CE103 Algorithms and Programming I

Introduction to Code Reusability and Automate Testing

Author: Asst. Prof. Dr. Uğur CORUH

Contents

List of Figures

List of Tables

0.1 CE103 Algorithms and Programming I

0.1.1 Week-4

0.1.1.1 Introduction to Code Reusability and Automated Testing

Download DOC[1], SLIDE[2], PPTX[3]

0.1.1.2 Outline

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

0.2 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.

0.3 Selected Development Environment

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

0.4 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

---

[1] ce103-week-4-test.en.md_doc.pdf
[2] ce103-week-4-test.en.md_slide.pdf
[3] ce103-week-4-test.en.md_slide.pptx

0.5 Shared Library Development

0.5.1 C Programming (Static Library)

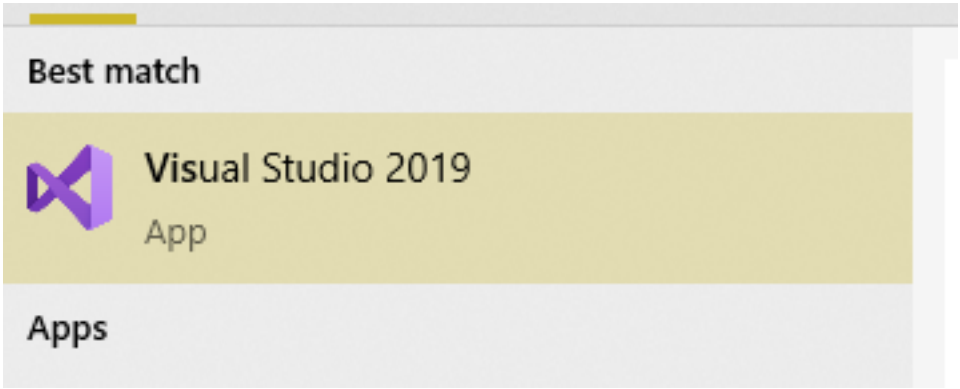
0.5.1.1 Visual Studio Community Edition

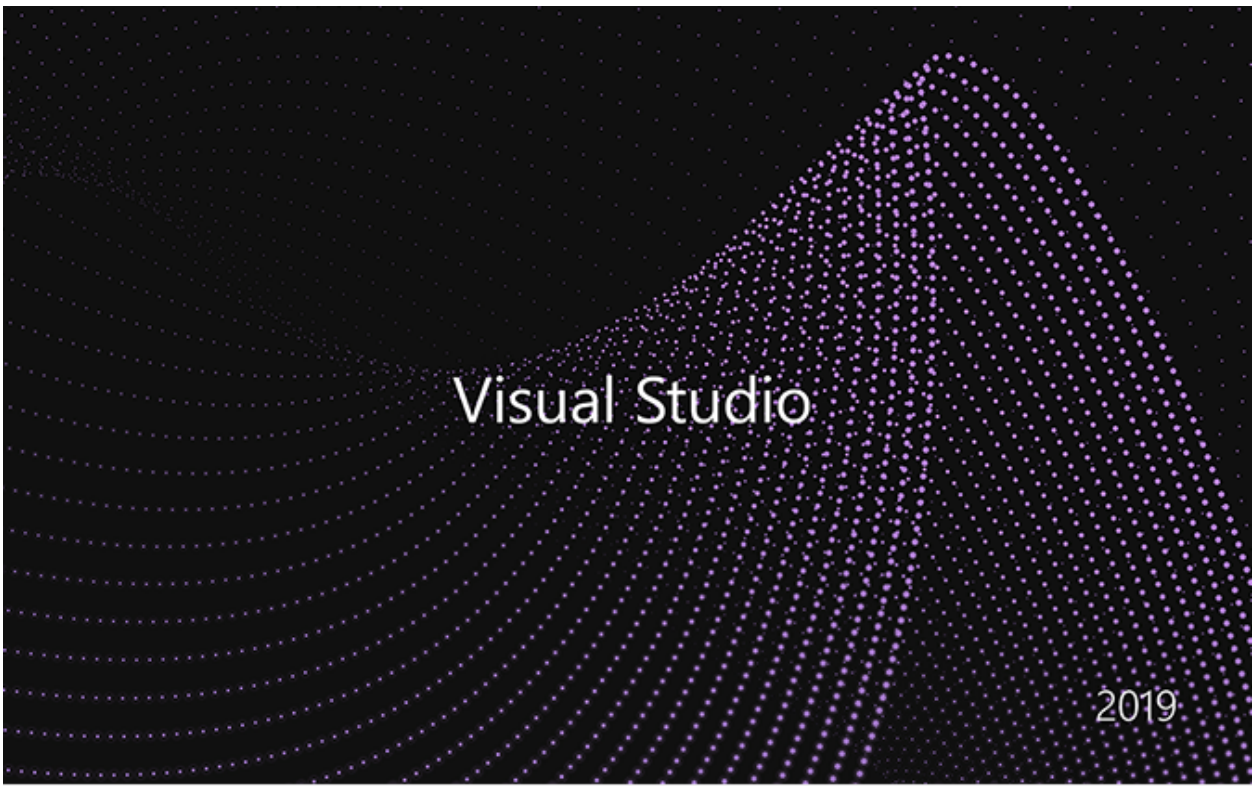
0.6 Shared Library Development - (VS C Static Library)-1

- 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[4]

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

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

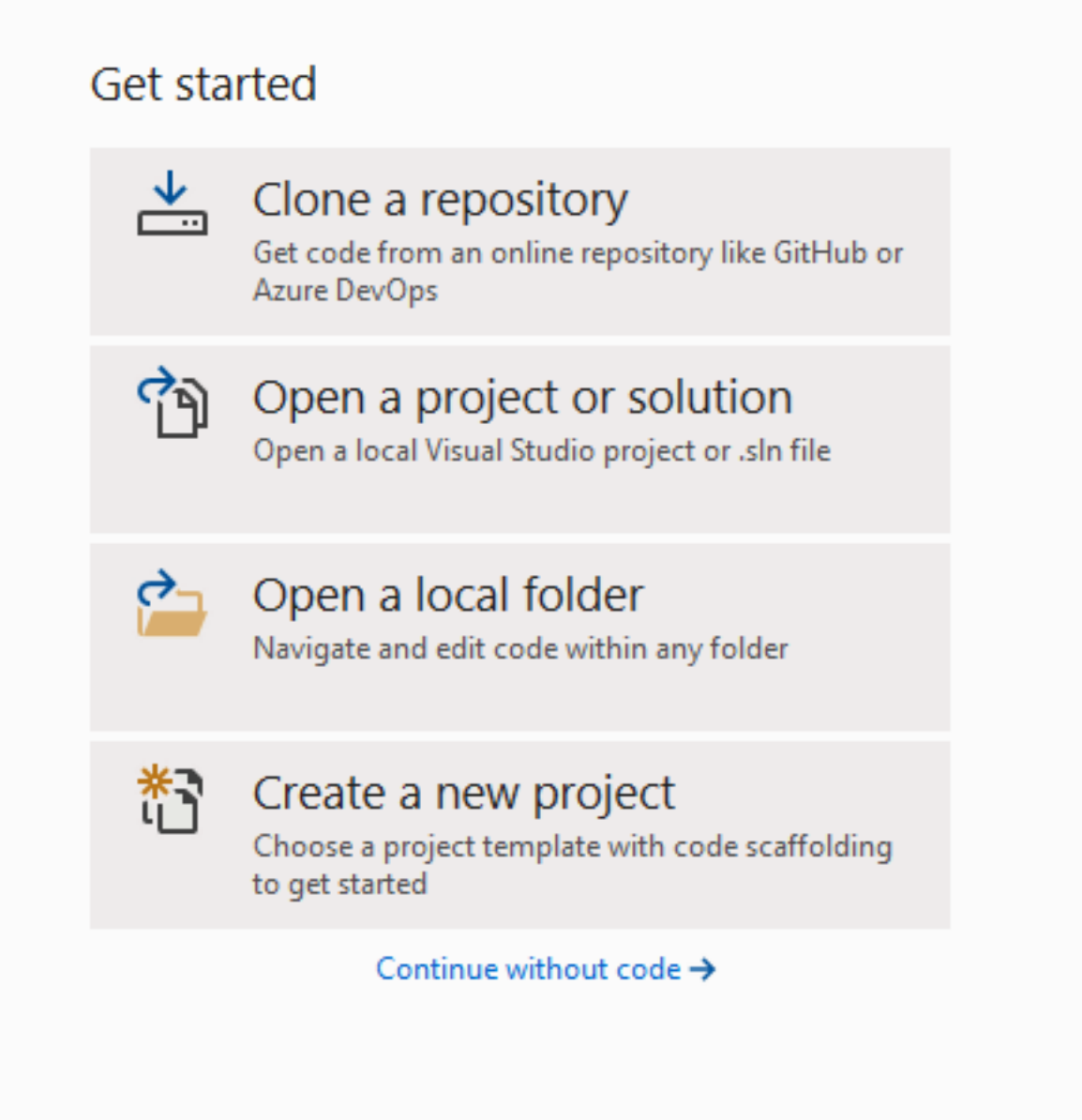




[4] https://visualstudio.microsoft.com/tr/vs/community/

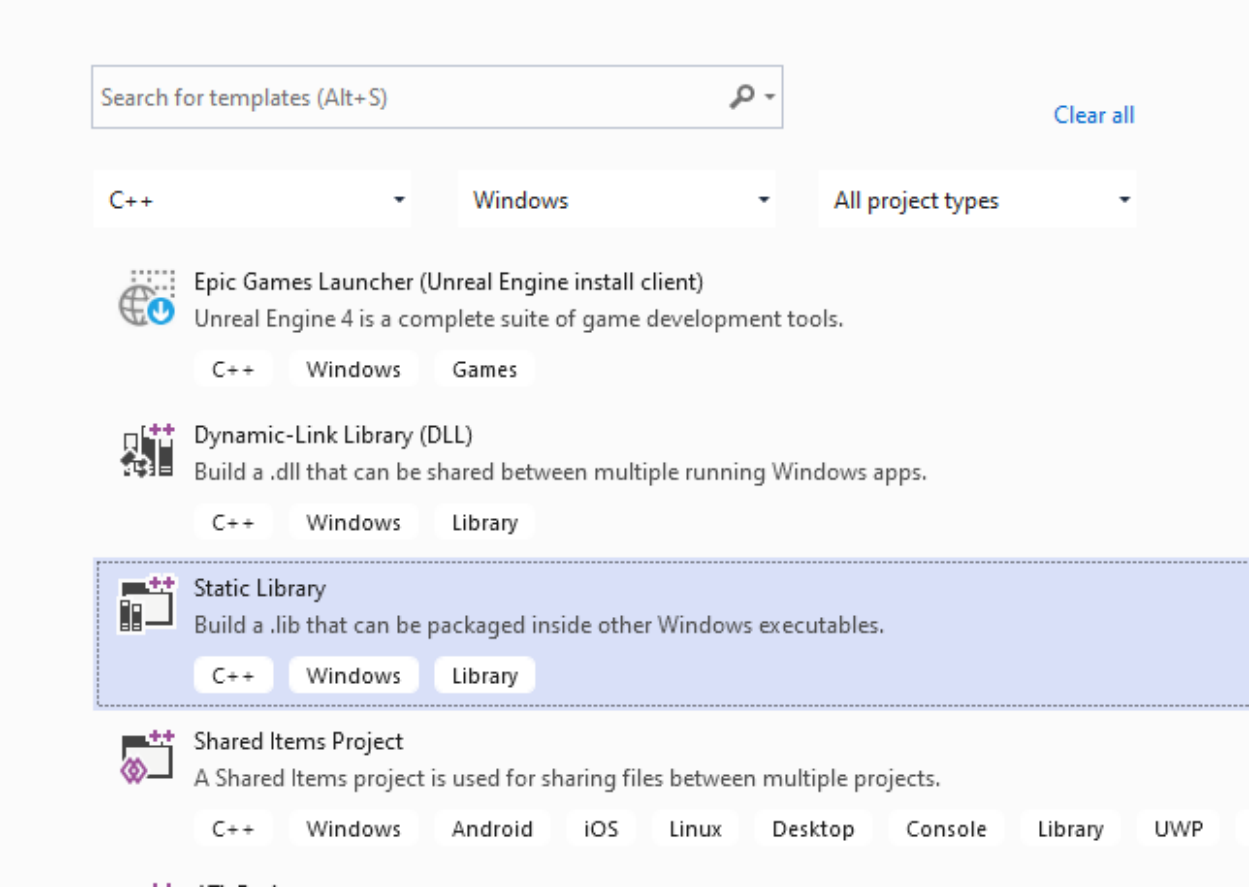
0.8 Shared Library Development - (VS C Static Library)-3

- Select create a new project



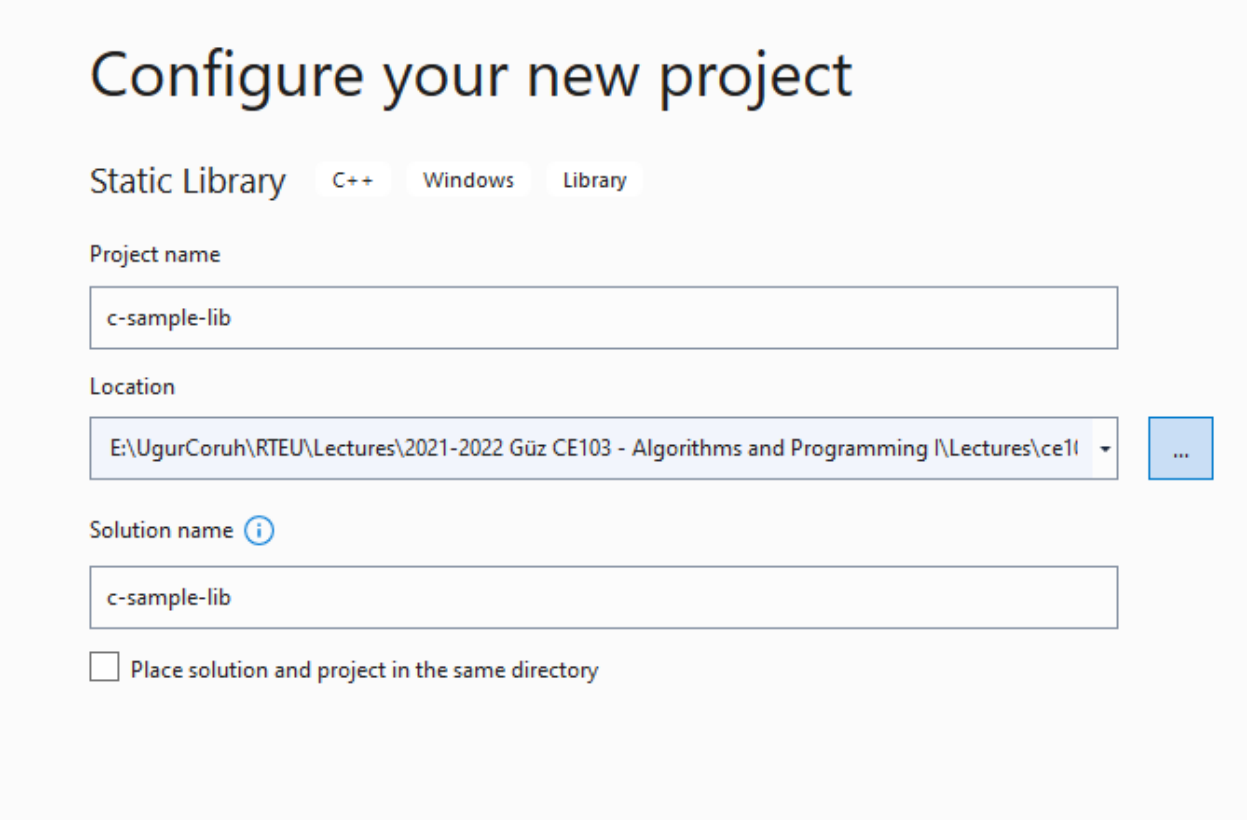
0.9 Shared Library Development - (VS C Static Library)-4

- Select C++ static library from the project list



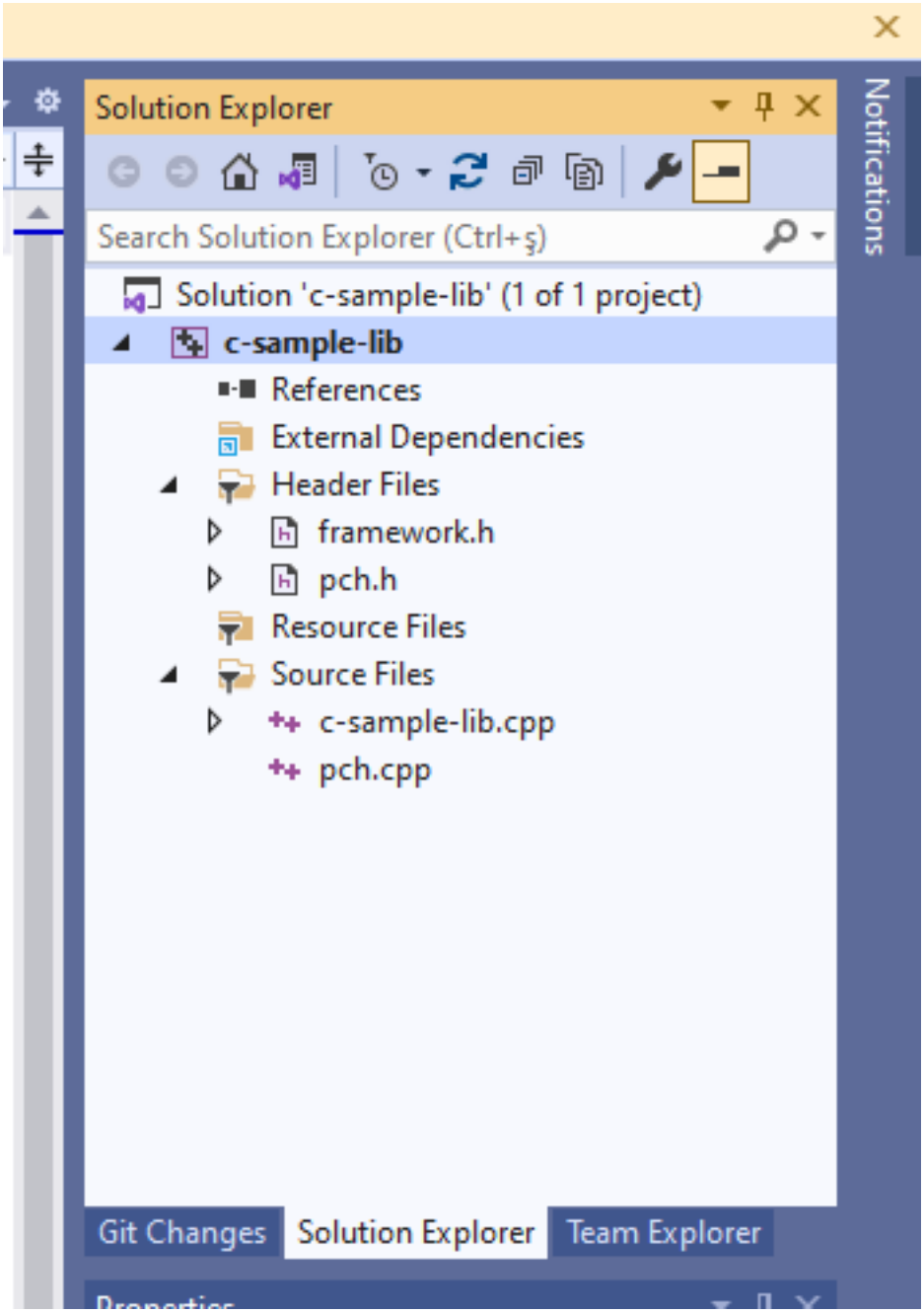
0.10 Shared Library Development - (VS C Static Library)-5

- Give static library project name



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

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



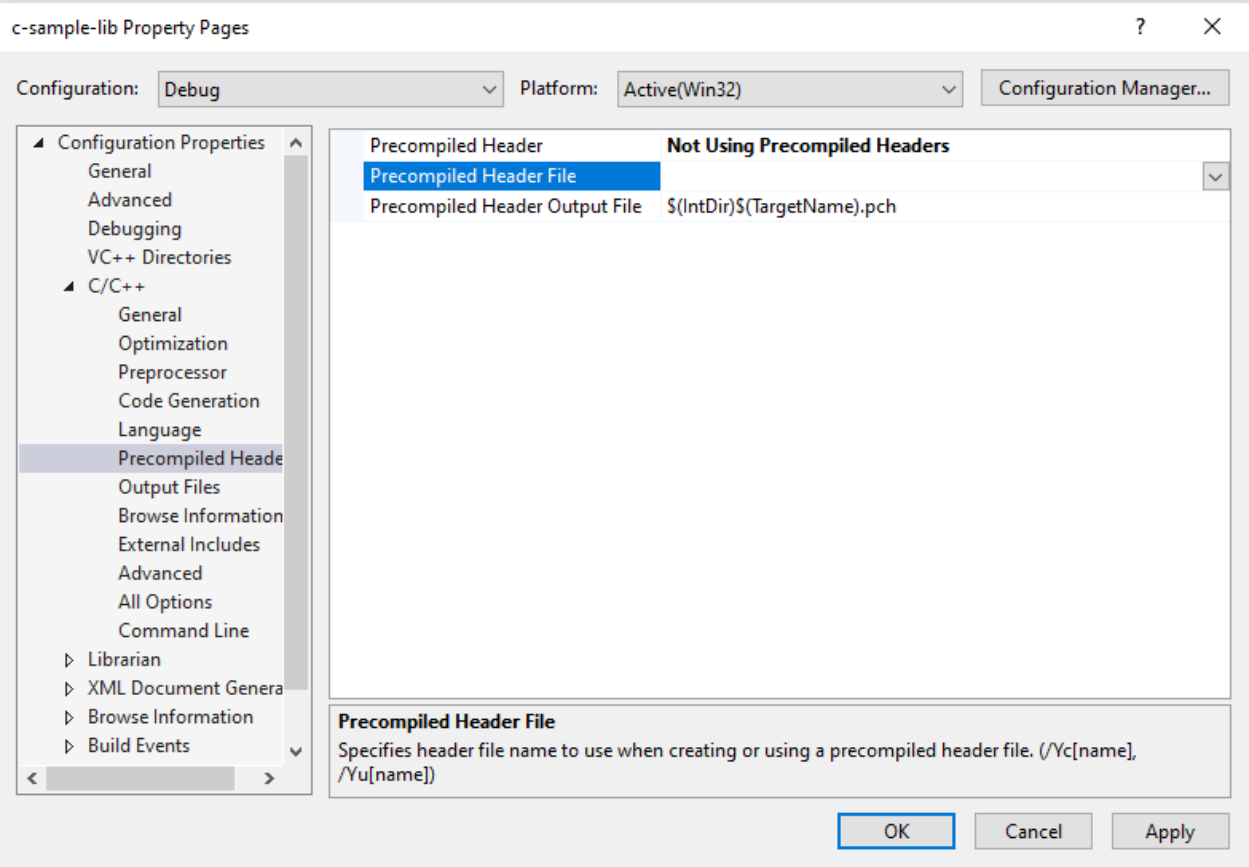
0.12 Shared Library Development - (VS C Static Library)-7

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

```
void fncsamplelib(){

}
```

0.13 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 Precomplied Headers”, also you can delete precomplied Header File.



