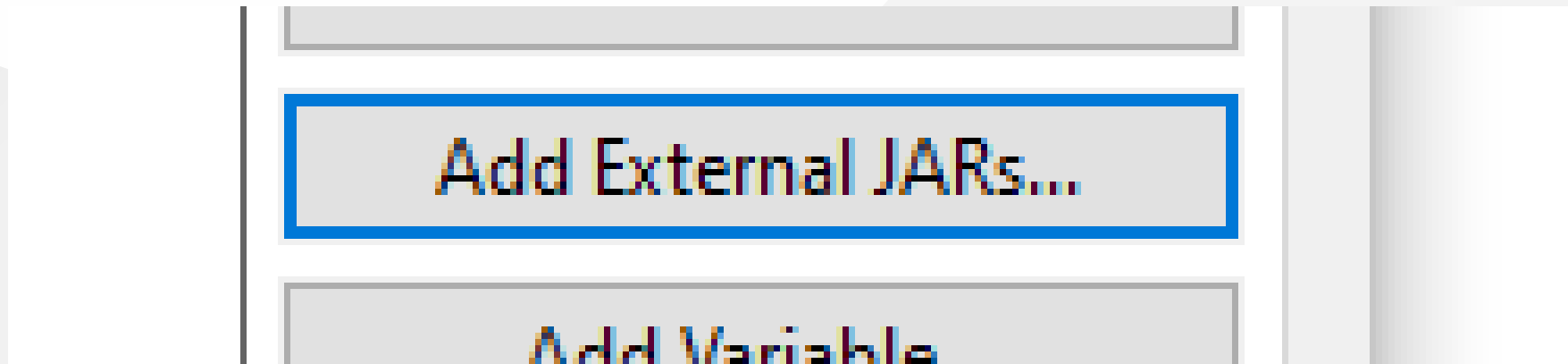
Shared Library Development - (Eclipse Java Jar Library)-57

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



Shared Library Development - (Eclipse Java Jar Library)-58

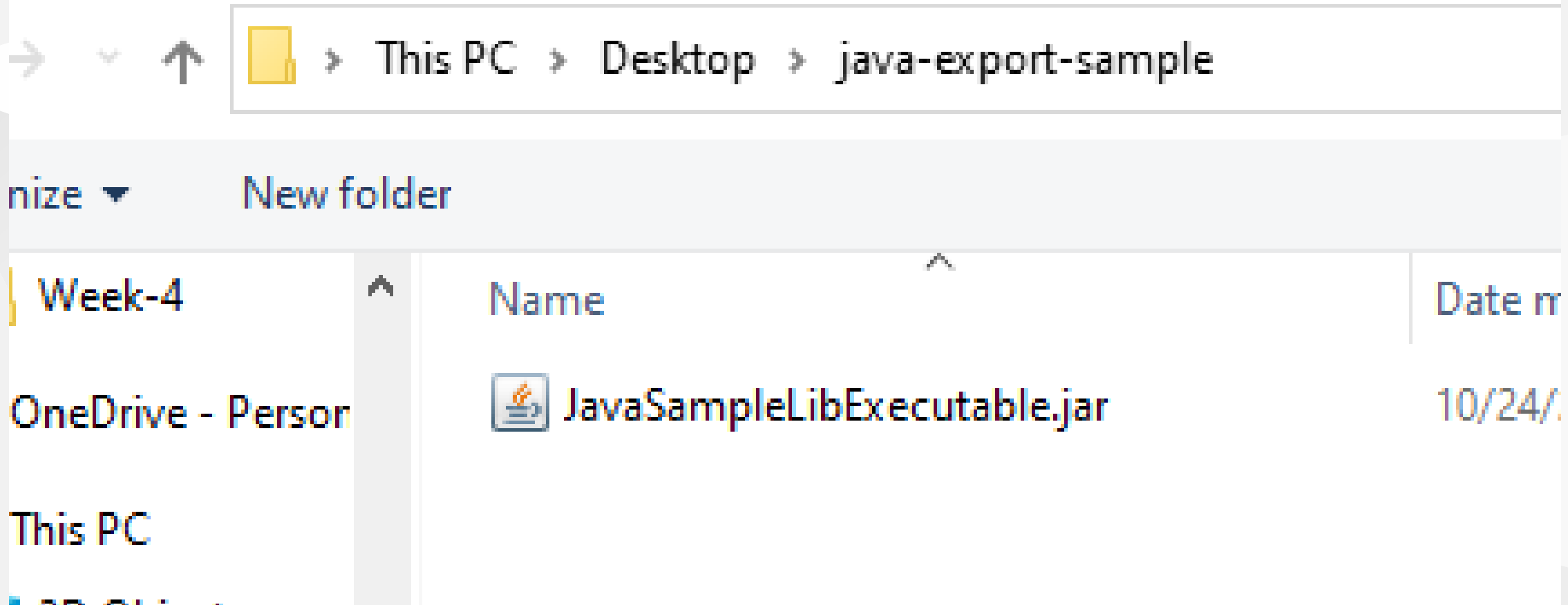
- Select Add External JARs...



Shared Library Development - (Eclipse Java Jar Library)-59

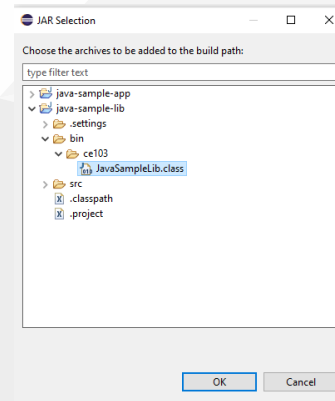
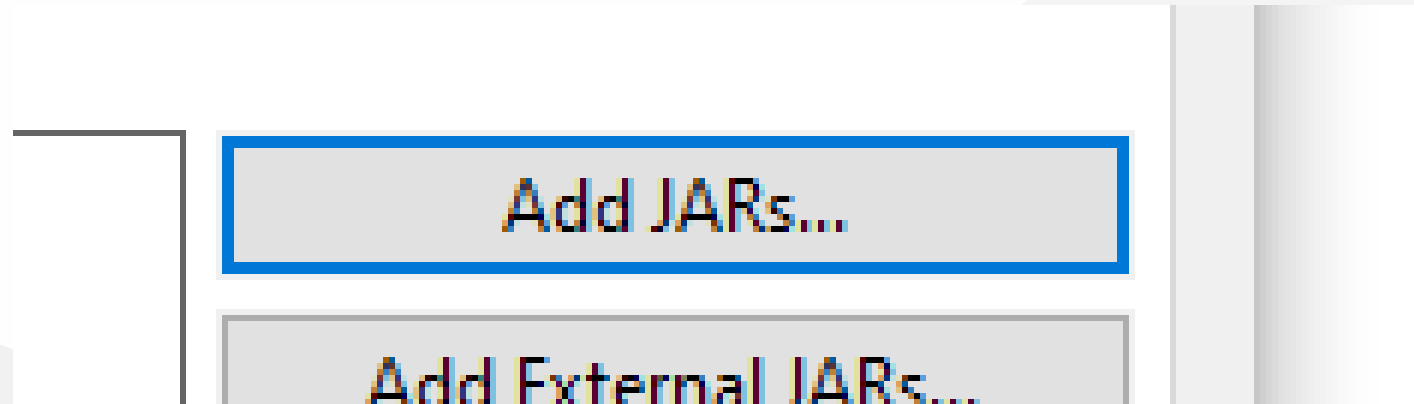
- Open Exported jar folder and select

{ Selection



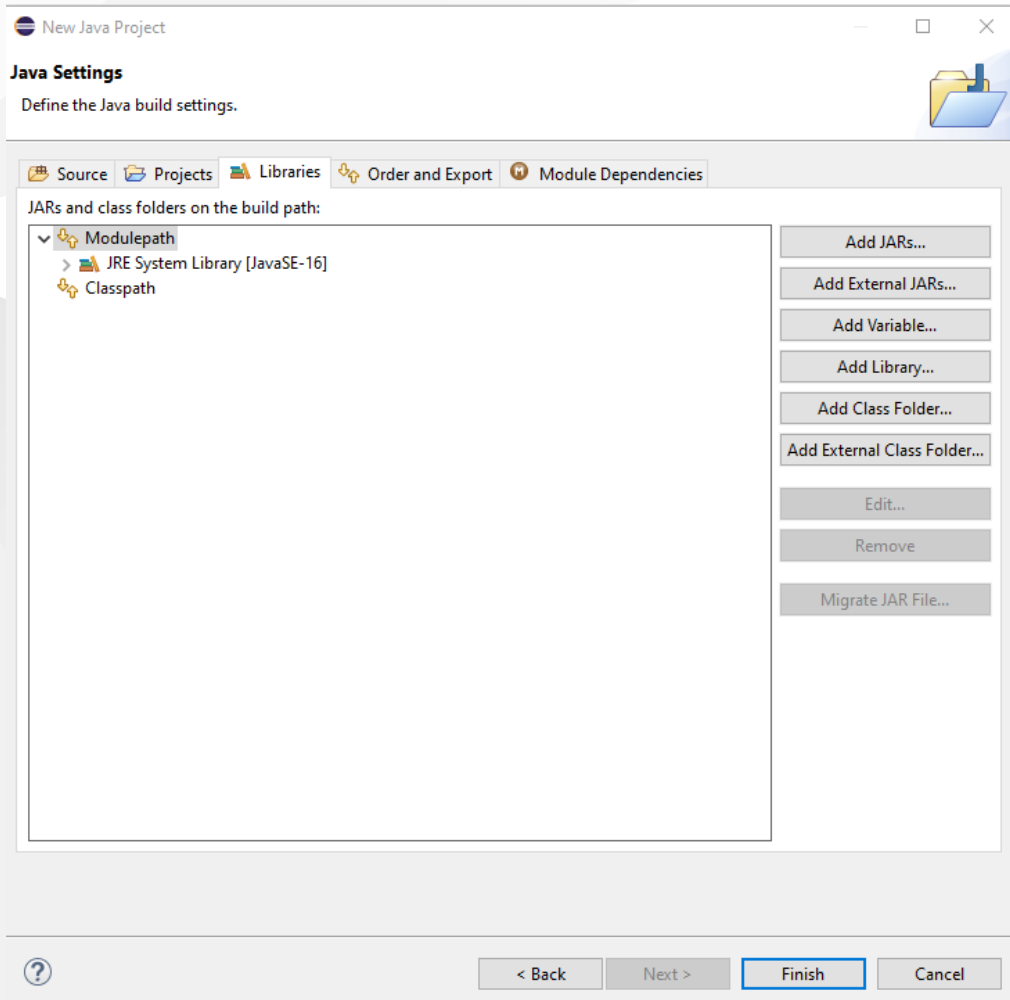
Shared Library Development - (Eclipse Java Jar Library)-60

- Or we can select by Add jar from current workspace



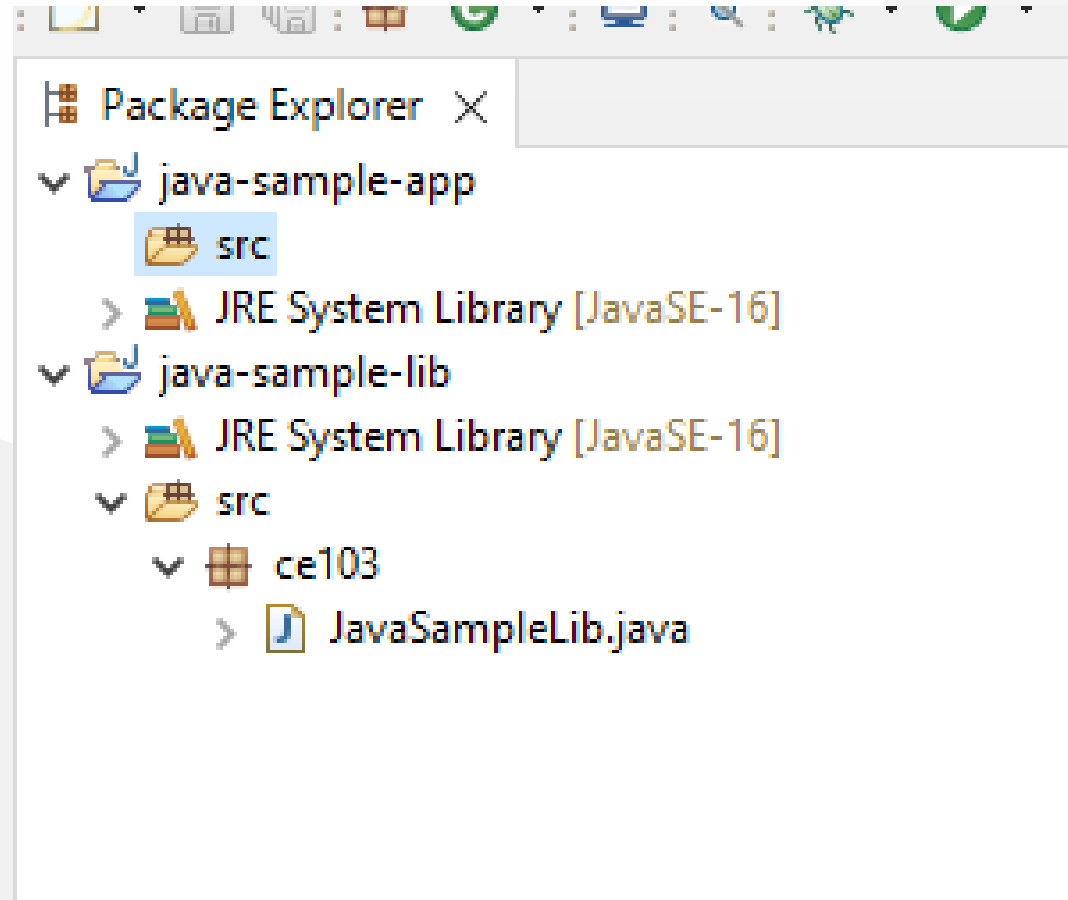
Shared Library Development - (Eclipse Java Jar Library)-61

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



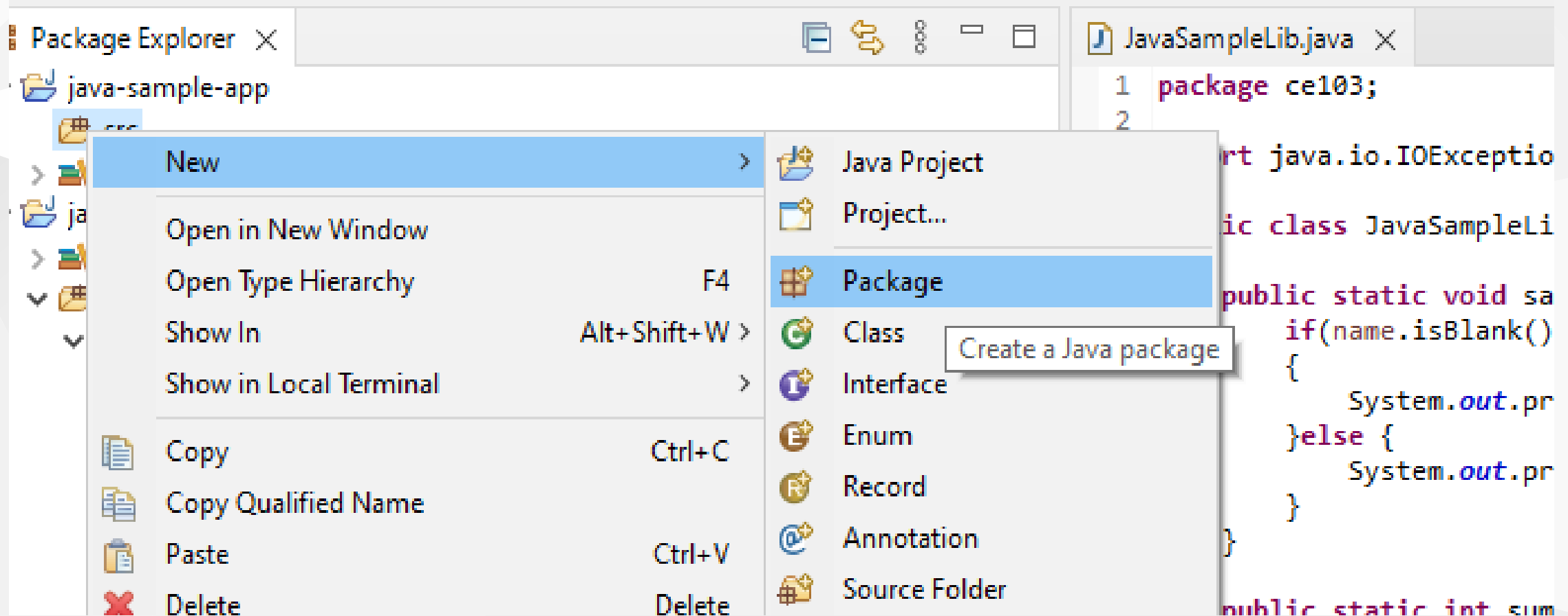
Shared Library Development - (Eclipse Java Jar Library)-62

- we will have the following project



Shared Library Development - (Eclipse Java Jar Library)-63

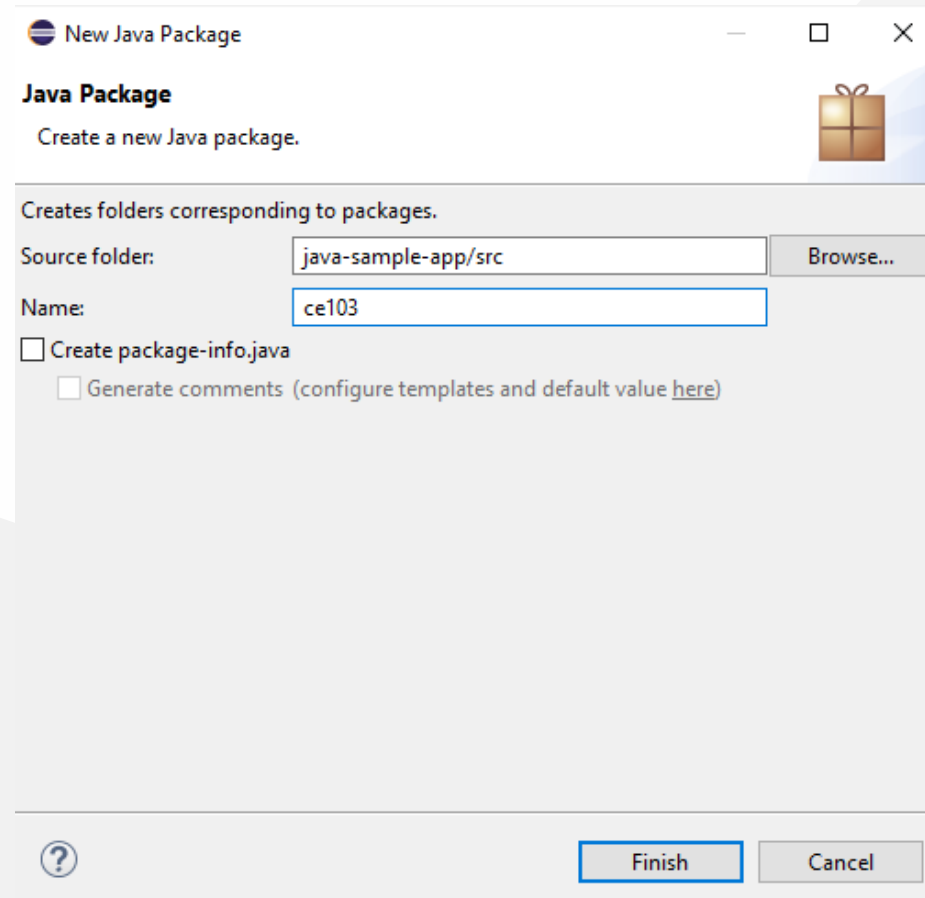
- lets create a package



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays a project named 'java-sample-app'. A context menu is open over a package icon, with the 'New' option selected. This opens a sub-menu where 'Package' is highlighted. A tooltip 'Create a Java package' is visible over the 'Package' option. The main editor window shows the code for 'JavaSampleLib.java', which includes the package declaration 'package ce103;' and the start of a class definition.