0.14 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;
}
```

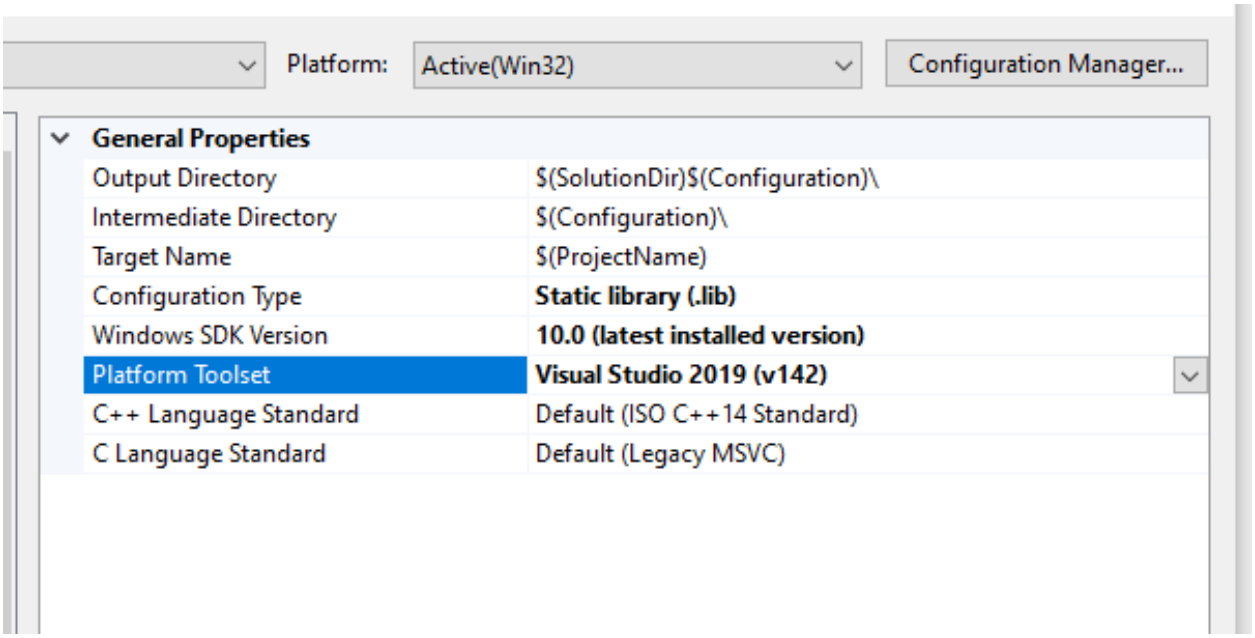
0.15 Shared Library Development - (VS C Static Library)-10

- 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);
```

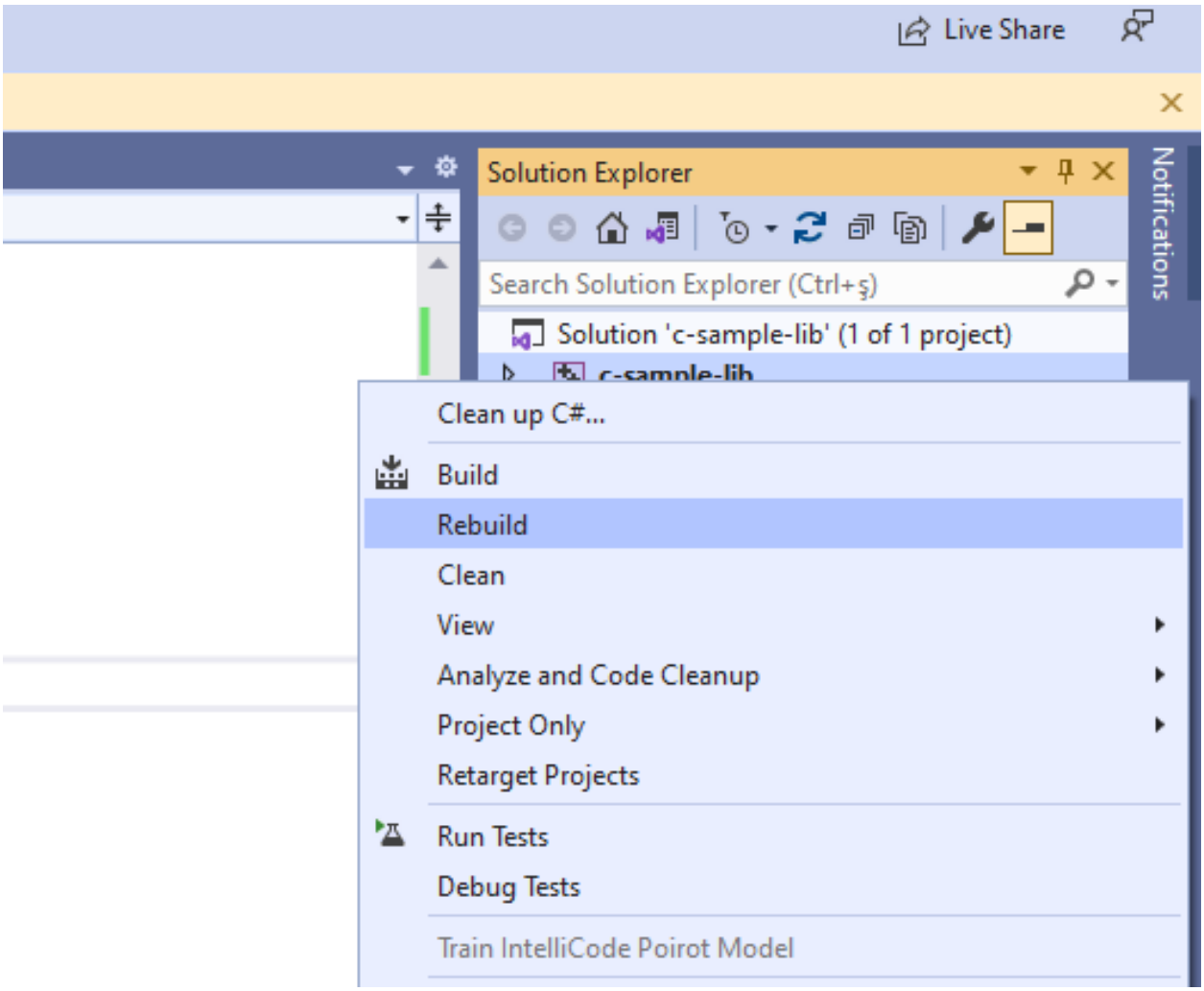
0.16 Shared Library Development - (VS C Static Library)-11

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



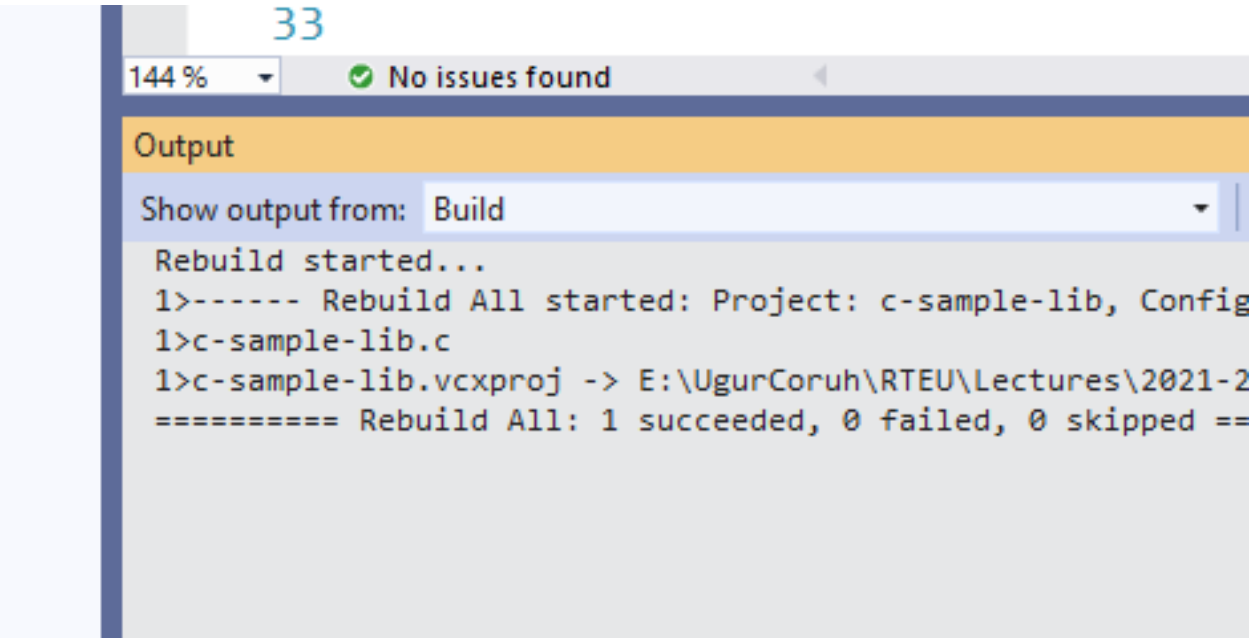
0.17 Shared Library Development - (VS C Static Library)-12

- Now we can compile our library



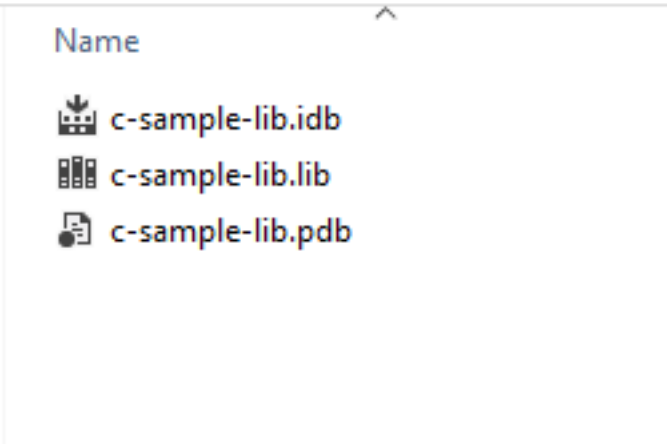
0.18 Shared Library Development - (VS C Static Library)-13

- You can follow operation from the output window



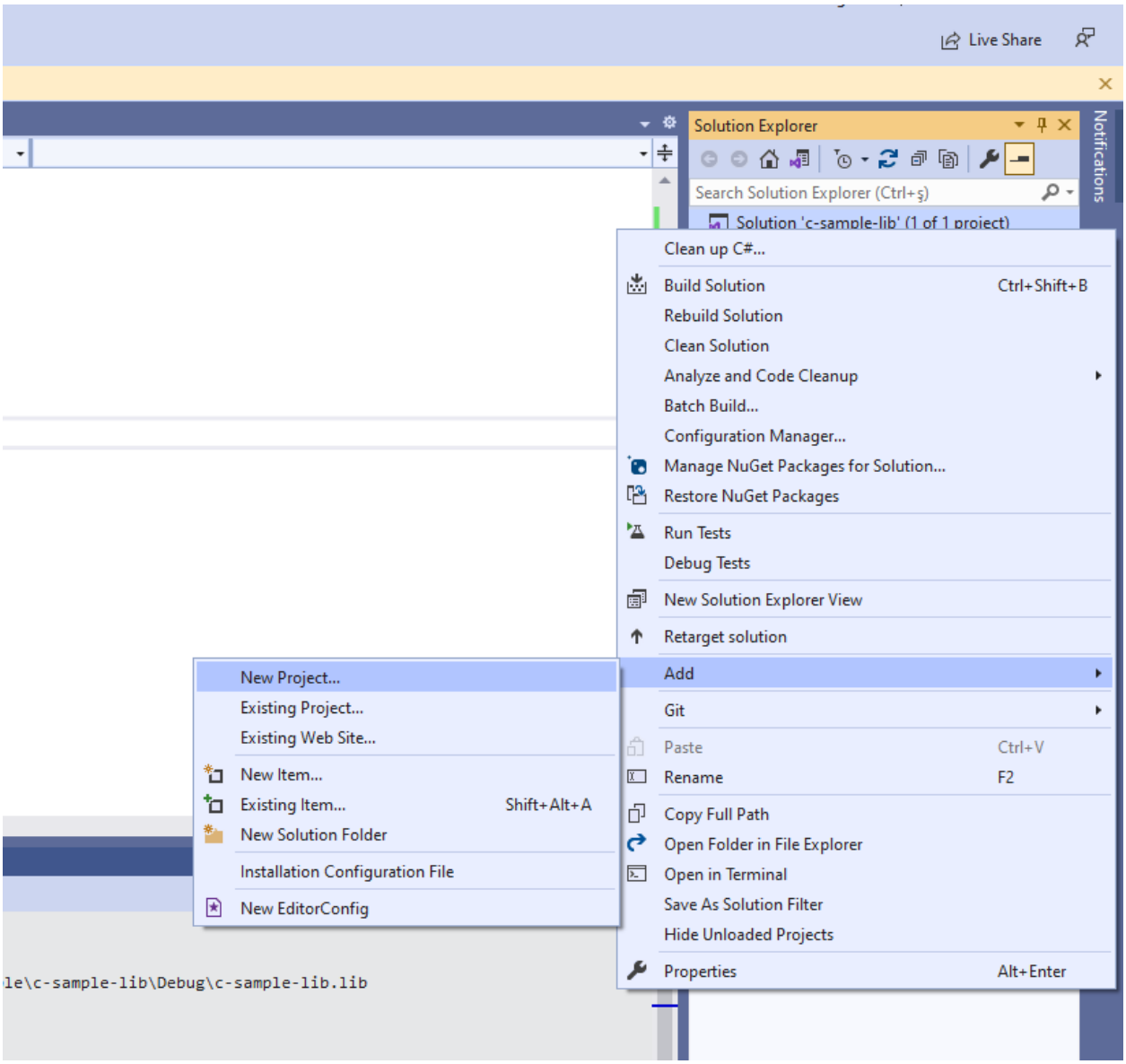
0.19 Shared Library Development - (VS C Static Library)-14

- In the debug folder, we will see our output



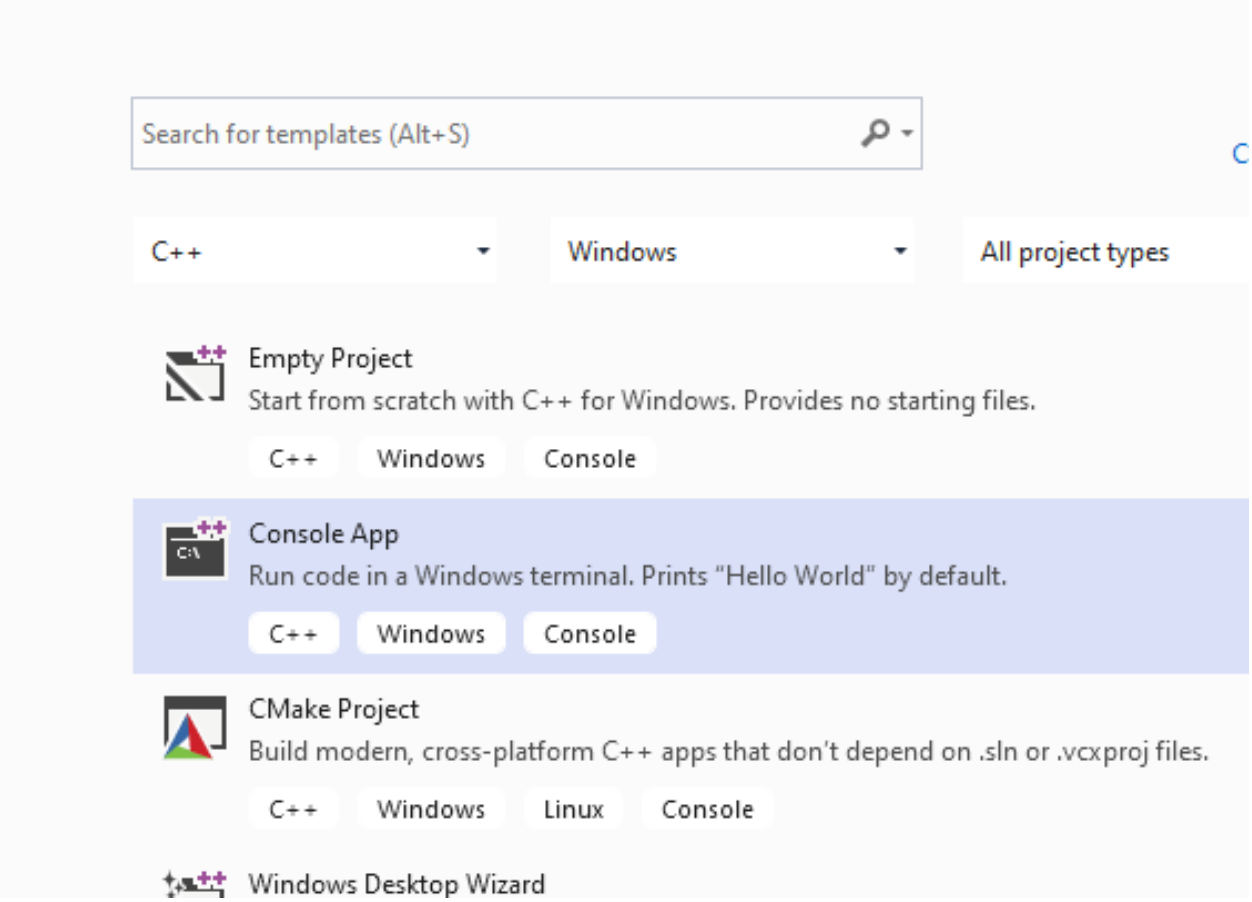
0.20 Shared Library Development - (VS C Static Library)-15

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



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

select C++ Windows Console Application from list



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

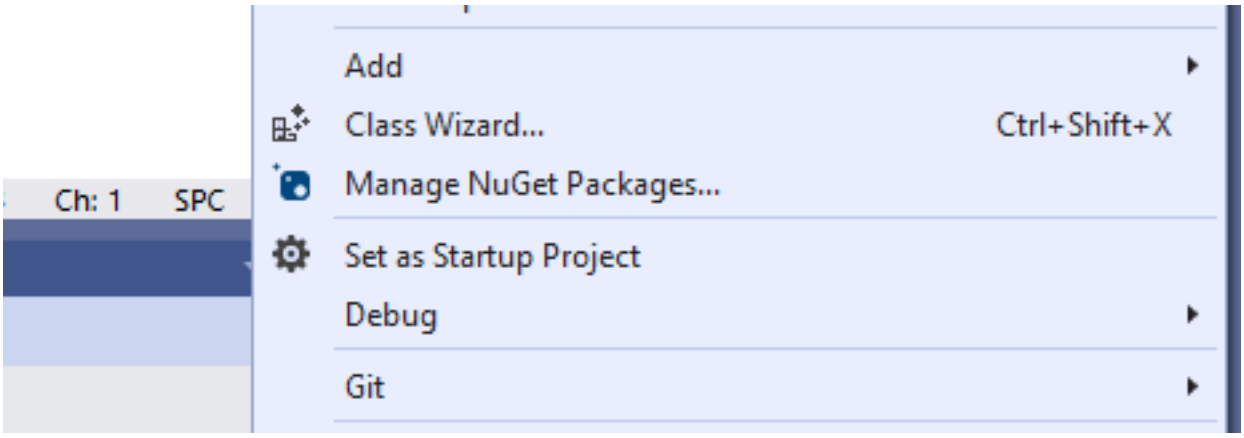
- C++ Console Application Selection will generate a C++ console project we can change extension to C to compile our application as C application.

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

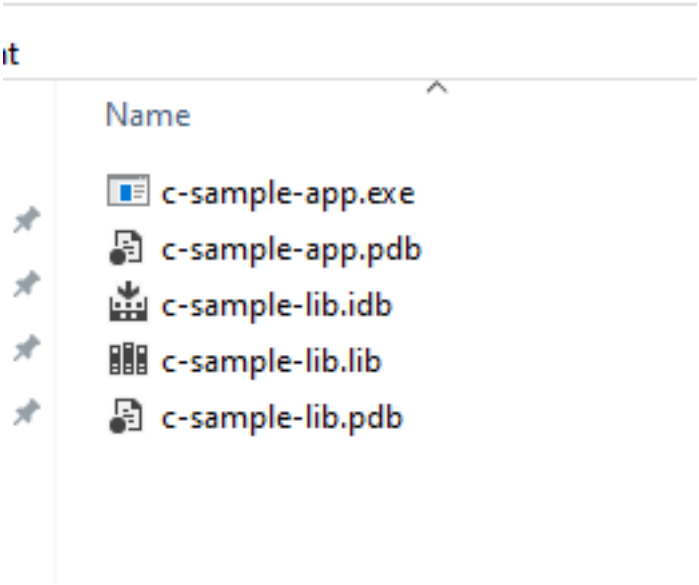
```
#include <stdio.h>

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

0.23 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"`

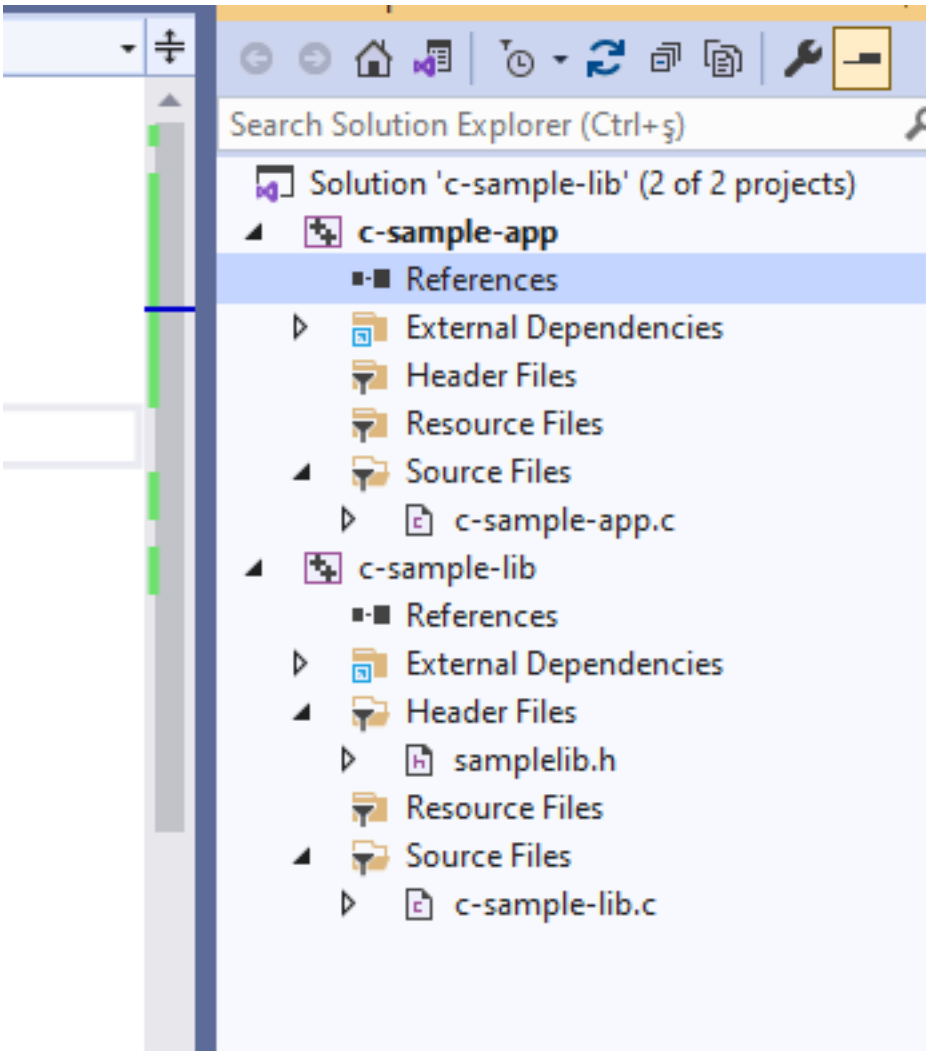
0.24 Shared Library Development - (VS C Static Library)-19

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

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

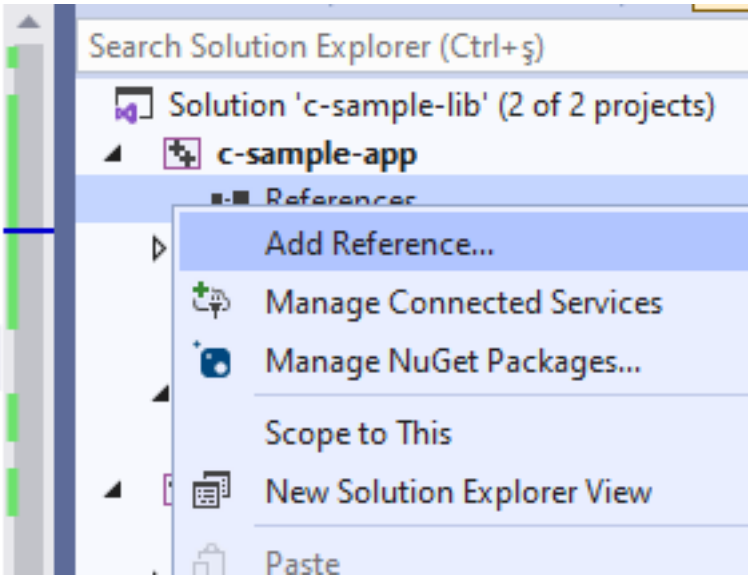
First option

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



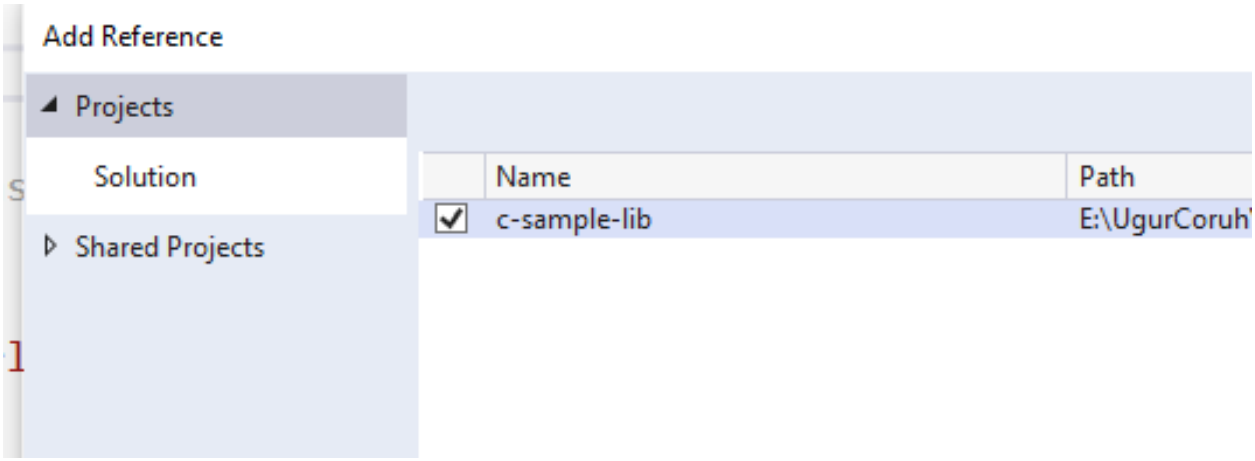
0.26 Shared Library Development - (VS C Static Library)-21

- Select Add Reference



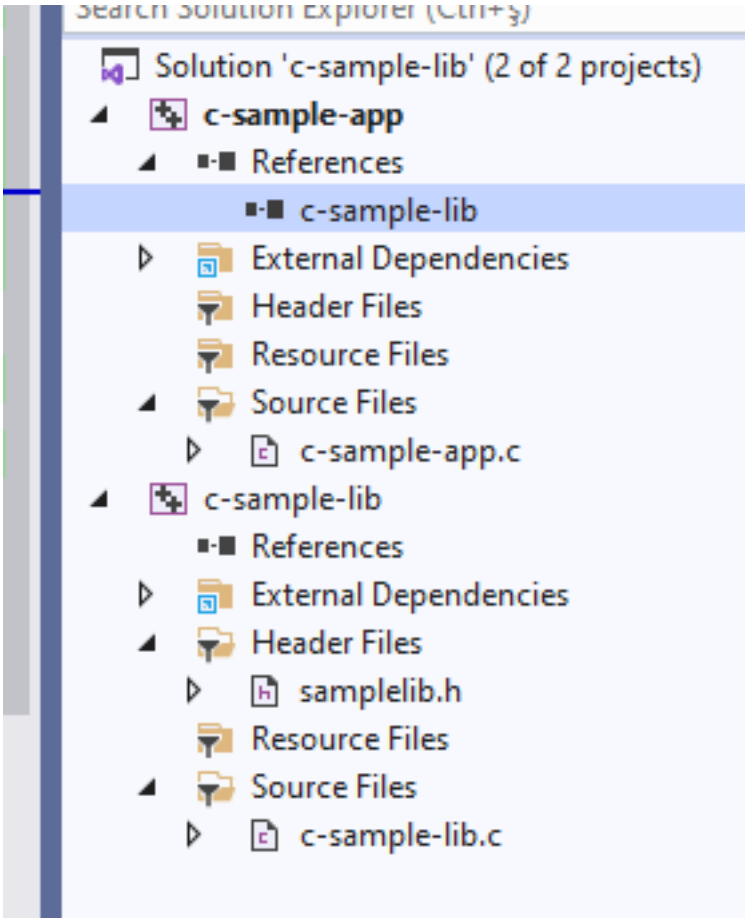
0.27 Shared Library Development - (VS C Static Library)-22

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



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

You can check added reference from references section



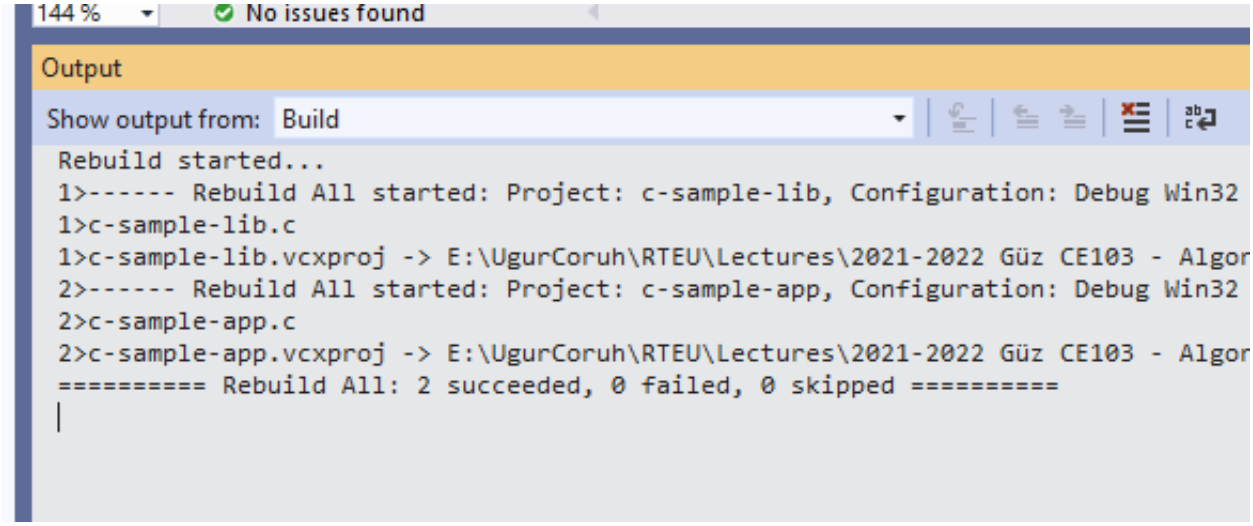
0.29 Shared Library Development - (VS C Static Library)-24

- 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");
}
```

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

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



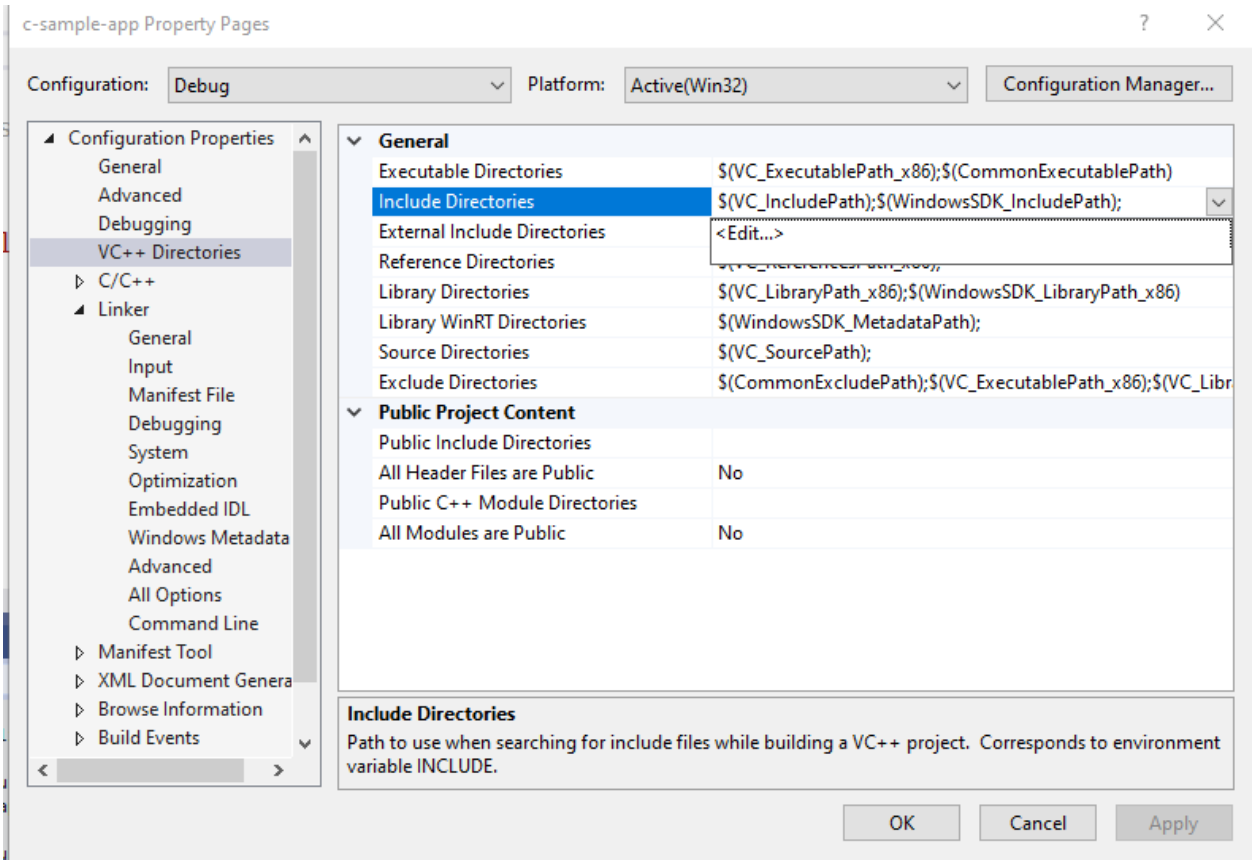
0.31 Shared Library Development - (VS C Static Library)-26

- Also we can only write header name

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

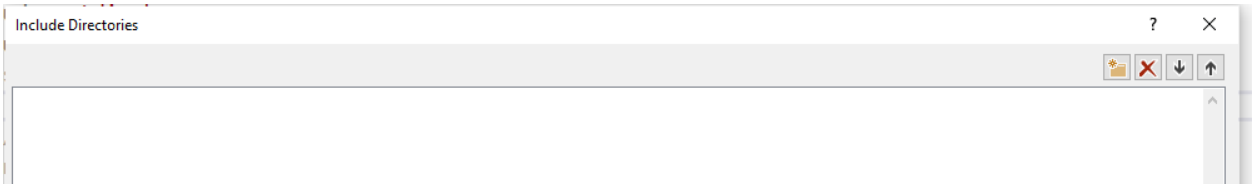
0.32 Shared Library Development - (VS C Static Library)-27

- For this option, we need to configure include directories

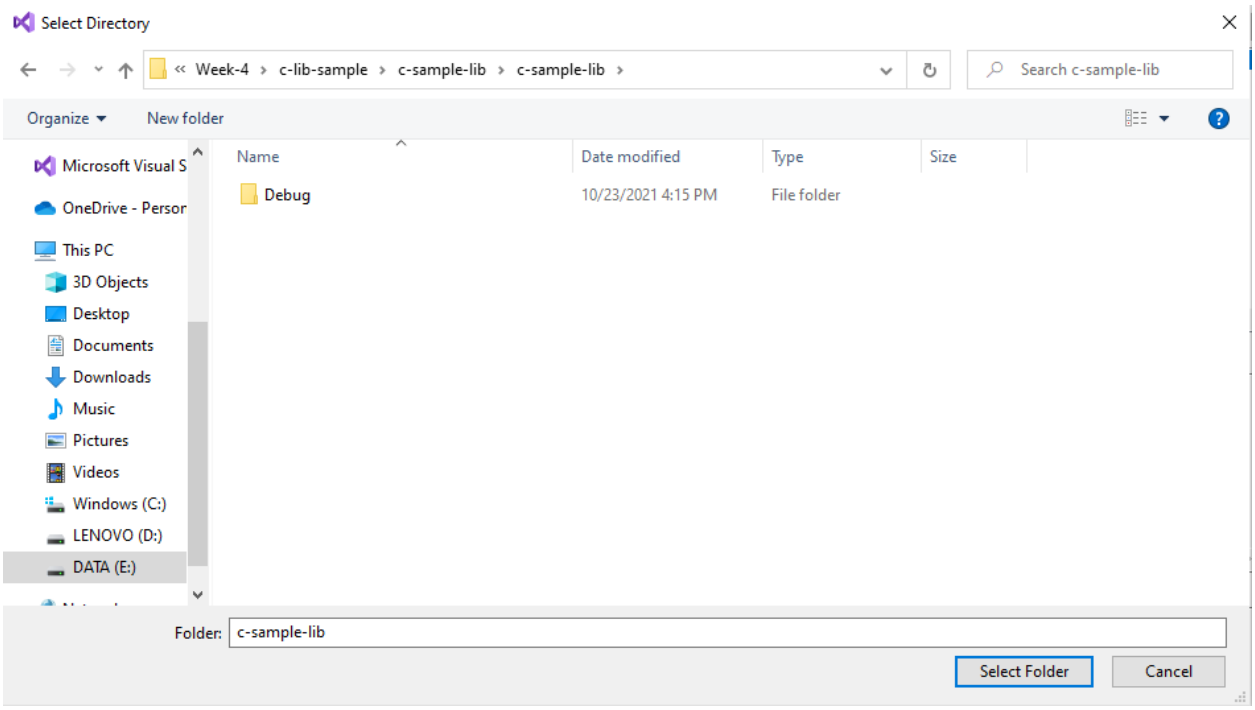


0.33 Shared Library Development - (VS C Static Library)-28

select c-sample-lib header file location

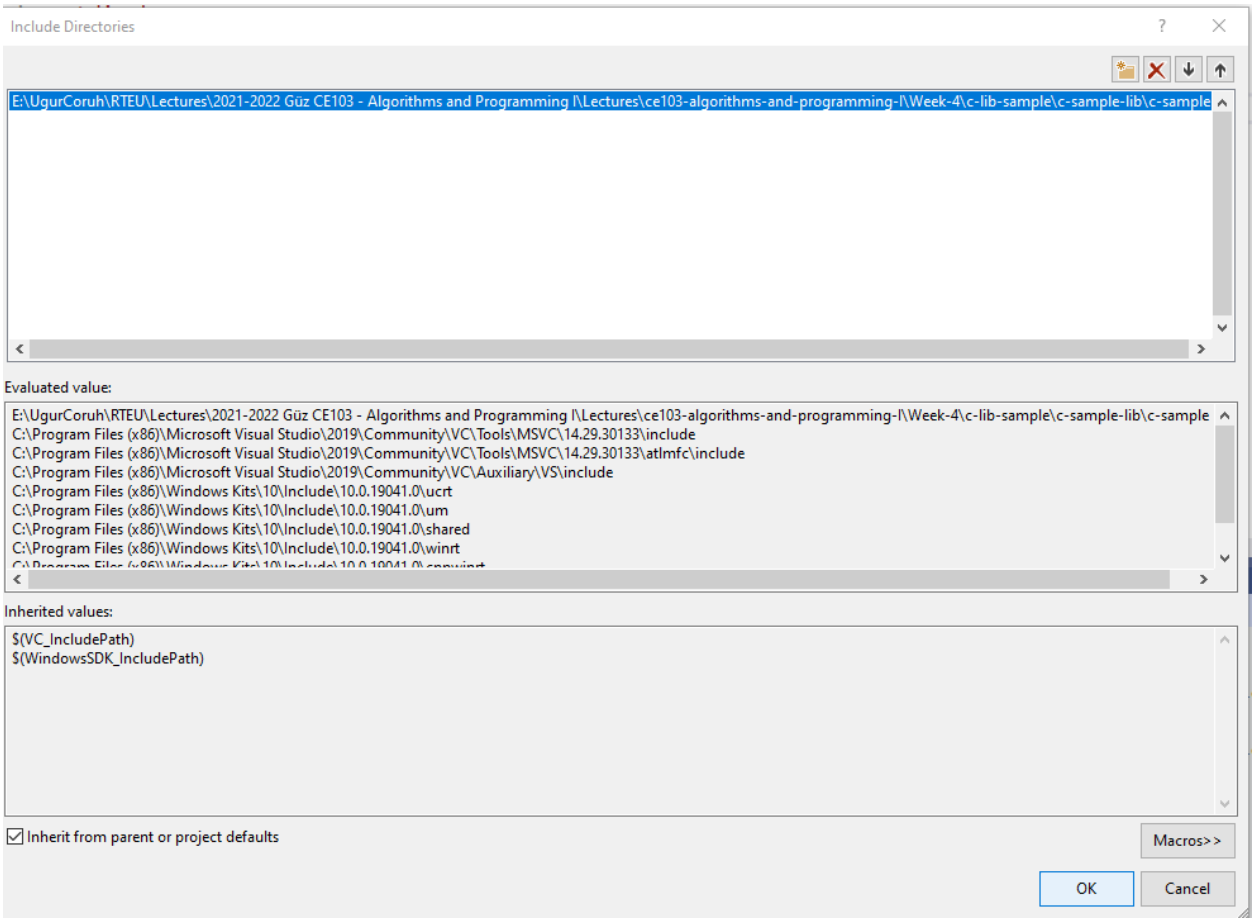


browse for folder



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

your full path will be added to your configuration



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

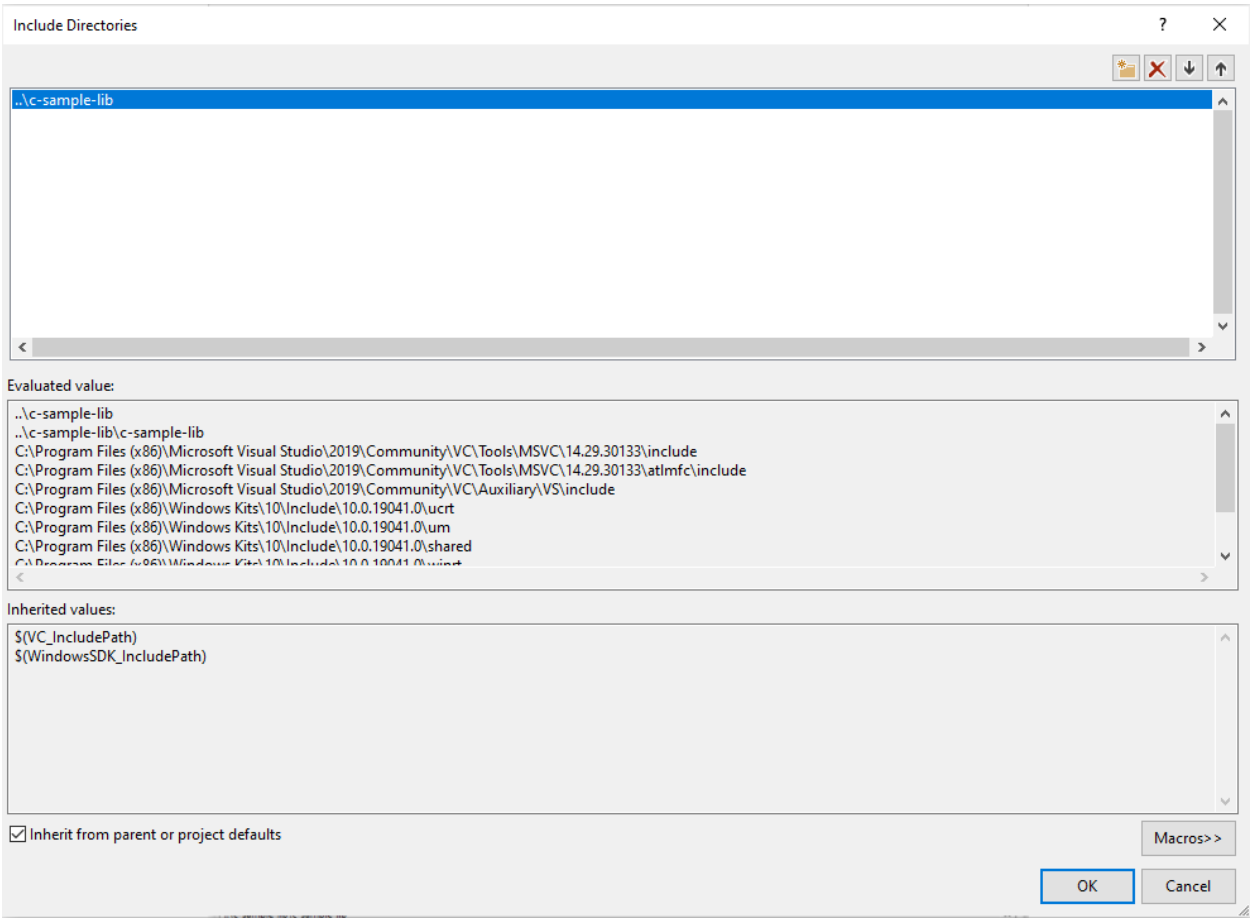
if you add header file paths to your configuration you can use header files by name in your source code