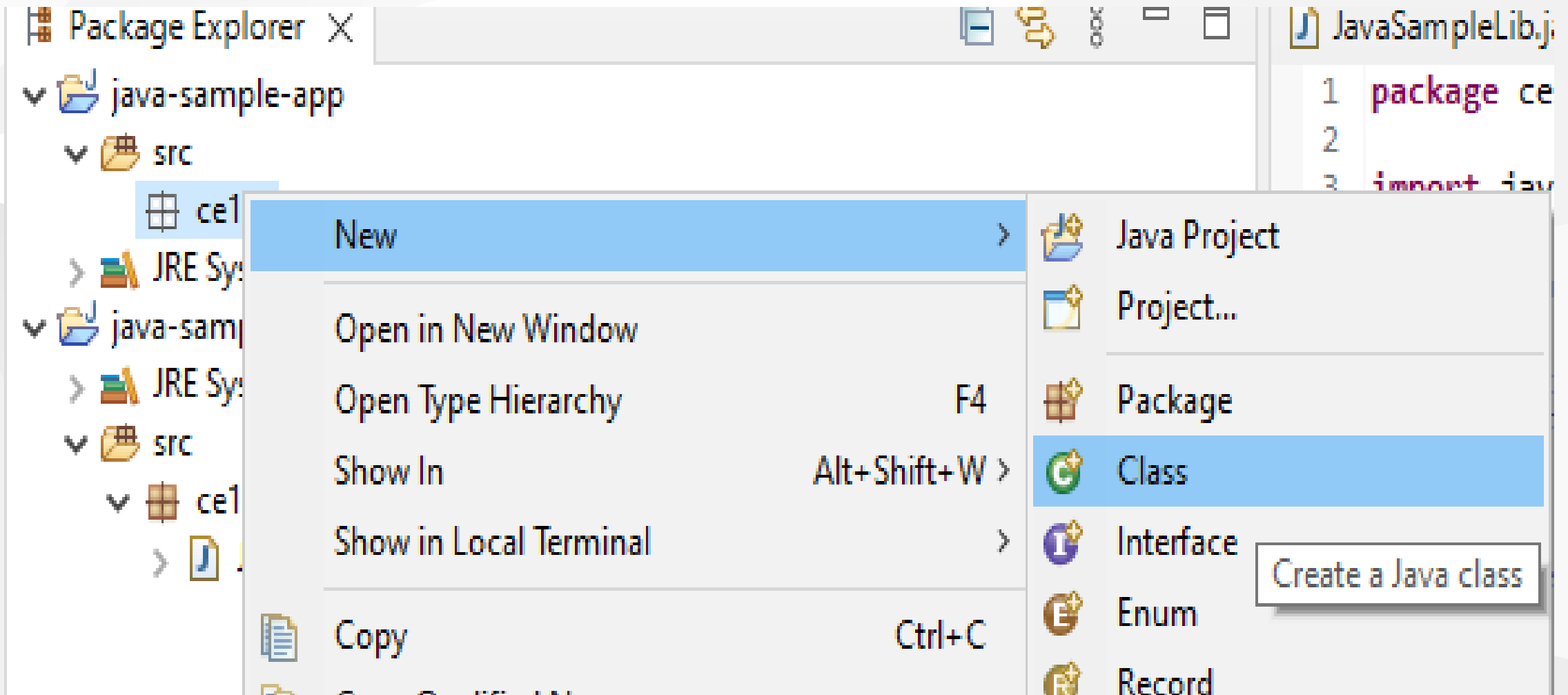
```
1 package ce103;  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

Shared Library Development - (Eclipse Java Jar Library)-64



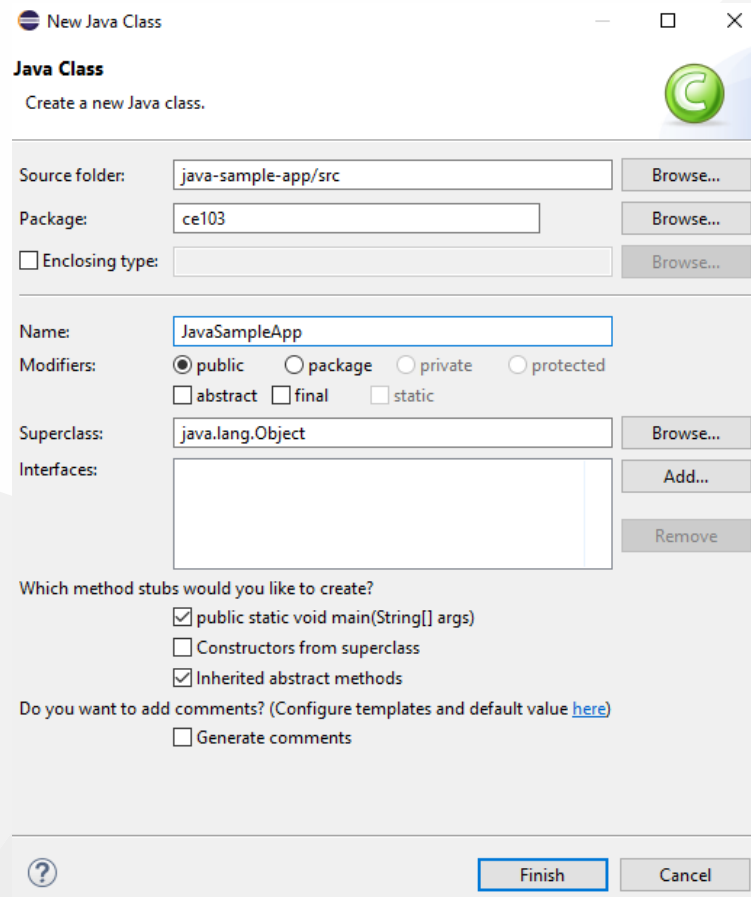
Shared Library Development - (Eclipse Java Jar Library)-65

- and lets create a main class for our application

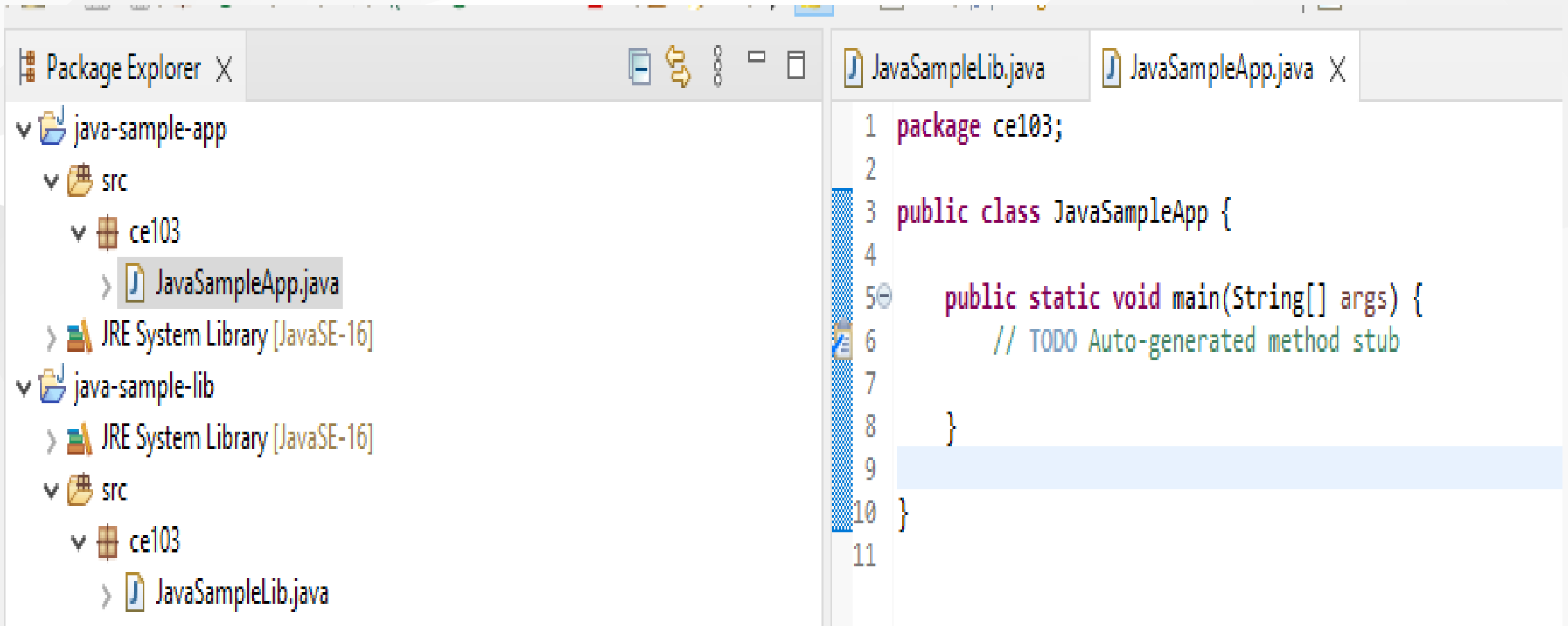


Shared Library Development - (Eclipse Java Jar Library)-66

- check create main function



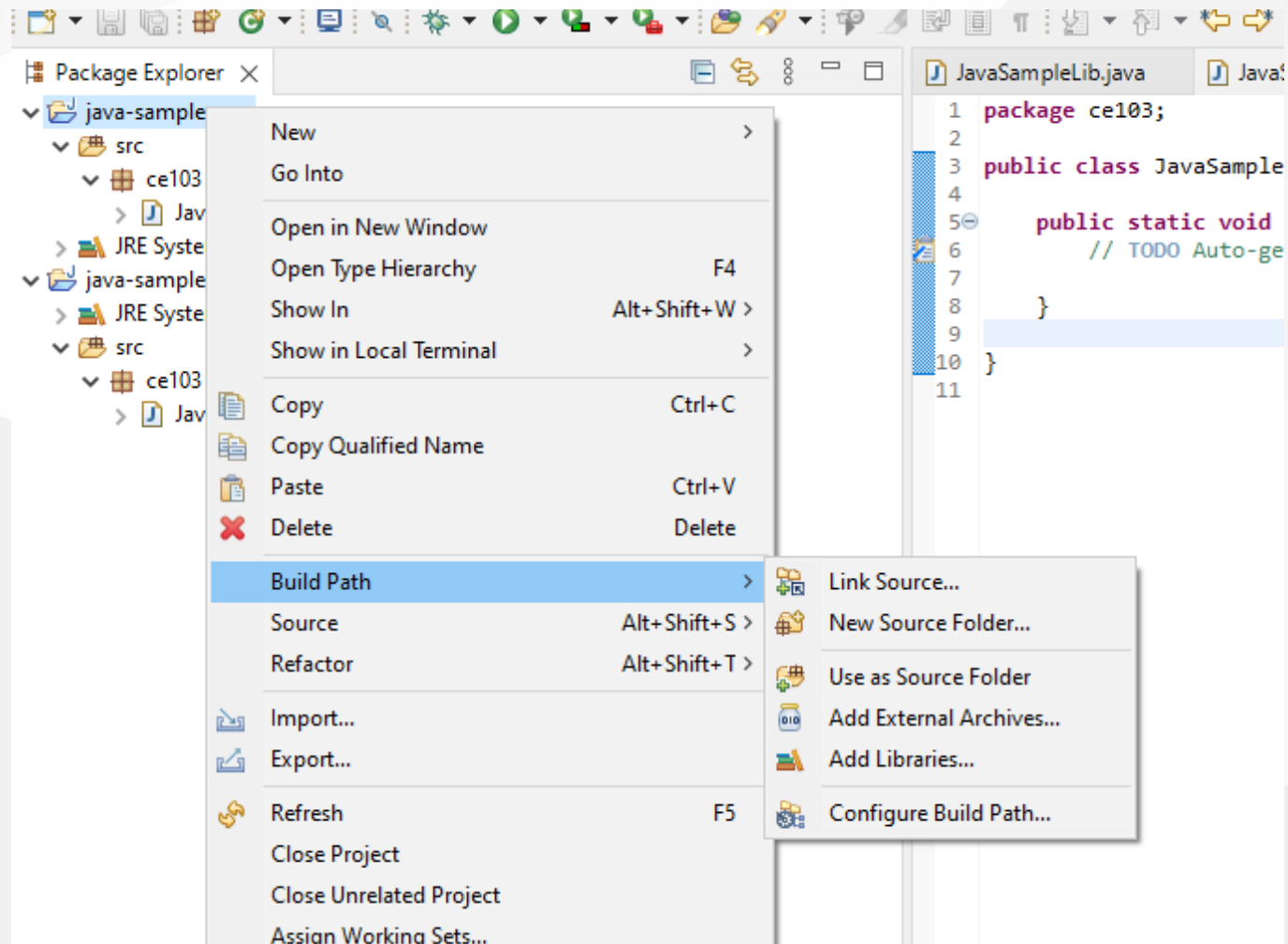
Shared Library Development - (Eclipse Java Jar Library)-67



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

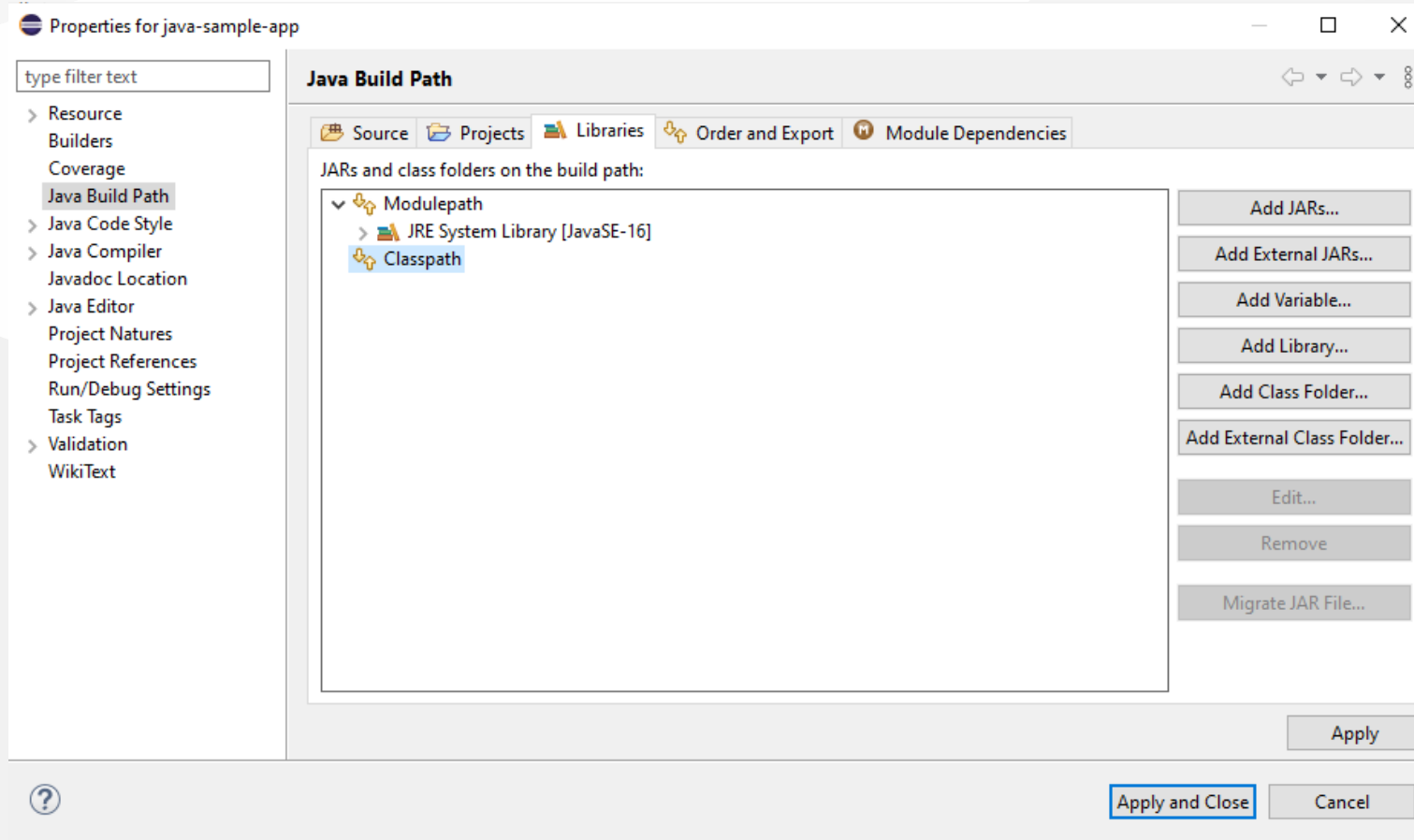
Shared Library Development - (Eclipse Java Jar Library)-68

- right click to project and add reference



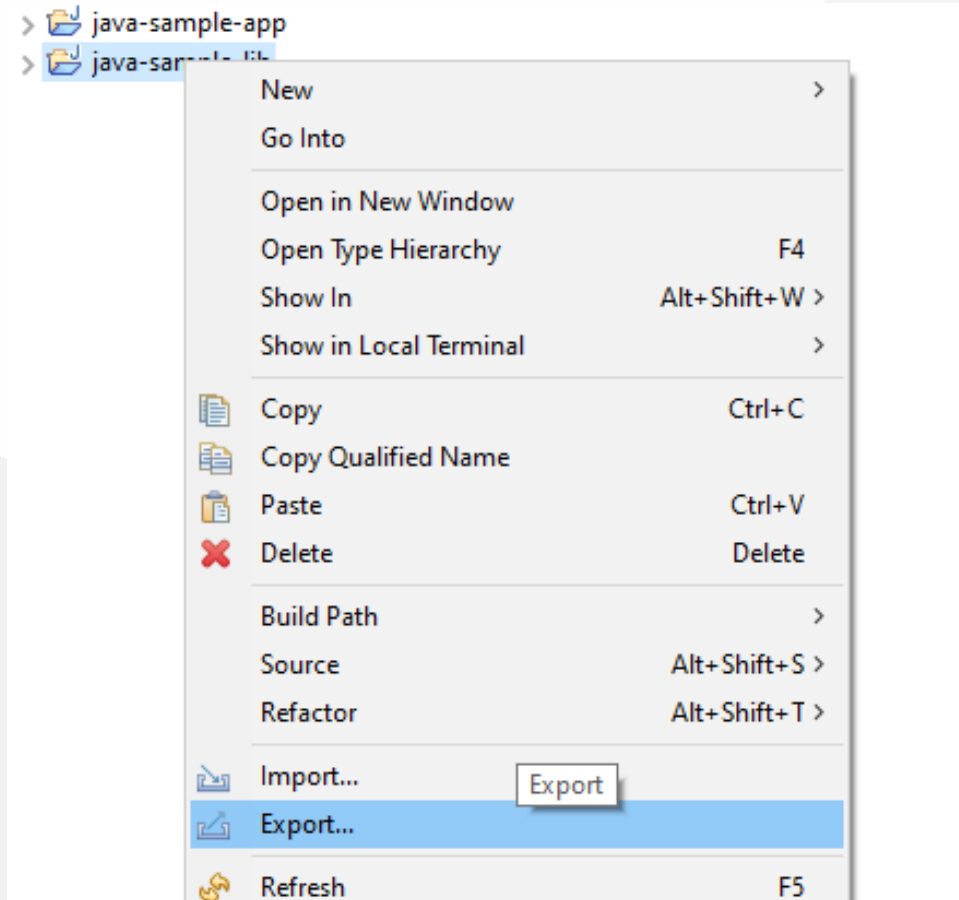
Shared Library Development - (Eclipse Java Jar Library)-69

- you can enter same configurations from project properties



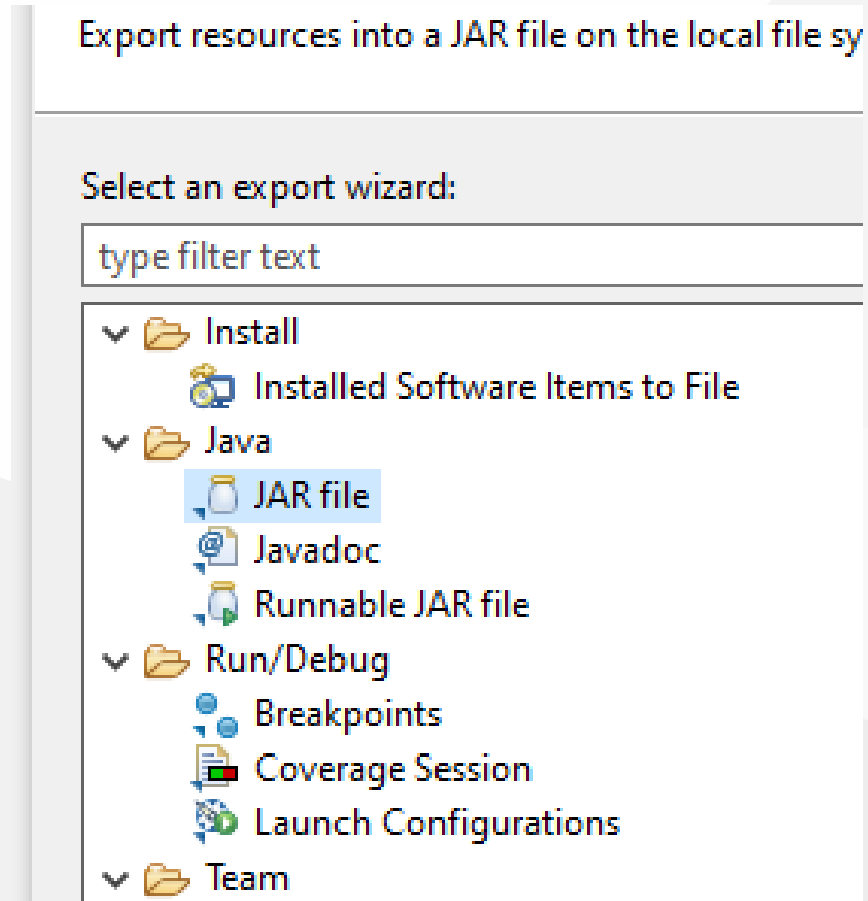
Shared Library Development - (Eclipse Java Jar Library)-70

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



Shared Library Development - (Eclipse Java Jar Library)-71

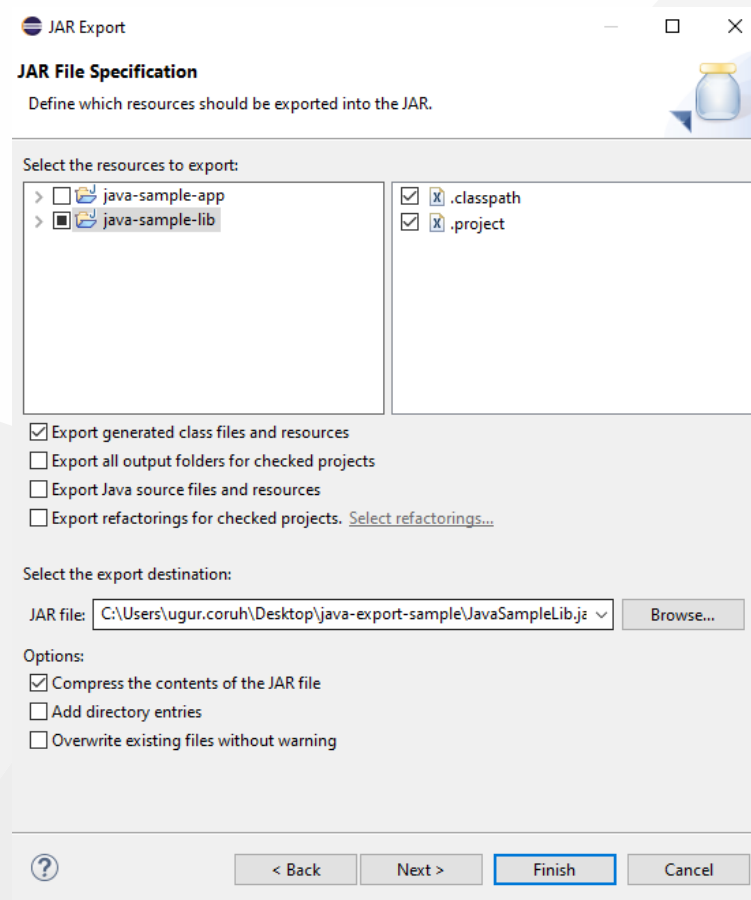
Select JAR file



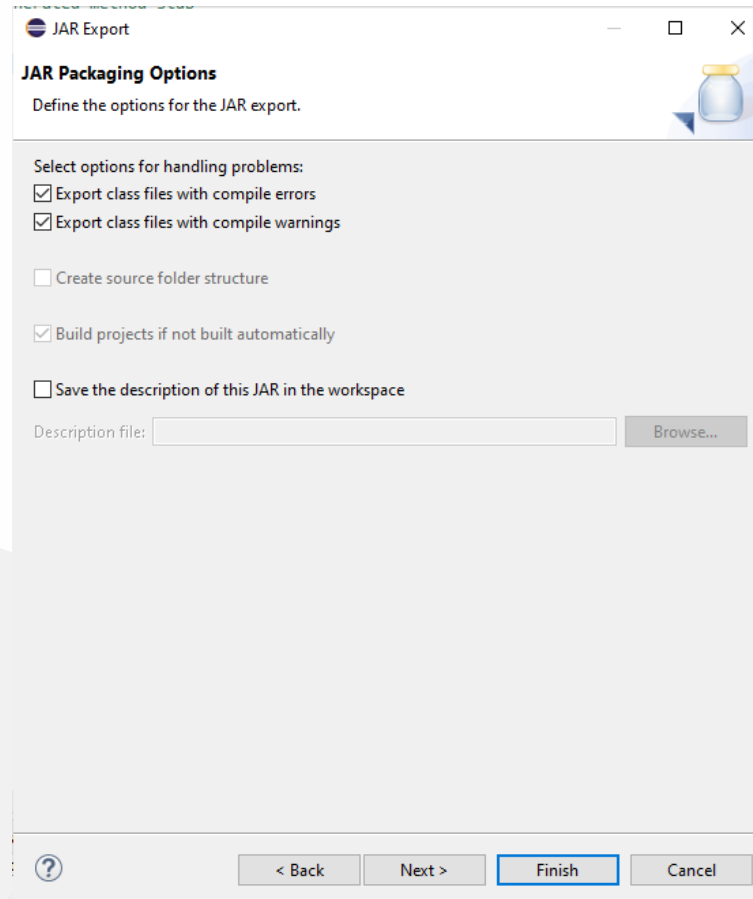
Shared Library Development - (Eclipse Java Jar Library)-72

we configured output as

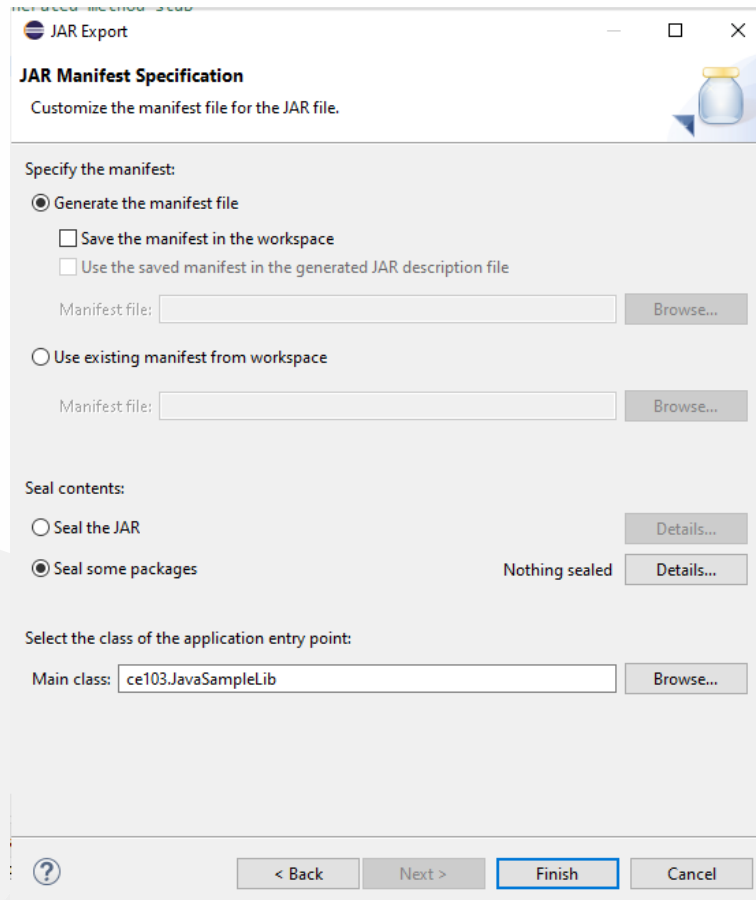
C:\Users\ugur.coruh\Desktop\java-export-sample\JavaSampleLib.jar



Shared Library Development - (Eclipse Java Jar Library)-73

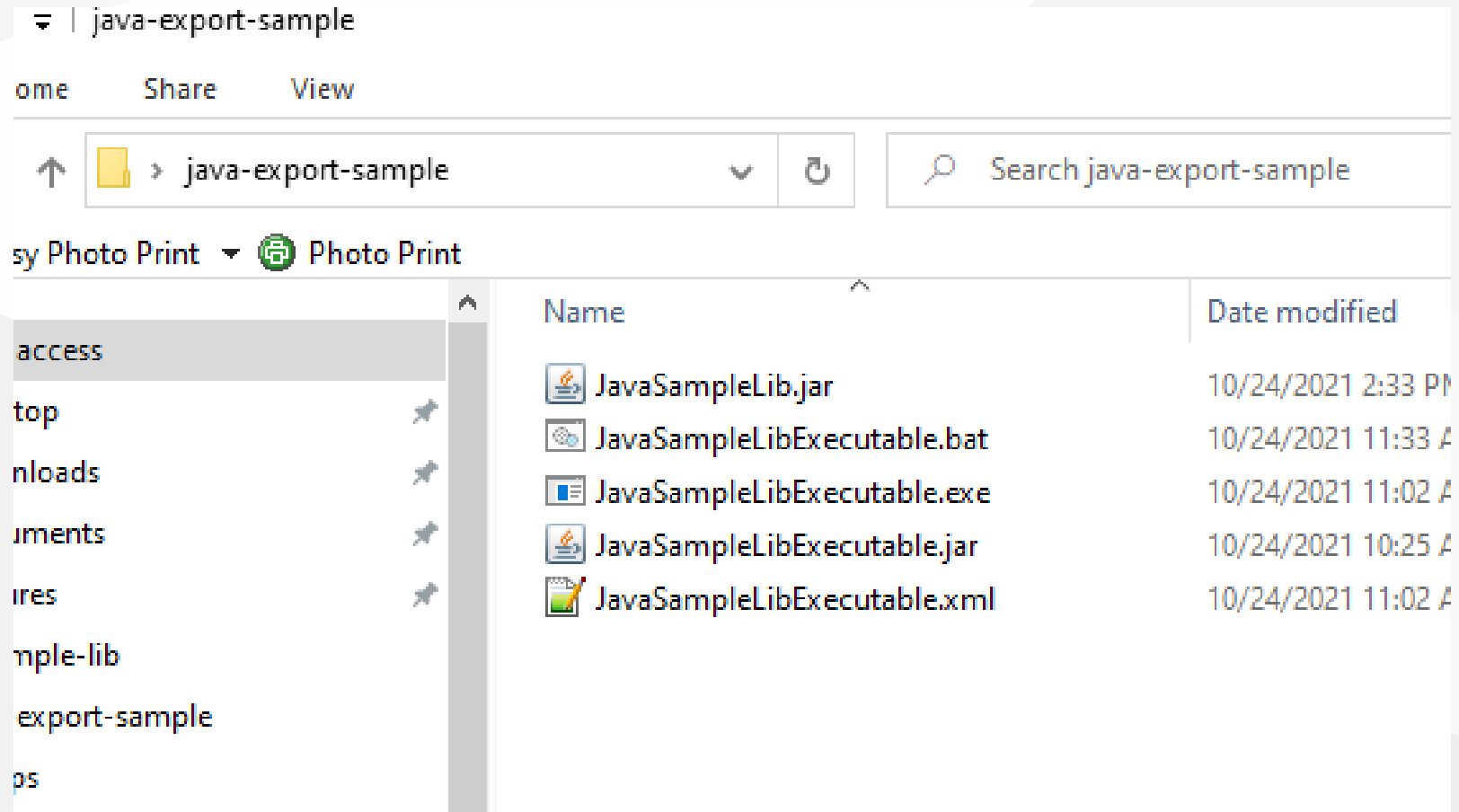


Shared Library Development - (Eclipse Java Jar Library)-74



Shared Library Development - (Eclipse Java Jar Library)-75

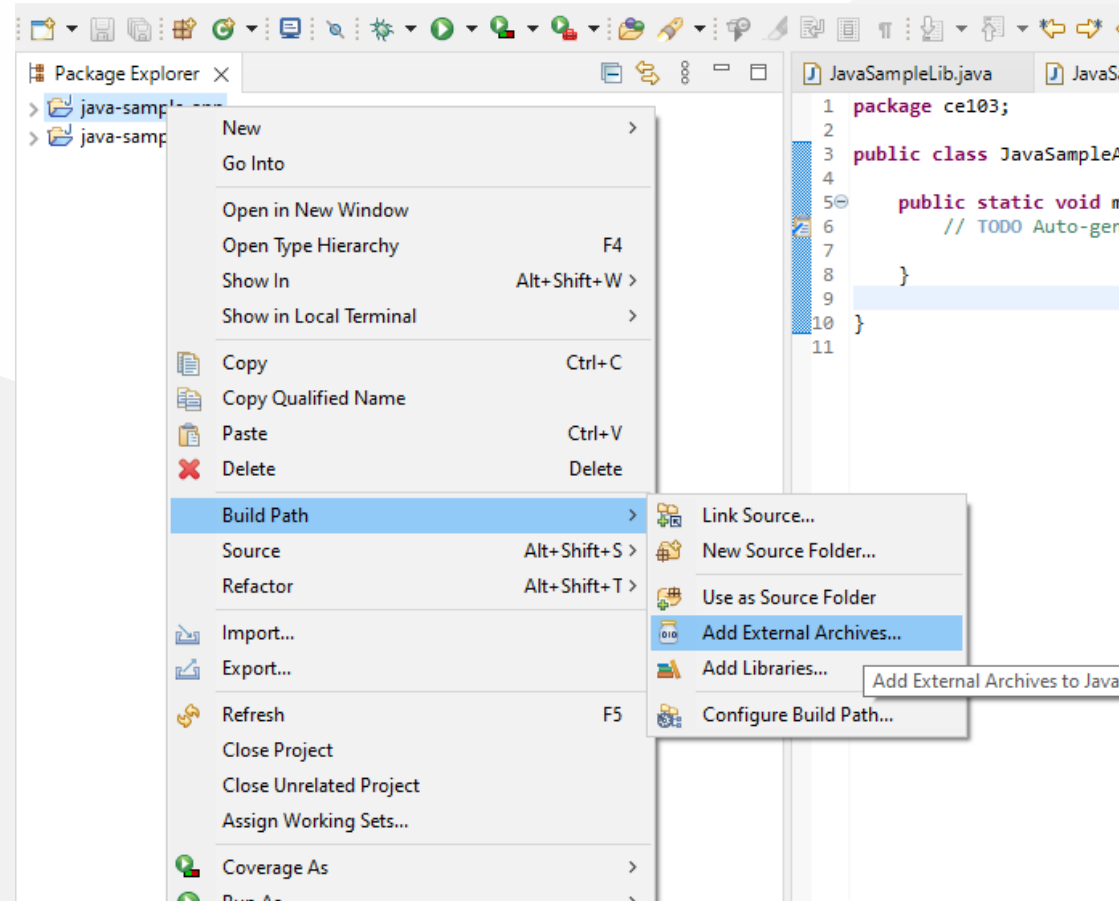
In the same export folder now we have JavaSampleLib.jar



Shared Library Development - (Eclipse Java Jar Library)-76

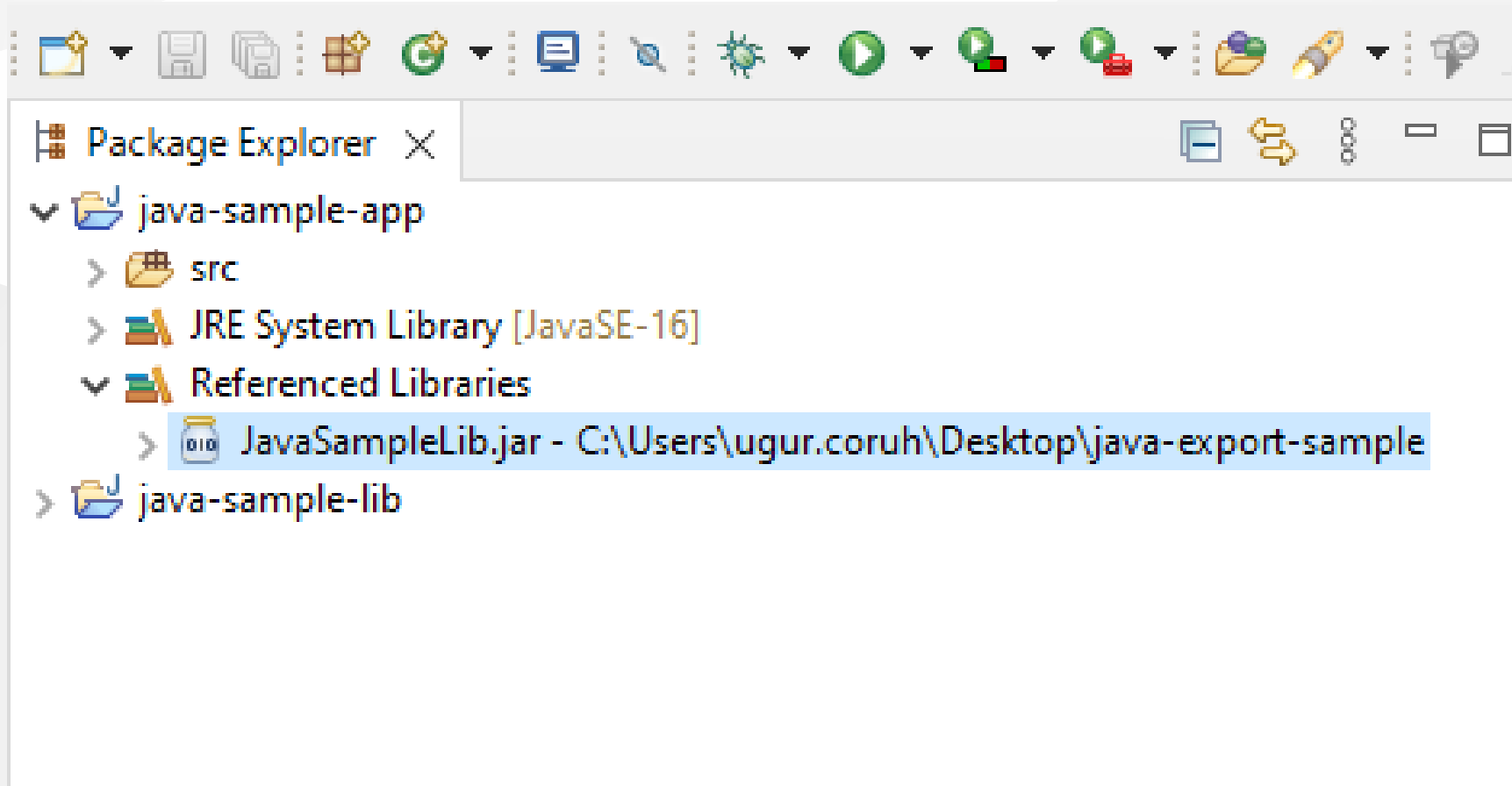
return back to java-sample-app and then add this jar file to our project

Build Path->Add External Archives



Shared Library Development - (Eclipse Java Jar Library)-77

you will see its added to reference libraries



Shared Library Development - (Eclipse Java Jar Library)-78

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!");

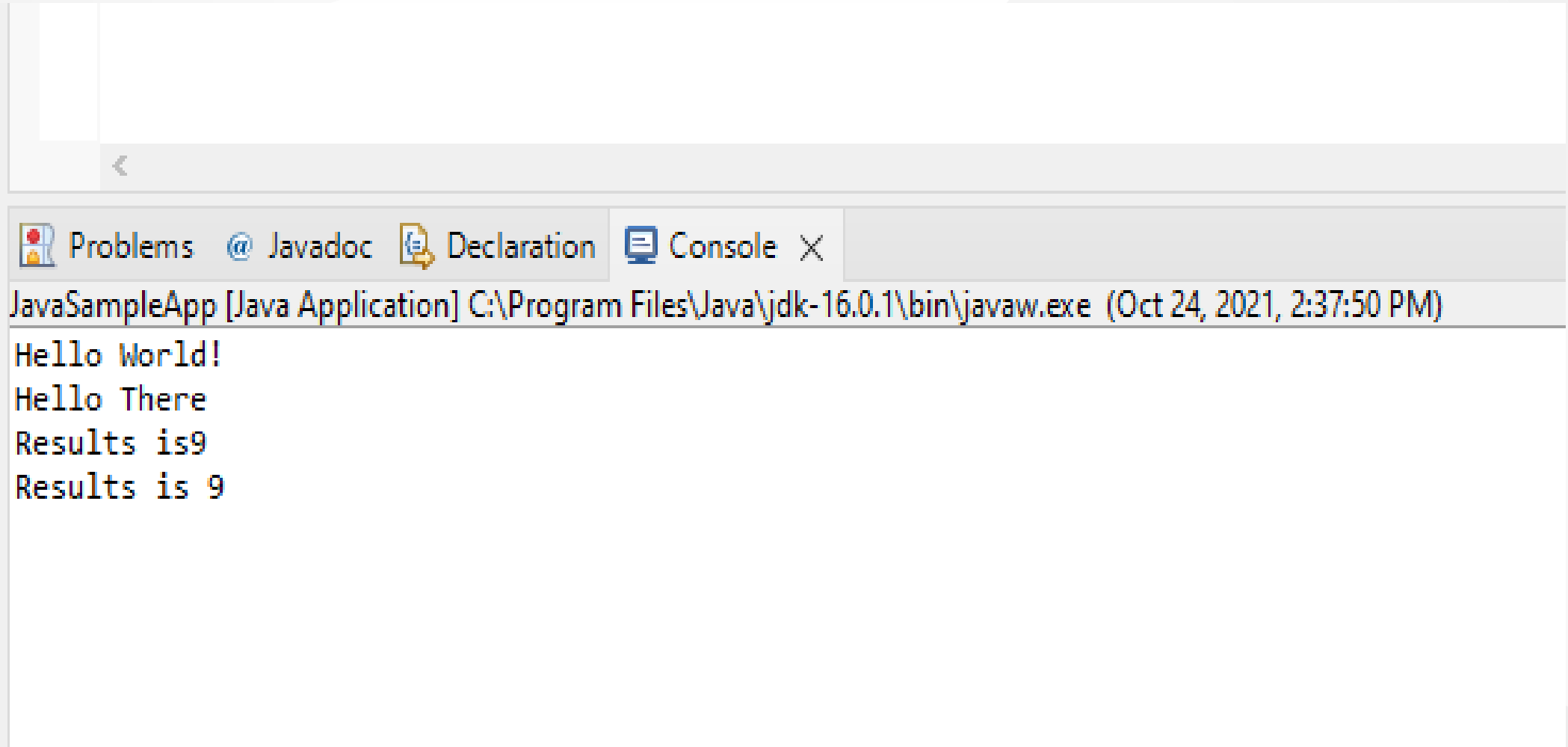
        JavaSampleLib.sayHelloTo("Computer");
        int result = JavaSampleLib.sum(5, 4);
        System.out.println("Results is" + result);
        System.out.printf("Results is %d \n", result);

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

    }
}
```

Shared Library Development - (Eclipse Java Jar Library)-79

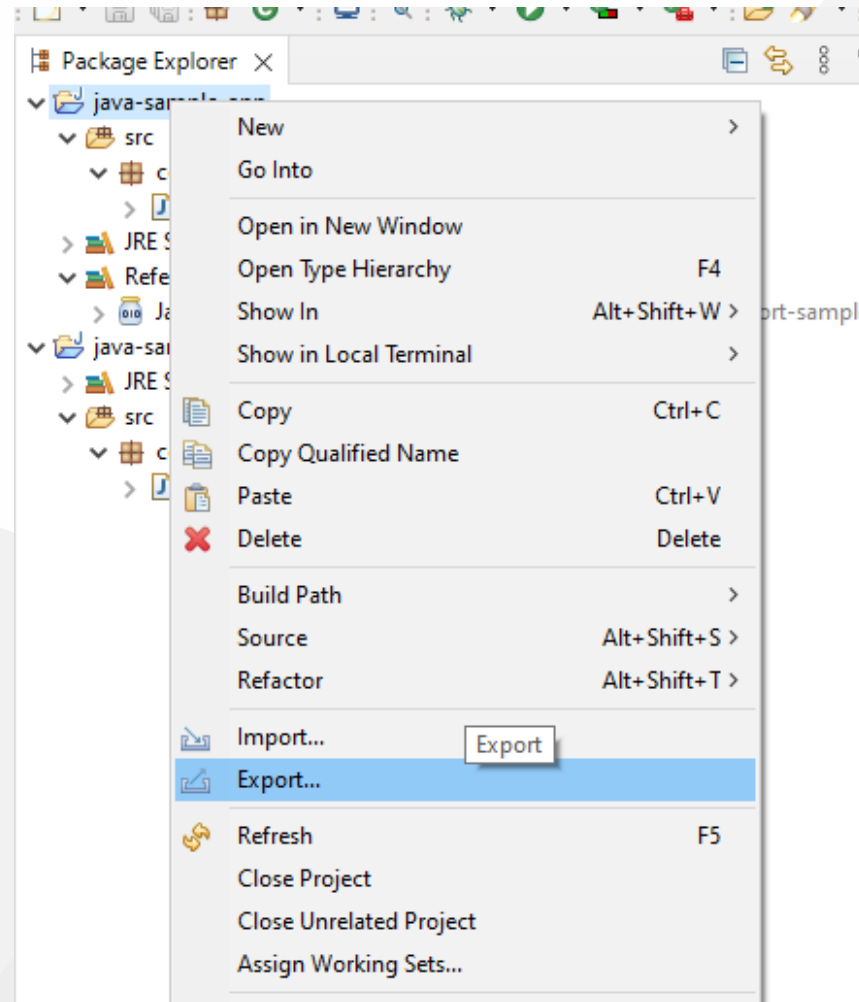
When we run application we will see similar output

A screenshot of the Eclipse IDE's console window. The window title bar shows 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console text reads: 'JavaSampleApp [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (Oct 24, 2021, 2:37:50 PM)' followed by four lines of output: 'Hello World!', 'Hello There', 'Results is9', and 'Results is 9'.