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

0.36 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`



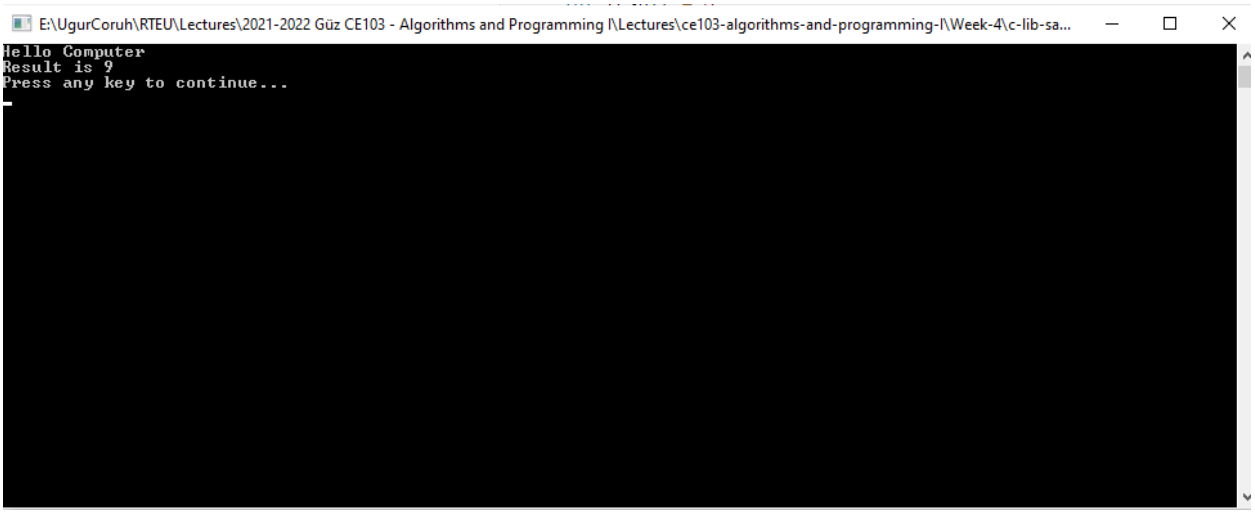
0.37 Shared Library Development - (VS C Static Library)-32

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

```c
#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;
}
```

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

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



0.39 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

---

0.40 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

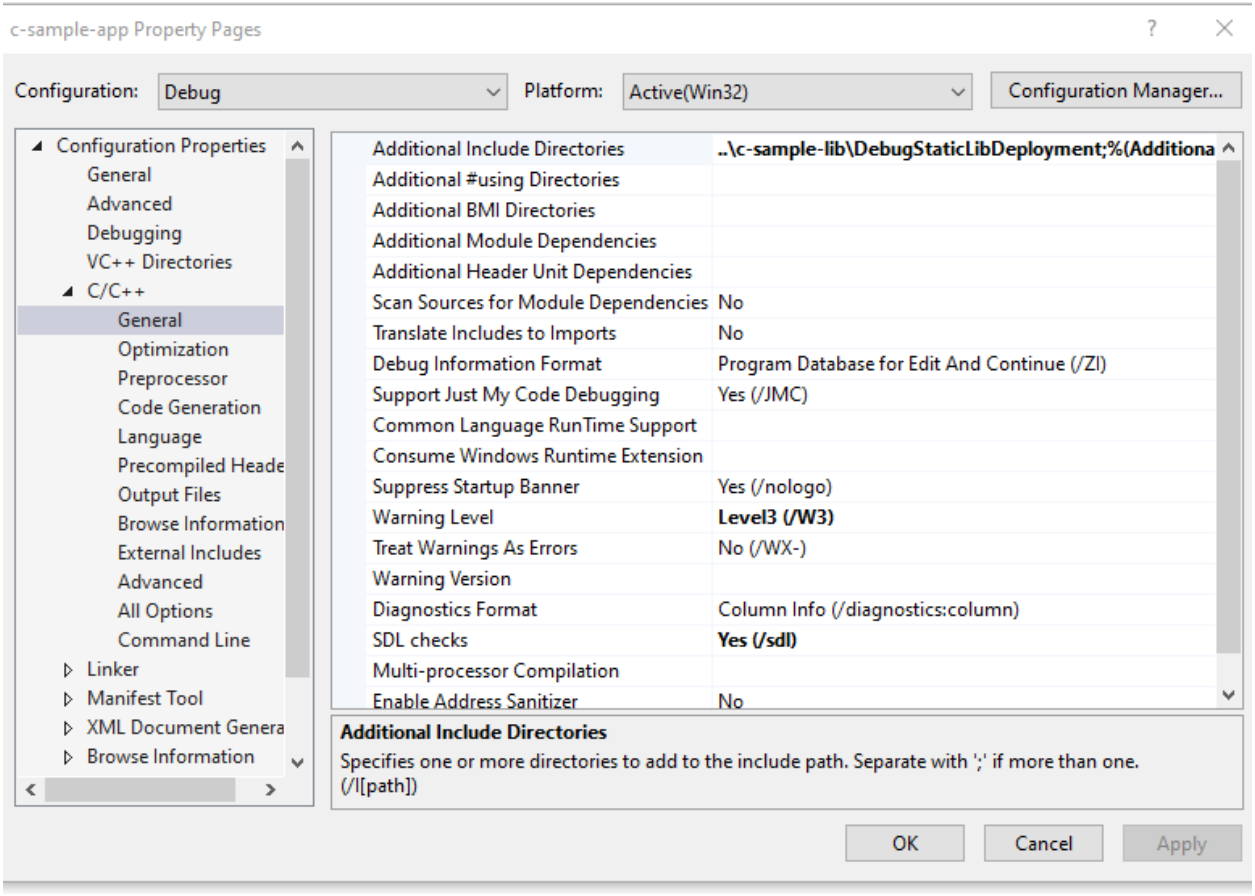
There is a bug in configurations and relative path not finding headers so for this reason we will set full path but this is not a good practice for team working

---

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

Not Working Solution

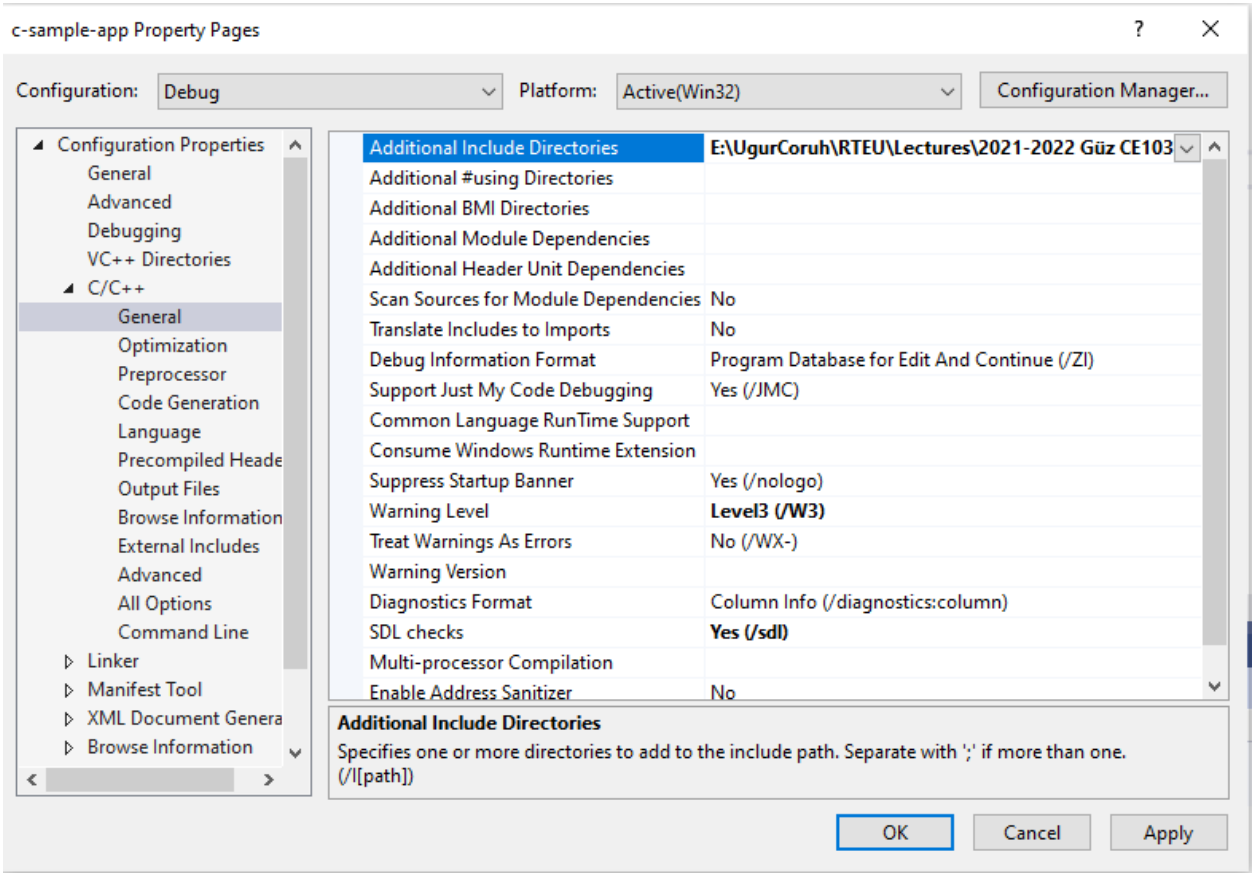
`..\c-sample-lib\DebugStaticLibDeployment`



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

Working Solution

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



0.43 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

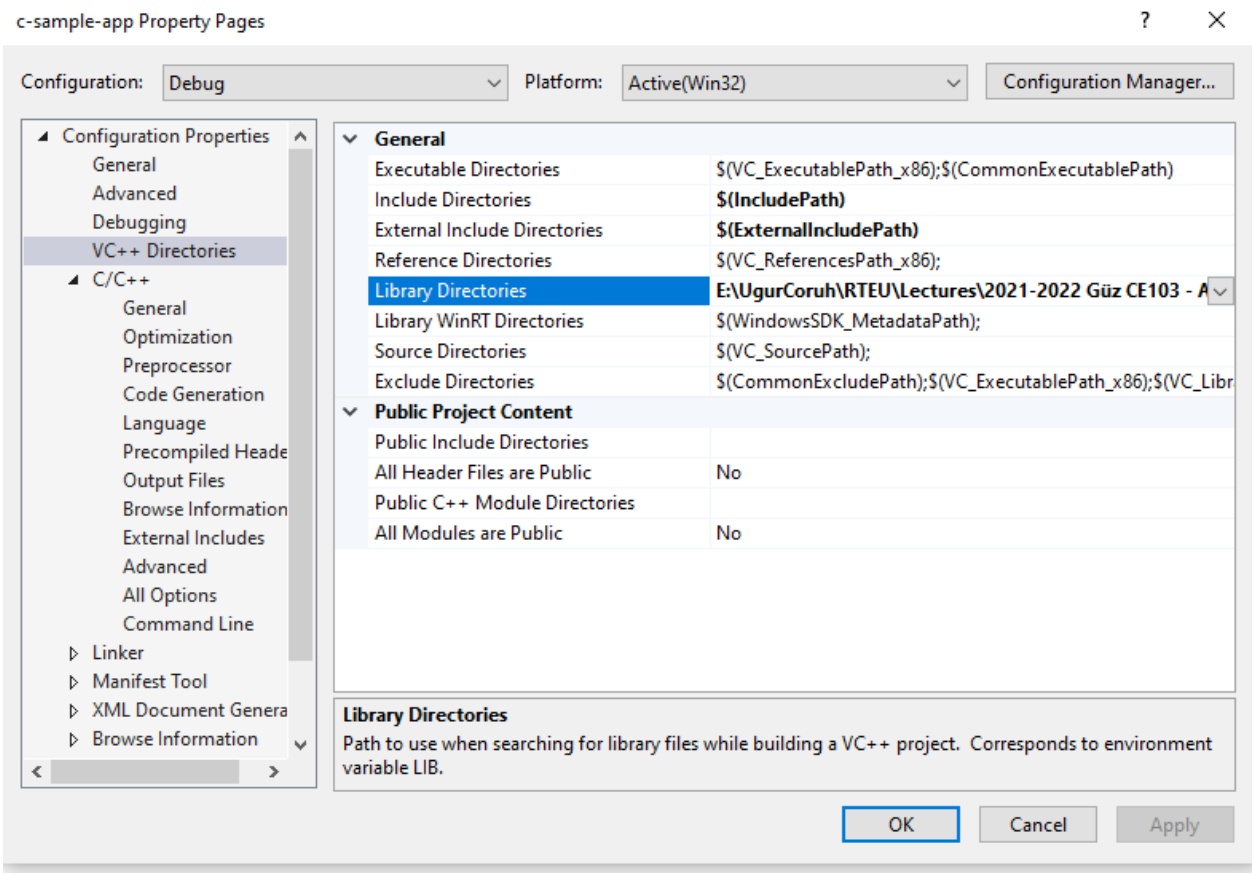
Here is the same issue if we use relative path it doesn't work we need to set full path for library folder

---

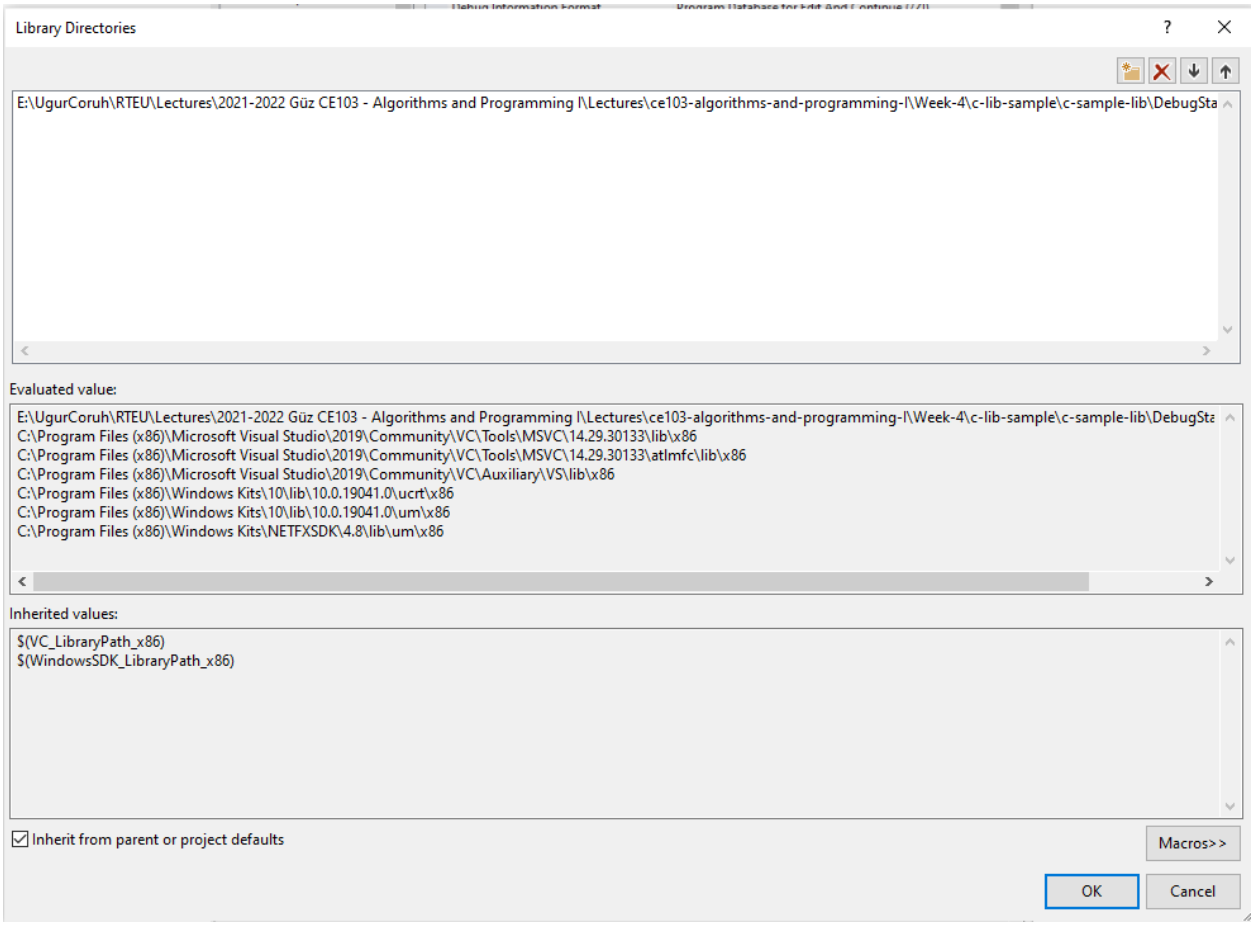
0.44 Shared Library Development - (VS C Static Library)-39

Working Solution

`E:\...\c-lib-sample\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++
General
Optimization
Preprocessor
Code Generation
Language
Precompiled Heade
Output Files
Browse Information
External Includes
Advanced
All Options
Command Line
Linker
Manifest Tool
XML Document Genera
Browse Information
General
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

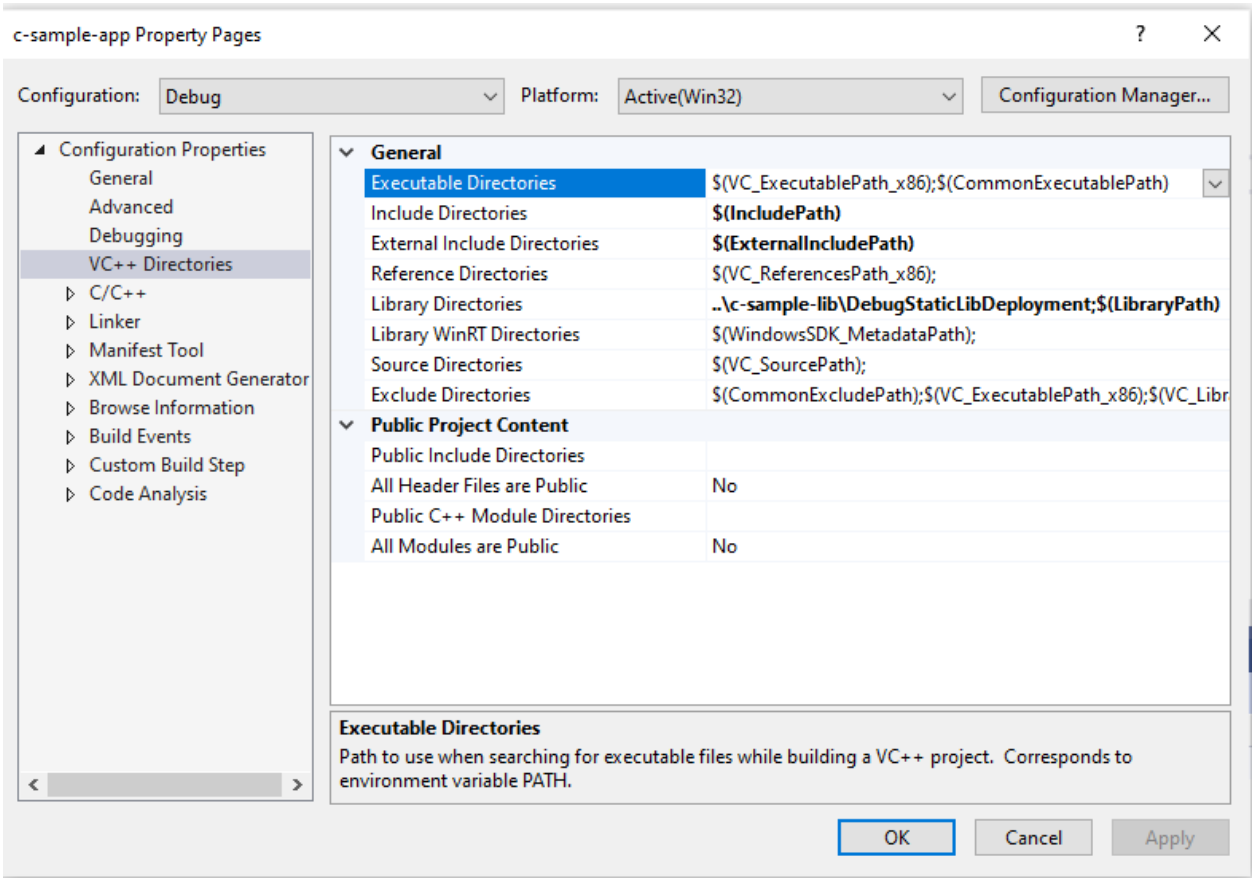
0.45 Shared Library Development - (VS C Static Library)-40



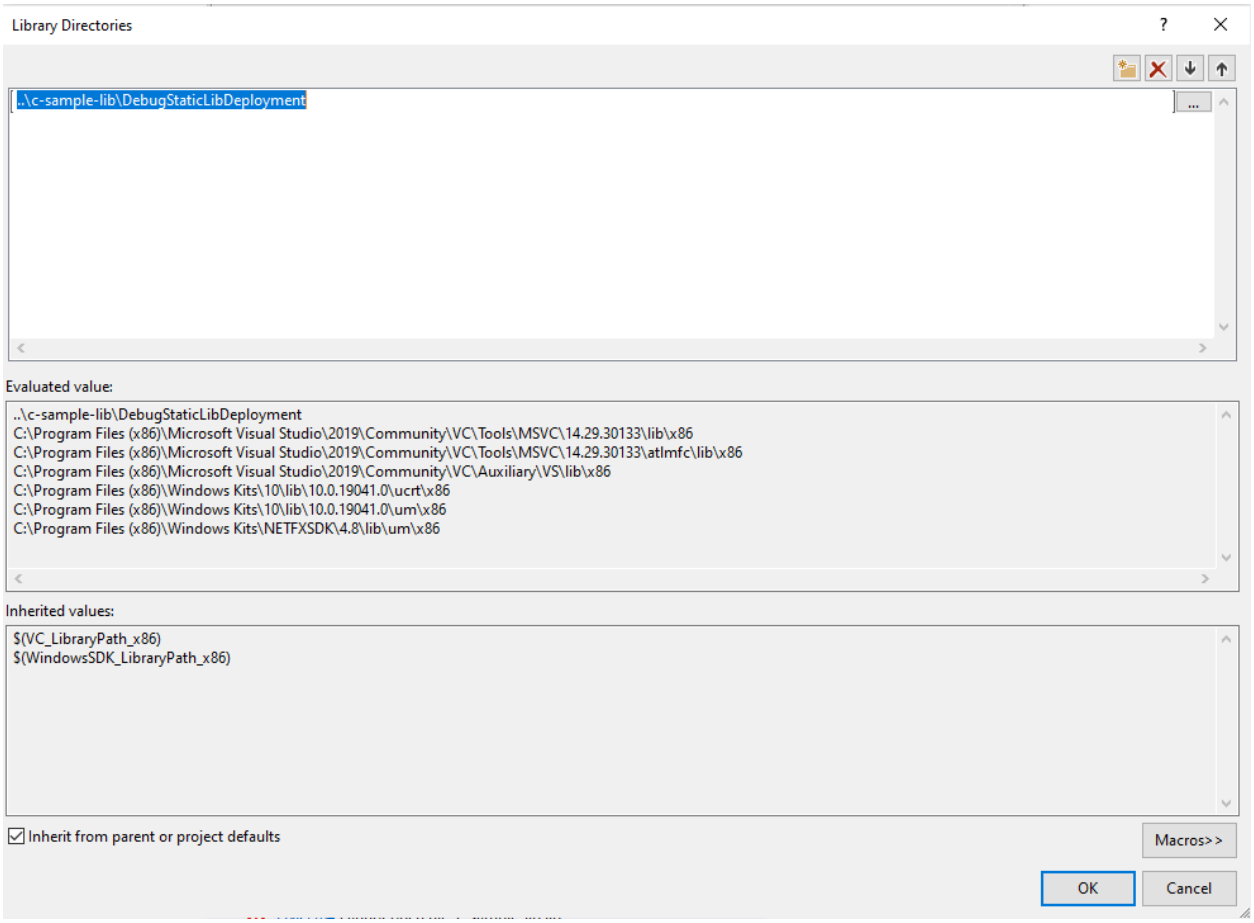
0.46 Shared Library Development - (VS C Static Library)-41

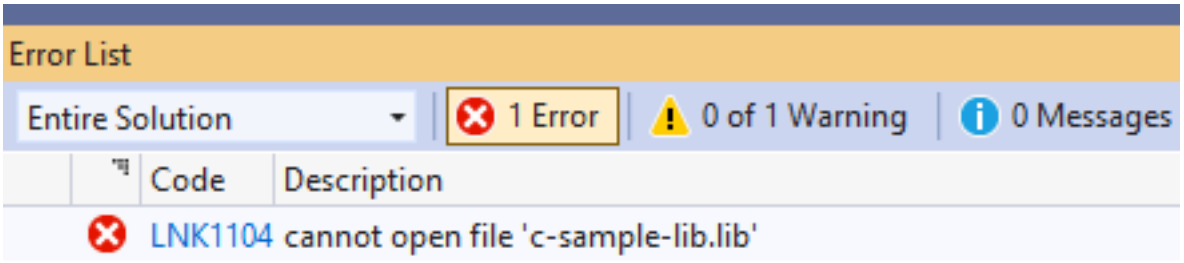
Not Working

`..\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

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

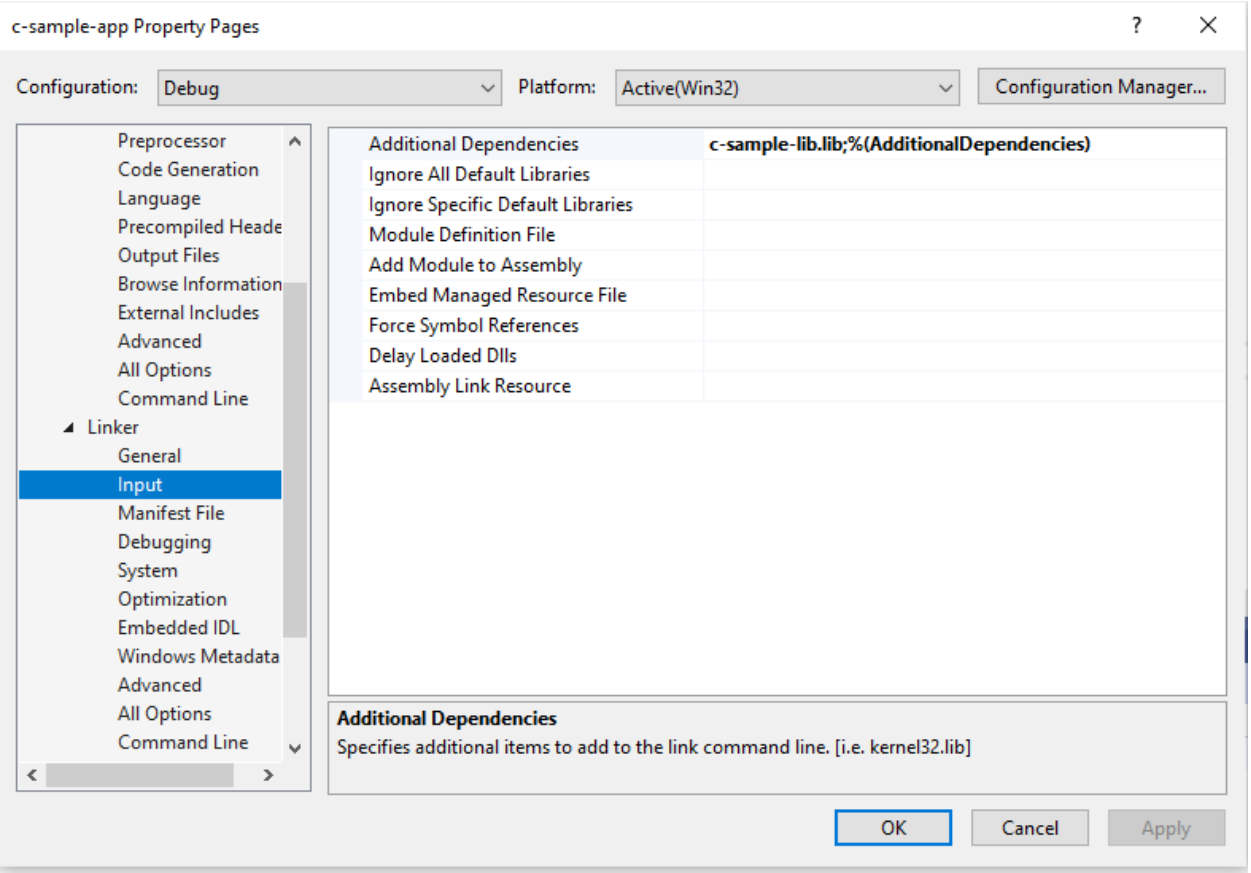




0.48 Shared Library Development - (VS C Static Library)-43

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.

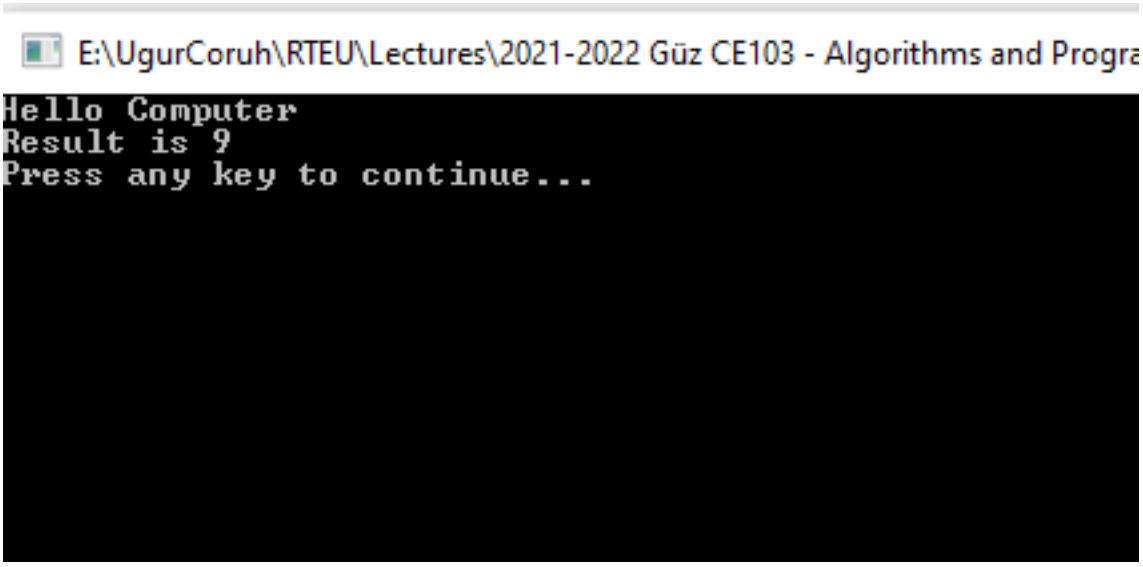
0.49 Shared Library Development - (VS C Static Library)-44

current source code will be like that nothing changed

```c
#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;
}
```

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

- and output will be as follow

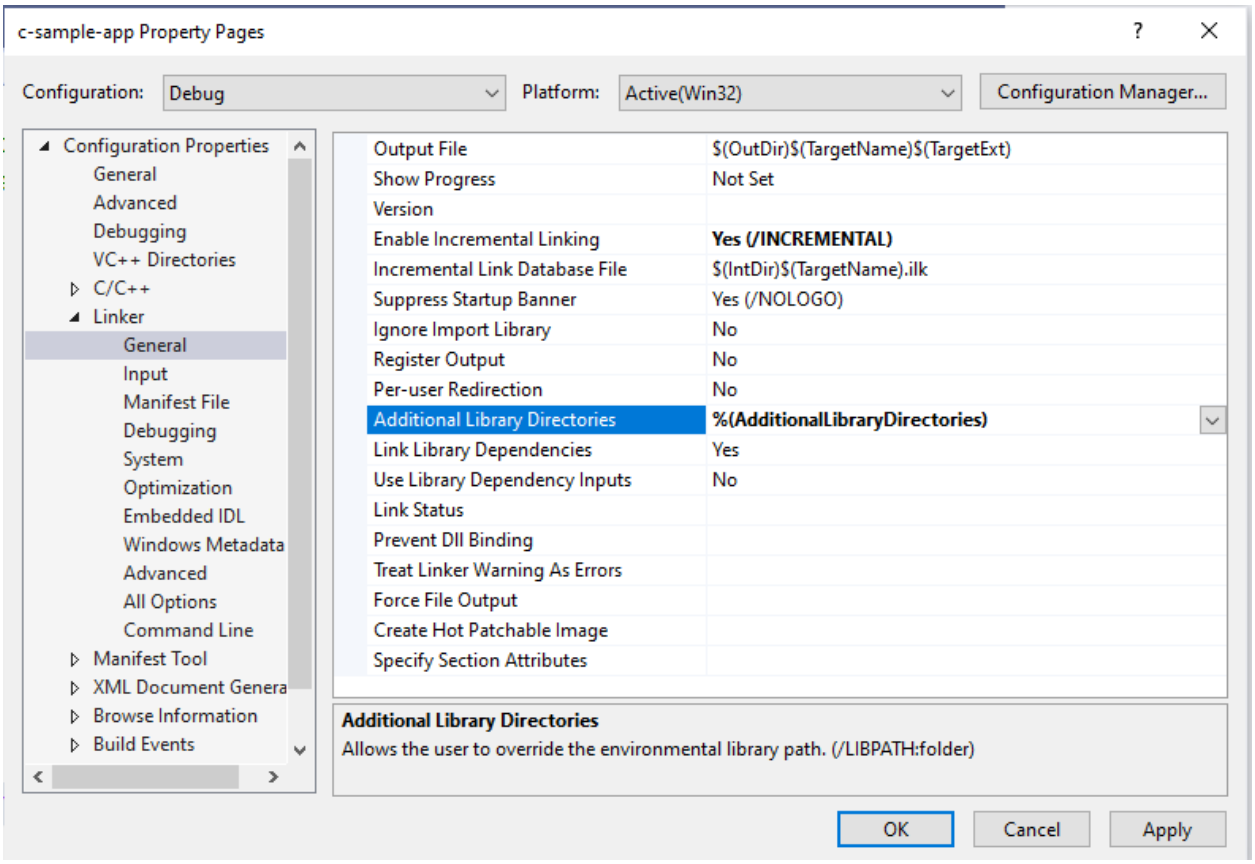


0.51 Shared Library Development - (VS C Static Library)-46

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

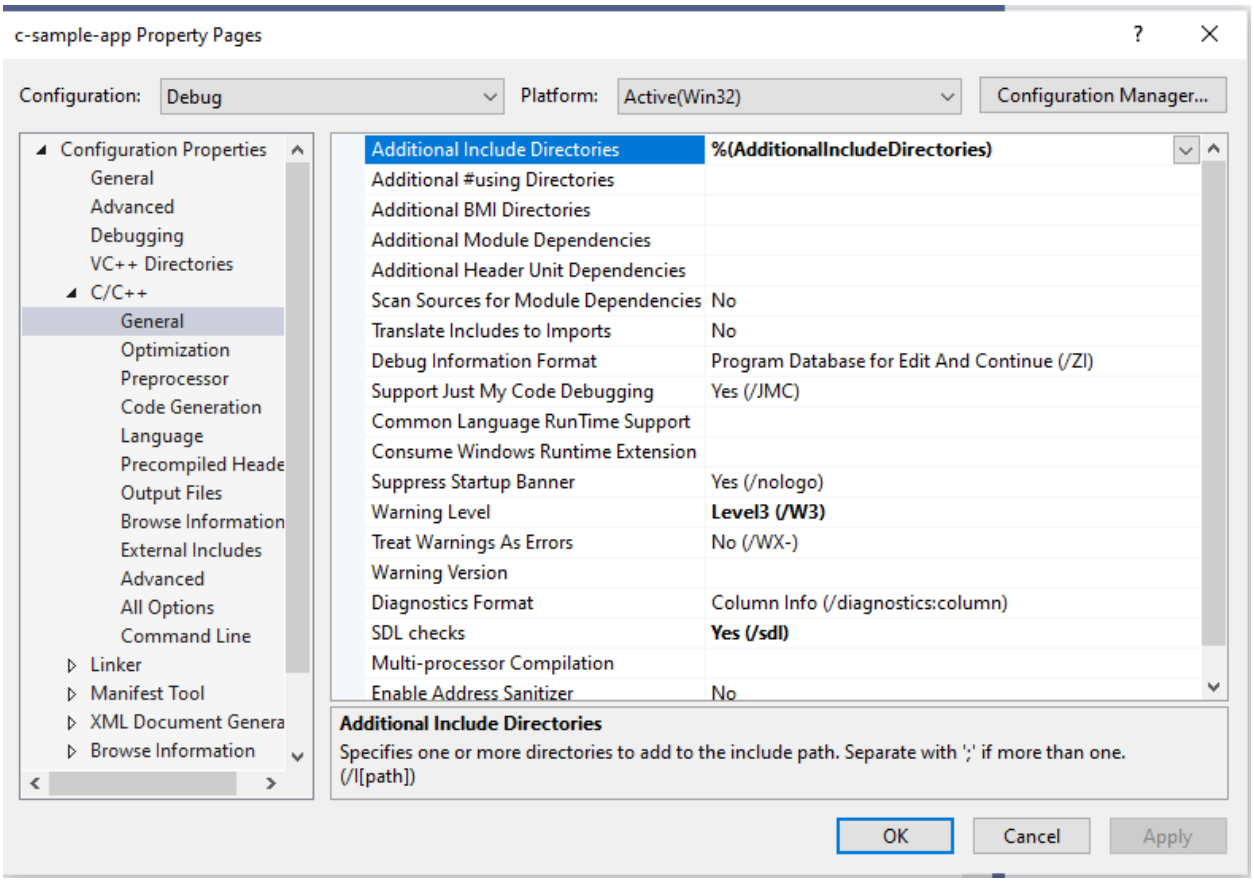
We will remove all library related settings from configurations and we will write them in source code

Clear linker->general->additional library directories



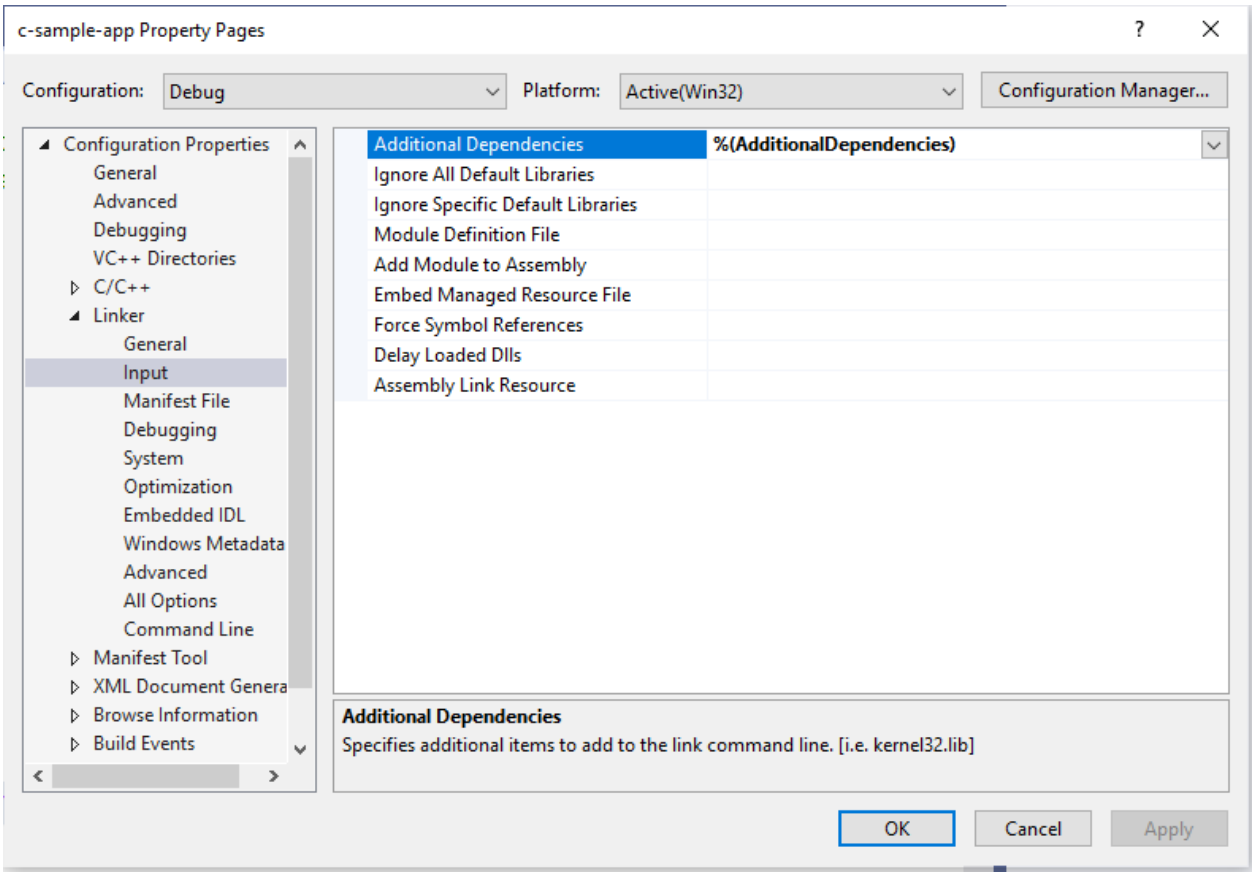
0.52 Shared Library Development - (VS C Static Library)-47

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



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

Clear Linker->Input->Additional Dependencies



0.54 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.

0.55 Shared Library Development

0.55.1 C++ Programming (Static Library)

0.55.1.1 Visual Studio Community Edition

0.56 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 complied version.

0.57 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;