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

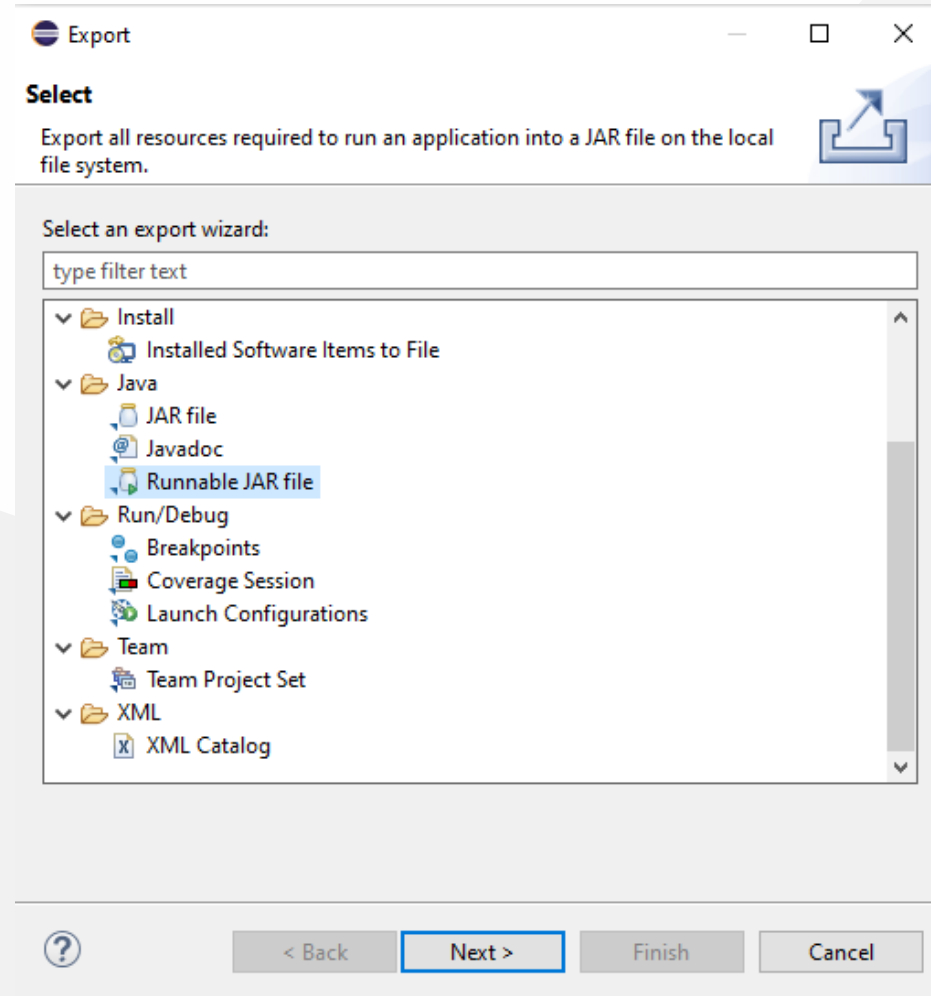
Shared Library Development - (Eclipse Java Jar Library)-80

Lets export this application with its dependent library



Shared Library Development - (Eclipse Java Jar Library)-81

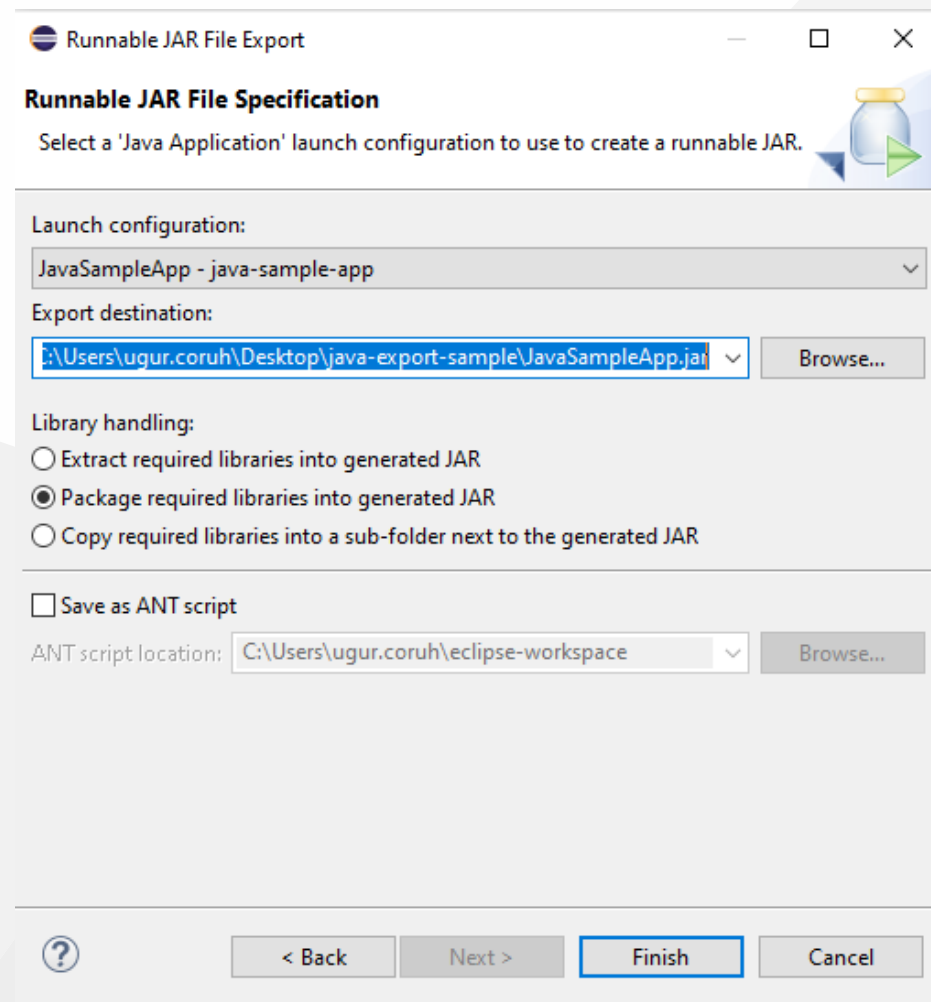
Select runnable jar



Shared Library Development - (Eclipse Java Jar Library)-82

Set Launch configuration and Export destination

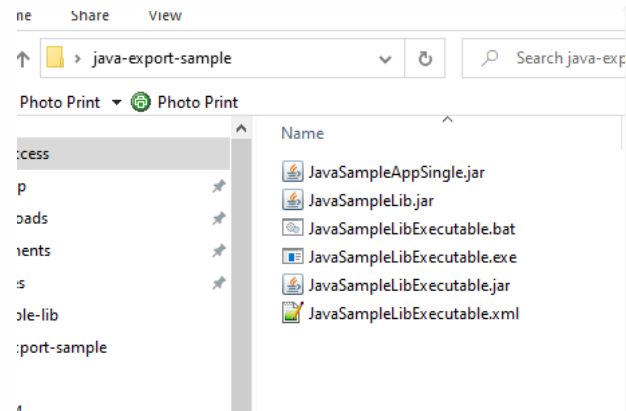
C:\Users\ugur.coruh\Desktop\java-export-sample\JavaSampleAppSingle.jar



Shared Library Development - (Eclipse Java Jar Library)-83

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

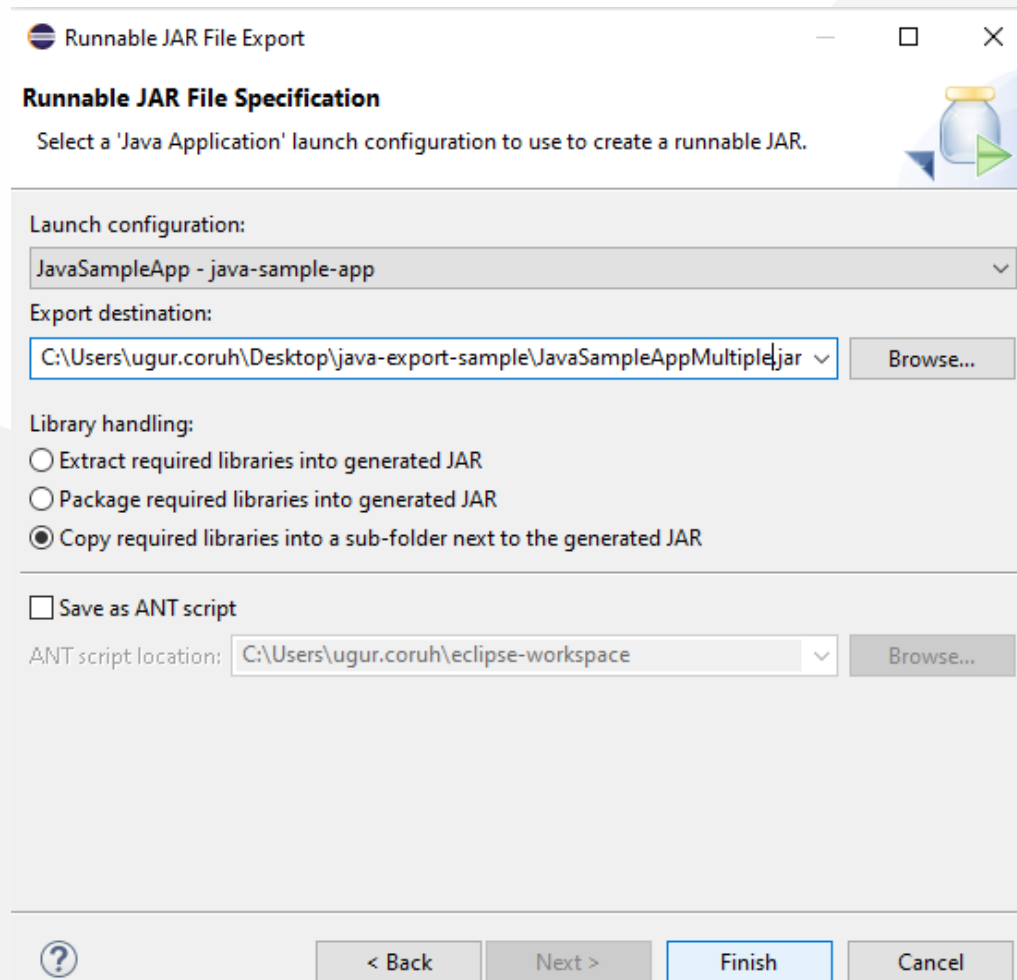
```
C:\Users\ugur.coruh\Desktop\java-export-sample>java -jar JavaSampleAppSingle.jar  
Hello World!  
Hello There  
Results is?  
Results is 9
```

```
24  
25 }  
26  
27 }  
28
```

Shared Library Development - (Eclipse Java Jar Library)-84

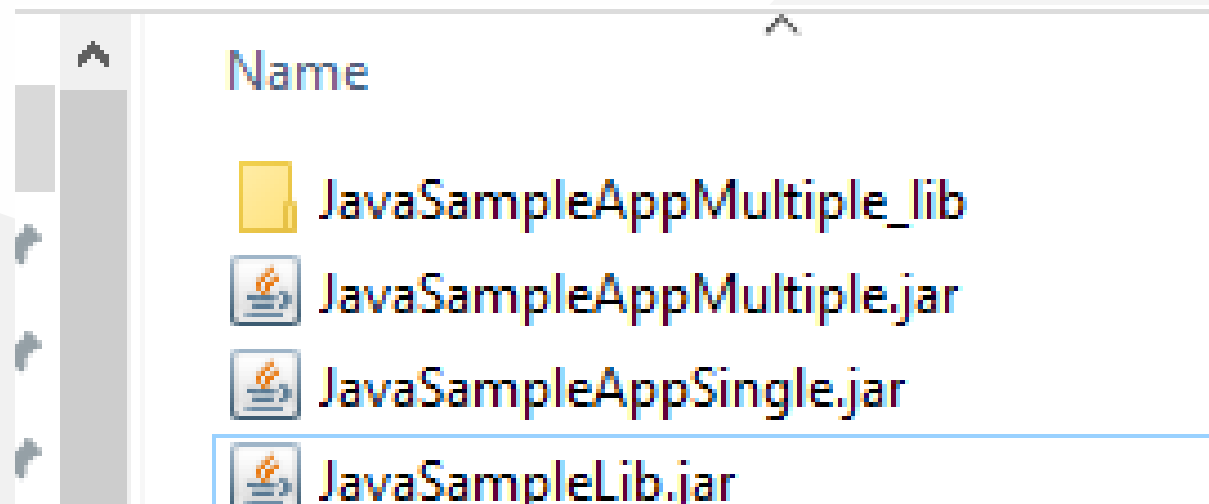
only change copy required libraries setting and then give a new name for new jar file and export

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



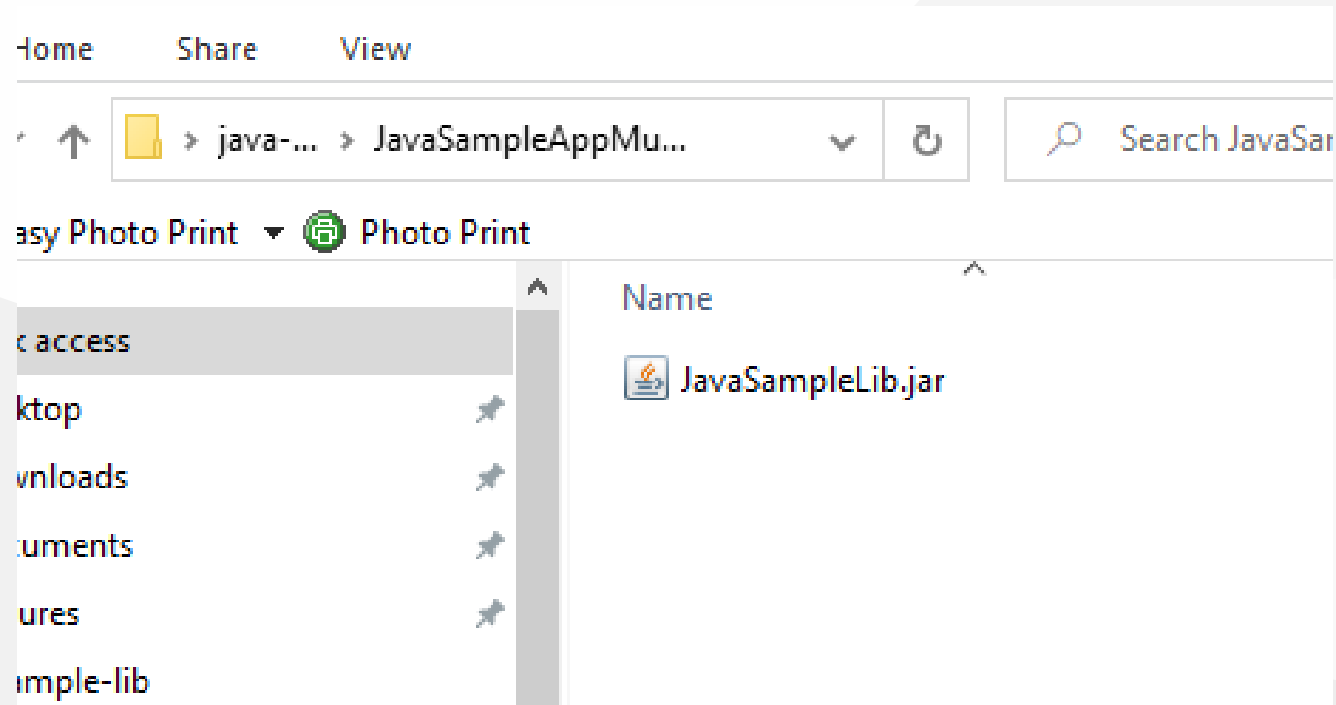
Shared Library Development - (Eclipse Java Jar Library)-85

now we have a folder that contains our libraries referenced



Shared Library Development - (Eclipse Java Jar Library)-86

in this file we can find our library



Shared Library Development - (Eclipse Java Jar Library)-87

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 is9
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>
```

Application Testing

- C
- C++
- C#
- Java

Unit Test Development

Wikipedia Unit Test Library List for Each Language

https://en.wikipedia.org/wiki/List_of_unit_testing_frameworks

Visual Studio Community Edition

C Unit Tests

Visual Studio Community Edition - C Unit Tests

- There is no direct C source testing but with additional frameworks. Visual Studio can test C sources.
- You can check the following entry
 - <https://stackoverflow.com/questions/65820/unit-testing-c-code>
- Recommended framework is Check
 - <https://libcheck.github.io/check/web/install.html>
 - <https://github.com/libcheck/check/releases>

Visual Studio Community Edition

C++ Unit Tests

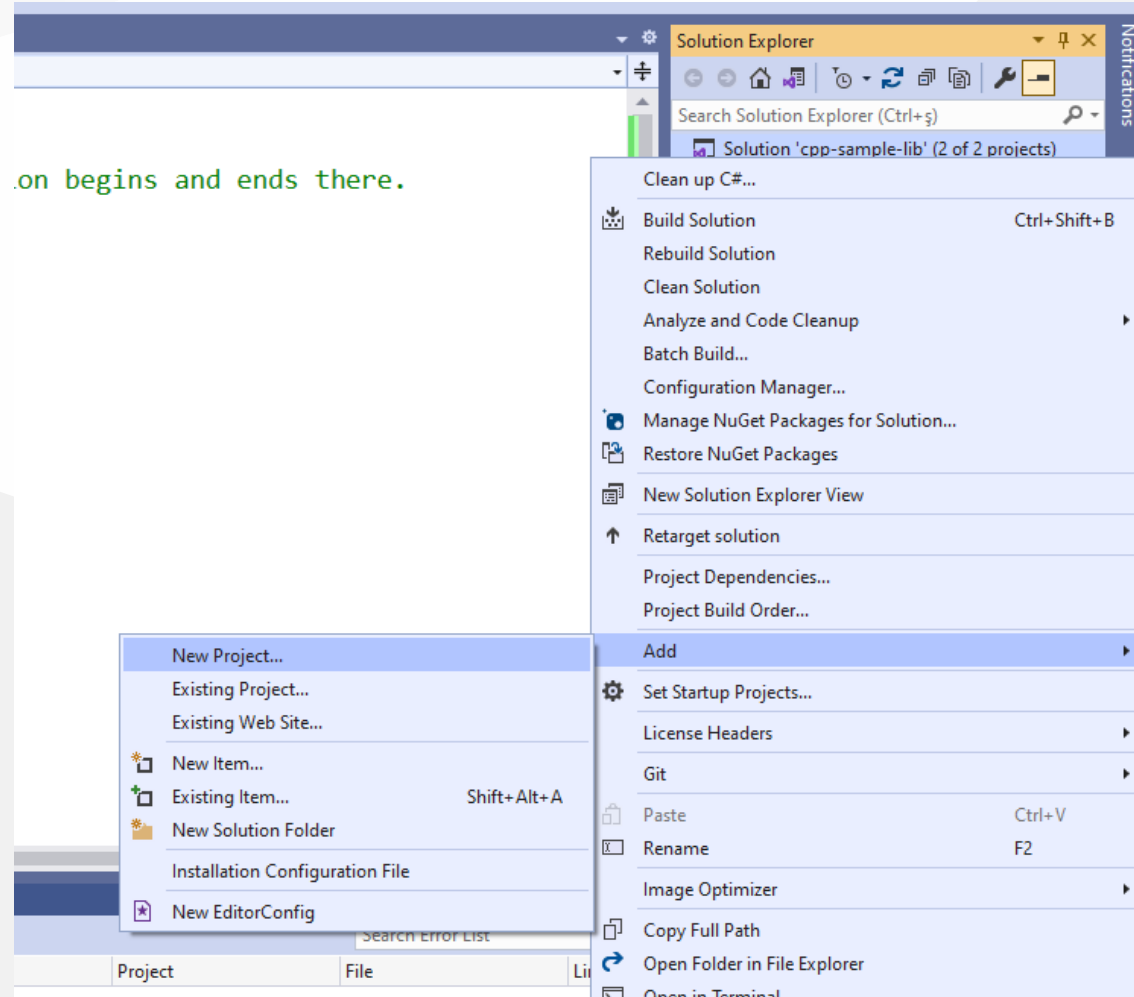
Visual Studio Community Edition - C++ Unit Tests-1

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

Visual Studio Community Edition - C++ Unit Tests-2

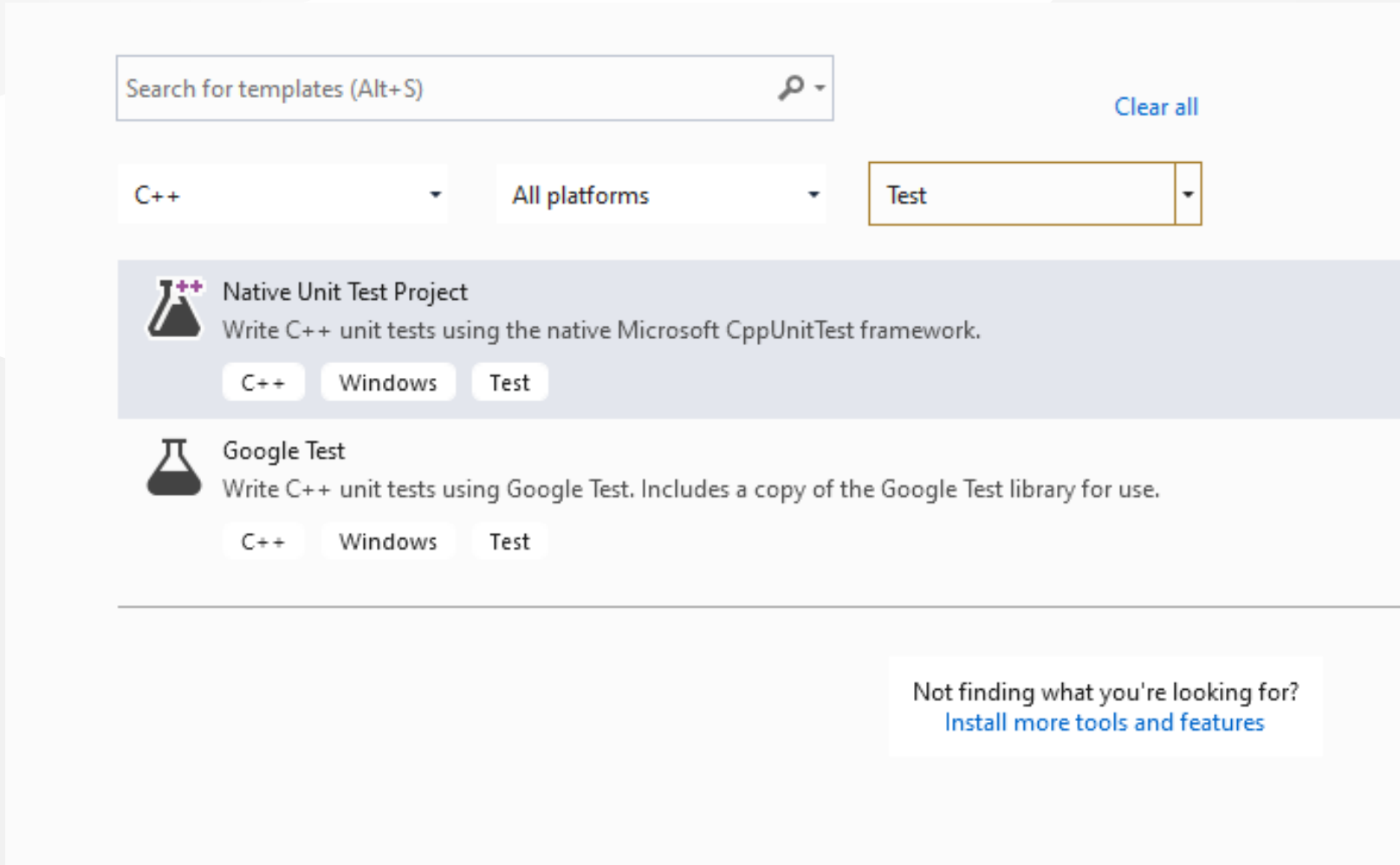
- Use cpp-sample-lib project and add

.on begins and ends there.



Visual Studio Community Edition - C++ Unit Tests-3

- Select Native Unit Test



Visual Studio Community Edition - C++ Unit Tests-4

- Set project path and name

Configure your new project

Native Unit Test Project C++ Windows Test

Project name

cpp-sample-test

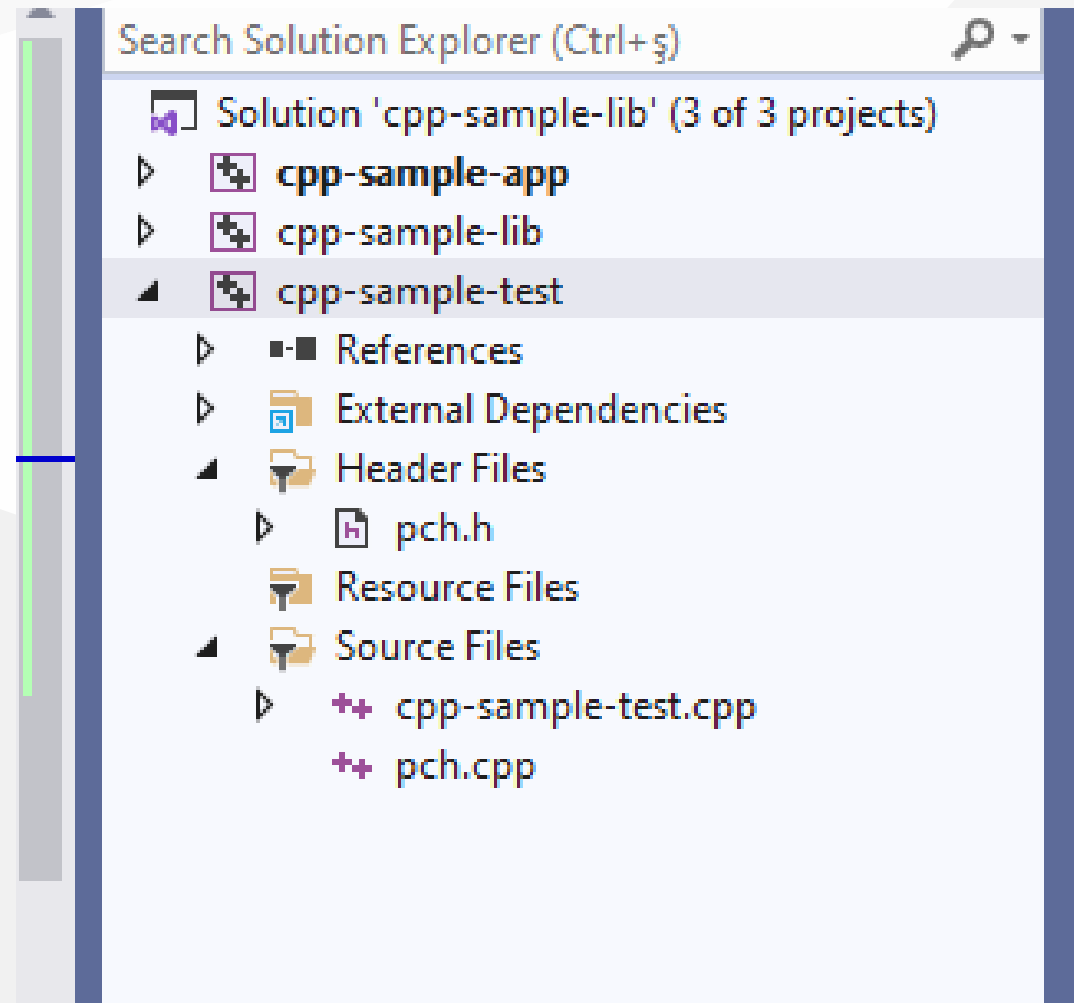
Location

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

...

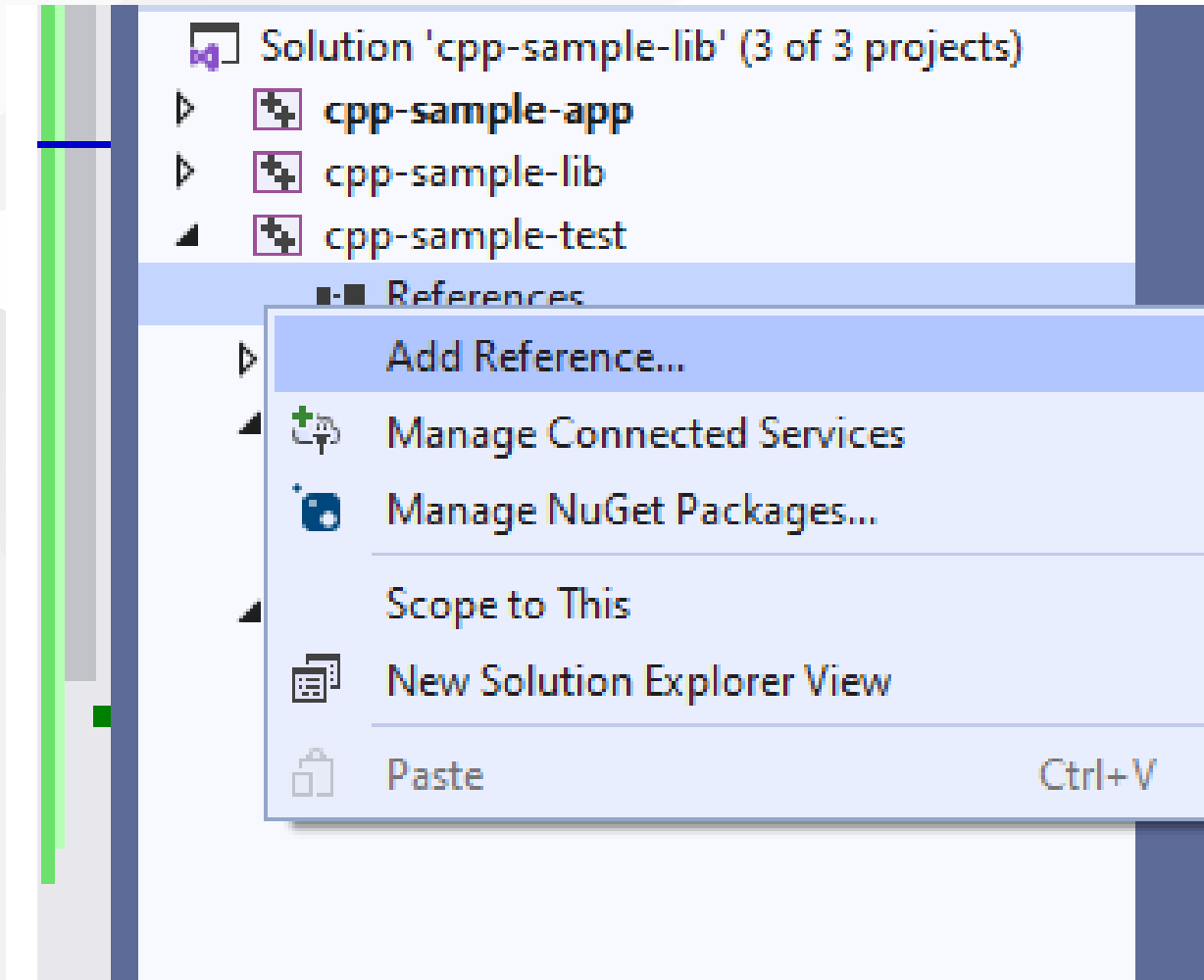
Visual Studio Community Edition - C++ Unit Tests-5

- You will have `cpp-sample-test` project



Visual Studio Community Edition - C++ Unit Tests-6

- Add library project from references



Visual Studio Community Edition - C++ Unit Tests-7

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

Add Reference

Projects

Solution

Shared Projects

	Name
	cpp-sample-app
<input checked="" type="checkbox"/>	cpp-sample-lib



Visual Studio Community Edition - C++ Unit Tests-8

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));
        }

    };
}
```

Visual Studio Community Edition - C++ Unit Tests-9

```
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
```

Test Explorer

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>

Visual Studio Community Edition

C# Unit Tests

- MSTest + .Net
- Fine Code Coverage
- NUnit + .NetCore

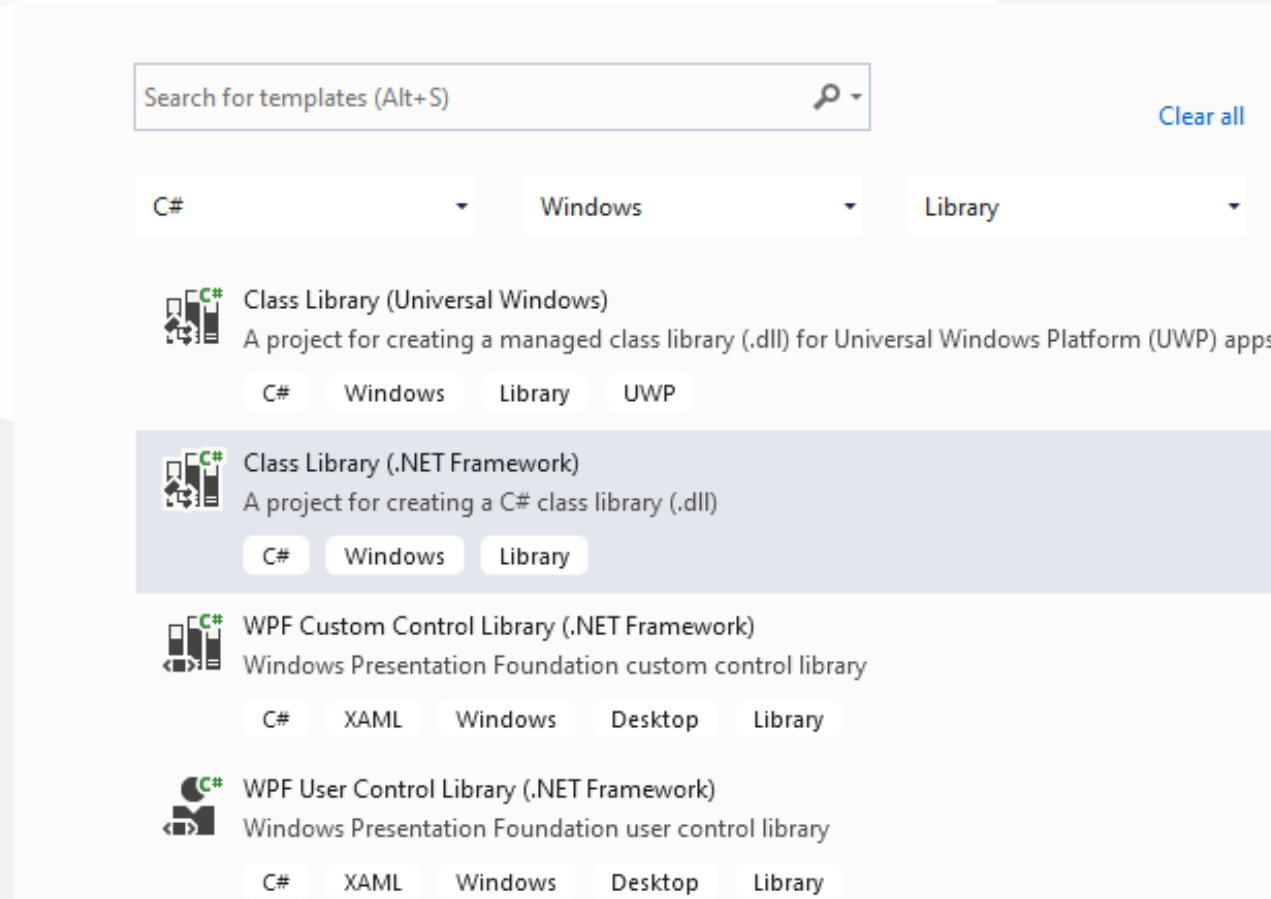
Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-1

- Install extension fine code coverage

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

Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-2

- Create a .Net Framework Library



Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-3

- Set project framework and path

Configure your new project

Class Library (.NET Framework) C# Windows Library

Project name

Location

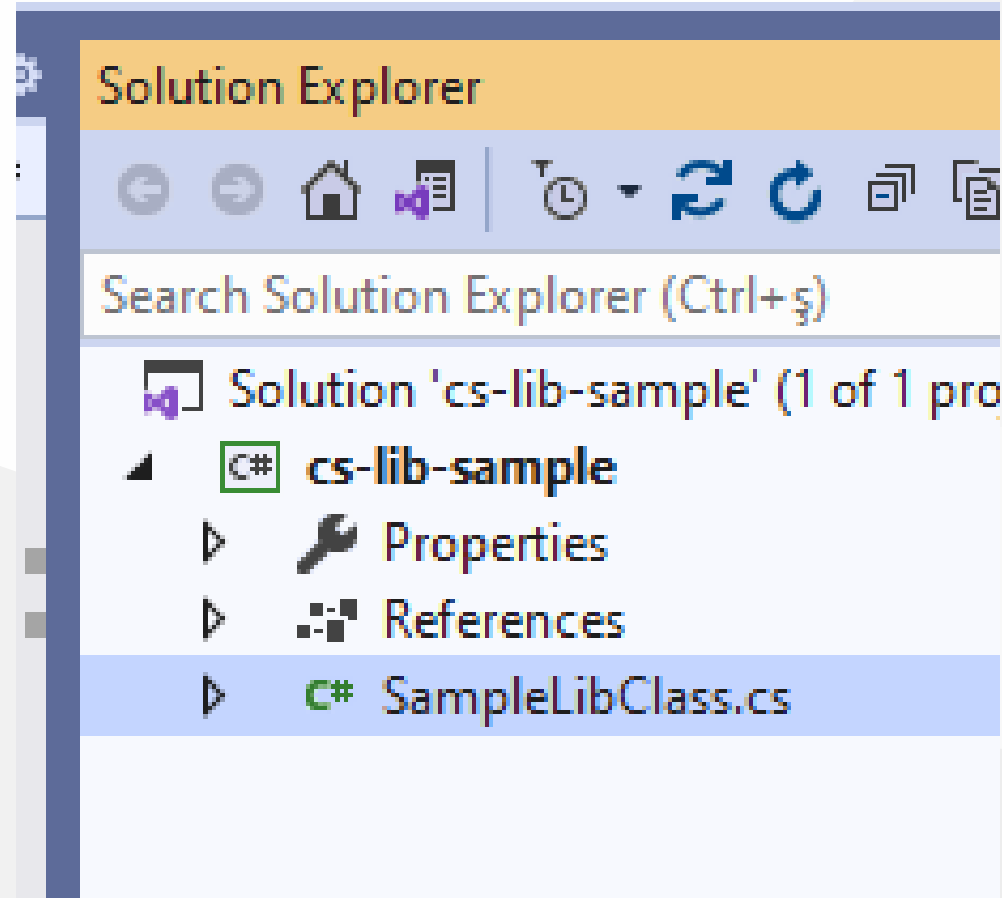
Solution name ⓘ

Place solution and project in the same directory

Framework

Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-4

- Create library functions



Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-5

```
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);

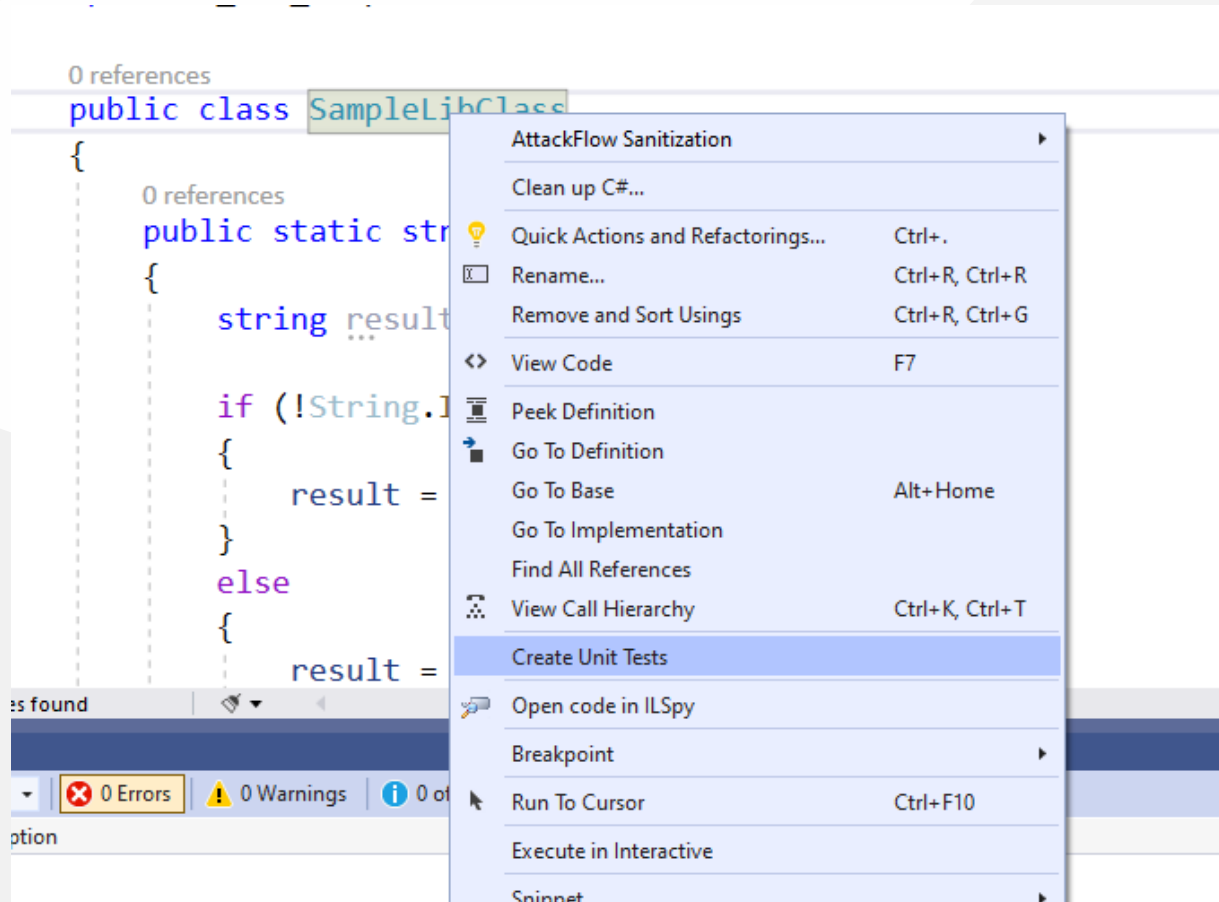
            return result;
        }

        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;
        }
    }
}
```

Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-6

- Right click and then create unit test project



Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-7

- Press OK

The screenshot shows the 'Create Unit Tests' dialog box in Visual Studio. The dialog has a title bar with a question mark and a close button. The main area contains several settings:

- Test Framework:** A dropdown menu set to 'MSTestv2'. To the right of this dropdown is a blue link that says 'Get Additional Extensions'.
- Test Project:** A dropdown menu set to '<New Test Project>'.
- Name Format for Test Project:** A text input field containing '[Project]Tests'.
- Namespace:** A text input field containing '[Namespace].Tests'.
- Output File:** A dropdown menu set to '<New Test File>'.
- Name Format for Test Class:** A text input field containing '[Class]Tests'.
- Name Format for Test Method:** A text input field containing '[Method]Test'.
- Code for Test Method:** A dropdown menu set to 'Assert failure'.

At the bottom right of the dialog, there are two buttons: 'OK' and 'Cancel'.

Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-8

- 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");
        }

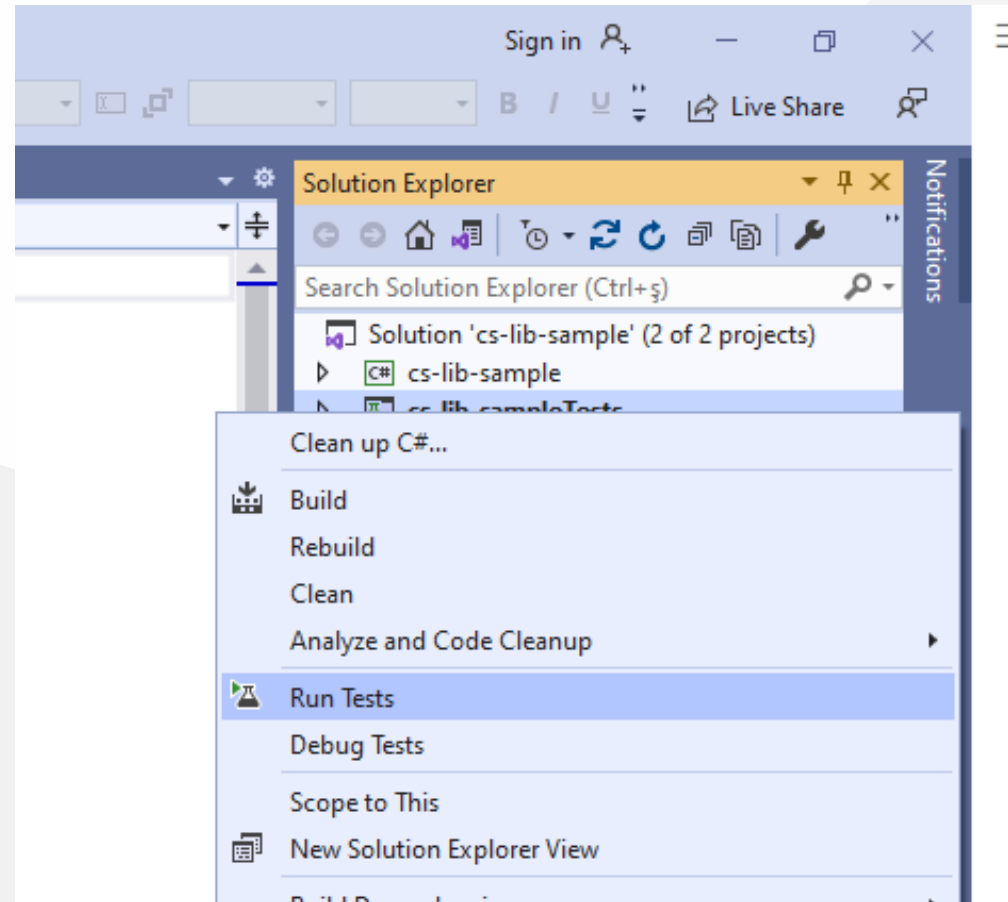
        [TestMethod()]
        public void testSumWrong()
        {
            Assert.AreEqual(10, SampleLibClass.sum(4, 5), "Regular sum shouldn't work");
        }

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

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

Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-9

- Run tests



Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-10

you will code coverage and entered or passed branches

```

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

144% No issues found 2 references | 1/2 passing Ln: 18 Ch: 14 SPC CRLF

Fine Code Coverage

Coverage Summary Risk Hotspots Rate & Review Log Issue/Suggestion Buy me a coffee

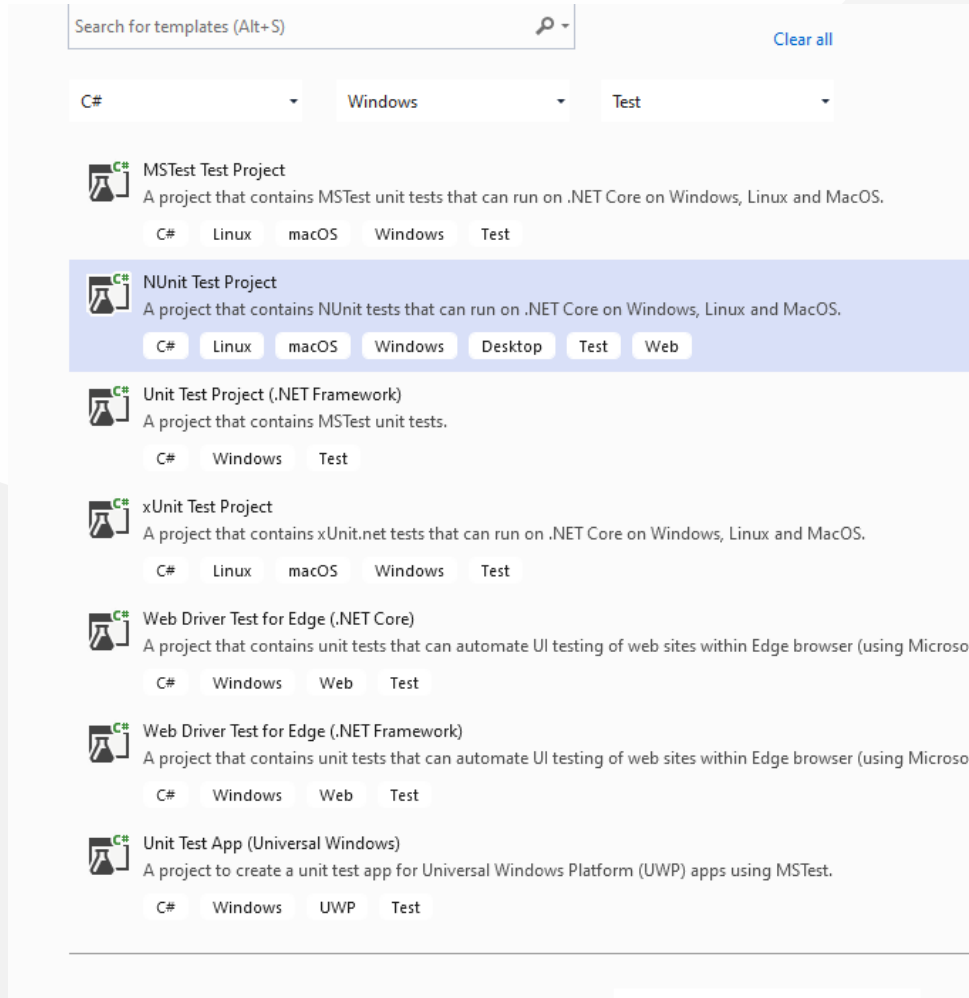
Name	Covered	Uncovered	Coverable	Total	Line coverage
- cs-lib-sample	17	3	20	39	85%
SampleLibClass	17	3	20	39	85%
- cs-lib-sampleTests	14	2	16	51	87.5%
SampleLibClassTests	14	2	16	51	87.5%

Visual Studio Community Edition

C# Unit Test + NUnit + .NETCore

Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-1

- Use `csharp-sample-lib` for this example
- Create and add a unit test project to solution



Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-2

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\ce10

...

Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-3

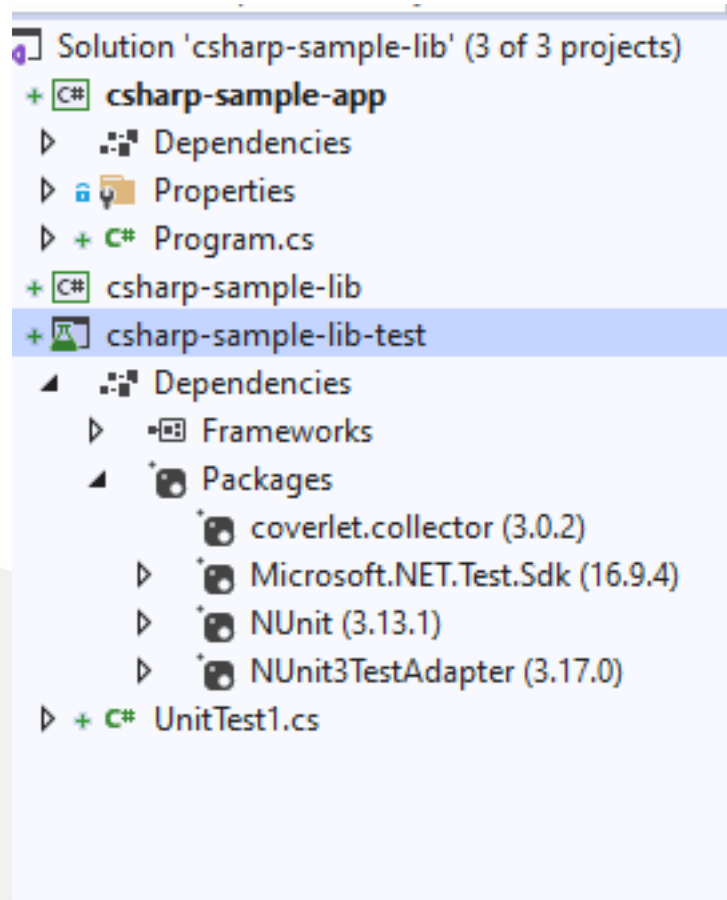
Additional information

NUnit Test Project C# Linux macOS Windows Desktop Test Web

Target Framework ⓘ

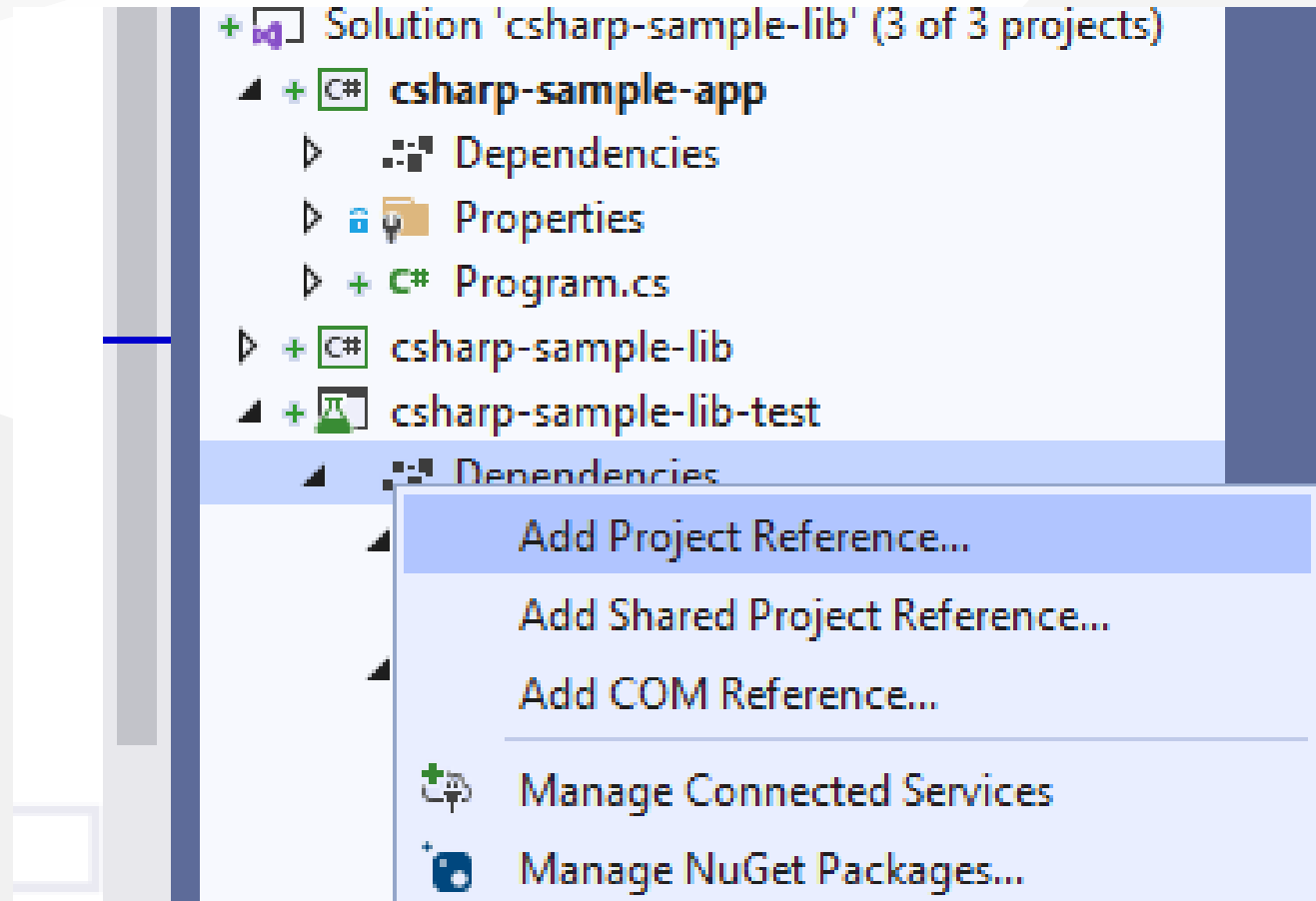
- .NET Core 3.1 (Long-term support)
- .NET Framework 4.0
- .NET Framework 4.5
- .NET Framework 4.5.1
- .NET Framework 4.5.2
- .NET Framework 4.6
- .NET Framework 4.6.1
- .NET Framework 4.6.2
- .NET Framework 4.7
- .NET Framework 4.7.1
- .NET Framework 4.7.2
- .NET Framework 4.8
- .NET Core 1.0 (Out of 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)

Visual Studio Community Edition (C# Unit Test+JUnit+.NETCore)-4



Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-5

- Add project reference



Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-6

Reference Manager - csharp-sample-lib-test

Projects

Solution

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

Shared Projects

COM

Browse

Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-7

SampleLibraryTestClass 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");
        }
    }
}
```

Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-8

- 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);

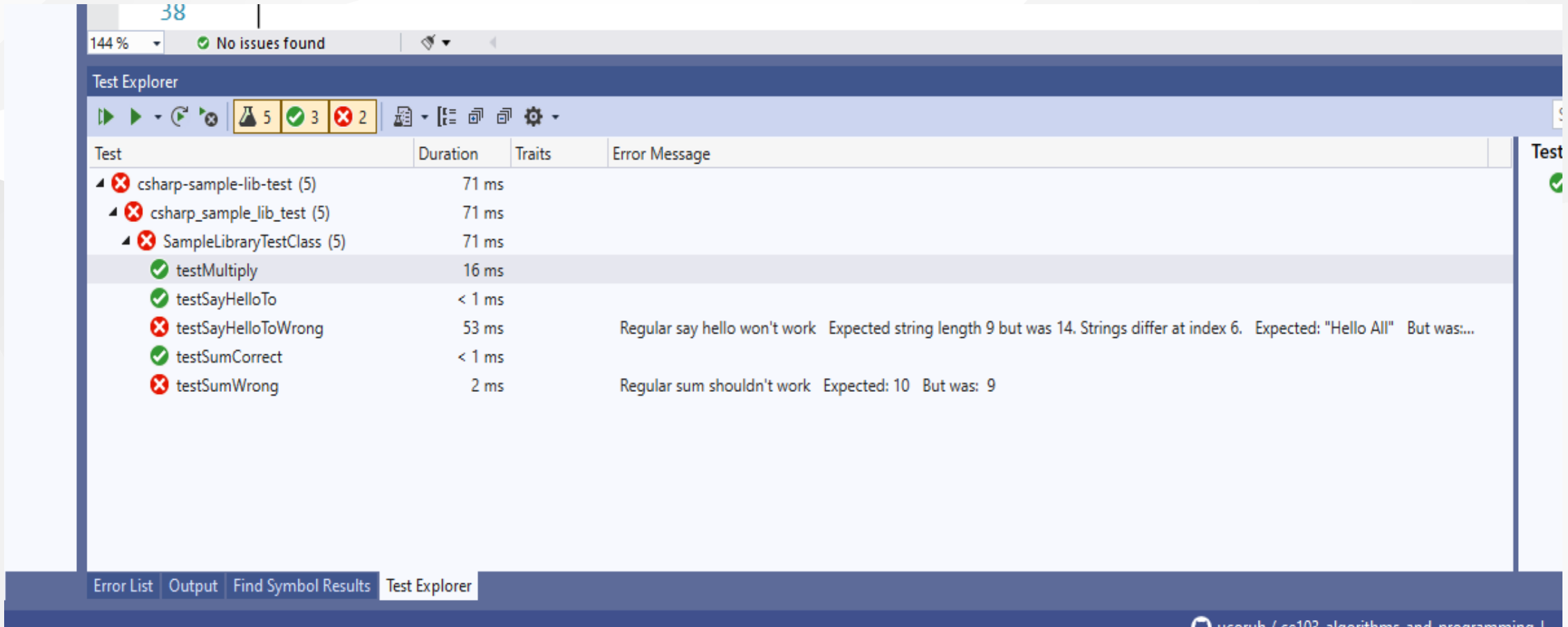
            return result;
        }

        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;
        }
    }
}
```

Visual Studio Community Edition (C# Unit Test+ NUnit+.NETCore)-9

- Open test explorer and run tests



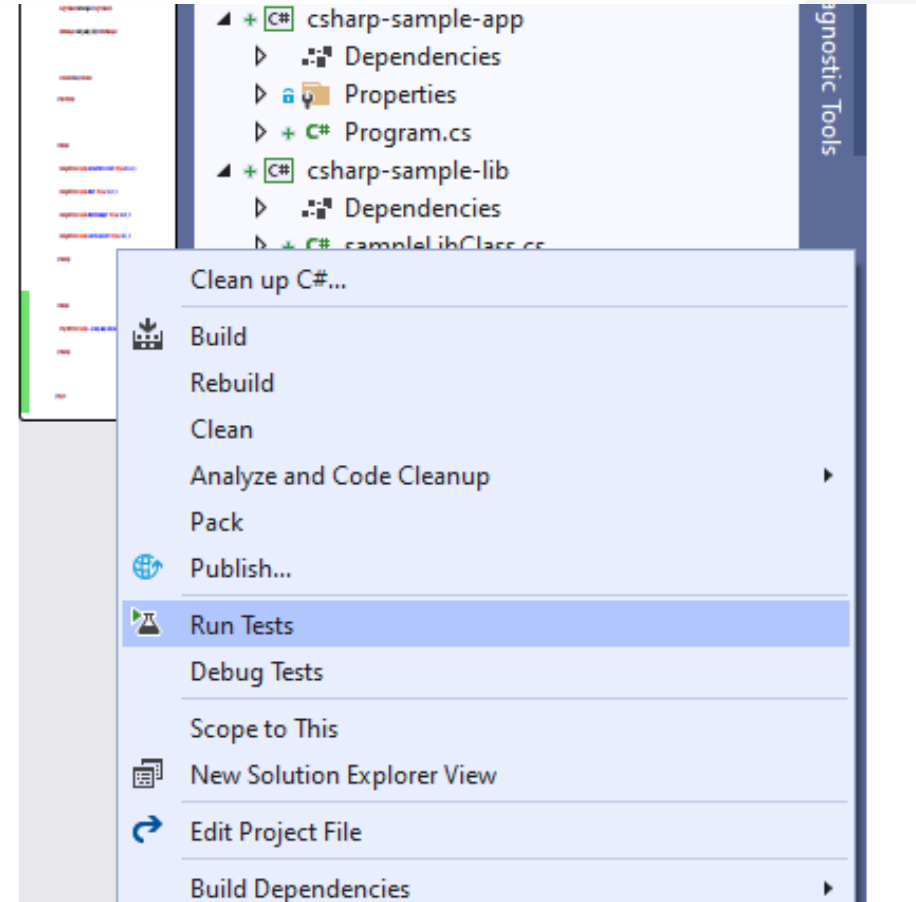
The screenshot shows the Visual Studio Test Explorer window. At the top, there are icons for running tests: a play button, a refresh button, a stop button, and a summary bar showing 5 failed tests (flask icon), 3 passed tests (checkmark icon), and 2 failed tests (X icon). Below the icons is a table with columns for Test, Duration, Traits, and Error Message.

Test	Duration	Traits	Error Message
✘ csharp-sample-lib-test (5)	71 ms		
✘ csharp_sample_lib_test (5)	71 ms		
✘ SampleLibraryTestClass (5)	71 ms		
✔ testMultiply	16 ms		
✔ testSayHelloTo	< 1 ms		
✘ testSayHelloToWrong	53 ms		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	2 ms		Regular sum shouldn't work Expected: 10 But was: 9

At the bottom of the window, there are tabs for Error List, Output, Find Symbol Results, and Test Explorer. The Test Explorer tab is currently active.

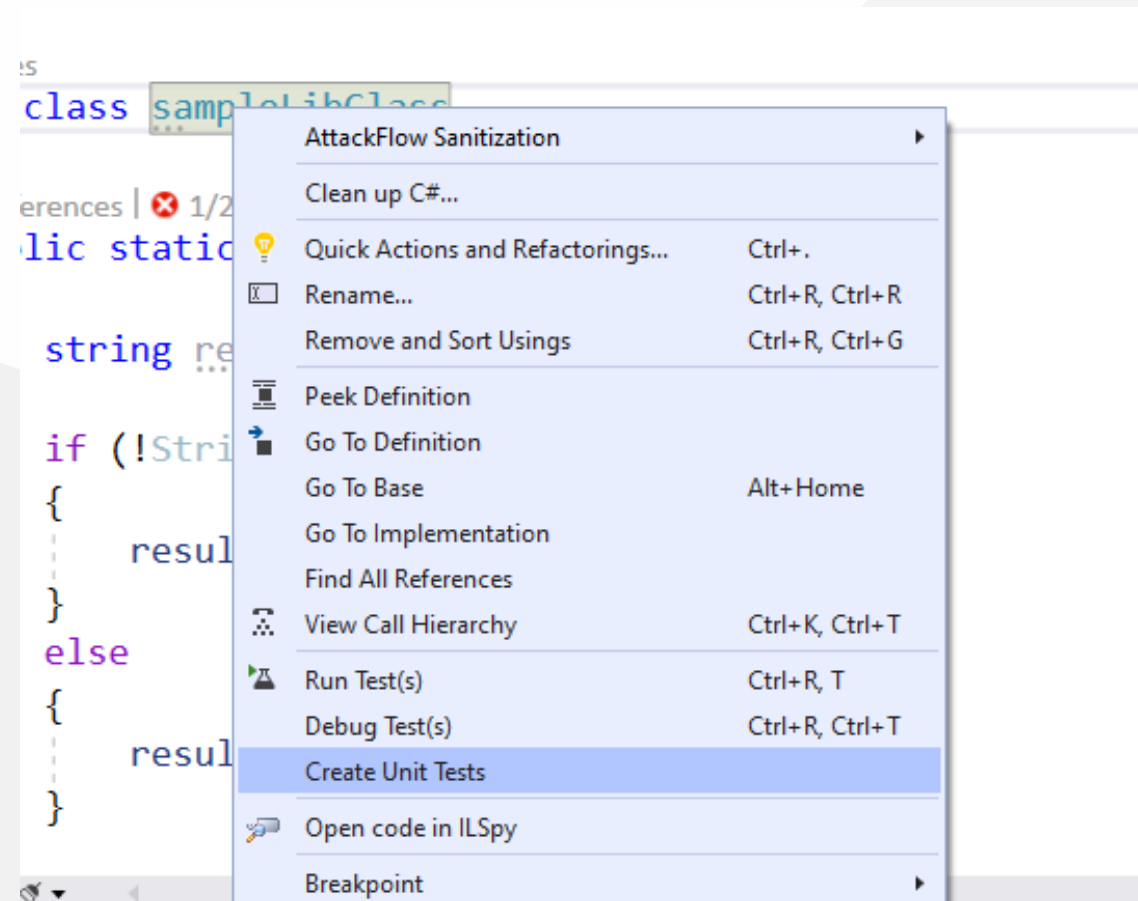
Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-10

- or you can run from project



Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-11

- 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.



Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-12

The screenshot shows the 'Create Unit Tests' dialog box in Visual Studio. The dialog has a title bar with a question mark and a close button. The main area contains several fields for configuring the test project:

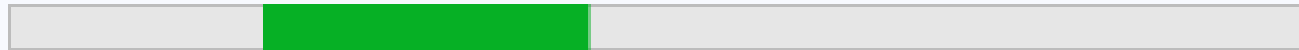
- Test Framework:** A dropdown menu set to 'MSTestv2'. A blue link '[Get Additional Extensions](#)' is visible to the right.
- Test Project:** A dropdown menu set to '<New Test Project>'. A blue link '[Get Additional Extensions](#)' is visible to the right.
- Name Format for Test Project:** A text input field containing '[Project]Tests'.
- Namespace:** A text input field containing '[Namespace].Tests'.
- Output File:** A dropdown menu set to '<New Test File>'.
- Name Format for Test Class:** A text input field containing '[Class]Tests'.
- Name Format for Test Method:** A text input field containing '[Method]Test'.
- Code for Test Method:** A dropdown menu set to 'Assert failure'.

At the bottom right of the dialog, there are two buttons: 'OK' and 'Cancel'.

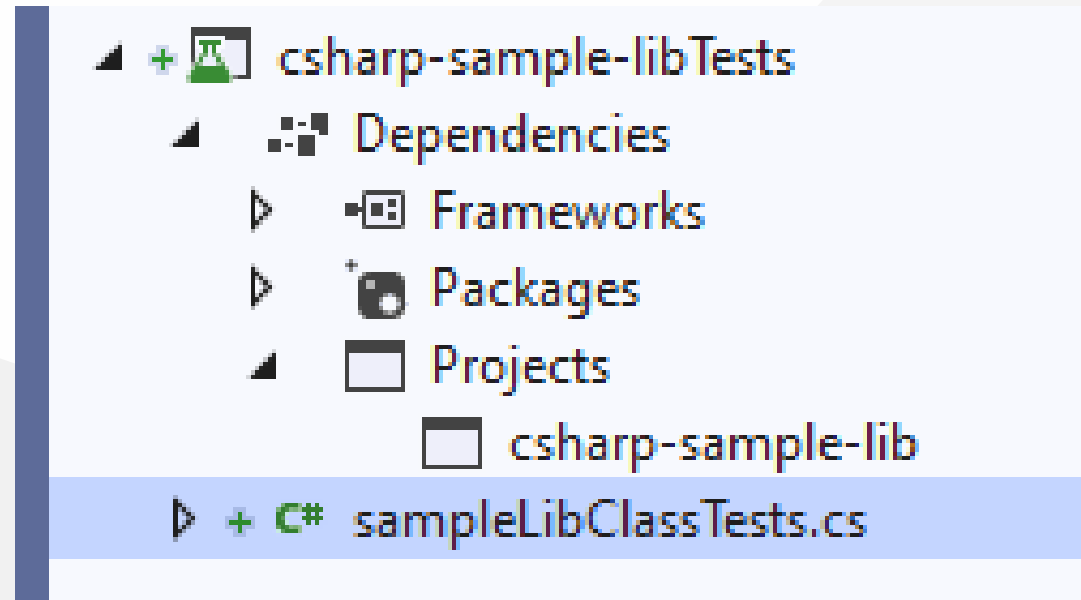
Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-13

Create Unit Tests

Creating Unit Tests



Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-14



Visual Studio Community Edition (C# Unit Test+ NUnit+.NETCore)-15