}
```

0.58 Shared Library Development

0.58.1 C++ Programming (Static Library)

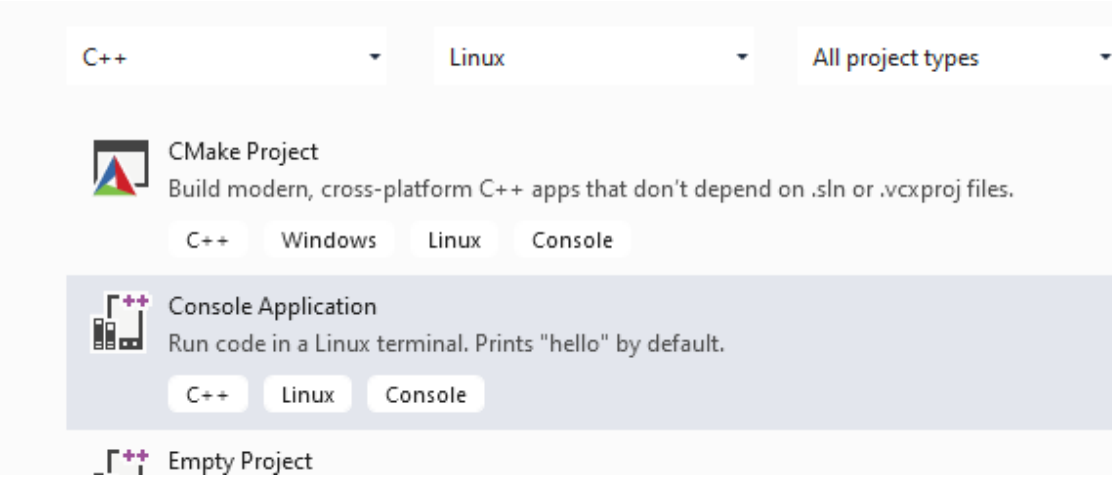
0.58.1.1 Visual Studio Community Edition WSL Option

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

- Install WSL2
  - GitHub - ucoruh/ns3-wsl-win10-setup: ns3 windows 10 WSL2 setup and usage[5]

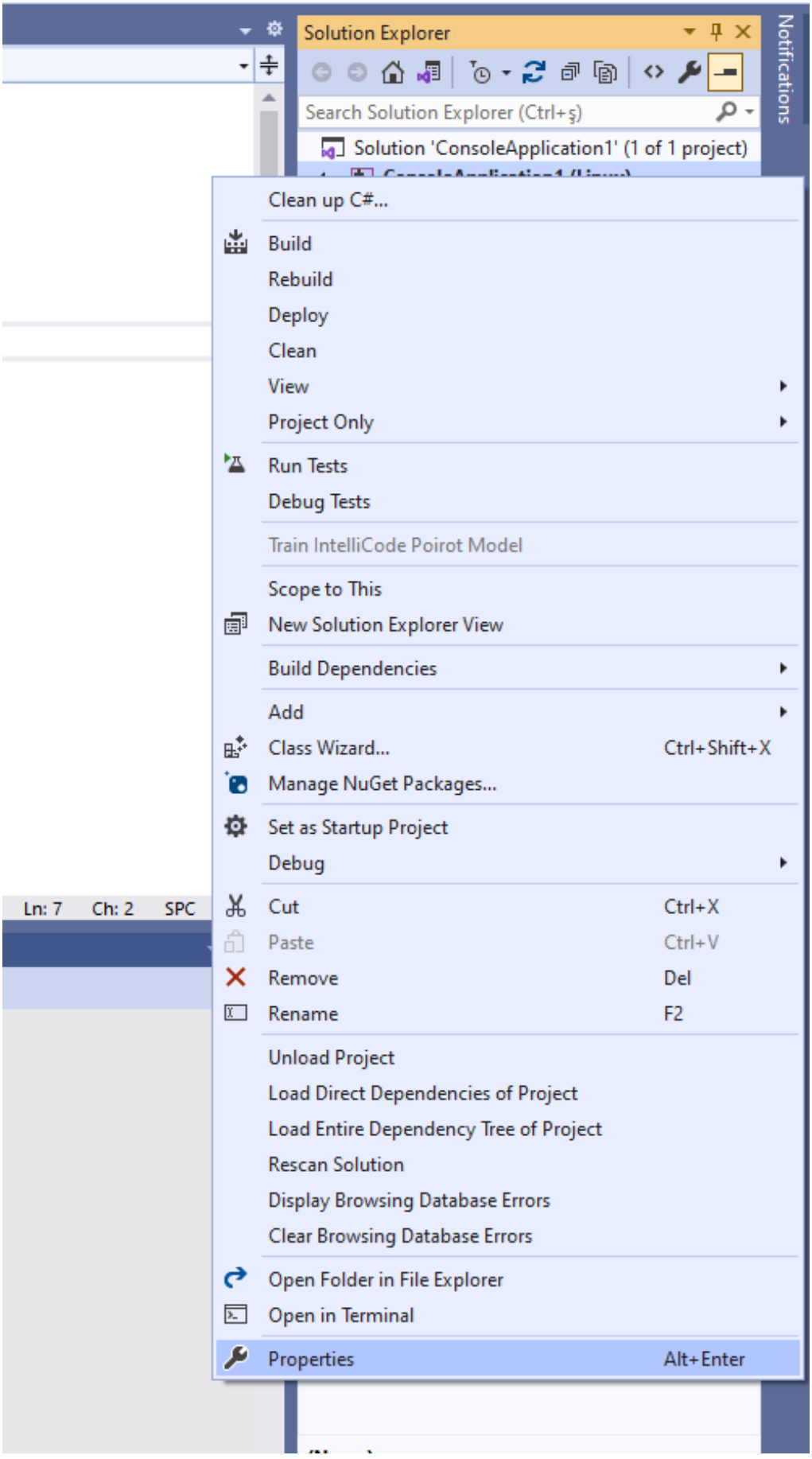
[5] https://github.com/ucoruh/ns3-wsl-win10-setup

- Create a Linux project



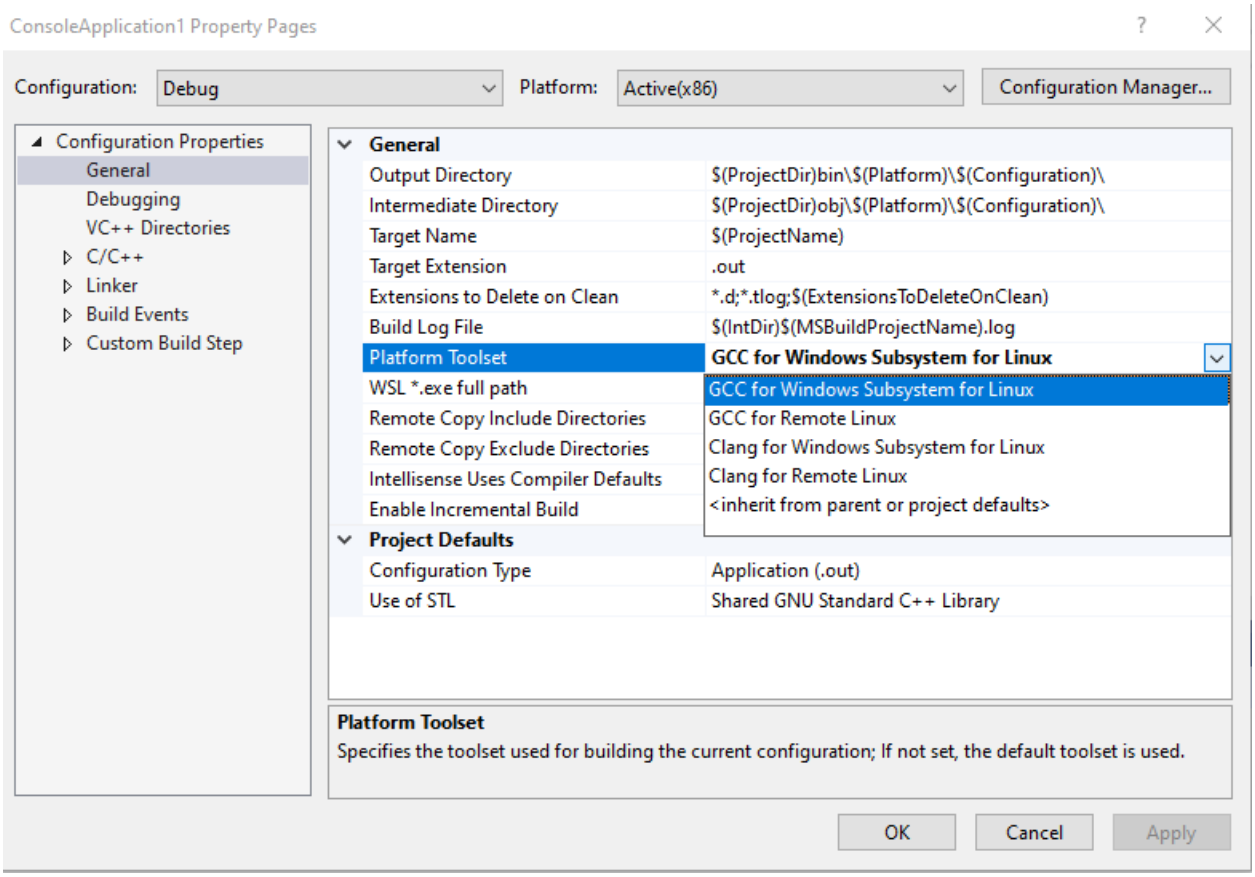
0.60 Shared Library Development - (VS Cpp WSL Static Library)-2

- Configure Platform Toolset to WSL


Solution Explorer
Search Solution Explorer (Ctrl+ş)
Solution 'ConsoleApplication1' (1 of 1 project)
Notifications
Clean up C#...
Build
Rebuild
Deploy
Clean
View
Project Only
Run Tests
Debug Tests
Train IntelliCode Poirot Model
Scope to This
New Solution Explorer View
Build Dependencies
Add
Class Wizard... Ctrl+Shift+X
Manage NuGet Packages...
Set as Startup Project
Debug
Cut Ctrl+X
Paste Ctrl+V
Remove Del
Rename F2
Unload Project
Load Direct Dependencies of Project
Load Entire Dependency Tree of Project
Rescan Solution
Display Browsing Database Errors
Clear Browsing Database Errors
Open Folder in File Explorer
Open in Terminal
Properties Alt+Enter
Ln: 7
Ch: 2
SPC

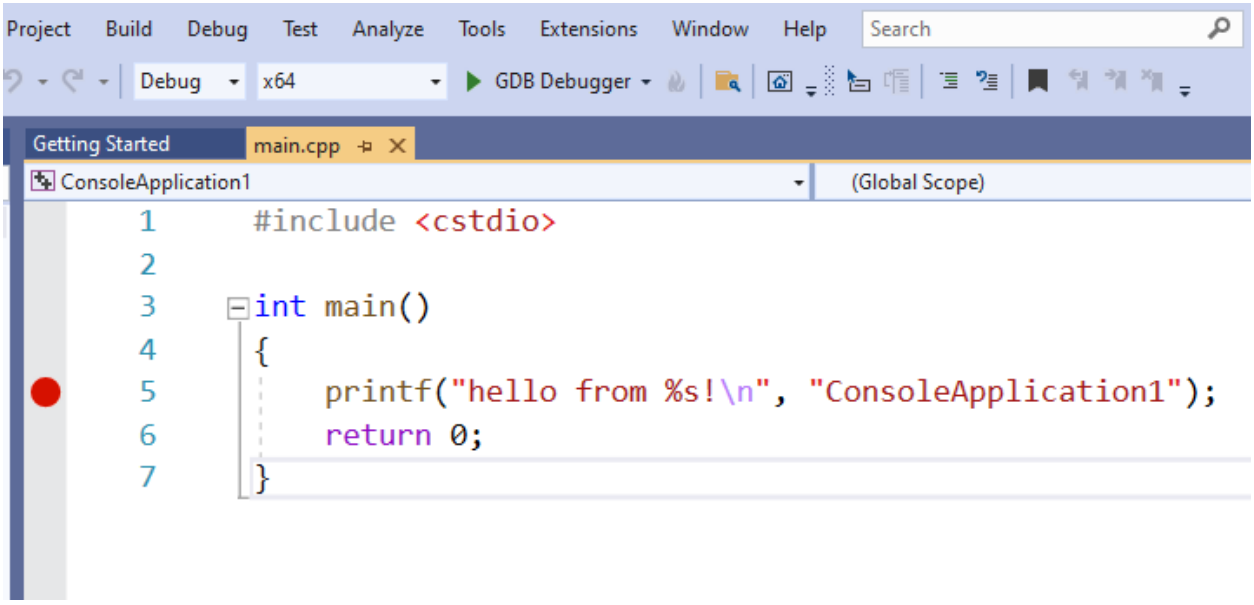
0.61 Shared Library Development - (VS Cpp WSL Static Library)-3

- Select GCC for Windows Subsystem for Linux



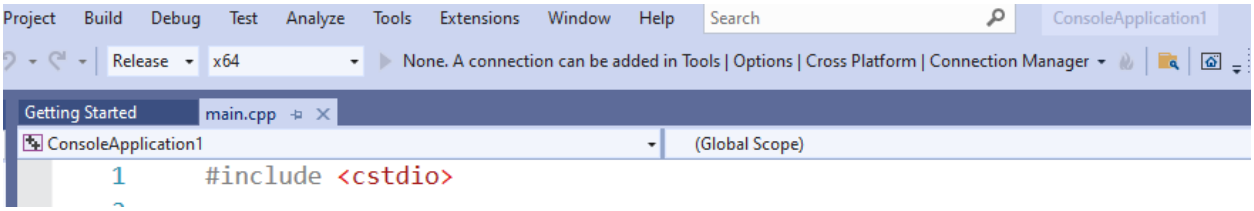
0.62 Shared Library Development - (VS Cpp WSL Static Library)-4

Put a breakpoint and run debugger



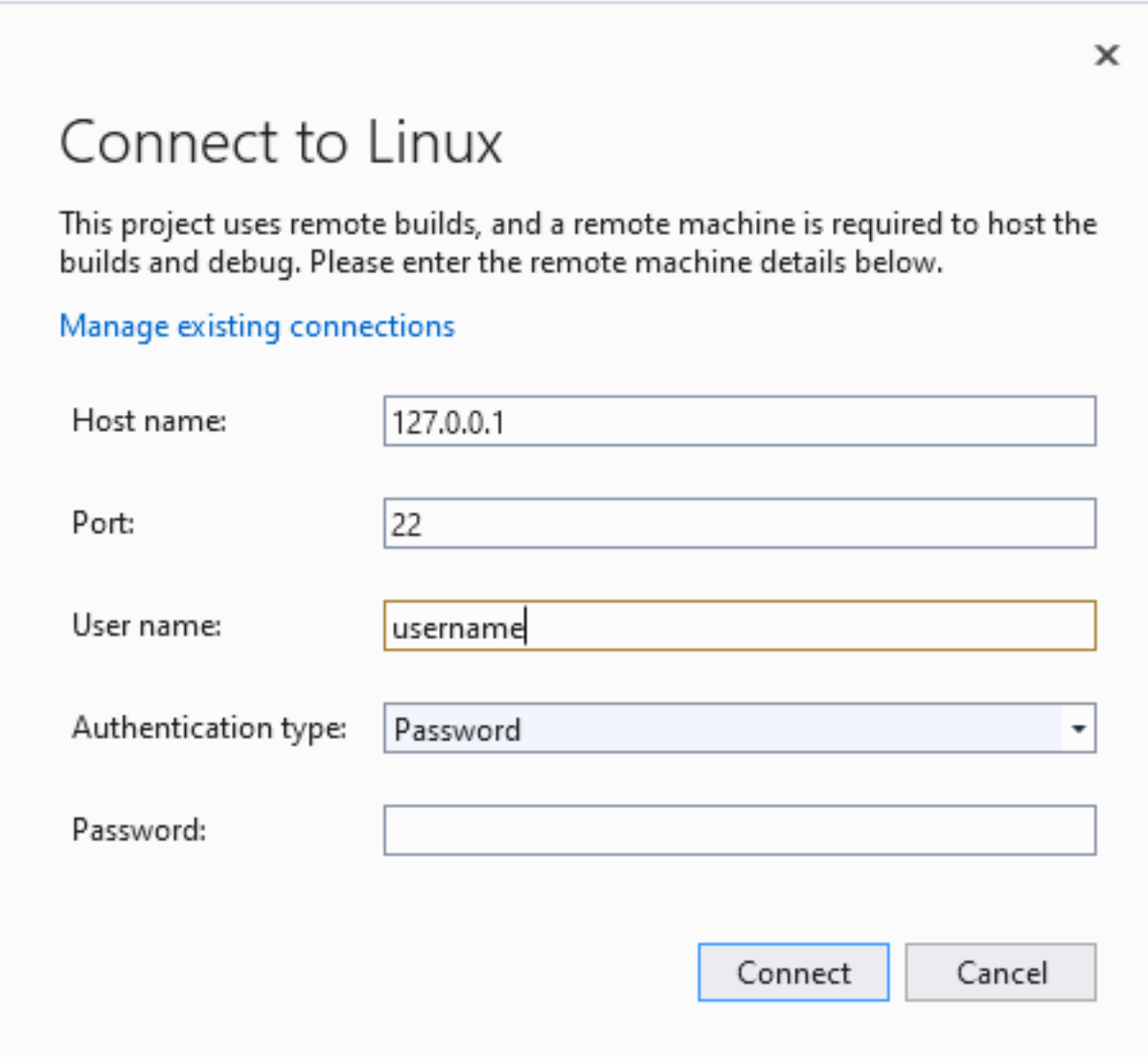
0.63 Shared Library Development - (VS Cpp WSL Static Library)-5

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.



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

- Configure SSH parameters



0.65 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[6]
  - Connect to Remote WSL Environment
    - Bağlan hedef Linux sisteminize Visual Studio | Microsoft Docs[7]

0.66 Shared Library Development

0.66.1 C# Programming (Dinamik Library)

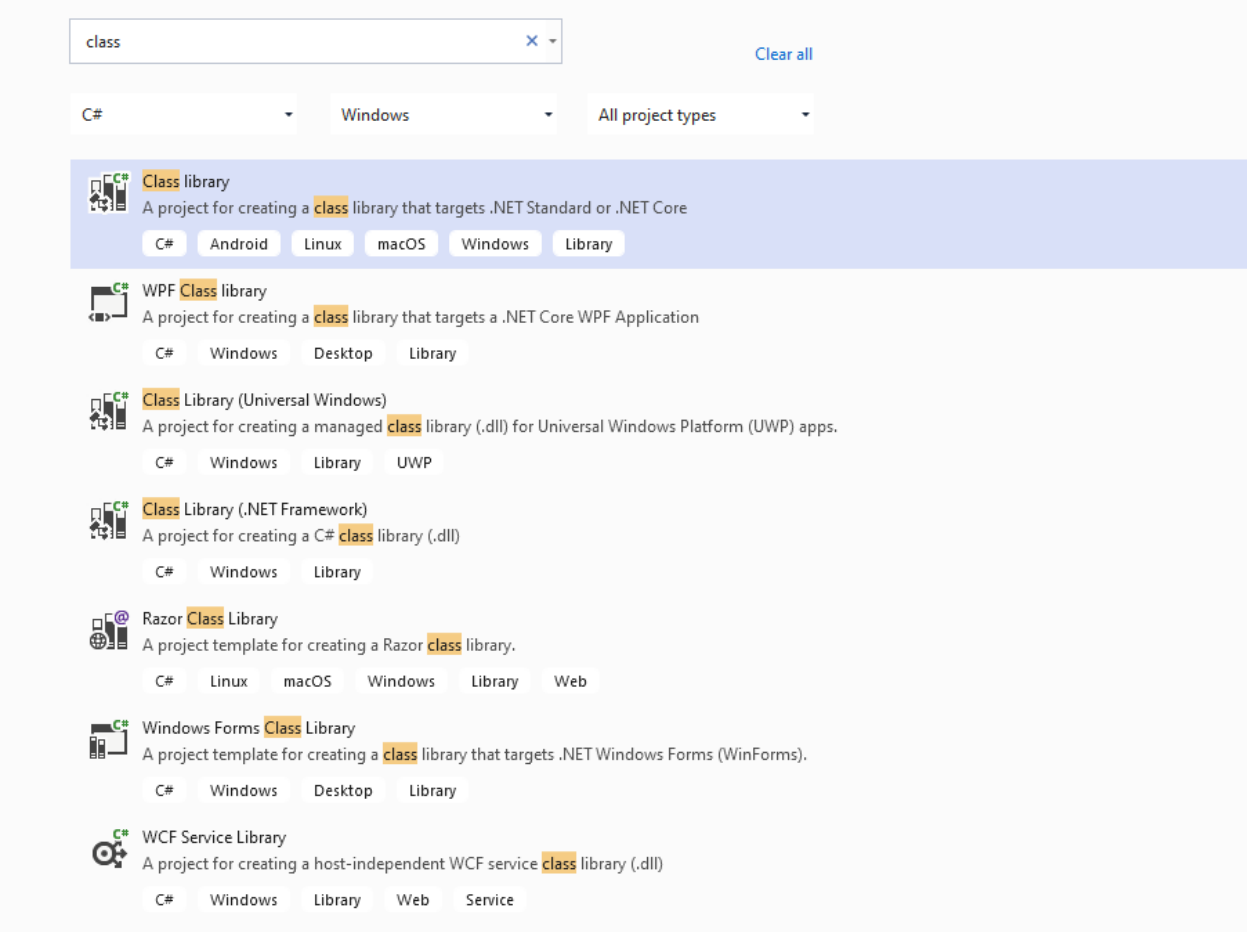
0.66.1.1 Visual Studio Community Edition

---

[6] https://www.illuminiastudios.com/dev-diaries/ssh-on-windows-subsystem-for-linux/
[7] https://docs.microsoft.com/tr-tr/cpp/linux/connect-to-your-remote-linux-computer?view=msvc-160

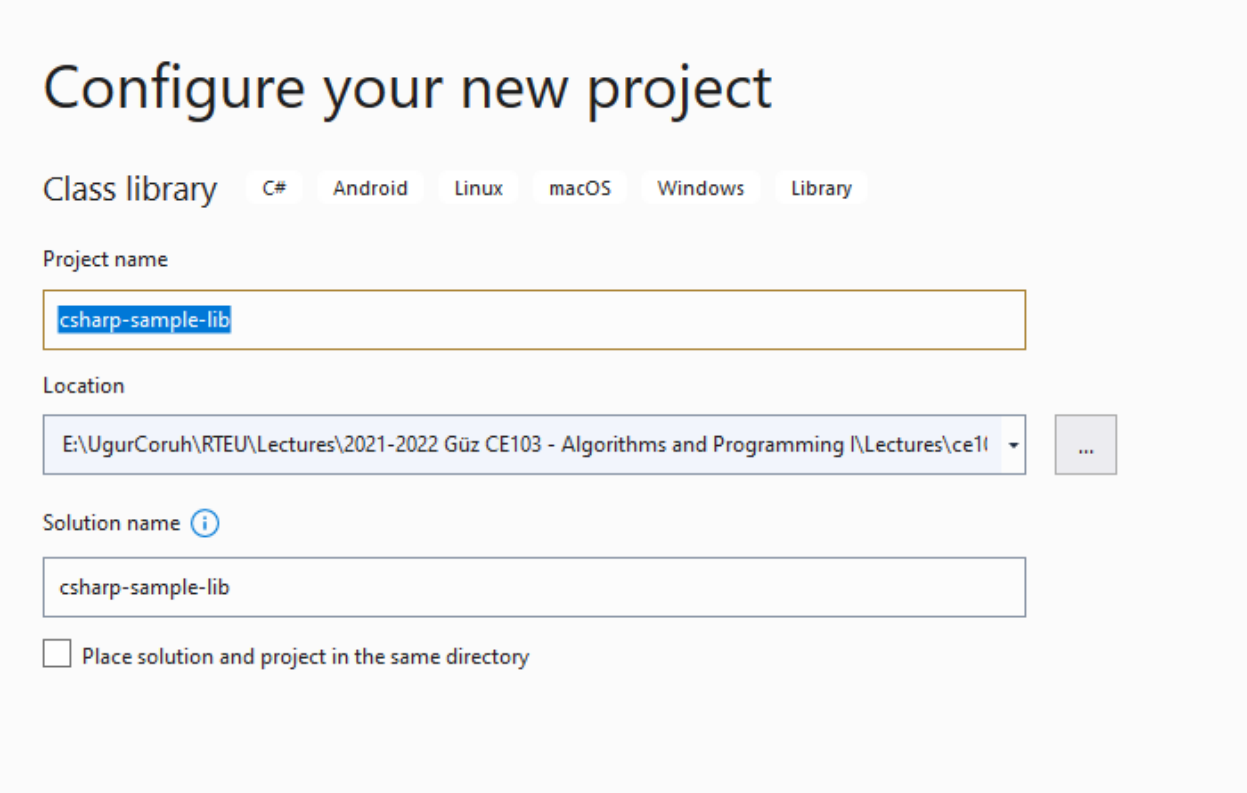
0.67 Shared Library Development - (VS Csharp Dynamic Library)-1

- 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



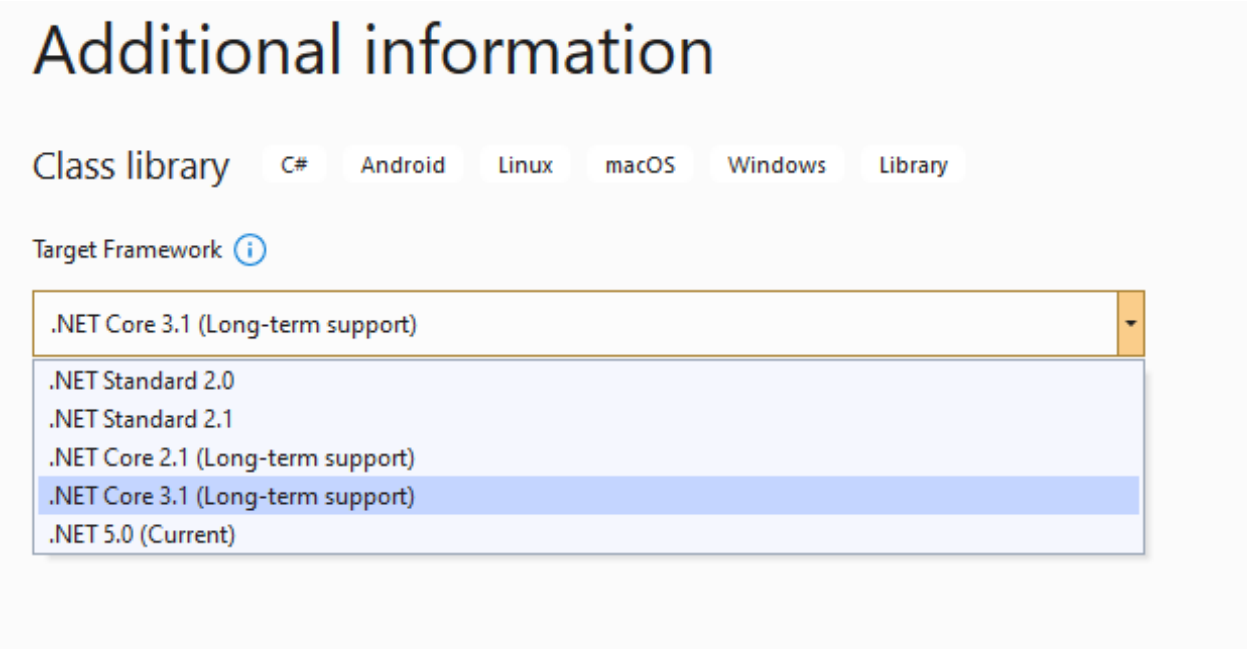
0.68 Shared Library Development - (VS Csharp Dynamic Library)-2

- There is no static library option



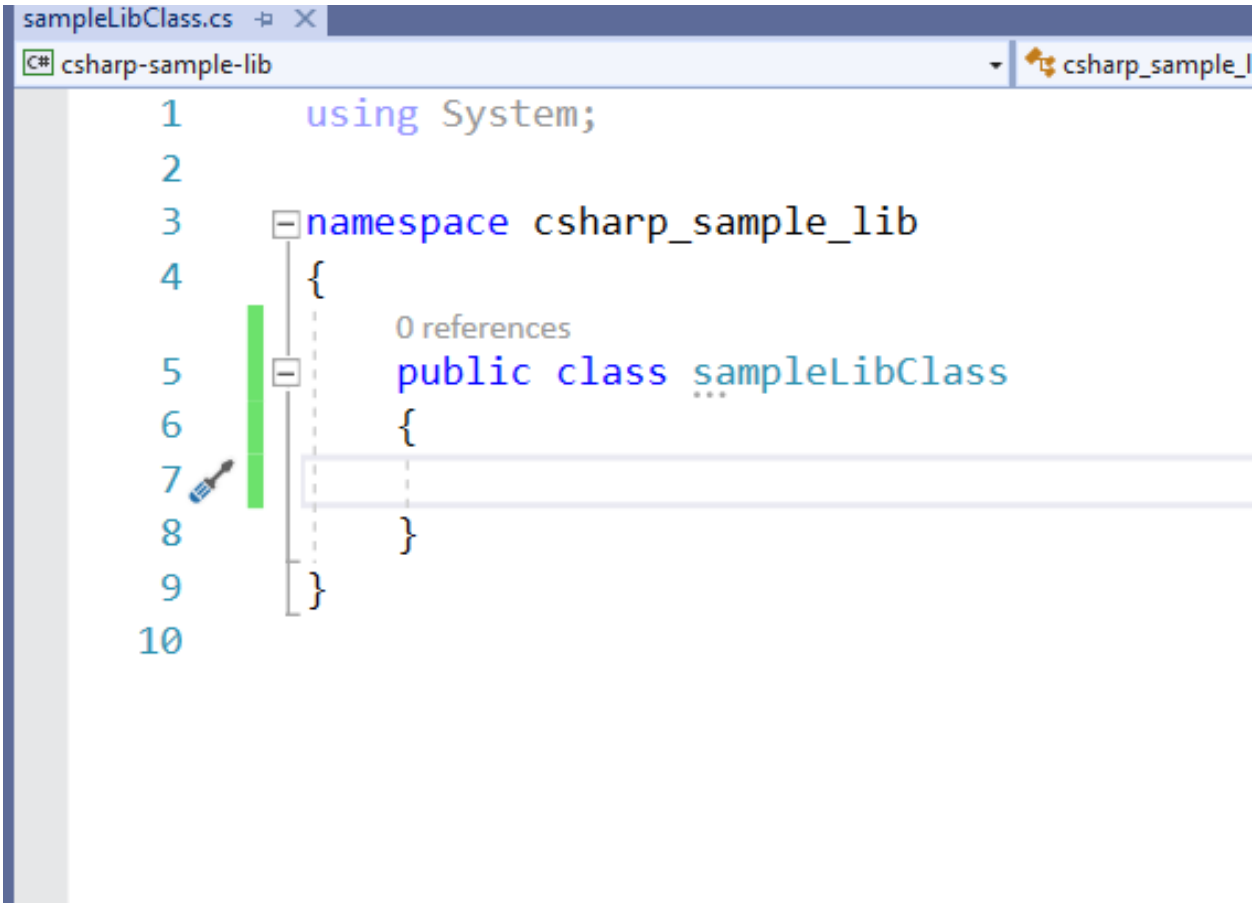
0.69 Shared Library Development - (VS Csharp Dynamic Library)-3

- We will select .Net Core 3.1



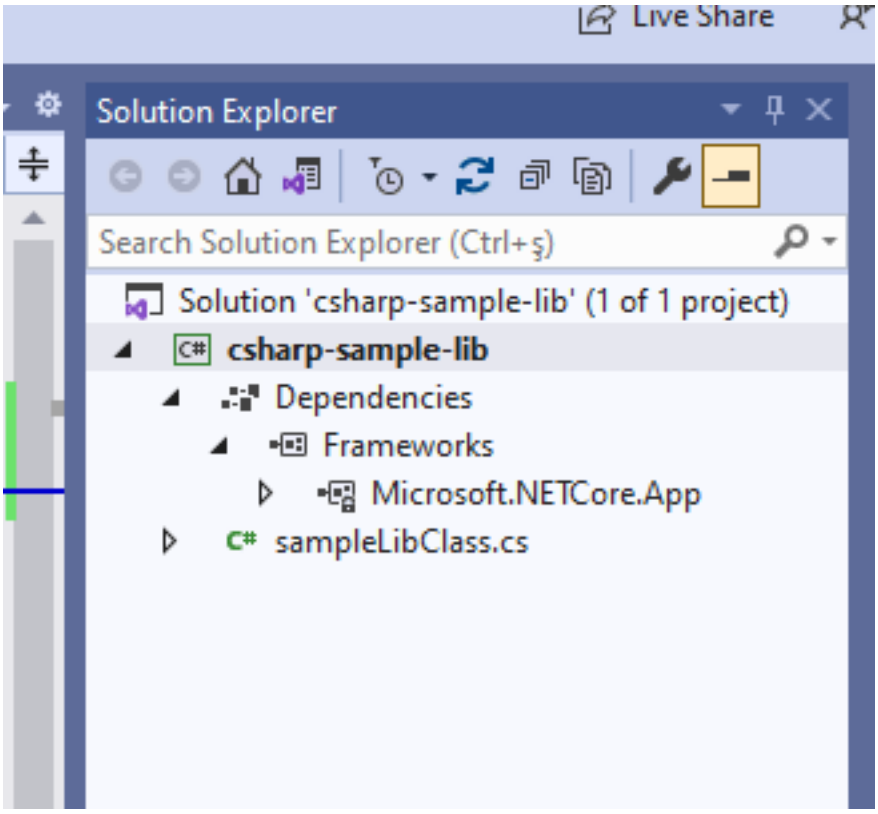
0.70 Shared Library Development - (VS Csharp Dynamic Library)-4

- You will have default empty class library file



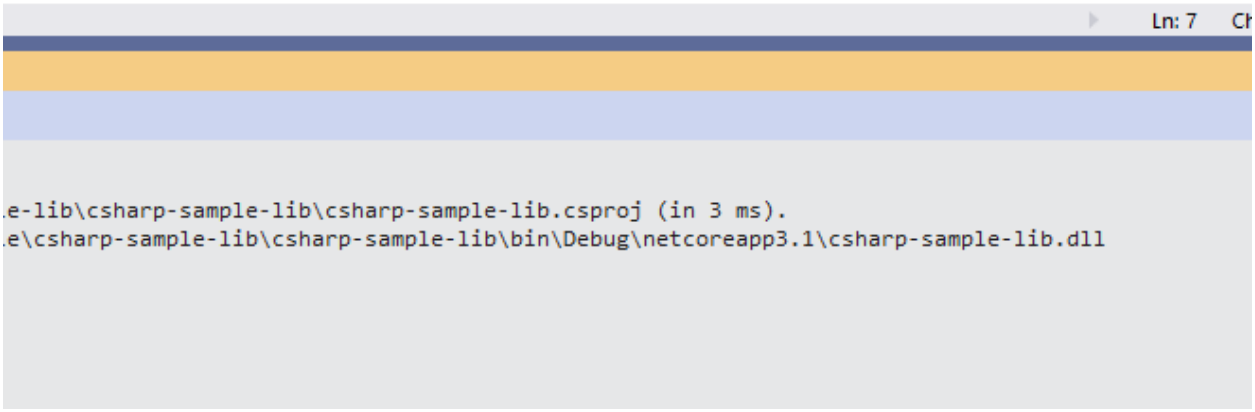
0.71 Shared Library Development - (VS Csharp Dynamic Library)-5

- In the project you can see .NETcore reference



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

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



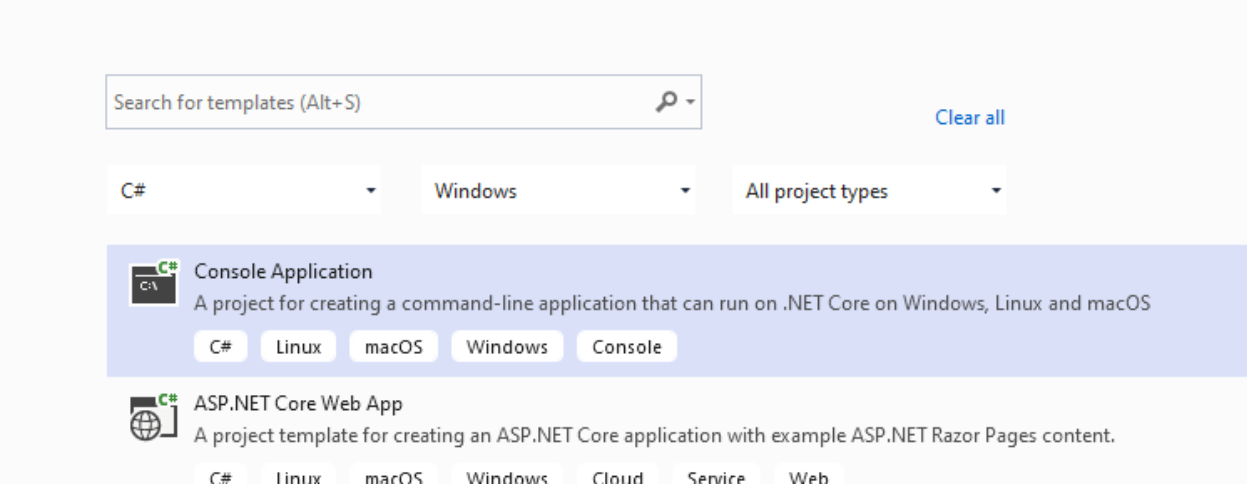
0.73 Shared Library Development - (VS Csharp Dynamic Library)-7

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

---

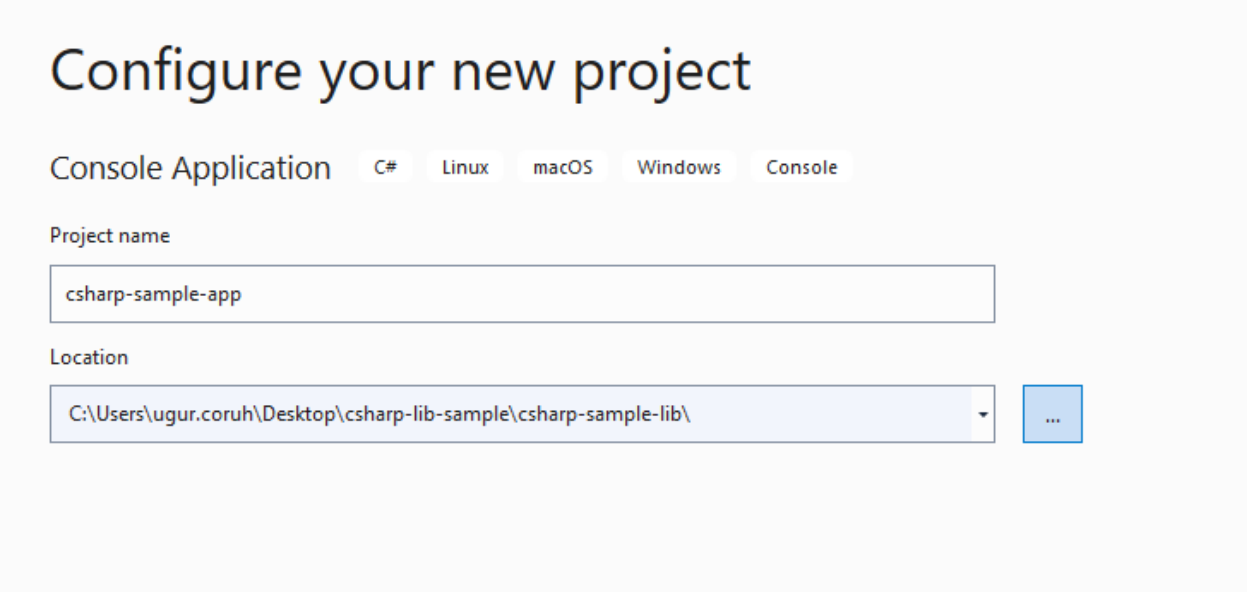
0.74 Shared Library Development - (VS Csharp Dynamic Library)-8

- Select New Project



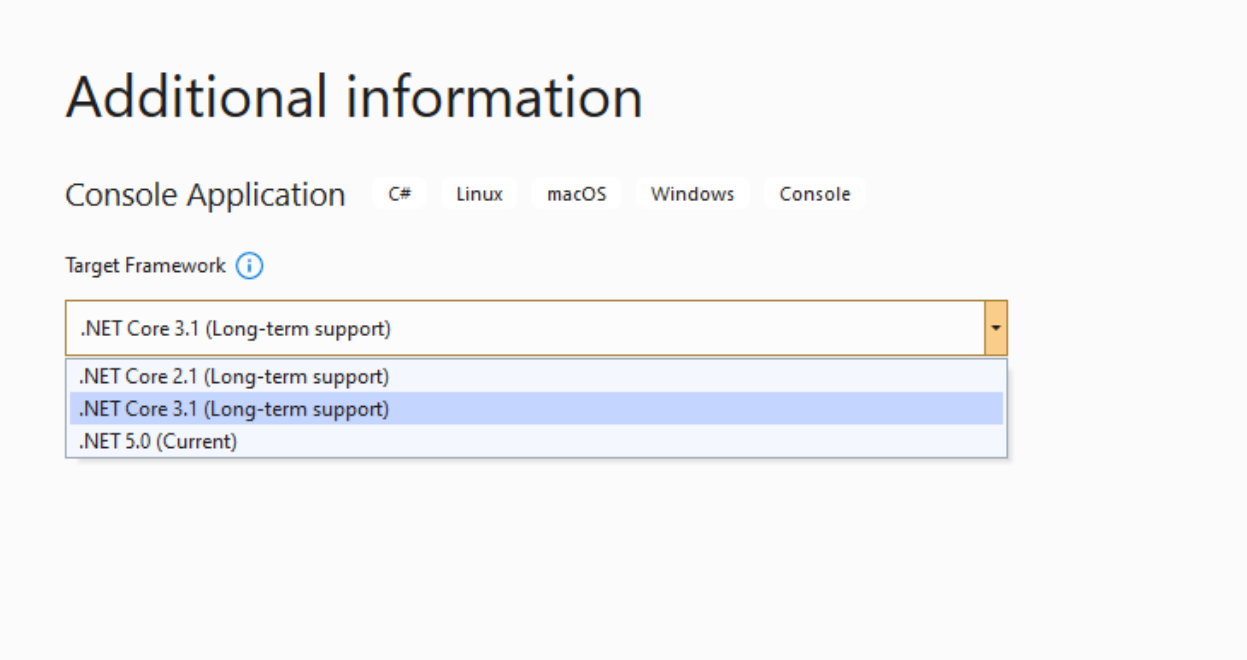
0.75 Shared Library Development - (VS Csharp Dynamic Library)-9

- Set project name



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

- Select .NETCore framework



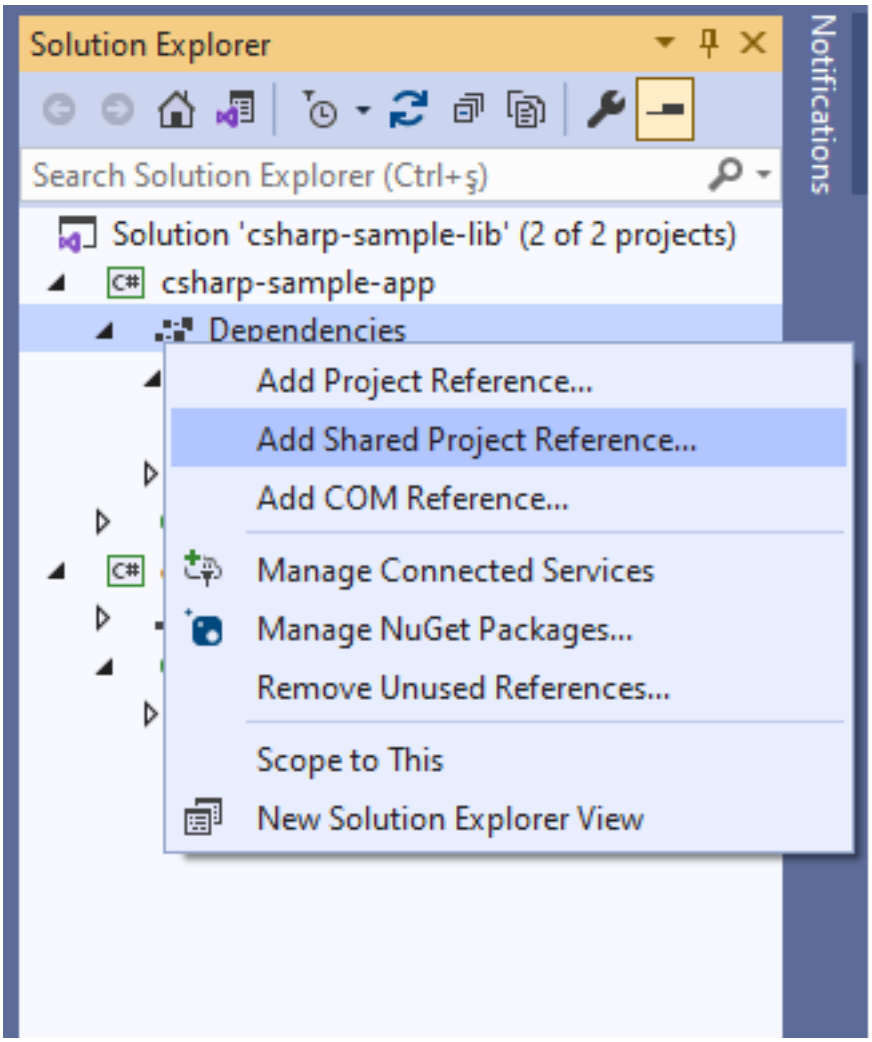
0.77 Shared Library Development - (VS Csharp Dynamic Library)-11

- 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!");
        }
    }
}
```