```
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();
        }
    }
}
```

Visual Studio Community Edition (C# Unit Test+ NUnit+.NETCore)-16

- 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 with the following components:

- Code Editor:** Displays the source code for `sampleLibClass.cs`. The code includes a namespace `csharp_sample_lib` and a public class `sampleLibClass` with a static method `sayHelloTo`. The method logic is as follows:


```

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";
            }
        }
    }
}

```
- Solution Explorer:** Shows the project structure with folders for `csharp-sample-lib` and `csharp-sample-lib-test`, and files `sampleLibClass.cs` and `SampleLibraryTestClass.cs`.
- Fine Code Coverage:** A table summarizing the coverage results for the code and tests.

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%

Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-17

The screenshot displays the Visual Studio Test Explorer interface. The top status bar shows '144 %' zoom and 'No issues found'. The Test Explorer toolbar includes icons for running tests, a summary of 5 tests (3 passed, 2 failed), and a search bar. The main area shows a tree view of test results:

Test	Duration	Traits	Error Message
✗ csharp-sample-lib-test (5)	197 ms		
✗ csharp_sample_lib_test (5)	197 ms		
✗ SampleLibraryTestClass (5)	197 ms		
✓ testMultiply	54 ms		
✓ testSayHelloTo	< 1 ms		
✗ testSayHelloToWrong	136 ms		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		Regular sum shouldn't work Expected: 10 But was: 9

On the right, the 'Group Summary' panel shows:

- Group: csharp-sample-lib-test
- Tests in group: 5
- Total Duration: 197 ms
- Outcomes: 3 Passed, 2 Failed

The bottom status bar includes tabs for 'Fine Code Coverage', 'Error List', 'Output', 'Find Symbol Results', and 'Test Explorer'.

- Inline code highlight is part of enterprise visual studio edition
 - [Analyzing code coverage in Visual Studio - DEV Community](#)

Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)

Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-1

TL;DR

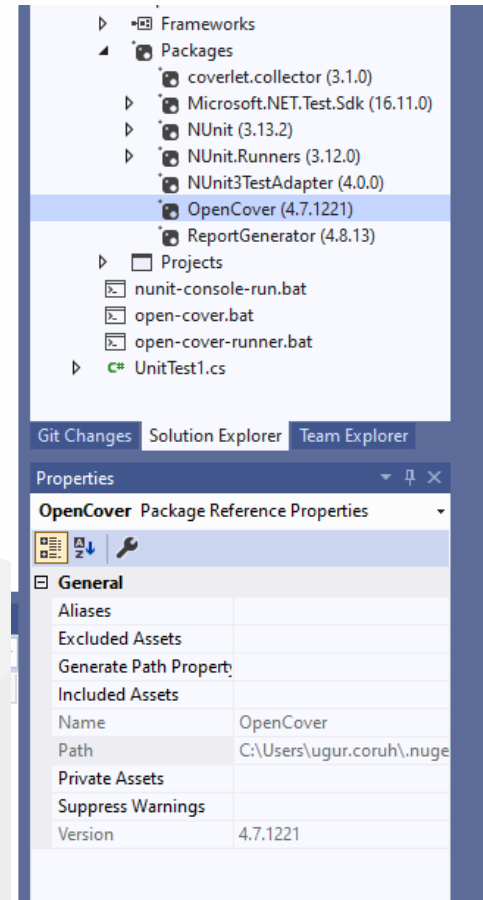
- Additional information you can use OpenCover + Nunit Runner + Report Generator together to setup a code coverage report but it has complex batch running process. After a few try I decided to use fine code coverage but here is the usage not tested well.
- First unit test runner tool doesn't support .Net Core

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

Visual Studio Community Edition (C# Unit Test+OpenCover + NUnit Runner + Report)-2

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



Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-3

- 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.console.runner\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
```


Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-5

- 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](#)

Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-6

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,

Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-7

OpenCover

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

OpenCover (Release) 4.7.1221 Latest

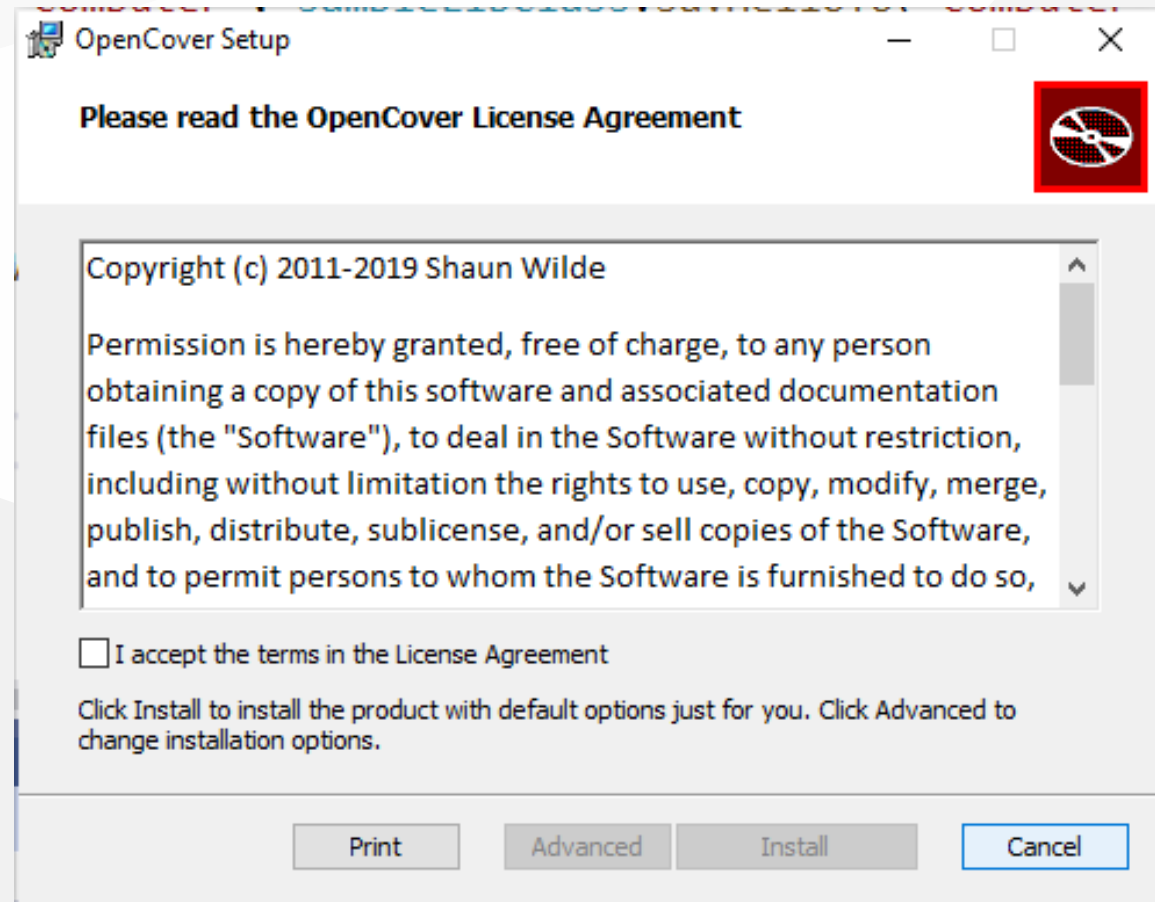
Merge pull request #1040 from sawilde/big/1029_chocolatey_dependency

add dependency to dotnet 4.7.2 to chocolatey packages

▼ **Assets** 6

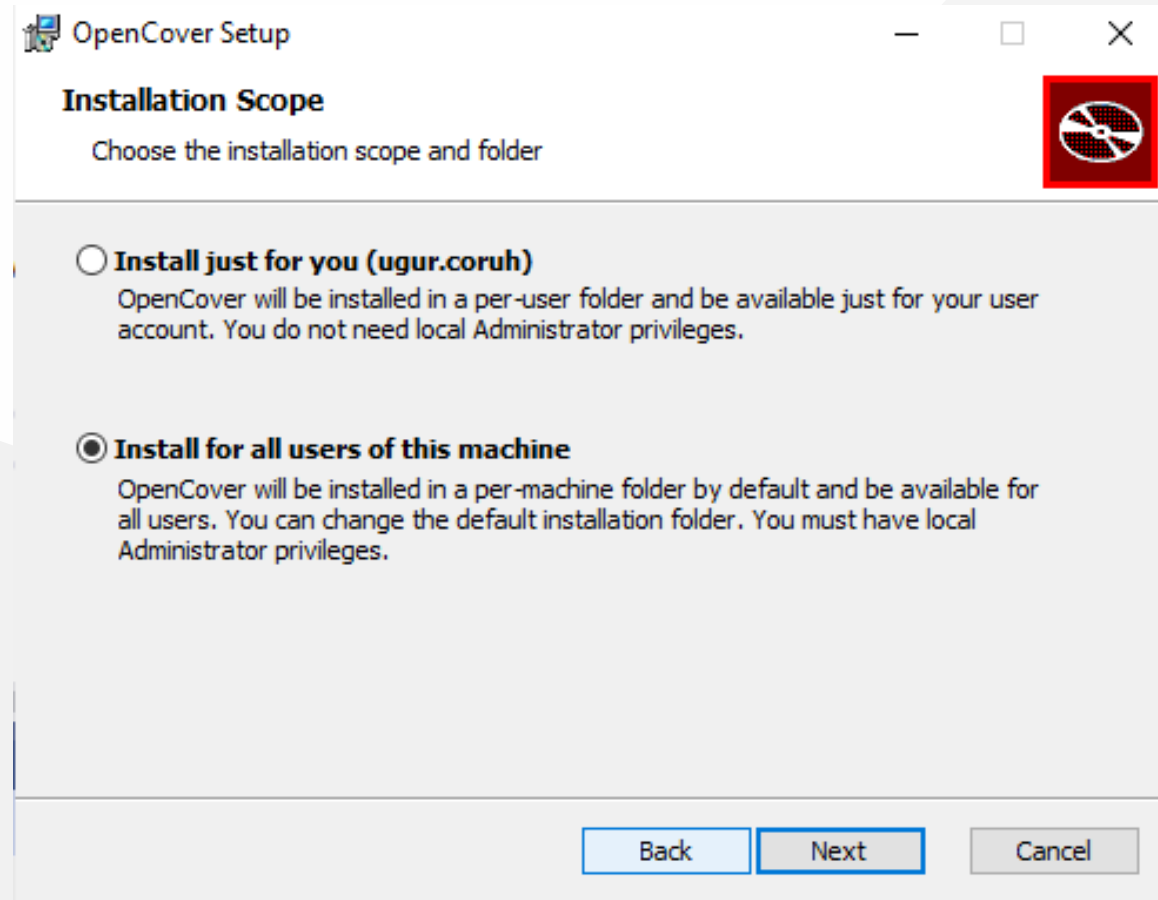
checksum.installer.txt	66 Bytes
checksum.zip.txt	66 Bytes
opencover.4.7.1221.msi	8.09 MB
opencover.4.7.1221.zip	7.76 MB
Source code (zip)	
Source code (tar.gz)	

Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-8

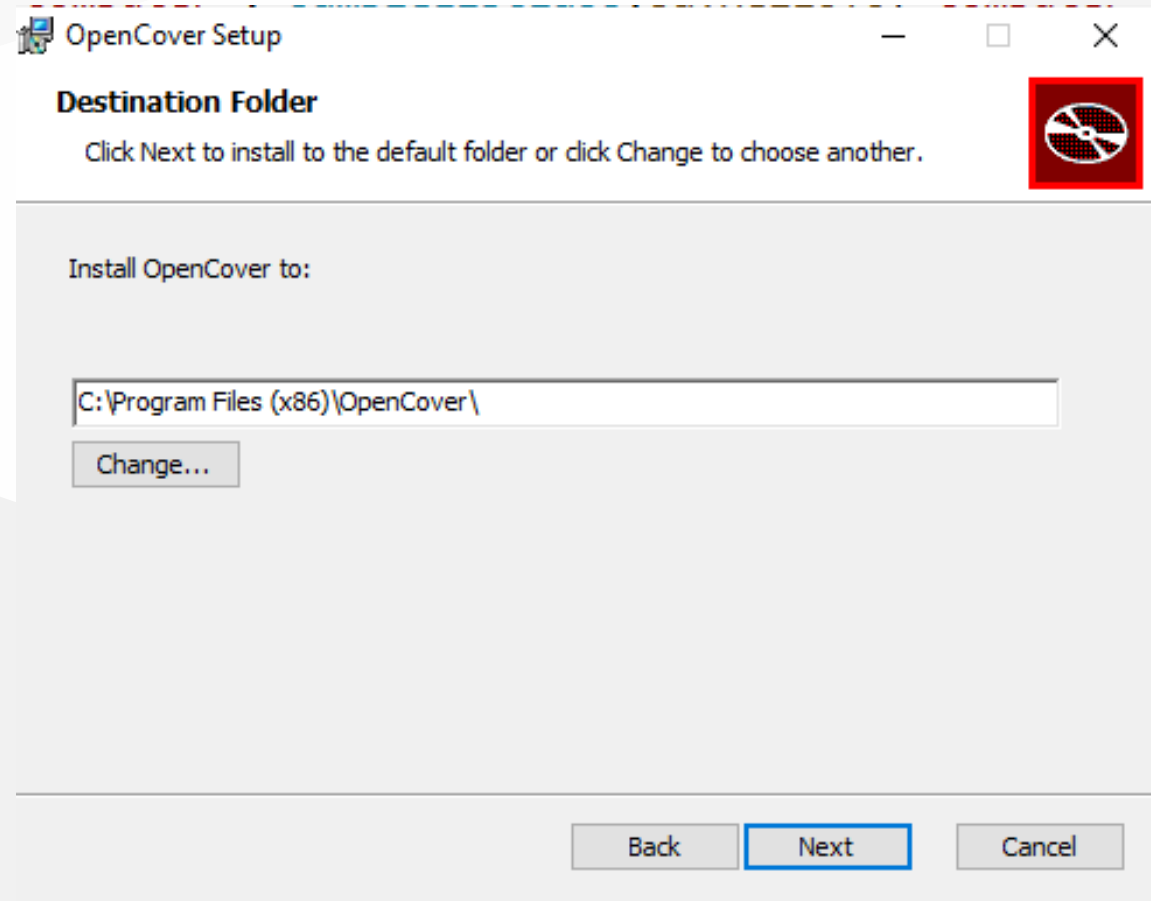


Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-9

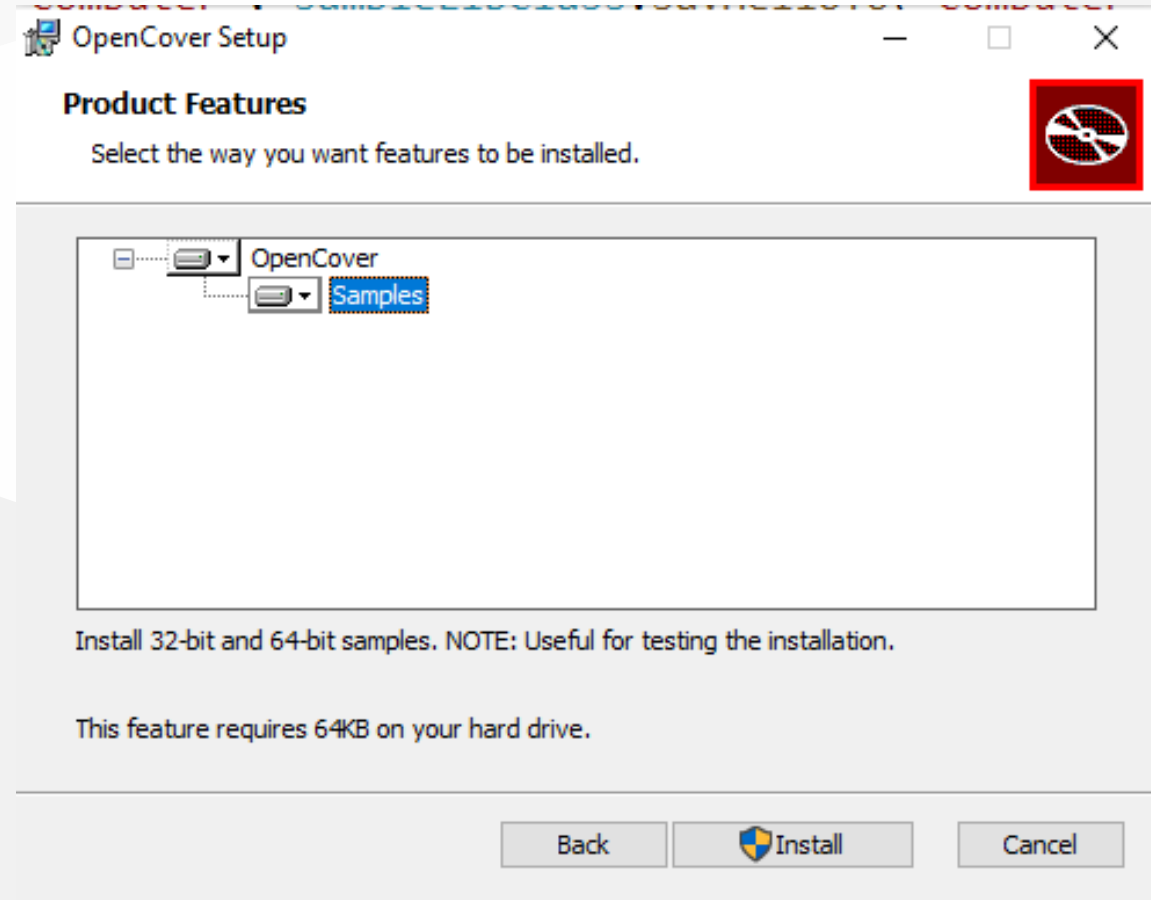
Select advanced and then install for all users



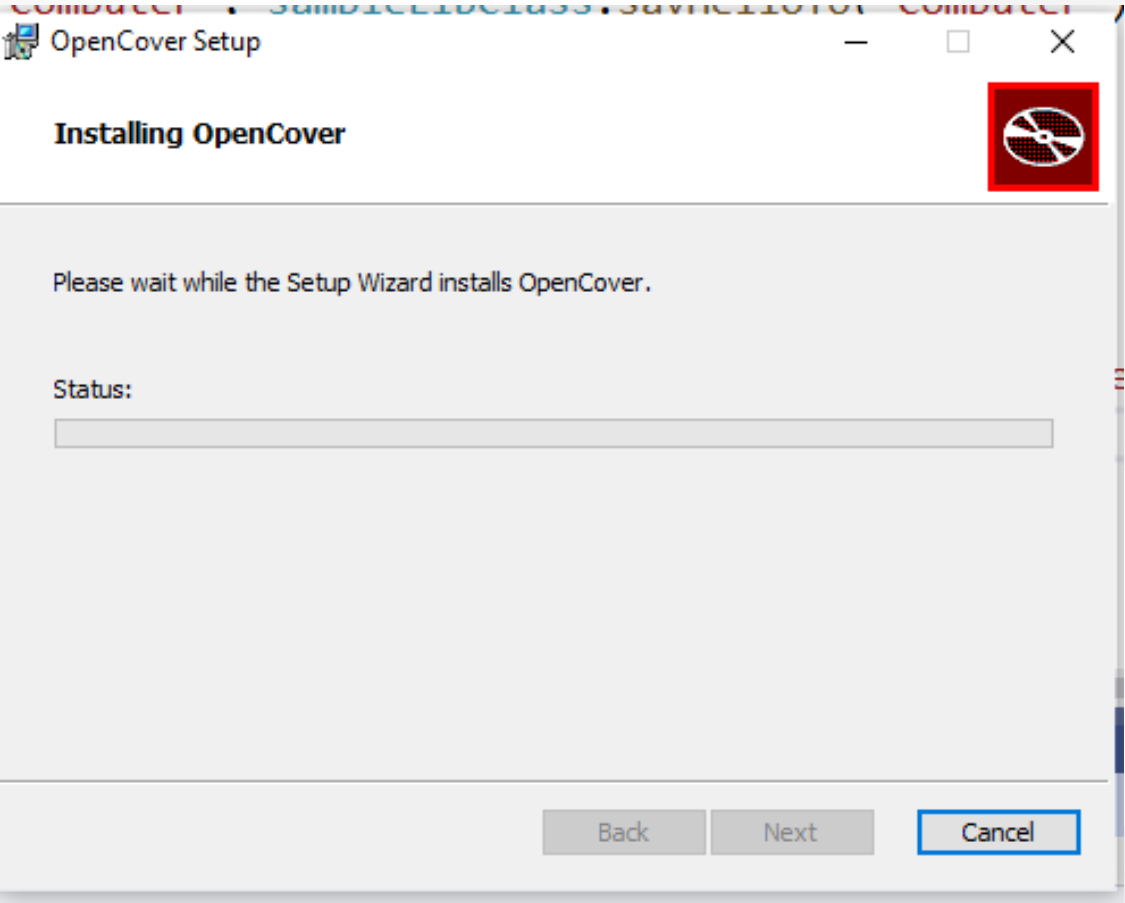
Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-10



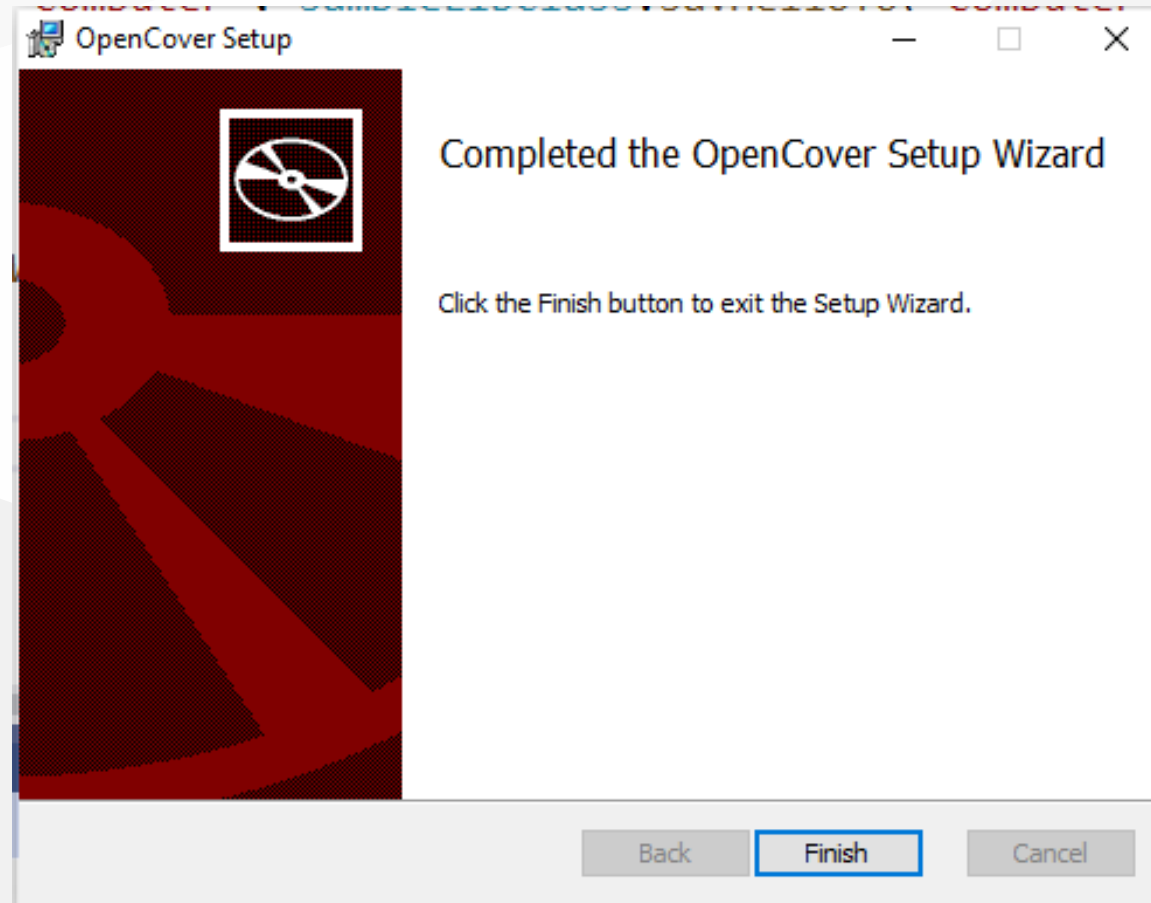
Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-11











Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-12



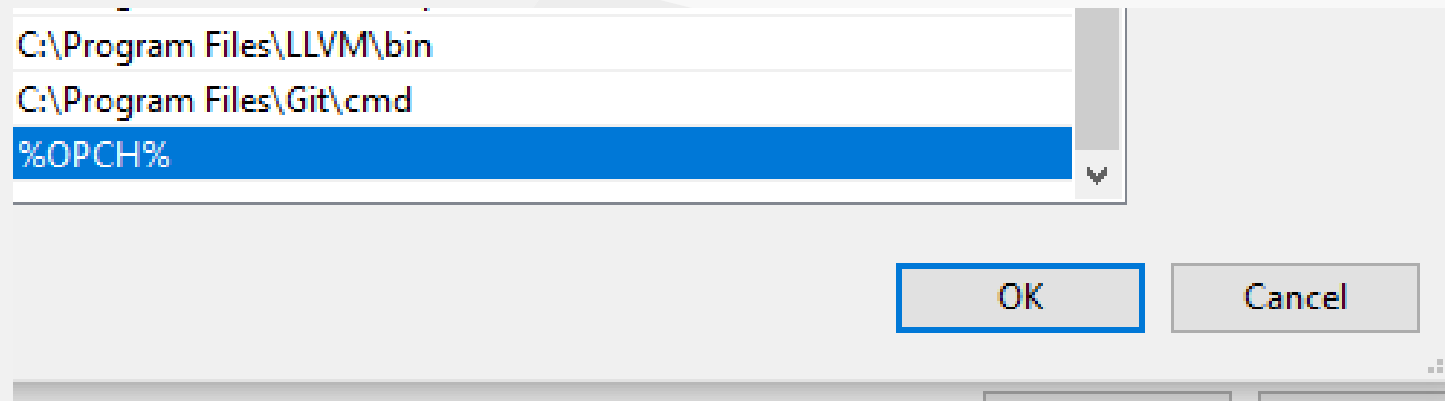
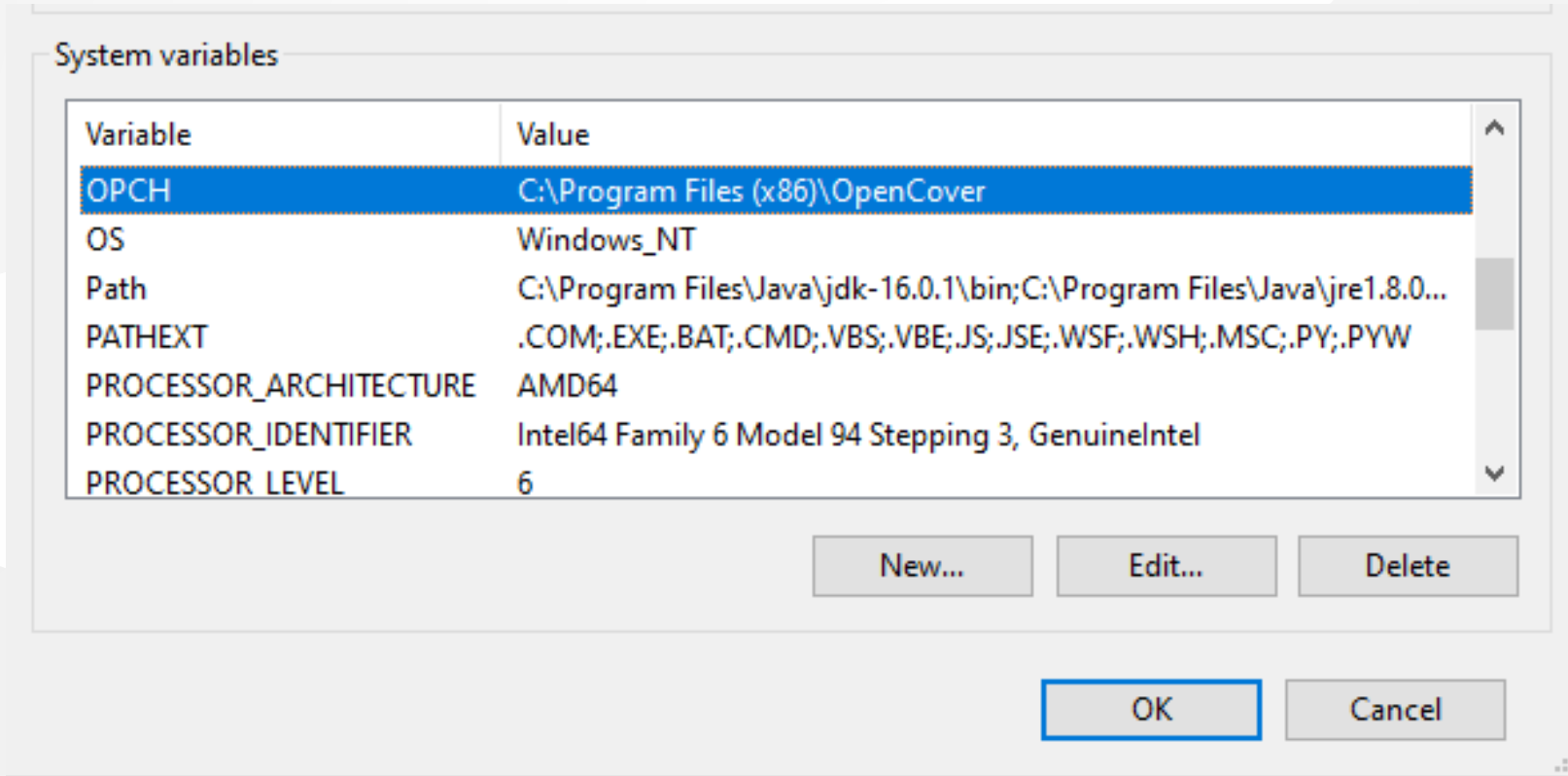
Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-13



Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-14

 Mono.Cecil.Pdb.dll	9/15/2021
 Mono.Cecil.Rocks.dll	9/15/2021
 Newtonsoft.Json.dll	11/9/2019
 OpenCover.Console.exe	6/19/2022
 OpenCover.Console.exe.config	6/19/2022
 OpenCover.Console.pdb	6/19/2022
 OpenCover.Extensions.dll	6/19/2022
 OpenCover.Extensions.pdb	6/19/2022

Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-15



Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-16

```
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
```

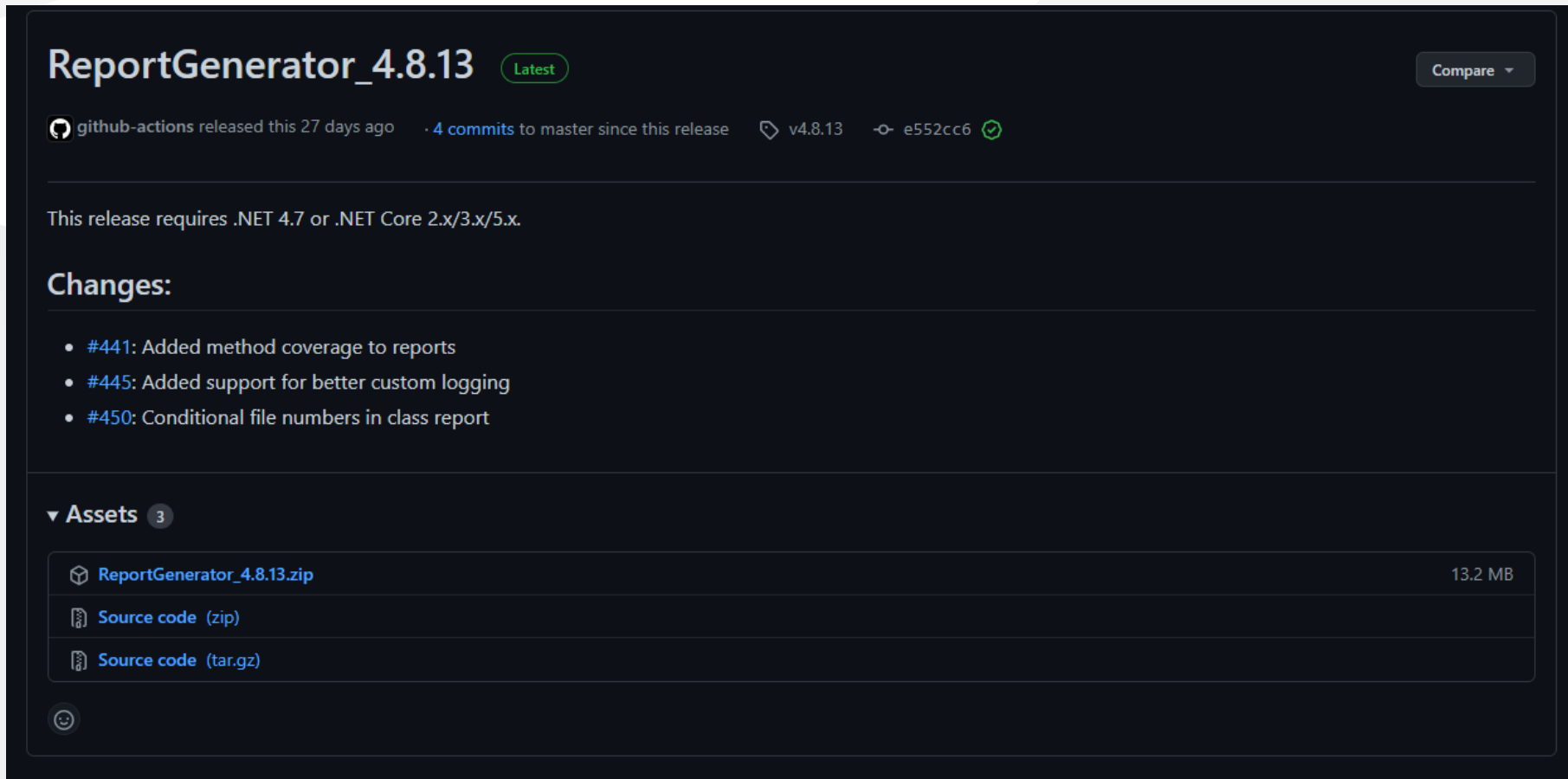
```
Usage:
```

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

Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-17

ReportGenerator

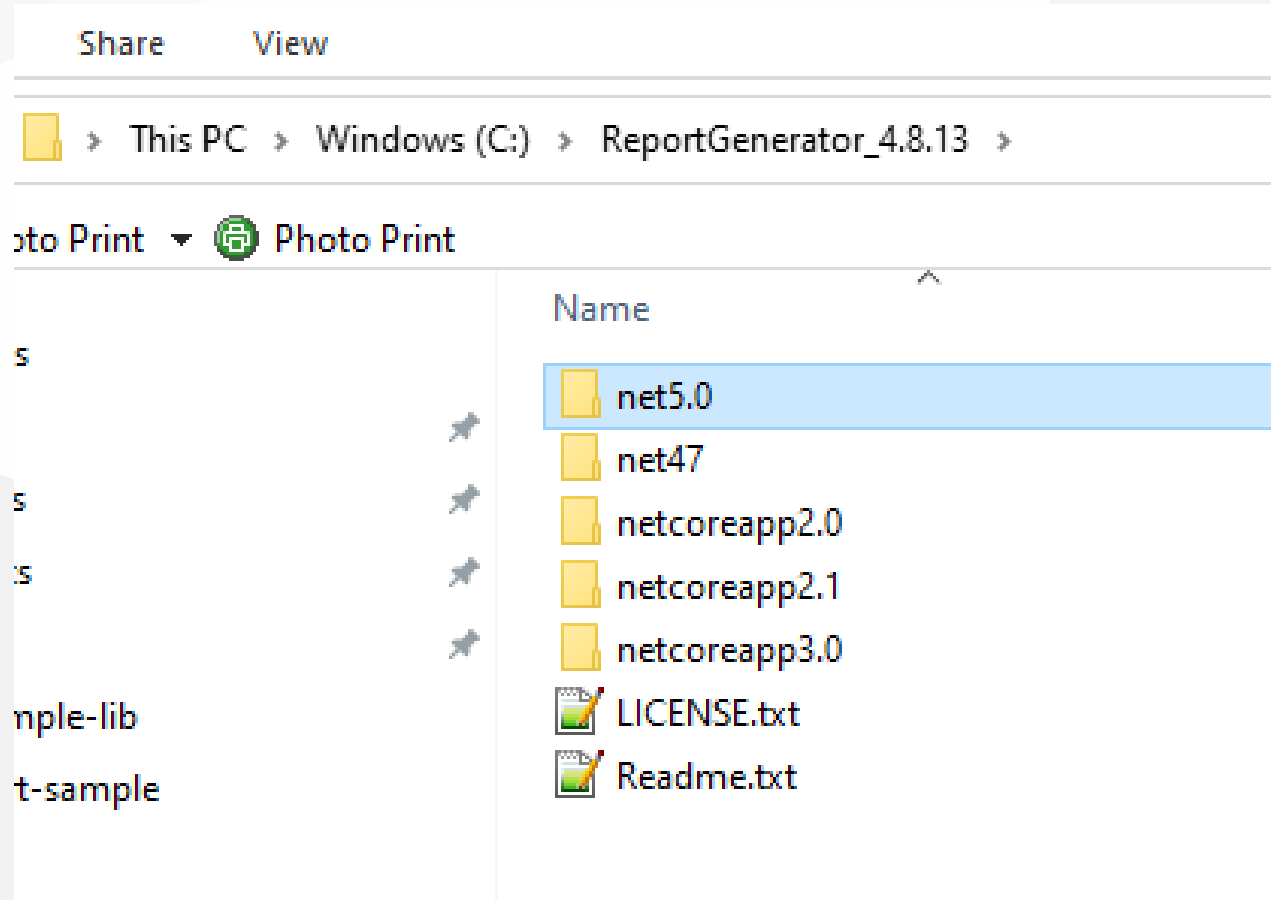
- [Release ReportGenerator_4.8.13](#) · [danielpalme/ReportGenerator](#) · [GitHub](#)



The screenshot shows the GitHub release page for **ReportGenerator_4.8.13**. The page is dark-themed and includes the following information:

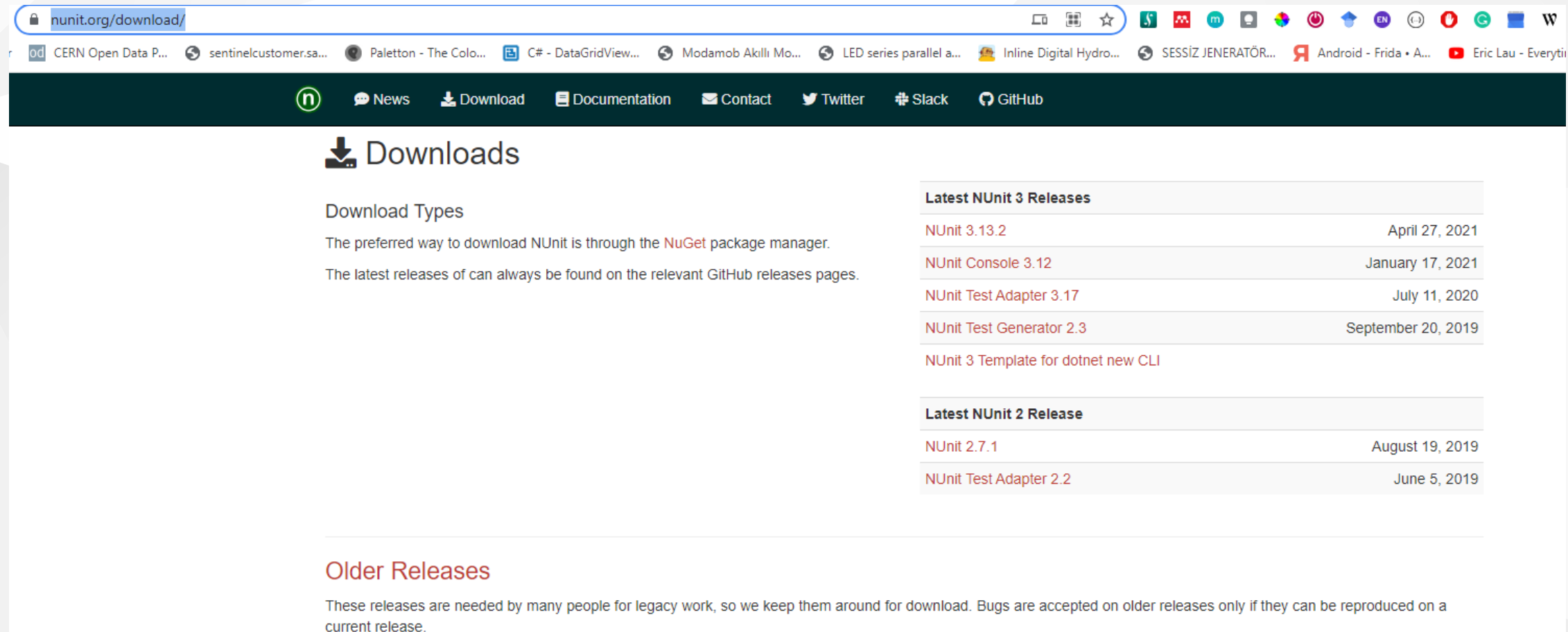
- Release Title:** ReportGenerator_4.8.13 (Latest)
- Author:** github-actions released this 27 days ago
- Commits:** 4 commits to master since this release
- Version:** v4.8.13
- Commit Hash:** e552cc6
- Requirements:** This release requires .NET 4.7 or .NET Core 2.x/3.x/5.x.
- Changes:**
 - #441: Added method coverage to reports
 - #445: Added support for better custom logging
 - #450: Conditional file numbers in class report
- Assets:** 3 assets are listed:
 - ReportGenerator_4.8.13.zip (13.2 MB)
 - Source code (zip)
 - Source code (tar.gz)

Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-18



NUnit Console

- Downloads



The screenshot shows the NUnit.org website's Downloads page. The browser address bar displays "nunit.org/download/". The page features a dark navigation bar with links for News, Download, Documentation, Contact, Twitter, Slack, and GitHub. The main content area is titled "Downloads" and includes a section for "Download Types" with instructions on using NuGet and finding releases on GitHub. To the right, there are two tables listing the latest releases for NUnit 3 and NUnit 2.

Downloads

Download Types

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

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

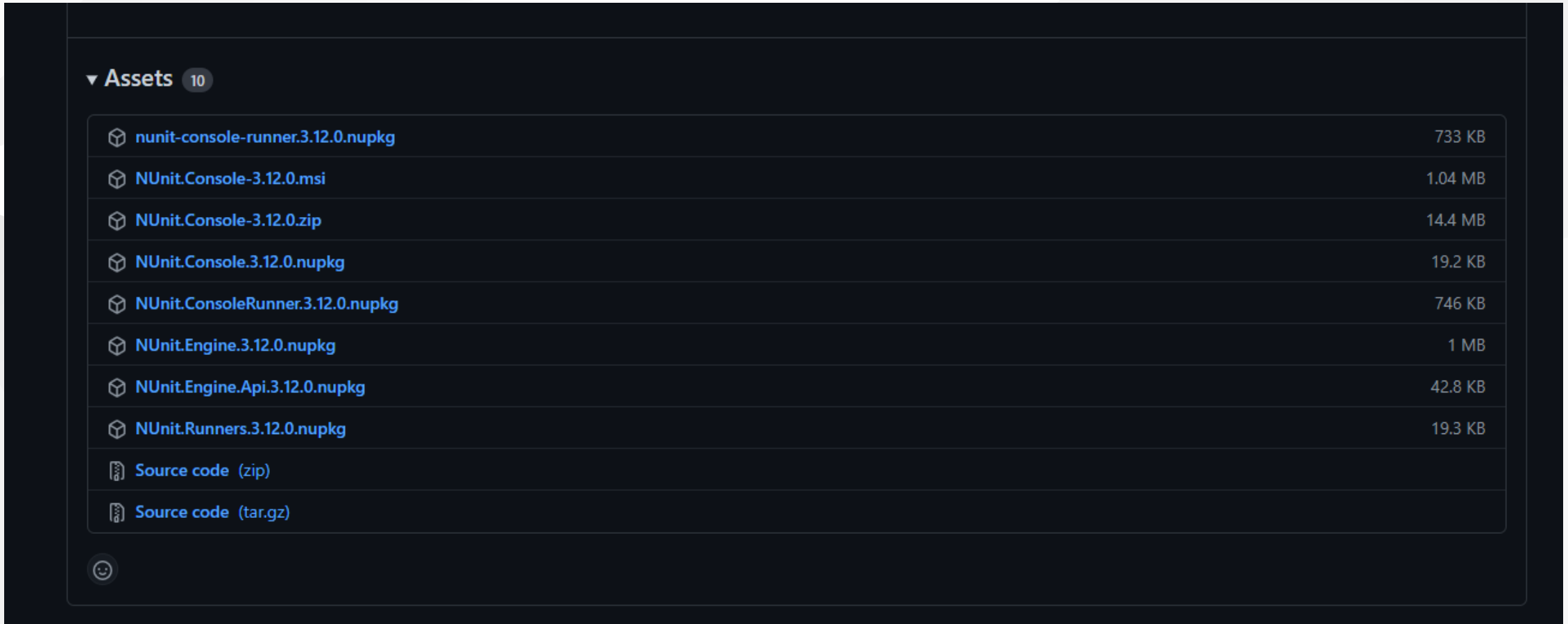
Latest NUnit 3 Releases	
NUnit 3.13.2	April 27, 2021
NUnit Console 3.12	January 17, 2021
NUnit Test Adapter 3.17	July 11, 2020
NUnit Test Generator 2.3	September 20, 2019
NUnit 3 Template for dotnet new CLI	

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.

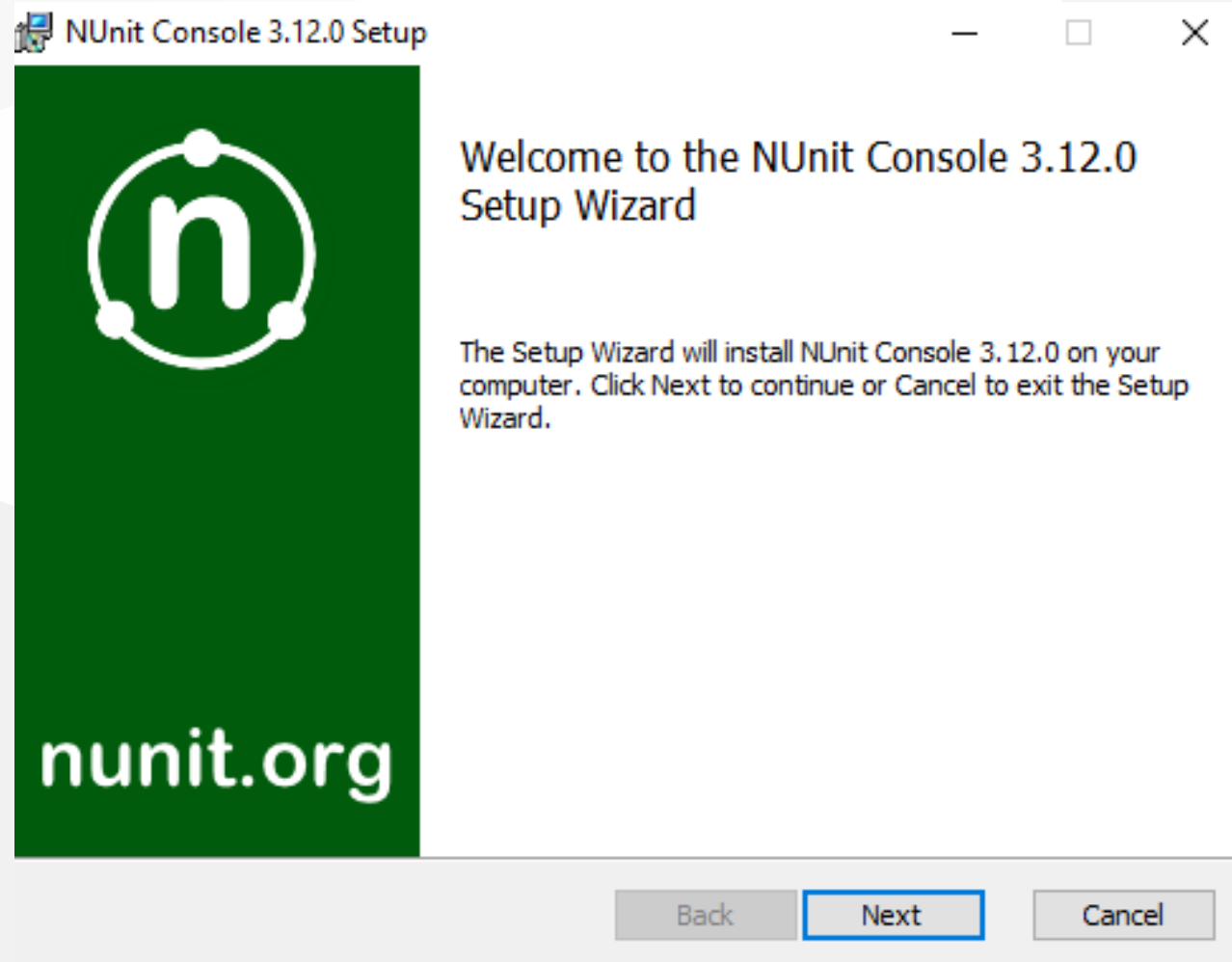
Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-20



The screenshot displays the Package Manager interface in Visual Studio, showing a list of assets for the NUnit package. The assets are listed in a table with columns for the asset name and its size.

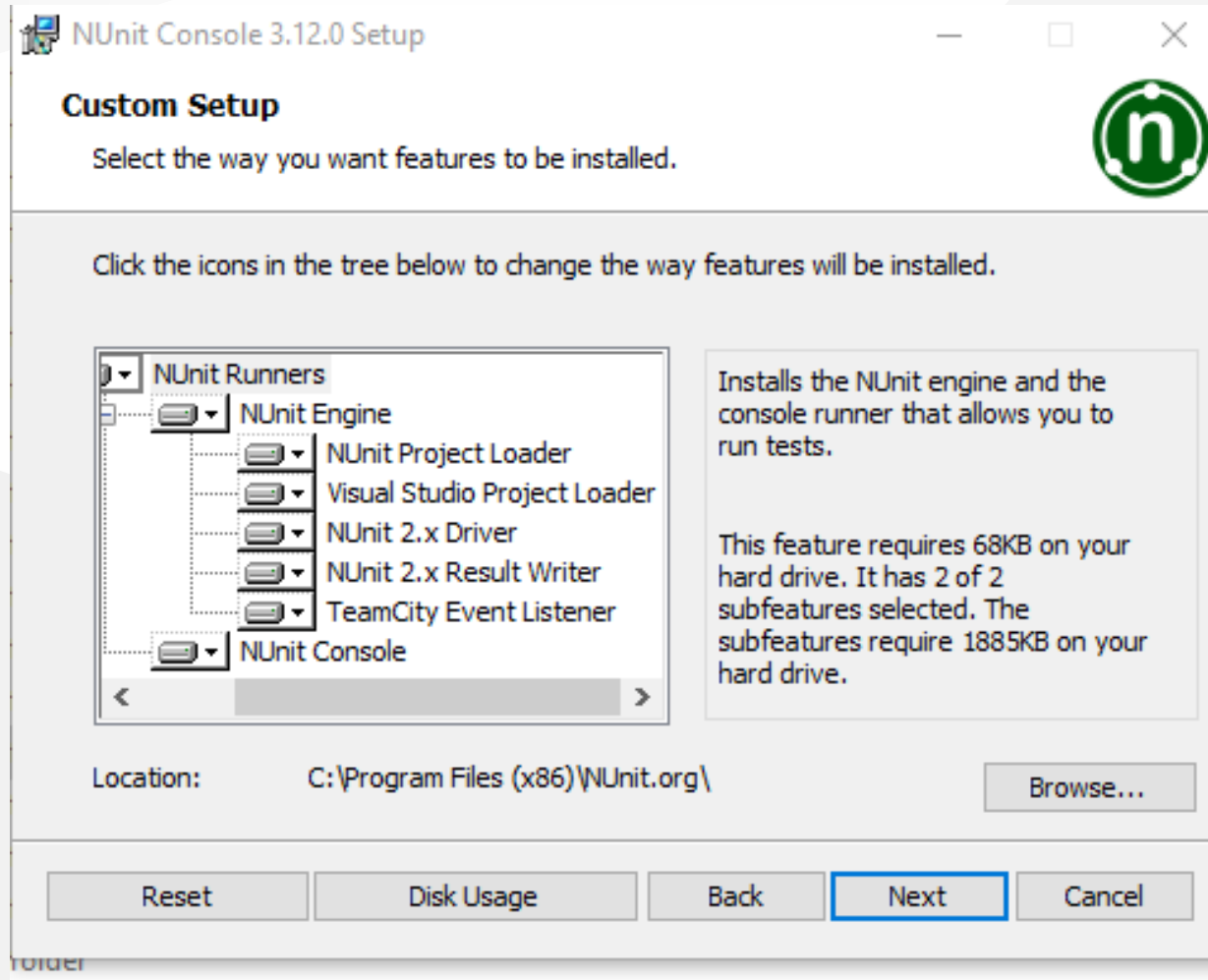
Asset Name	Size
nunit-console-runner.3.12.0.nupkg	733 KB
NUnit.Console-3.12.0.msi	1.04 MB
NUnit.Console-3.12.0.zip	14.4 MB
NUnit.Console.3.12.0.nupkg	19.2 KB
NUnit.ConsoleRunner.3.12.0.nupkg	746 KB
NUnit.Engine.3.12.0.nupkg	1 MB
NUnit.Engine.Api.3.12.0.nupkg	42.8 KB
NUnit.Runners.3.12.0.nupkg	19.3 KB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

Visual Studio Community Edition (C# Unit Test+OpenCover + NUnit Runner + Report)-21



Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-22

- Download setup



Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-23

- Install setup

Share View

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

Print Photo Print

Name	Date modified	Type	Size
addins	10/24/2021 11:30 PM	File folder	
agents	10/24/2021 11:30 PM	File folder	
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

Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-24

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)

Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

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](#)

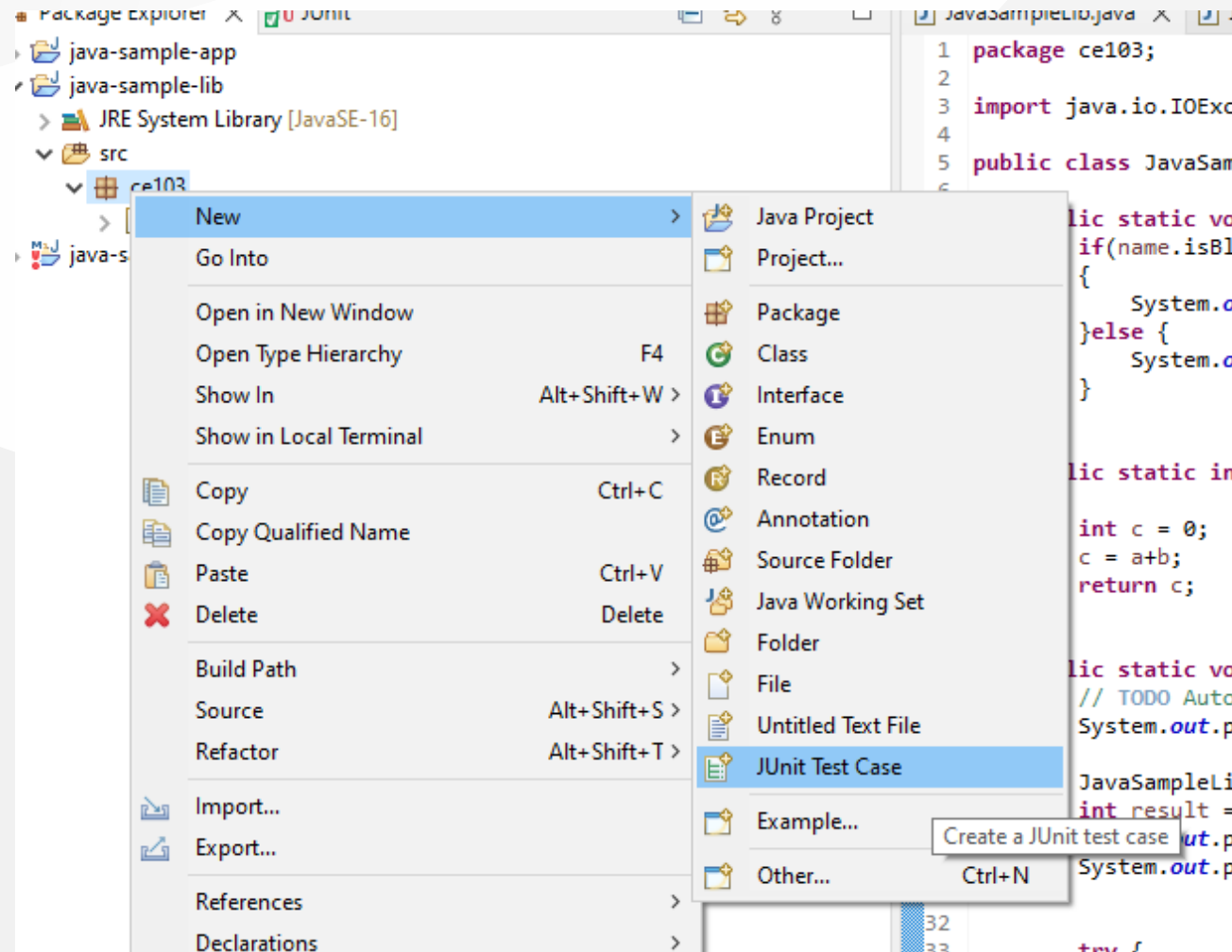
<https://www.eclemma.org/>

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

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

Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

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



Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

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?

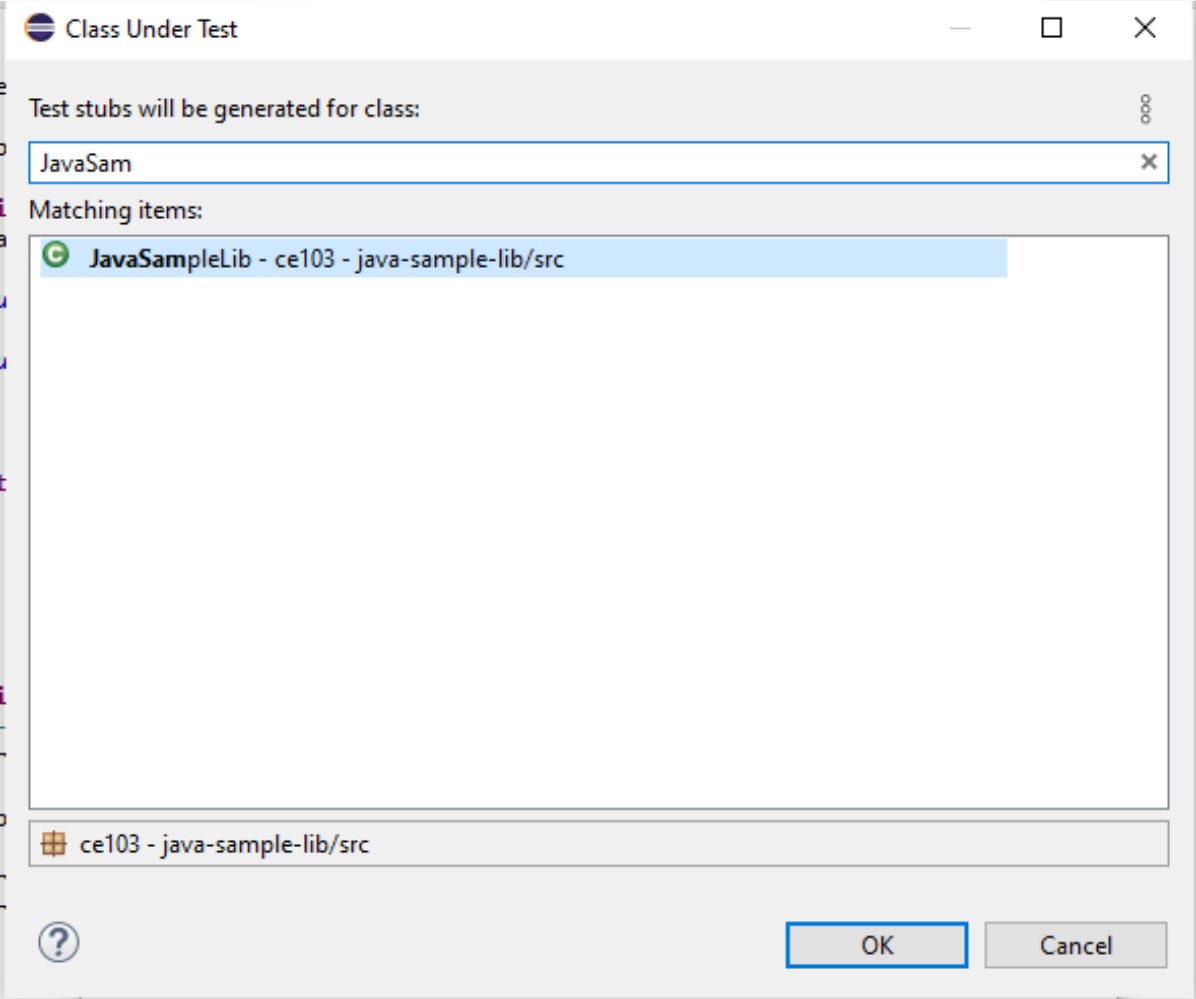
@BeforeAll setUpBeforeClass() @AfterAll tearDownAfterClass()
 @BeforeEach setUp() @AfterEach tearDown()
 constructor

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

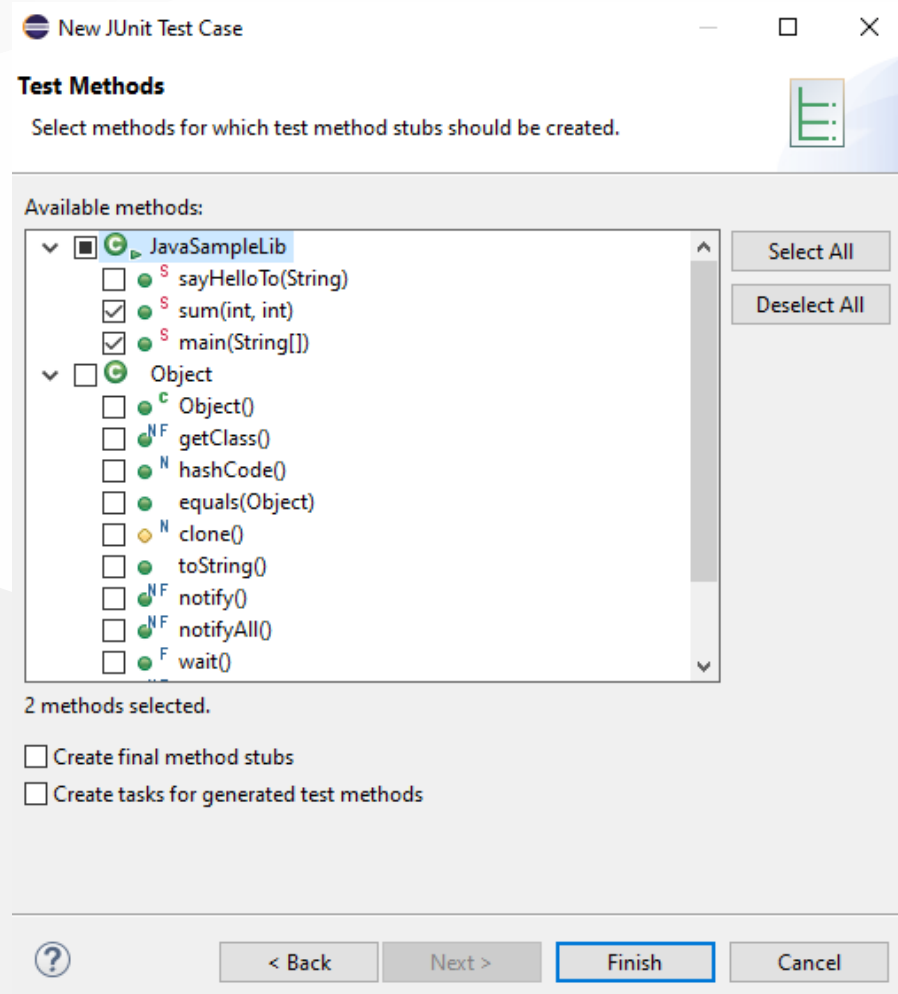
Generate comments

Class under test:

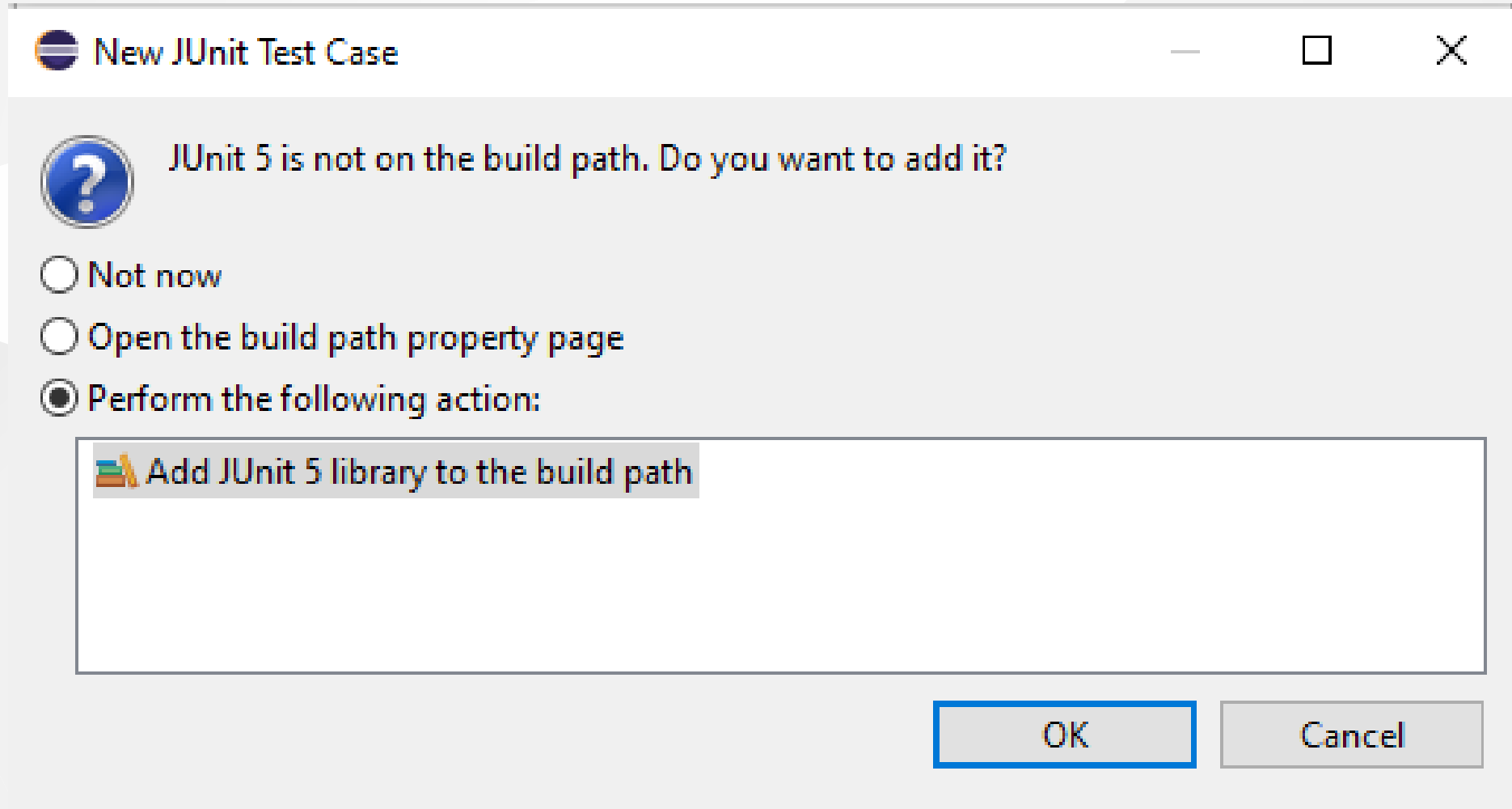
Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test



Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

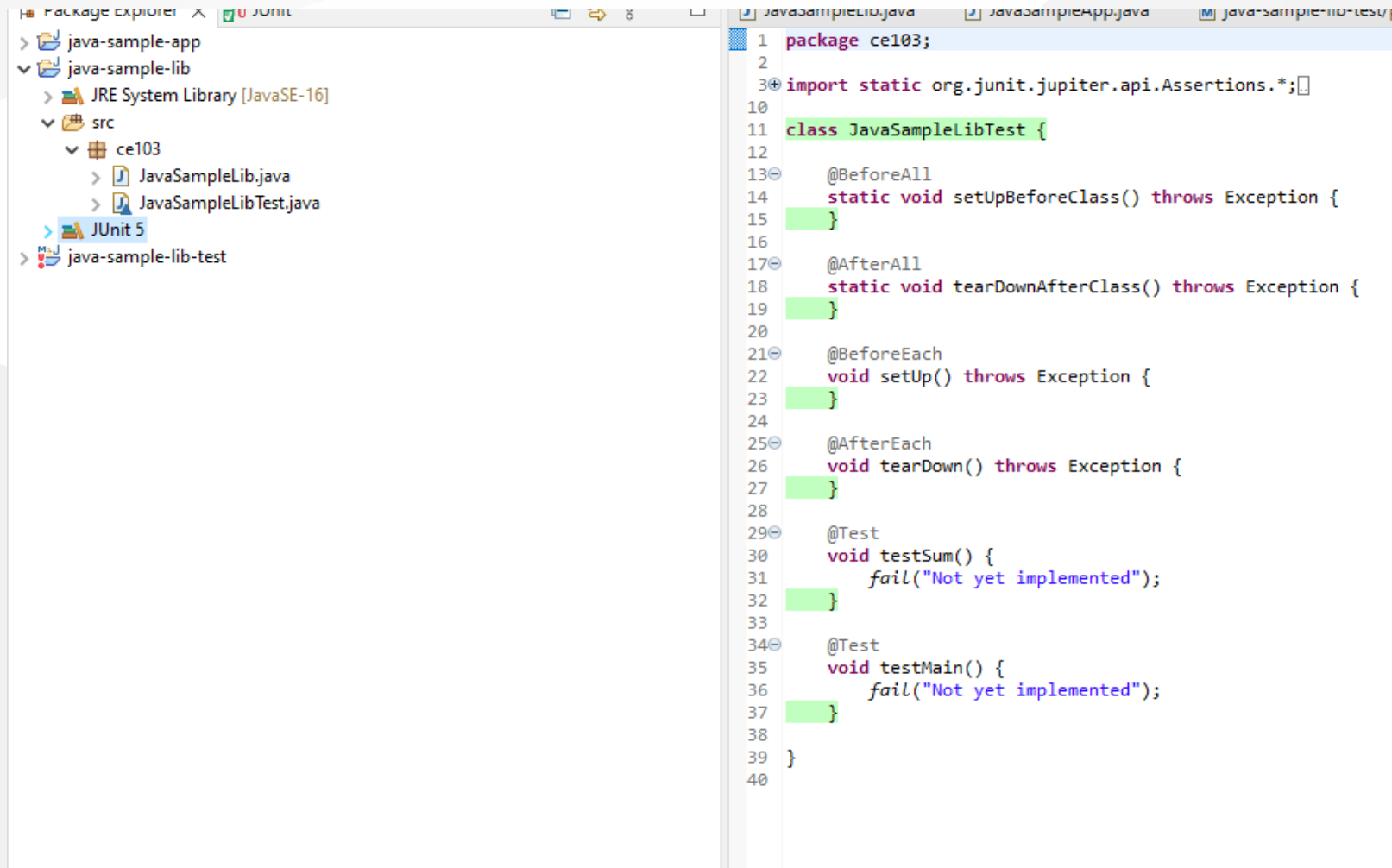


Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test



Eclipse IDE (JUnit4, JUnit5) + Java Unit Test

and you will have the following test class



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays a project named 'java-sample-app' containing a sub-project 'java-sample-lib'. Inside 'java-sample-lib', there is a 'src' folder with a package 'ce103' containing 'JavaSampleLib.java' and 'JavaSampleLibTest.java'. A 'JUnit 5' icon is also visible. On the right, the Java editor shows the code for 'JavaSampleLibTest.java'.

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

We need to cover all code branches that we coded

I have updated `JavaSampleLib.java` as follows 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();
    //     }
    // }
```

Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test 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");
    }

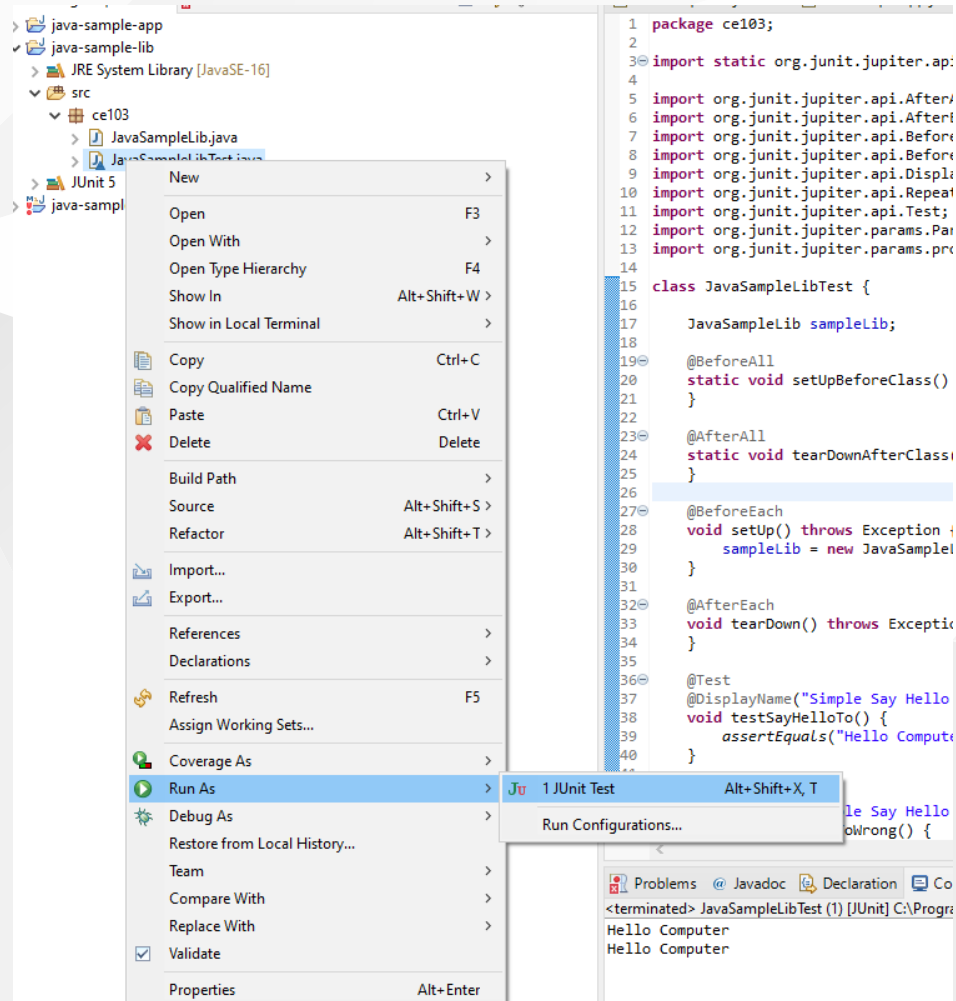
    public static int[][] data() {
        return new int[][] { { 1, 2, 2 }, { 5, 3, 15 }, { 121, 4, 484 }, { 2, 2, 2 } };
    }

    @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));
    }
}

```

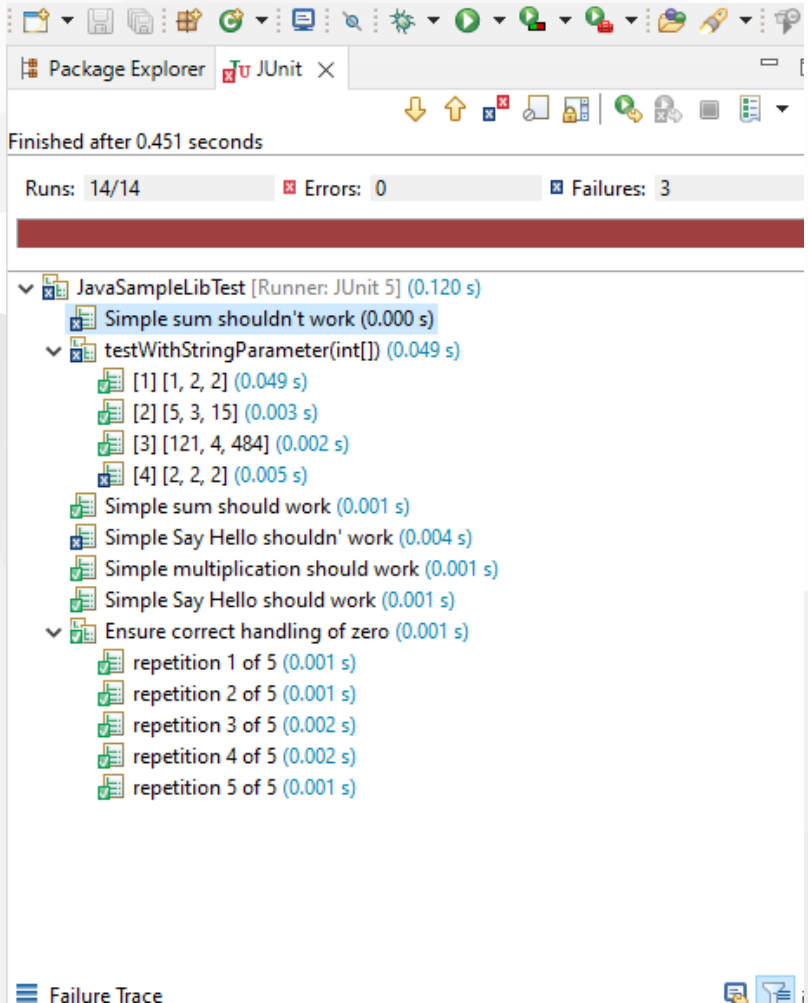

Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

if we run tests



Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

we will see all results there



Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

also we can see the code coverage of tests

Element	Coverage	Covered Instructio...	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

Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

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

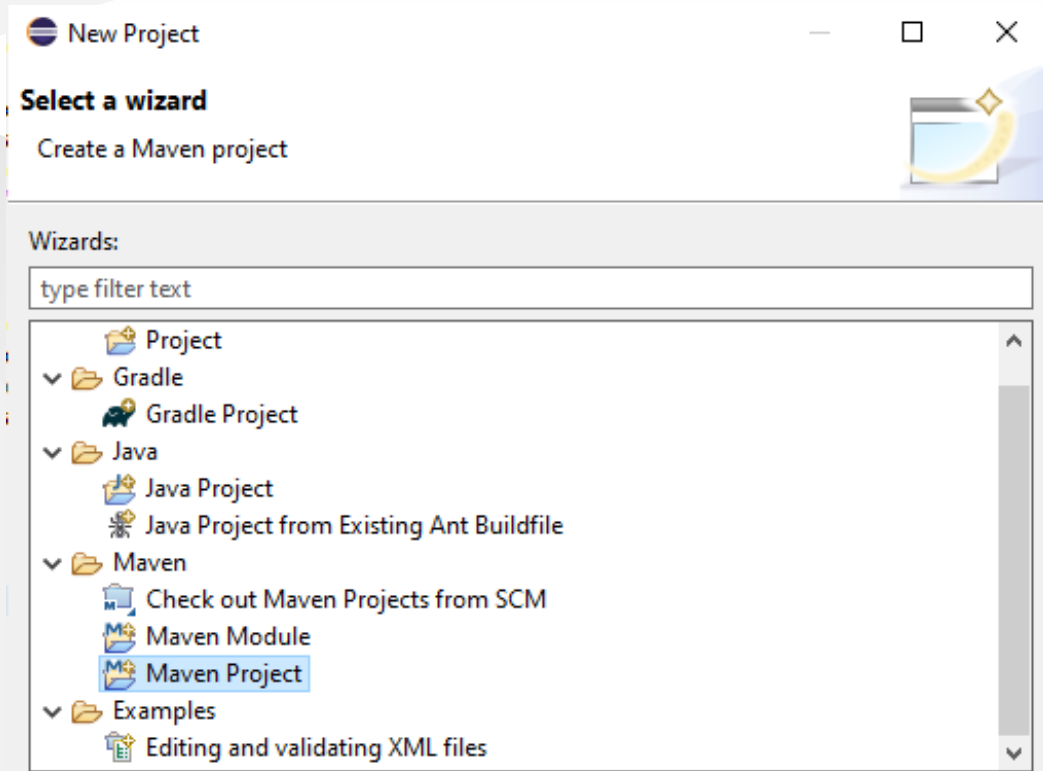
```
1 package ce103;
2
3 public class JavaSampleLib {
4
5     public static String sayHelloTo(String name) {
6
7         String output = "";
8
9         if(!name.isBlank() && !name.isEmpty()){
10            output = "Hello "+name;
11        }else {
12            output = "Hello There";
13        }
14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

Maven Java Application + JUnit

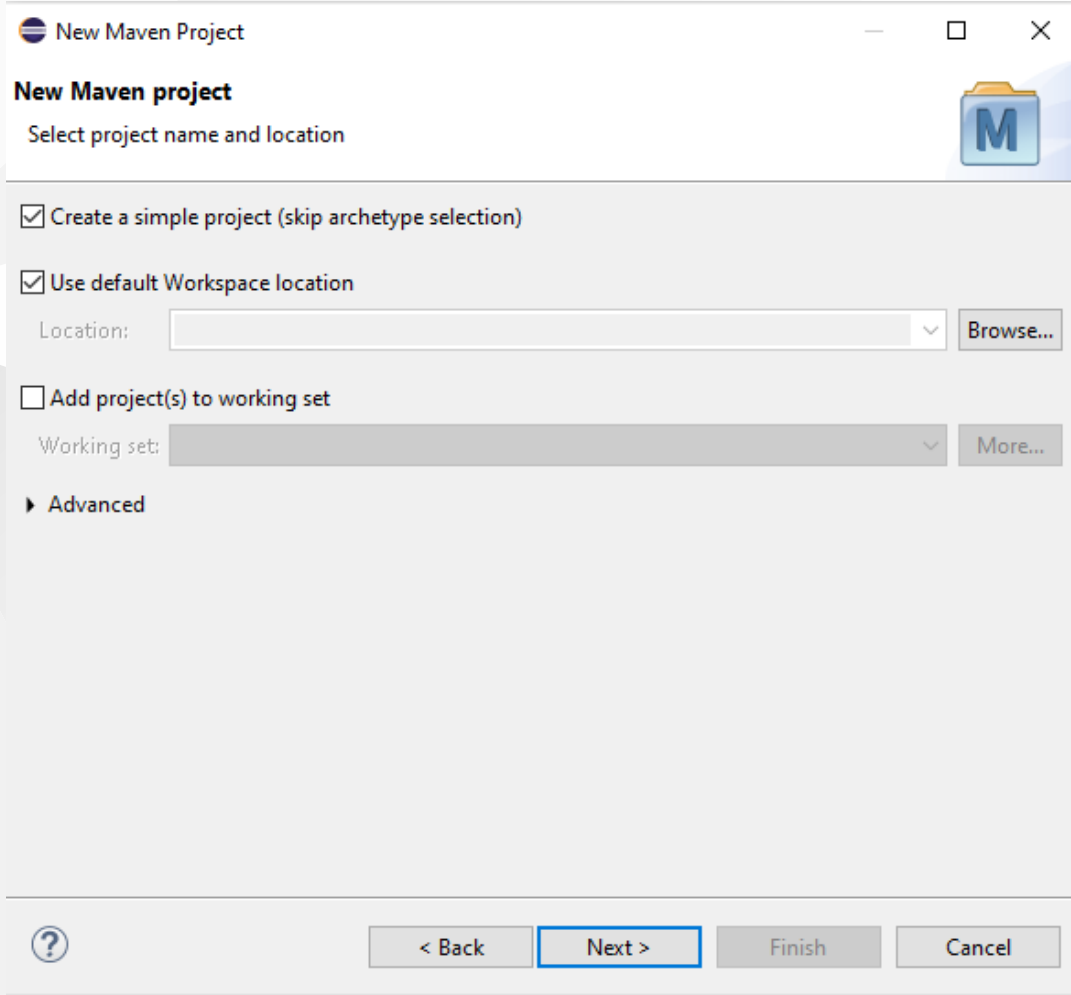
Lets create Maven project with tests

Create a maven project

File -> New -> Maven Project



Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test



Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

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)

Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

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

New Maven Project
Configure project

Artifact

Group Id:

Artifact Id:

Version:

Packaging:

Name:

Description:

Parent Project

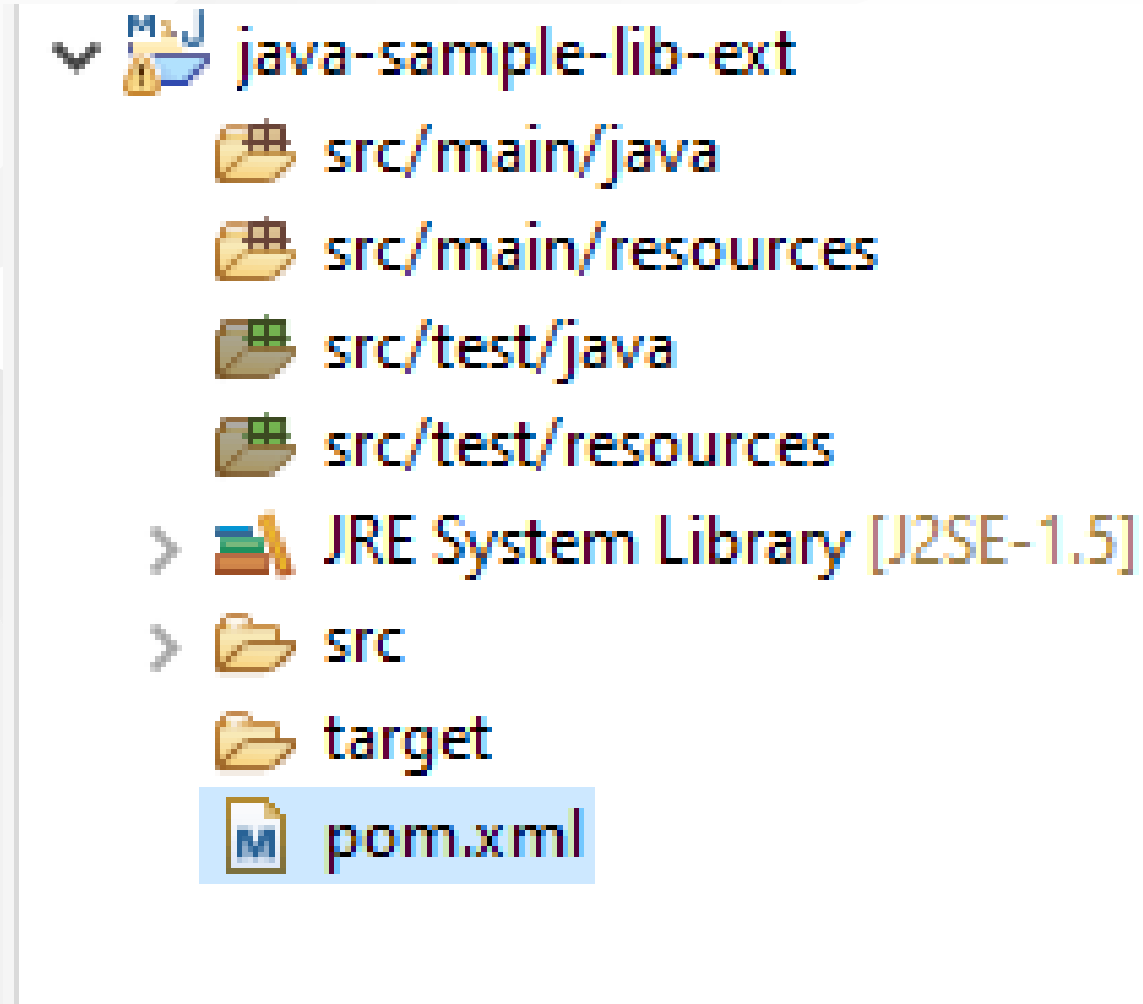
Group Id:

Artifact Id:

Version:

▶ **Advanced**

Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test



Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

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>
```

Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

we will add JUnit 5 for our project

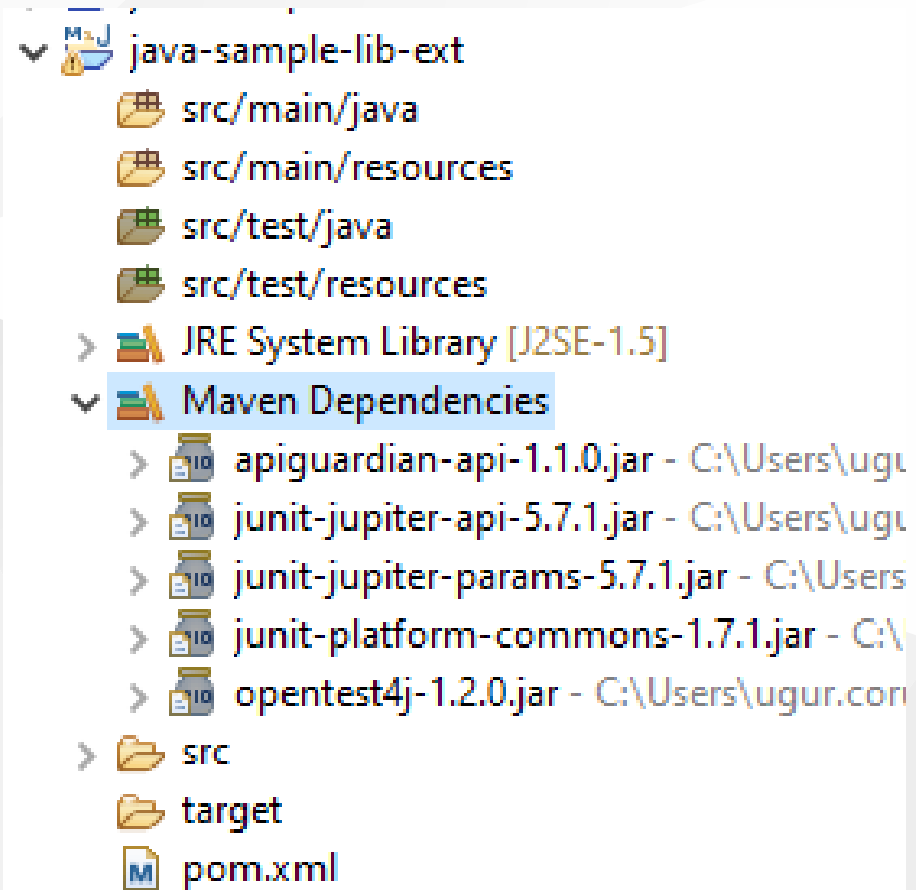
```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.ce103</groupId>
  <artifactId>java-sample-lib-ext</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>

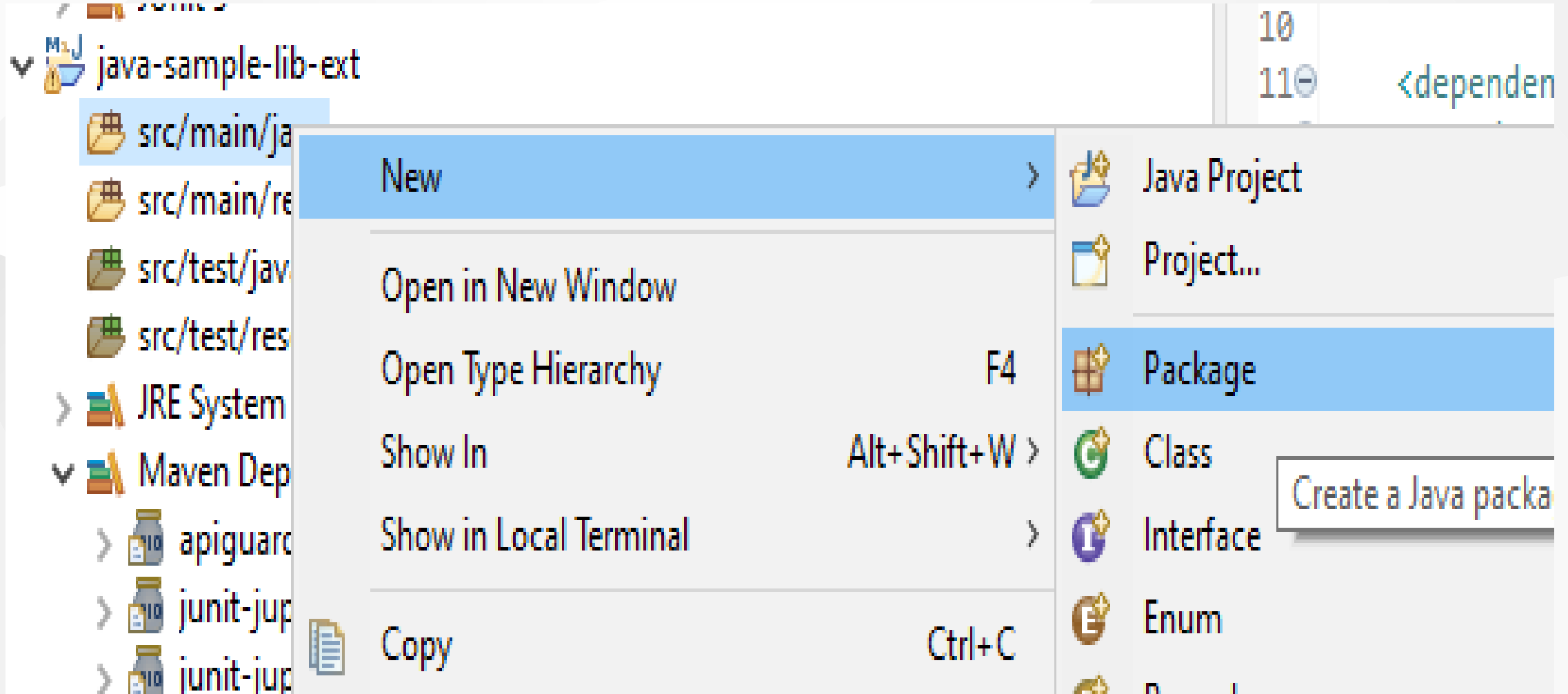
</project>
```

Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

it will automatically download libraries

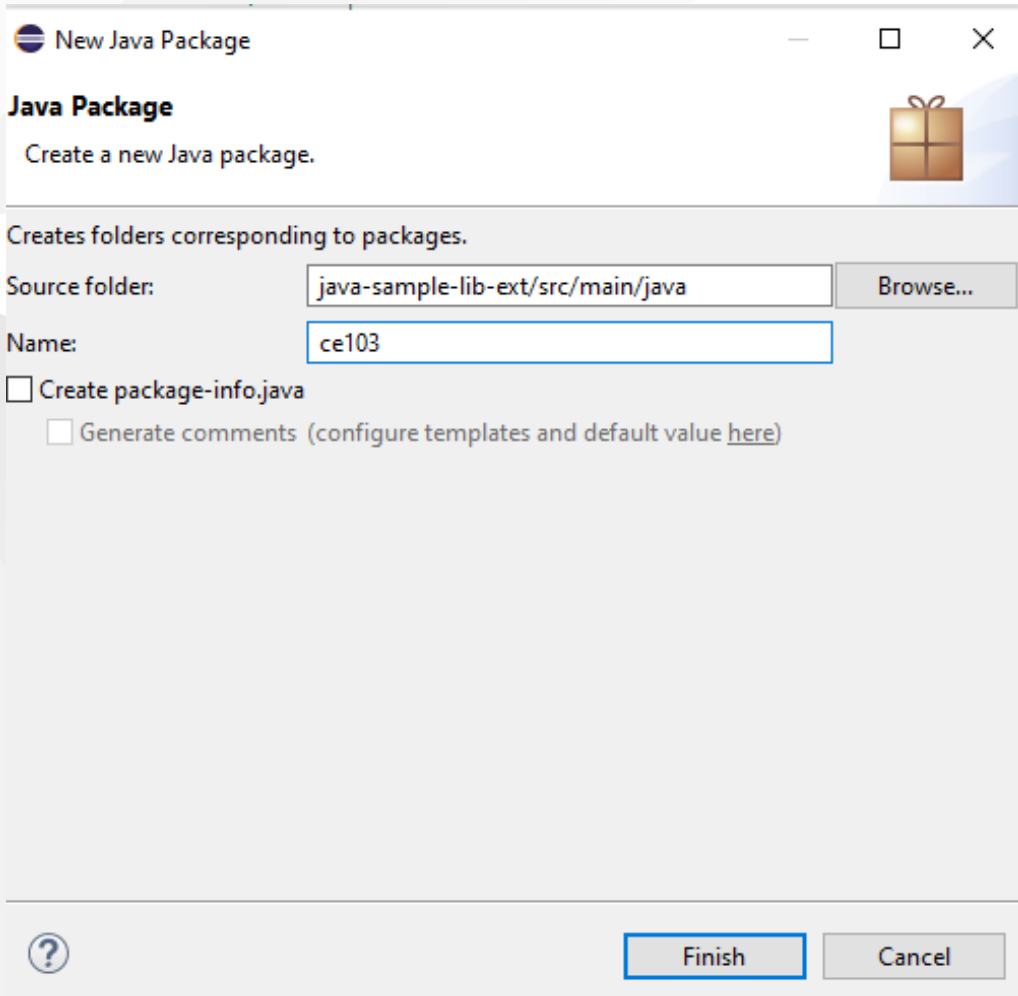


Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test



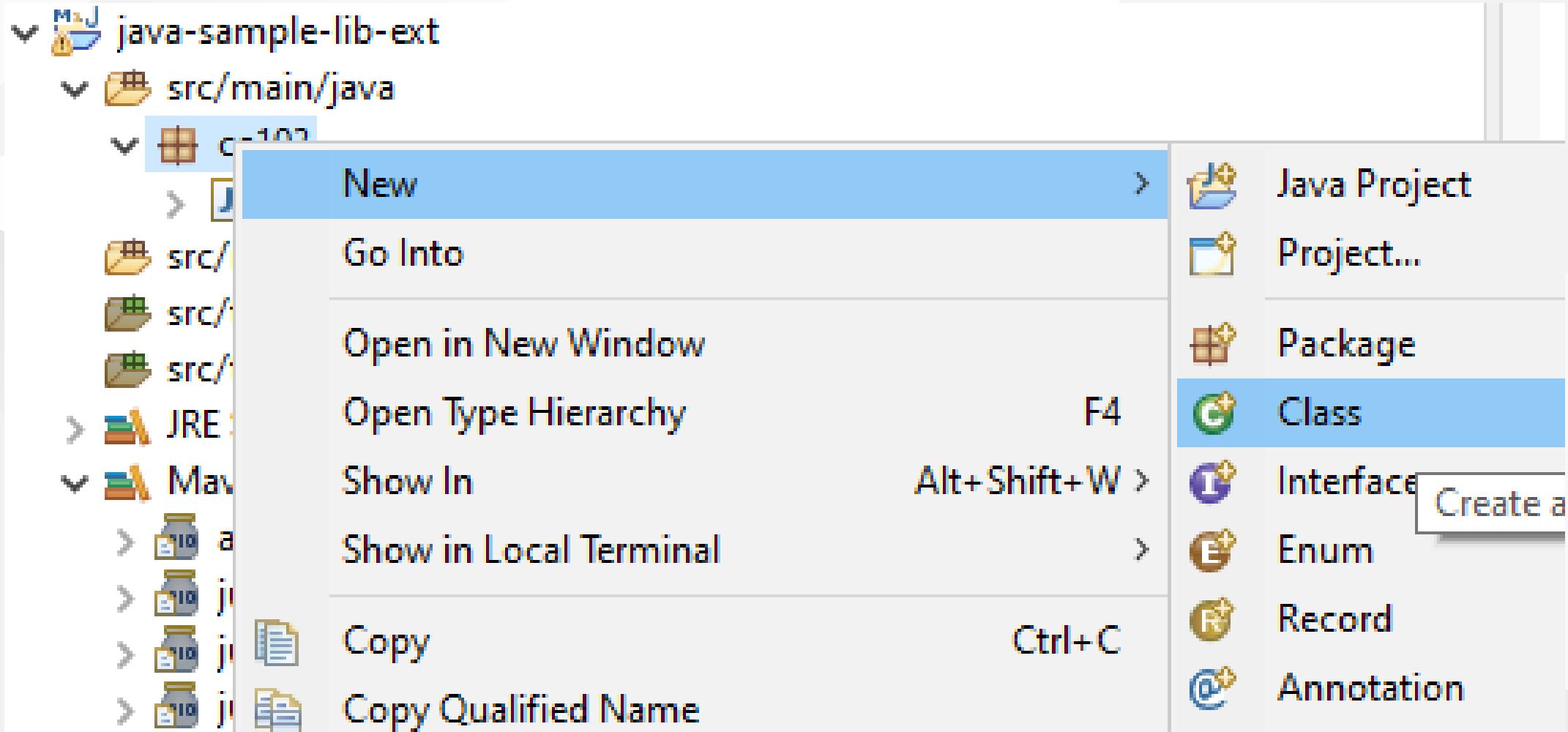
Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