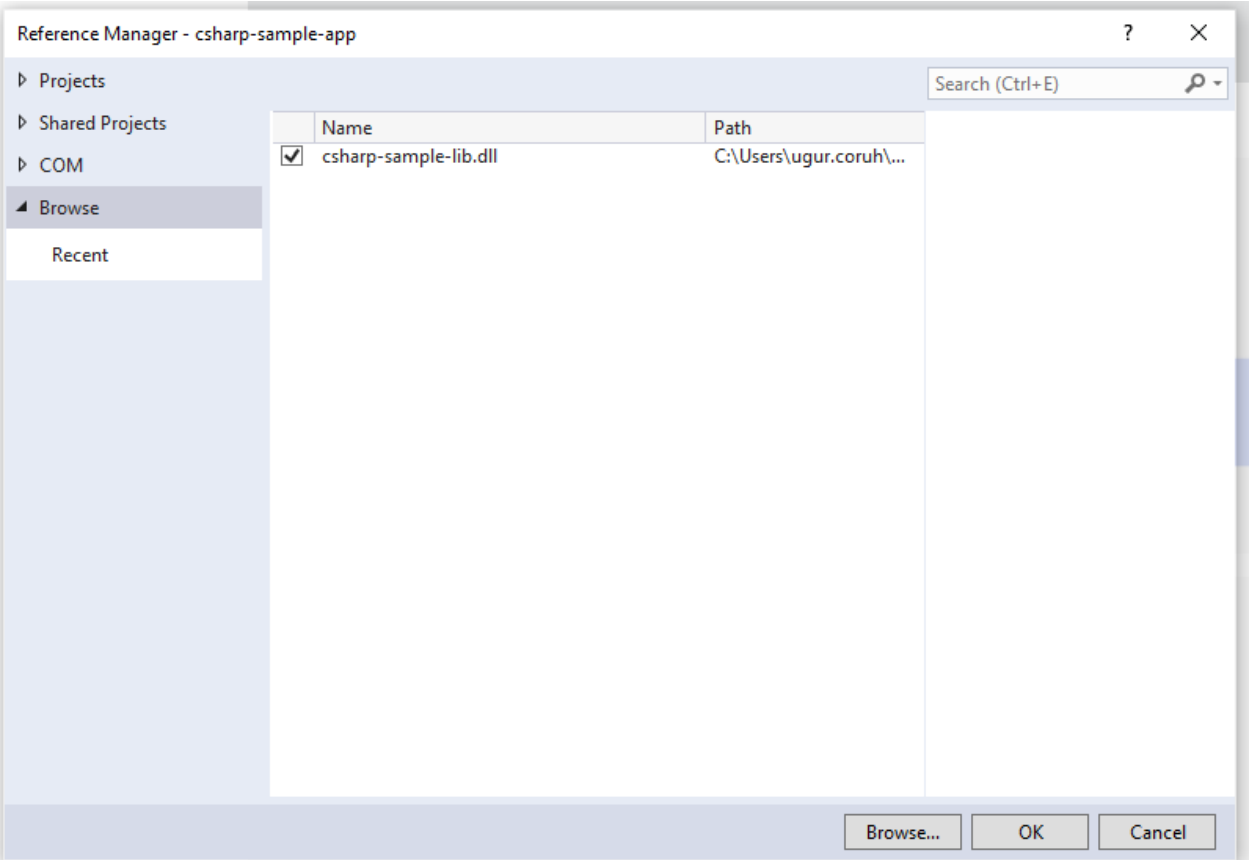
0.78 Shared Library Development - (VS Csharp Dynamic Library)-12

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



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

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



0.80 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

---

0.81 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;
        }
    }
}
```

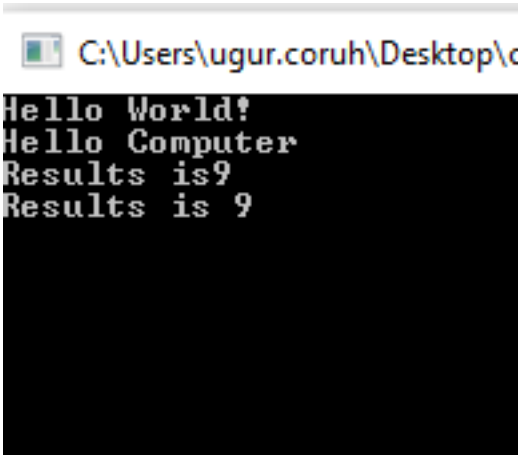
0.82 Shared Library Development - (VS Csharp Dynamic Library)-16

- 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();
        }
    }
}
```

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

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



0.84 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

0.85 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[8]

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
```

0.86 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
```