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

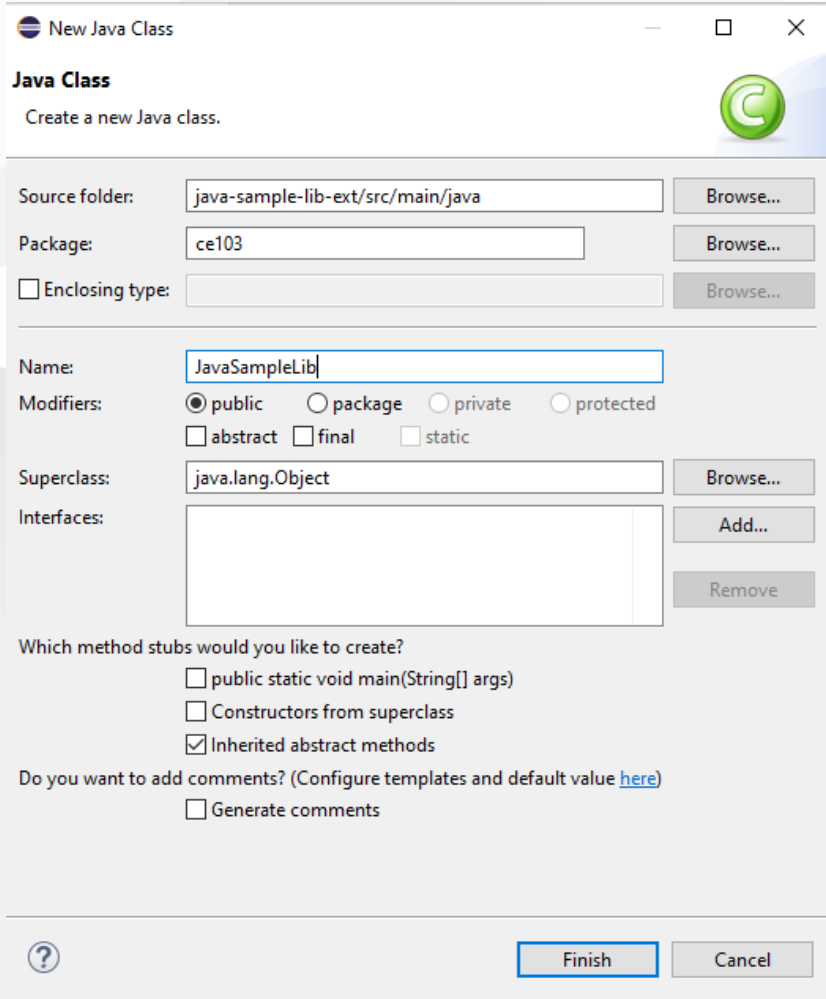


Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

In this package create library class



Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test



Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

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);

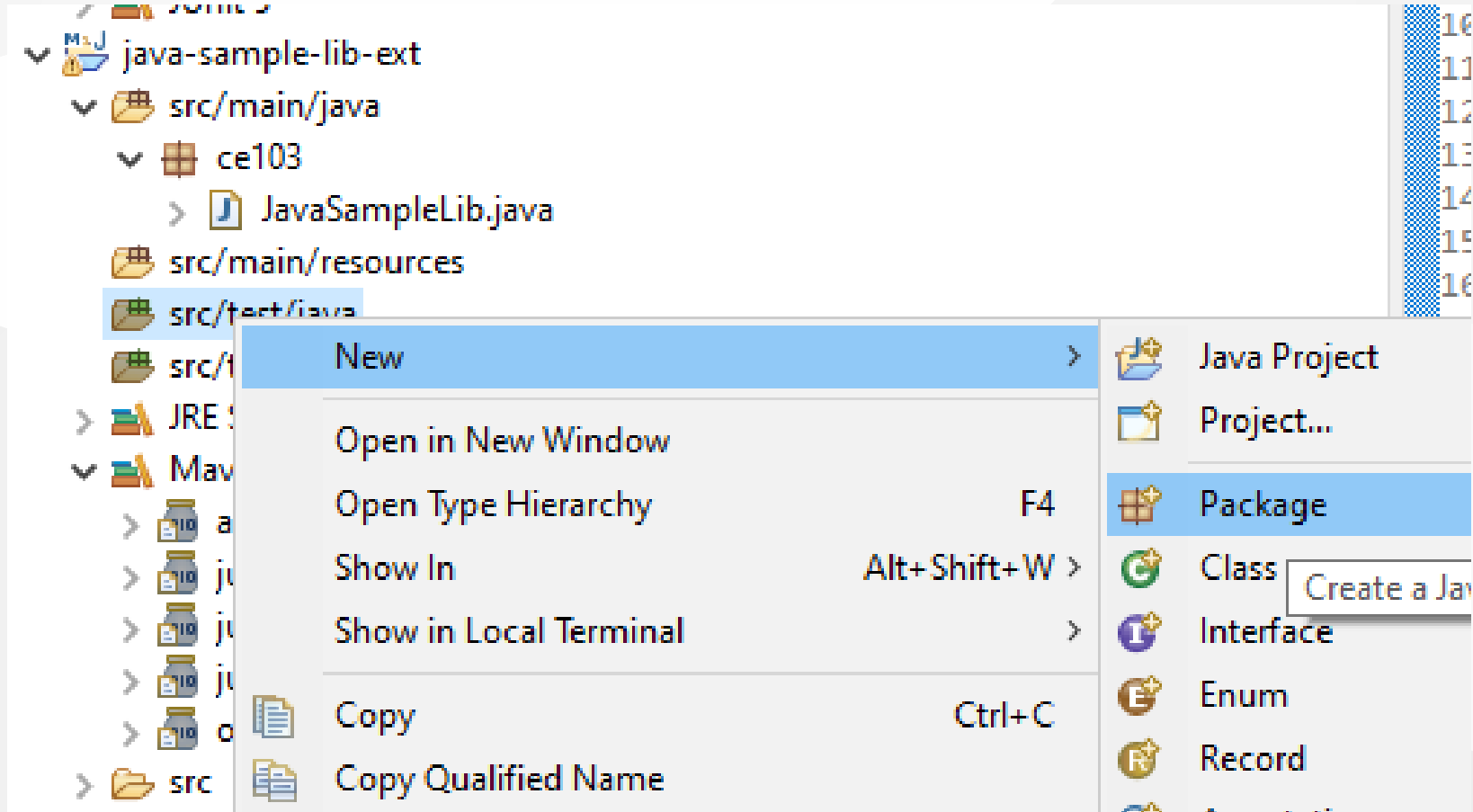
        return output;
    }

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

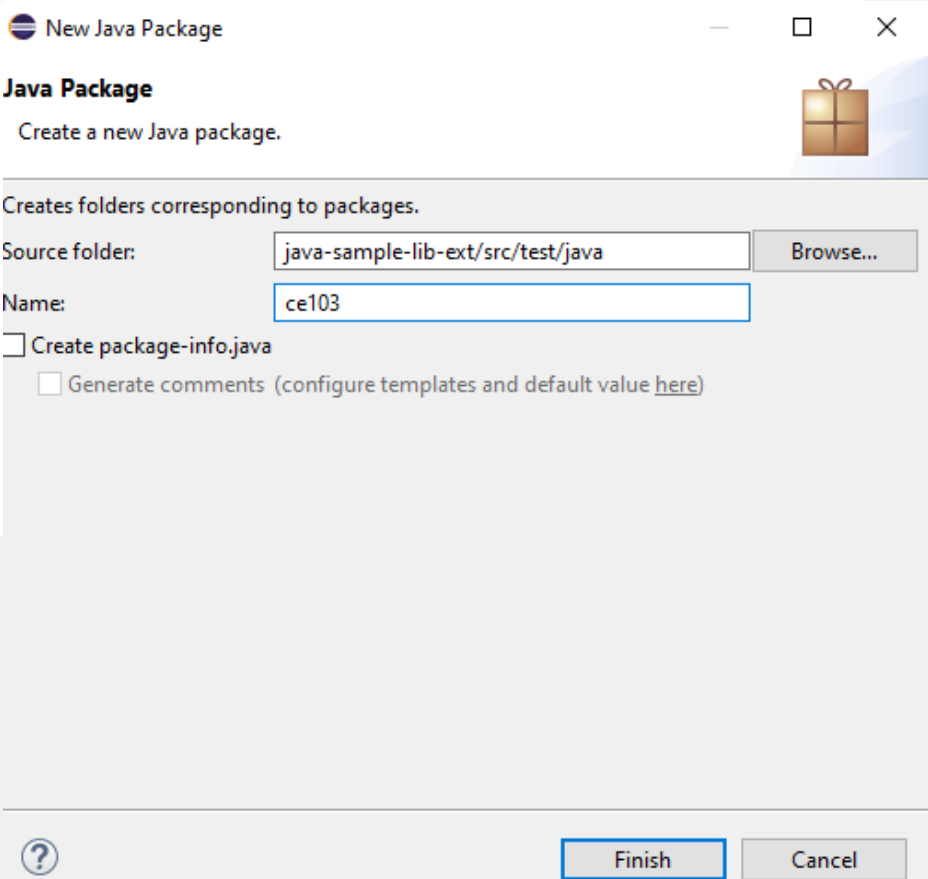
    public int multiply(int a, int b) {
        return a * b;
    }
}
```

Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

Now lets create tests inf src/test/java

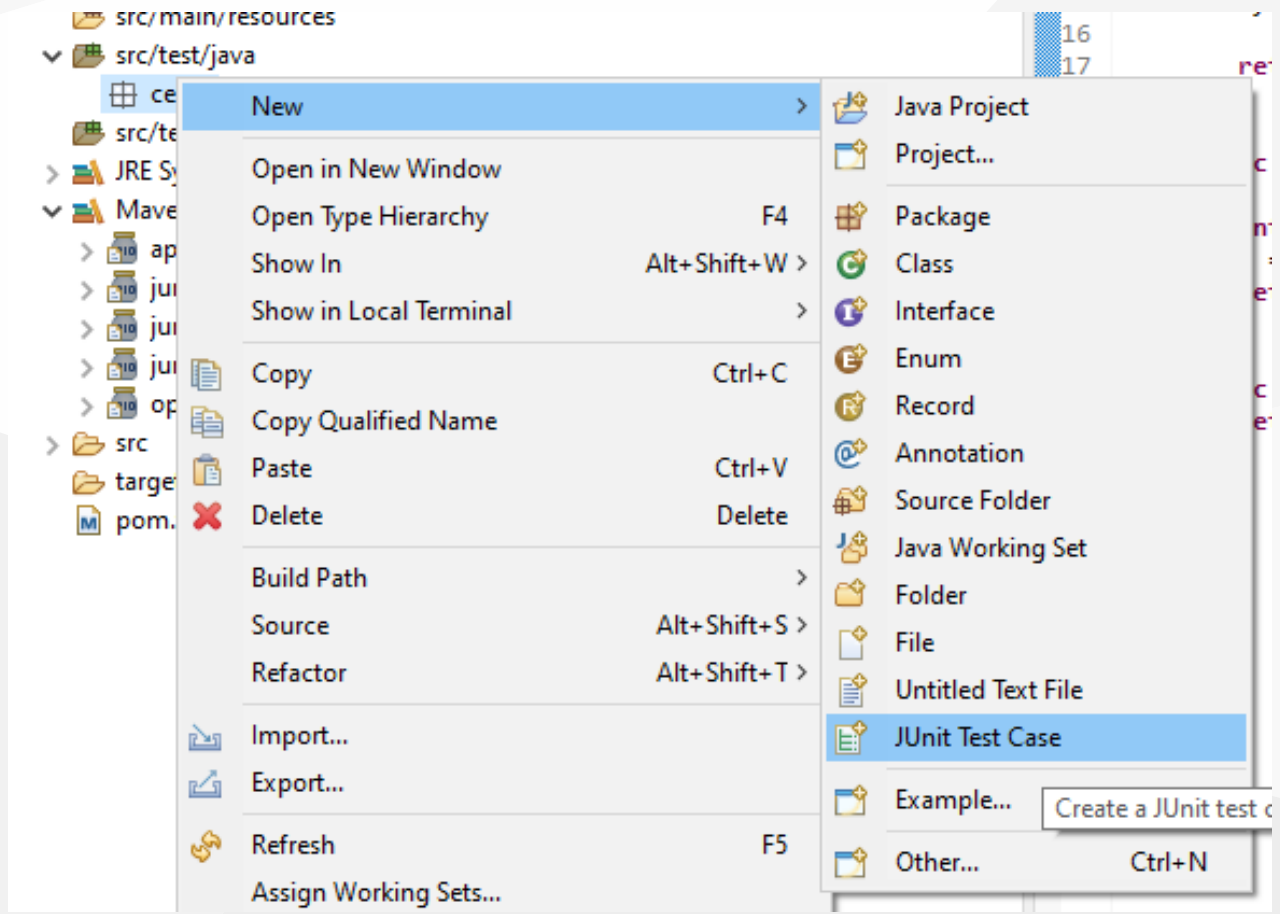


Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

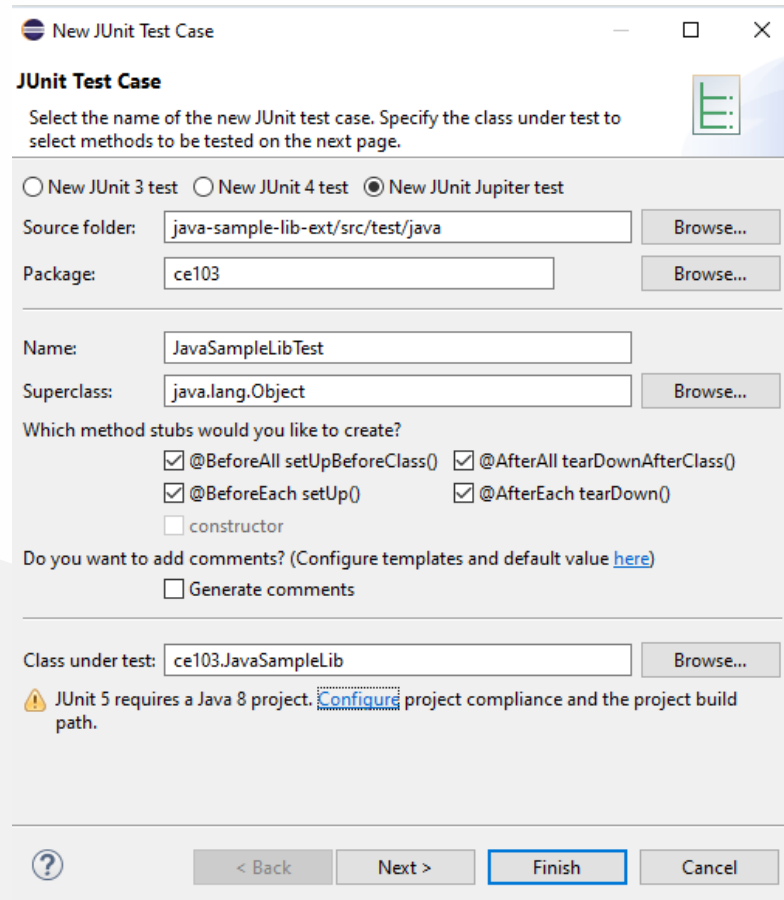


Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

create a JUnit Case



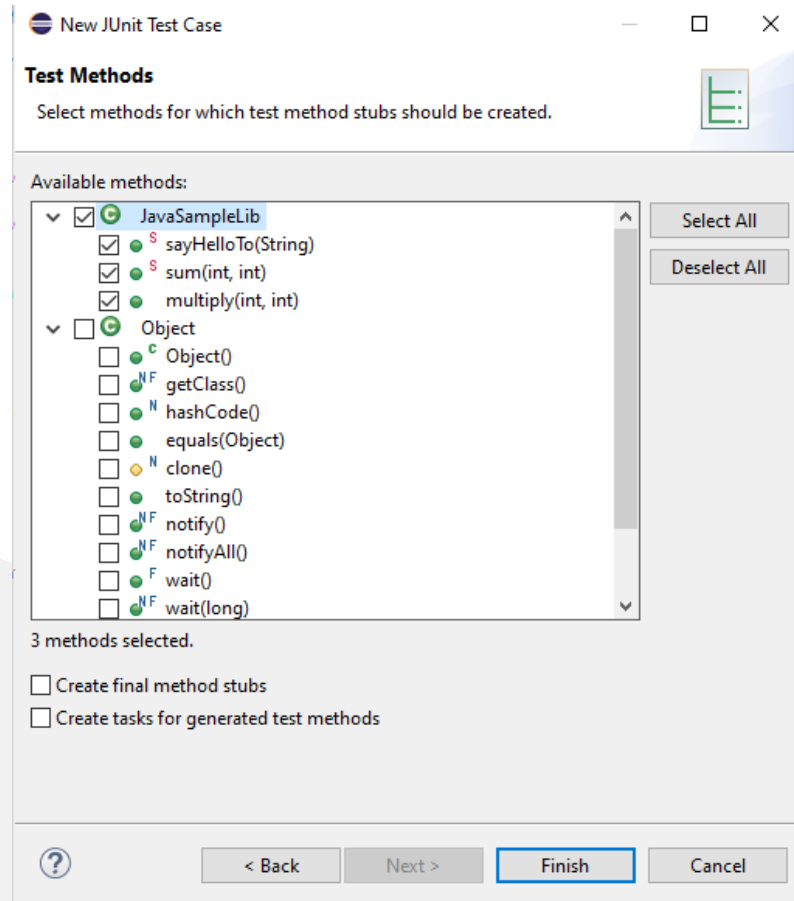
Eclipse IDE (JUnit4, JUnit5) + Java Unit Test



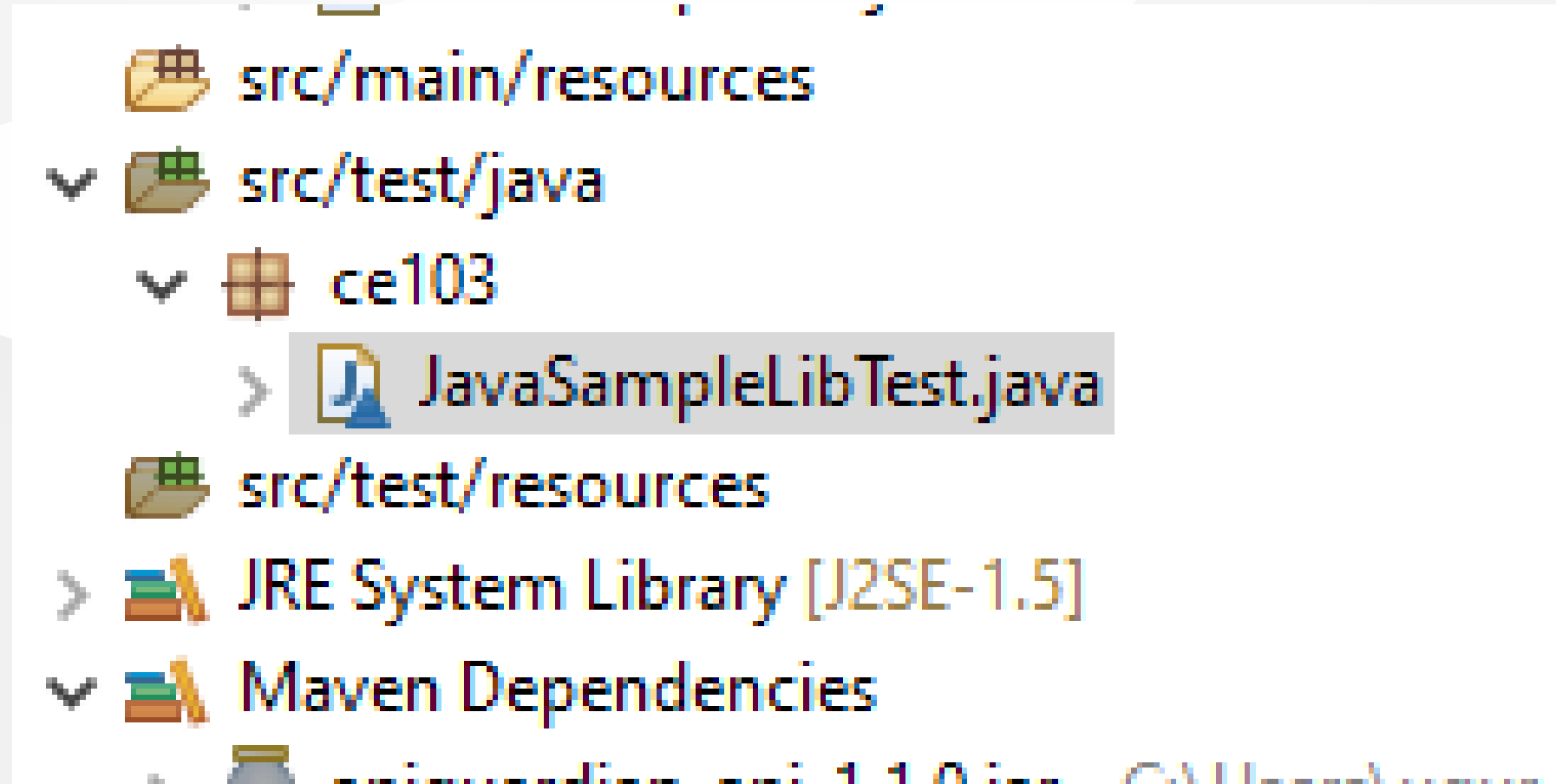
The screenshot shows the 'New JUnit Test Case' dialog box in Eclipse IDE. The dialog is titled 'New JUnit Test Case' and contains the following fields and options:

- 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.
- Test Type:** Radio buttons for 'New JUnit 3 test', 'New JUnit 4 test', and 'New JUnit Jupiter test'. 'New JUnit Jupiter test' is selected.
- Source folder:** Text field containing 'java-sample-lib-ext/src/test/java' and a 'Browse...' button.
- Package:** Text field containing 'ce103' and a 'Browse...' button.
- Name:** Text field containing 'JavaSampleLibTest'.
- Superclass:** Text field containing 'java.lang.Object' and a 'Browse...' button.
- 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:** Text field containing 'ce103.JavaSampleLib' and a 'Browse...' button.
- Warning:** A yellow warning icon followed by the text: 'JUnit 5 requires a Java 8 project. [Configure](#) project compliance and the project build path.'
- Navigation:** A question mark icon, '< Back' button, 'Next >' button, 'Finish' button (highlighted in blue), and 'Cancel' button.

Eclipse IDE (JUnit4, JUnit5) + Java Unit Test



Eclipse IDE (JUnit4, JUnit5) + Java Unit Test



Eclipse IDE (JUnit4, JUnit5) + Java Unit Test

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");
    }
}
```


Eclipse IDE (JUnit4, JUnit5) + Java Unit Test

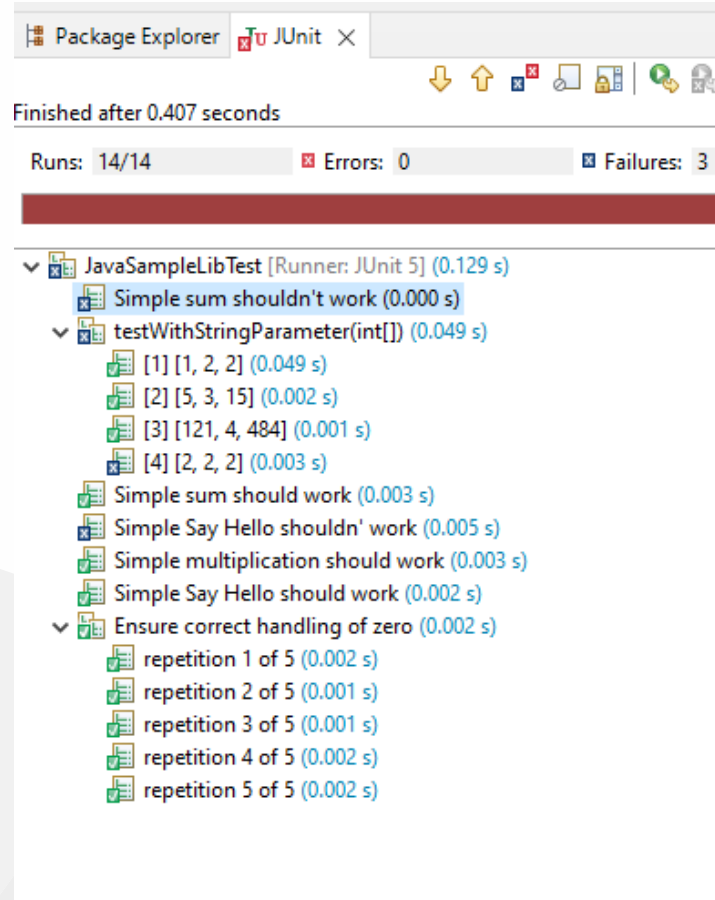
now lets copy tests from other projects

Convert source codes to java codes...

Eclipse IDE (JUnit4, JUnit5) + Java Unit Test

The screenshot displays the Eclipse IDE interface. On the left, the Project Explorer shows a project named 'java-sample-lib-ext' with a package 'ce103' containing 'JavaSampleLibTest.java'. A context menu is open over this file, listing various actions such as 'New', 'Open', 'Copy', 'Paste', 'Delete', 'Run As', and 'Debug As'. The 'Run As' option is highlighted. The main editor window shows the code for 'JavaSampleLibTest.java', which includes JUnit 5 annotations like '@BeforeAll', '@AfterAll', '@BeforeEach', and '@AfterEach', along with test methods like 'setUpBet', 'tearDown', 'testSayHelloTo', and 'testSumCorrect'. The bottom of the IDE shows the 'Run Configurations' dialog box, indicating the selected configuration for running the test.

Eclipse IDE (JUnit4, JUnit5) + Java Unit Test



Eclipse IDE (JUnit4, JUnit5) + Java Unit Test

The screenshot displays the Eclipse IDE interface. The main editor shows the source code for `JavaSampleLib.java` with the following content:

```

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
31 }
32
33
    
```

The JUnit test runner shows the following test results:

- JavaSampleLibTest [Runner: JUnit 5] (0.278 s)
 - Simple sum shouldn't work (0.017 s) ✗
 - testWithStringParameter(int[])
 - 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) ✓

The Failure Trace shows the following error:

```

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)
    
```

The Coverage window shows the following data:

Element	Coverage	Covered Instructio...	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

Eclipse IDE (JUnit4, JUnit5) + Java Unit Test

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

TDD (Test Driven Development)

- Test Driven Development (TDD)
 - https://en.wikipedia.org/wiki/Test-driven_development
- Acceptance Test Driven Development (ATDD)
 - https://en.wikipedia.org/wiki/Acceptance_test-driven_development
- Also check out
 - https://en.wikipedia.org/wiki/Kent_Beck
- Extreme Programming
 - https://en.wikipedia.org/wiki/Extreme_programming
- Software Design Patterns
 - https://en.wikipedia.org/wiki/Software_design_pattern

Test and Deployment Automation Management

There are several Continuous-Integration services online as follow;

- Travis-CI
- Appveyor
- Jenkins
- CircleCI
- GitLab
- Pantheon
- GitHub
- Bitrise
- Flosum
- Buddy
- Semaphore

Test and Deployment Automation Management

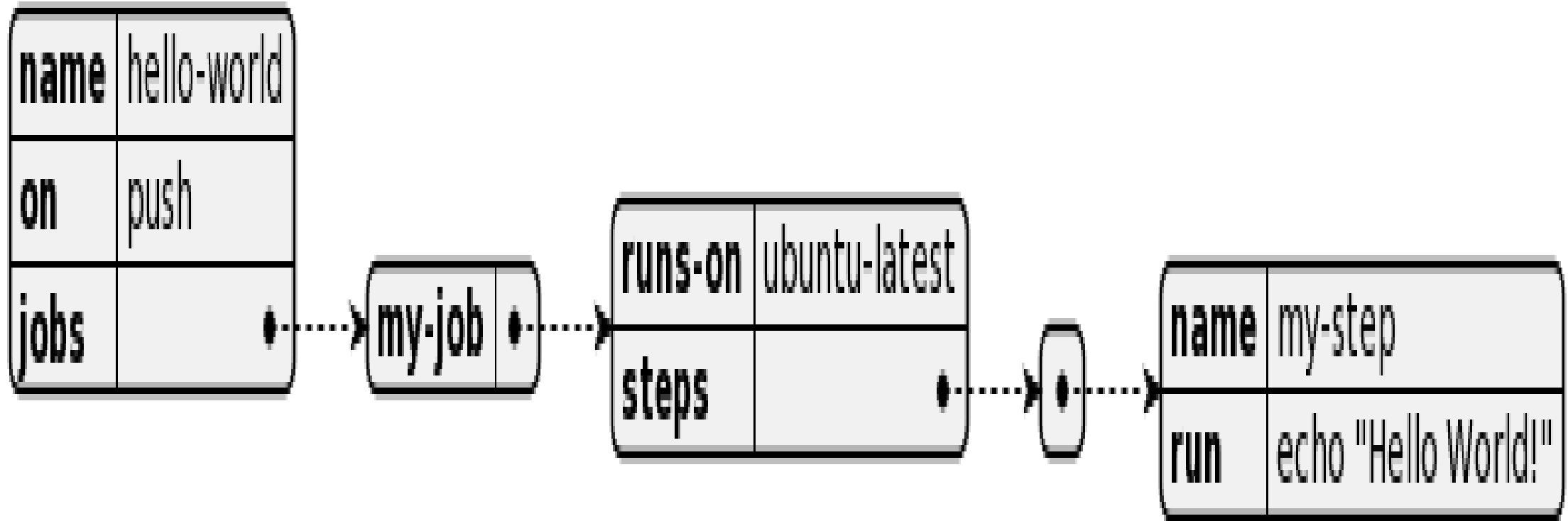
- Github provides Github Actions for Releases and Tests
- Jenkins has on promise solutions private development

Test and Deployment Automation Management

- GitHub Actions provide several actions and marketplace
 - <https://github.com/marketplace/actions/build-c-project>
- Also, we Can Provide Our Custom Actions

```
name: hello-world
on: push
jobs:
  my-job:
    runs-on: ubuntu-latest
    steps:
      - name: my-step
        run: echo "Hello World!"
```

Test and Deployment Automation Management

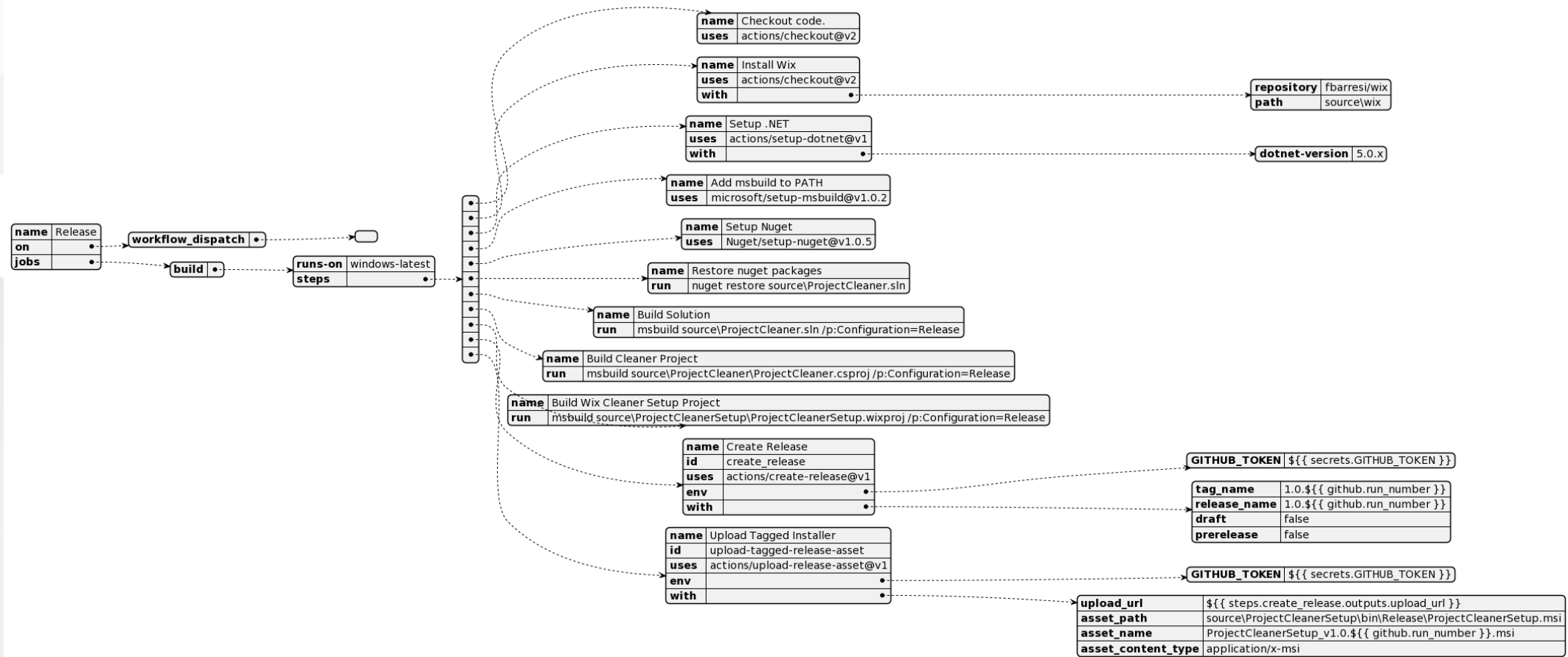


Test and Deployment Automation Management

- <https://github.com/ucoruh/project-cleaner/blob/main/.github/workflows/dotnet-desktop.yml>

This action build c# application and generates setup manually.

- Also there is a nice web example
 - <https://dev.to/geromegrignon/github-actions-full-ci-cd-javascript-workflow-39om>



References

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

End – Of – Week – 4