0.87 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
```

0.88 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.

0.89 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:

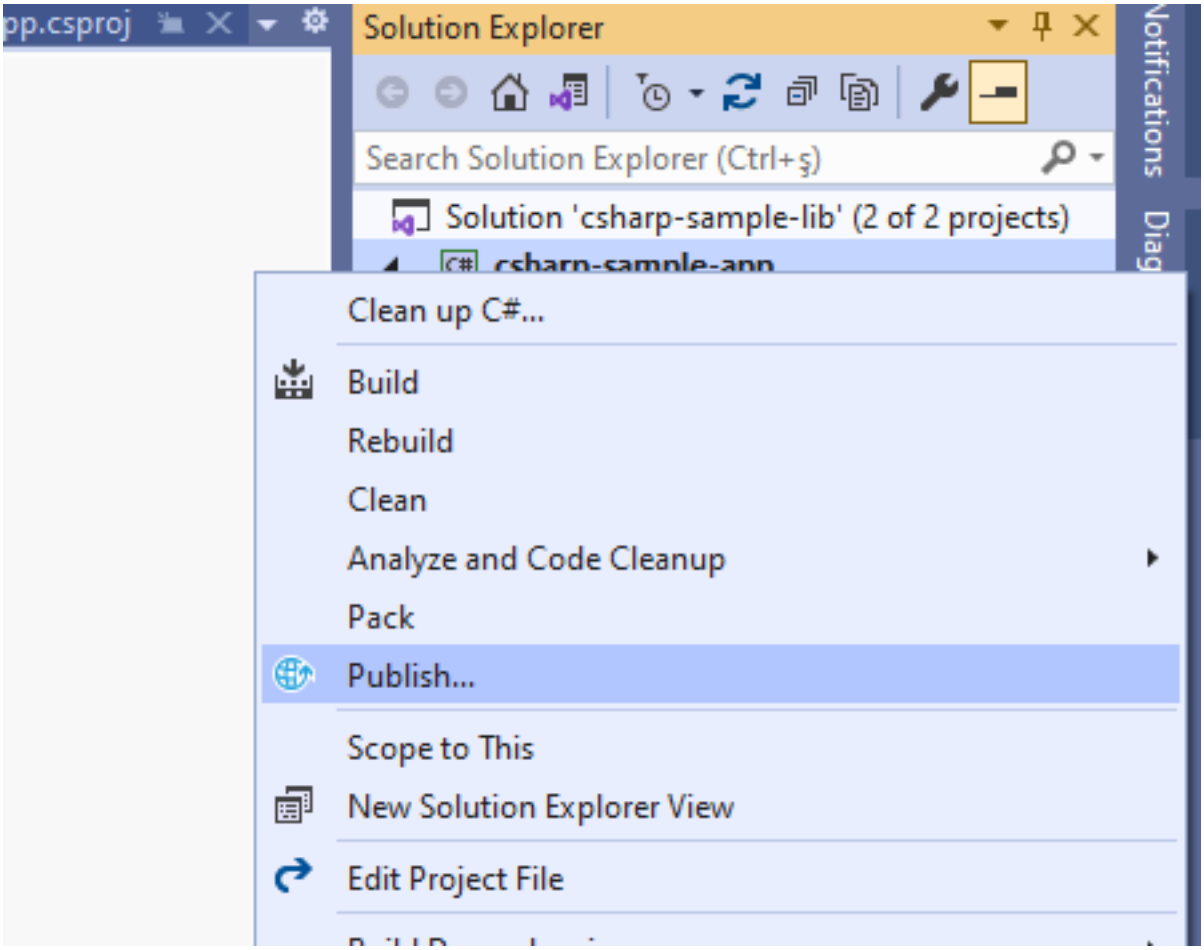
```
dotnet --version
```

[8] https://tecadmin.net/how-to-install-net-core-on-ubuntu-20-04/

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

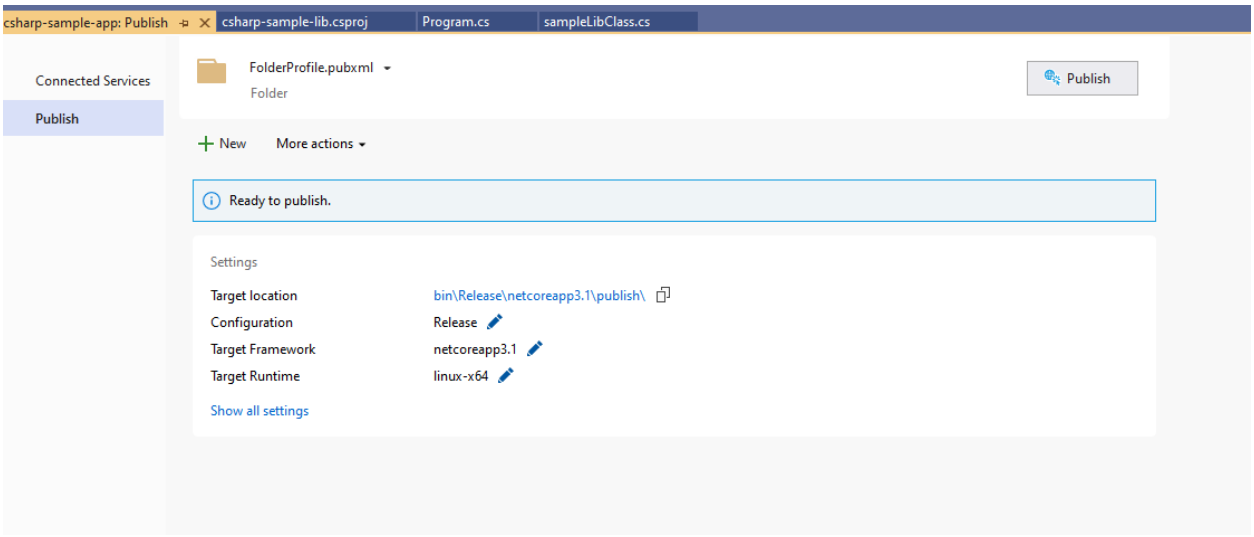
0.90 Shared Library Development - (VS Csharp Dynamic Library)-24

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



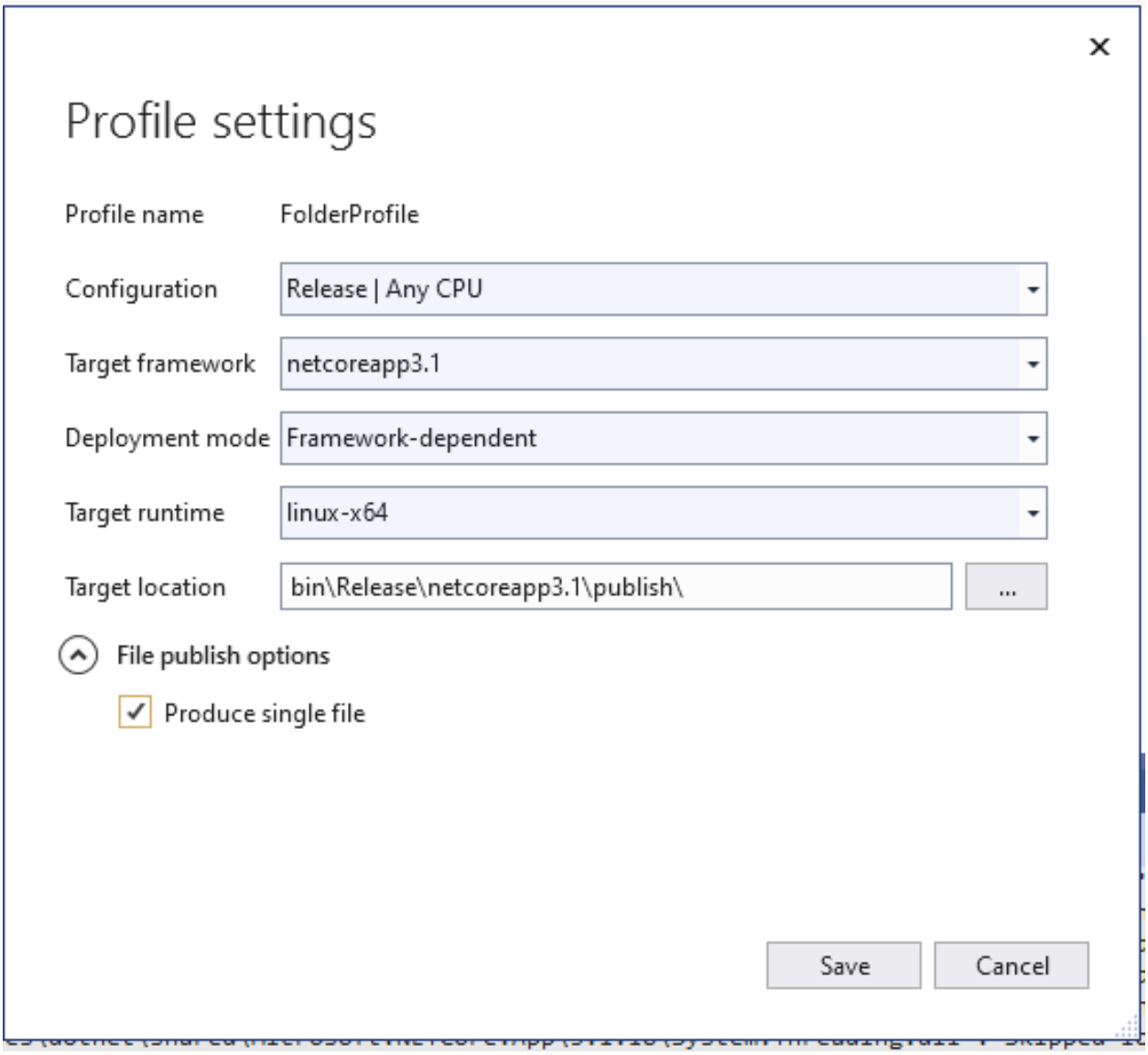
0.91 Shared Library Development - (VS Csharp Dynamic Library)-25

- Select netcoreapp3.1 and Release for linux-x64



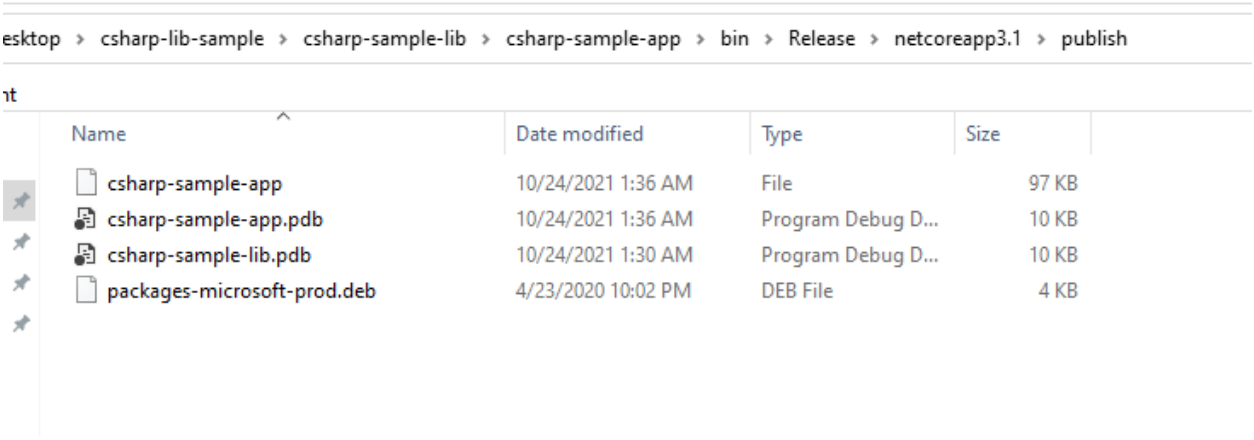
0.92 Shared Library Development - (VS Csharp Dynamic Library)-26

- Select produce single file



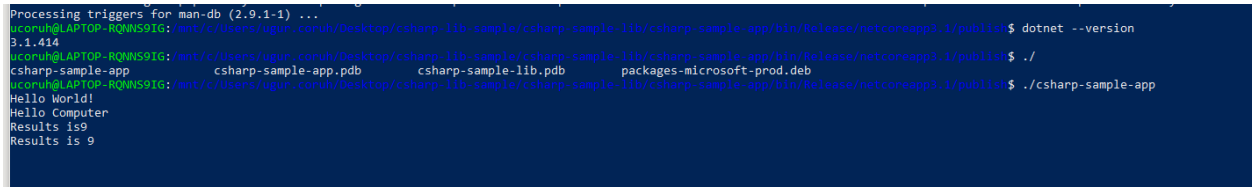
0.93 Shared Library Development - (VS Csharp Dynamic Library)-27

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



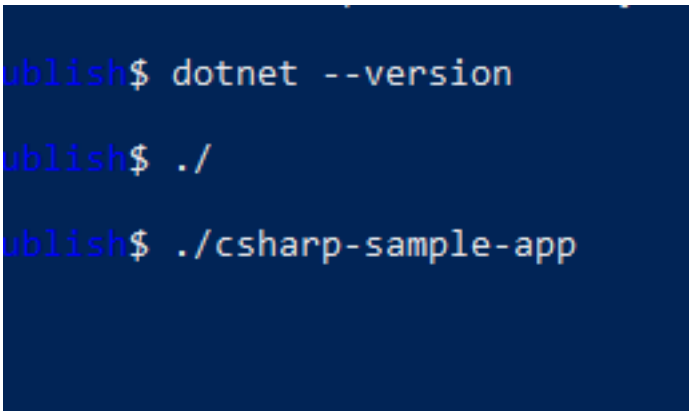
0.94 Shared Library Development - (VS Csharp Dynamic Library)-28

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

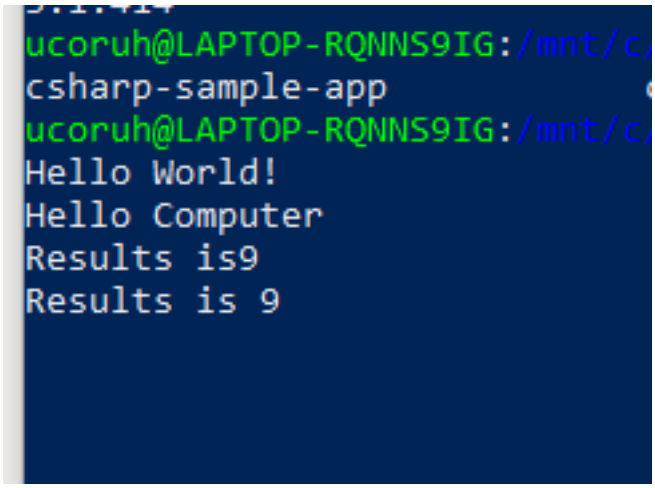


0.95 Shared Library Development - (VS Csharp Dynamic Library)-29

check dotnet –version and then run application



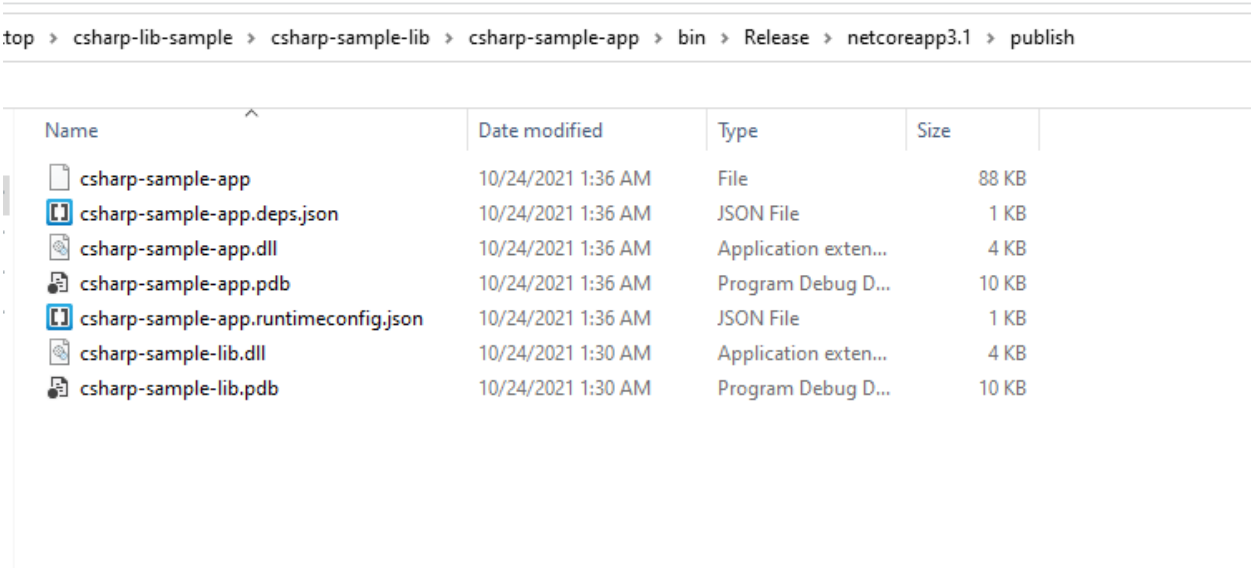
- you will see similar output



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

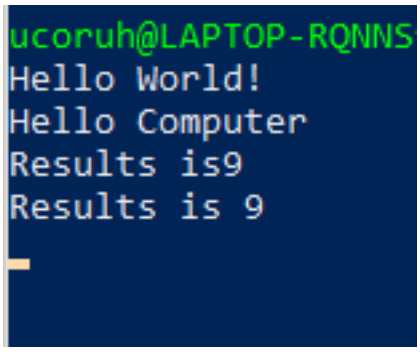
In this sample we created single application from settings lets try with shared library located option uncheck the “produce single file” option and publish again.

Then you will have the following outputs



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

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



0.98 Shared Library Development

0.98.1 Java Programming

0.98.1.1 Eclipse IDE

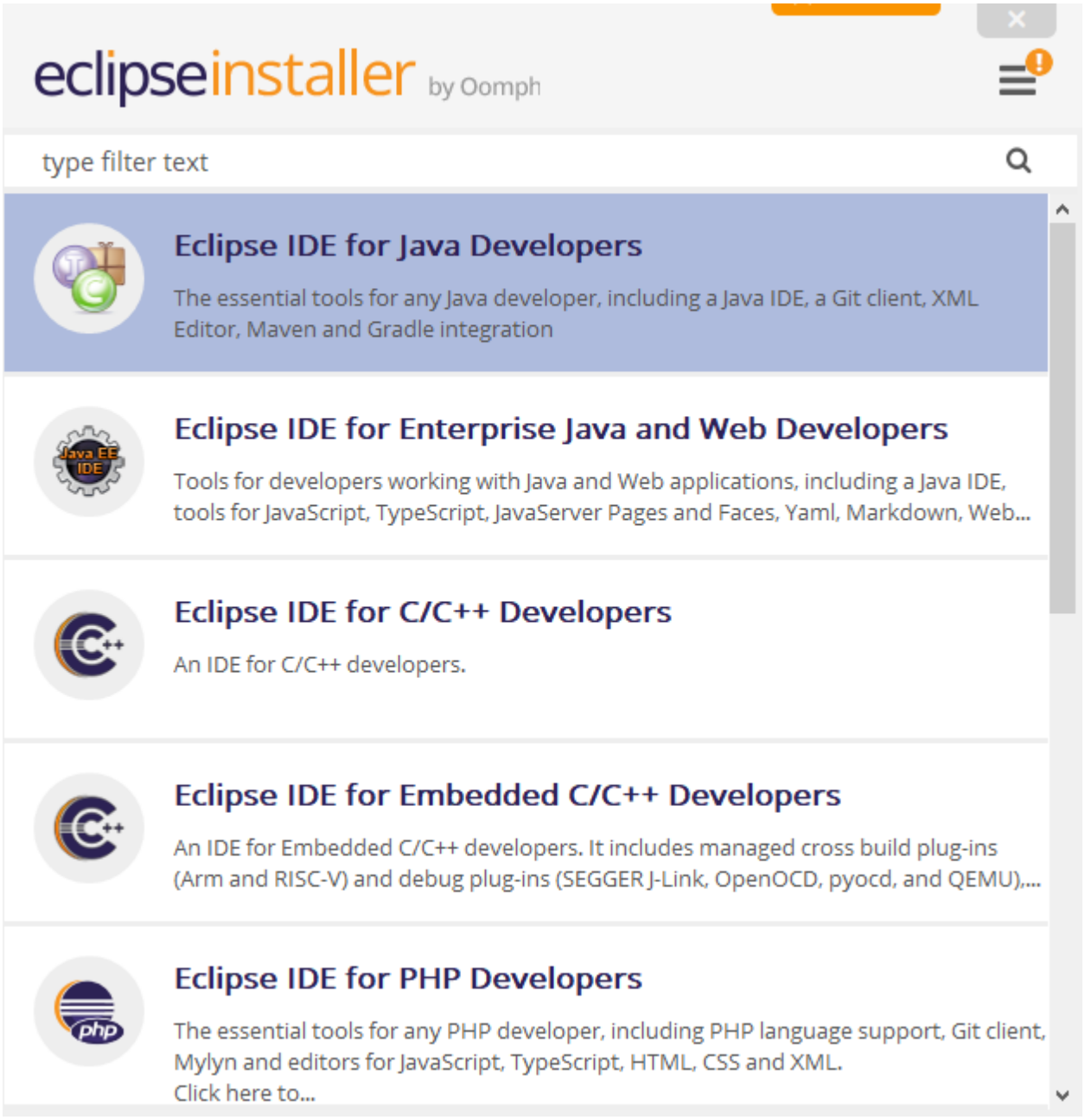
0.99 Shared Library Development - (Eclipse Java Jar Library)-1

- You should download and install eclipse installer and then you should select Eclipse IDE for Java Developers
  - Eclipse Installer 2021-09 R | Eclipse Packages[9]

[9] https://www.eclipse.org/downloads/packages/installer


eclipseinstaller
by Oomph

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

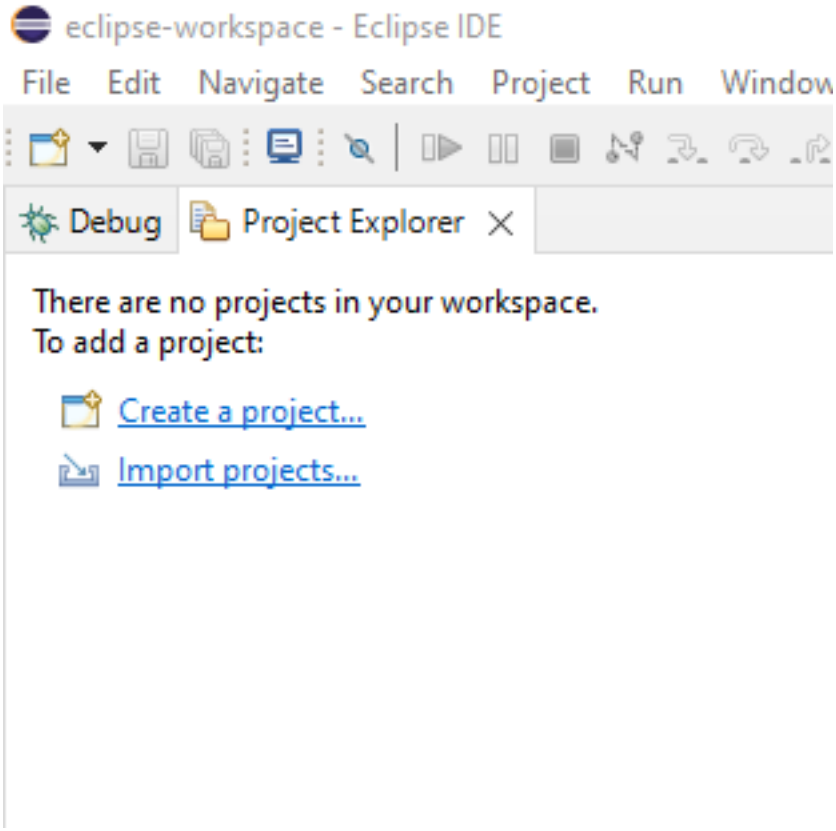


0.101 Shared Library Development - (Eclipse Java Jar Library)-3

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

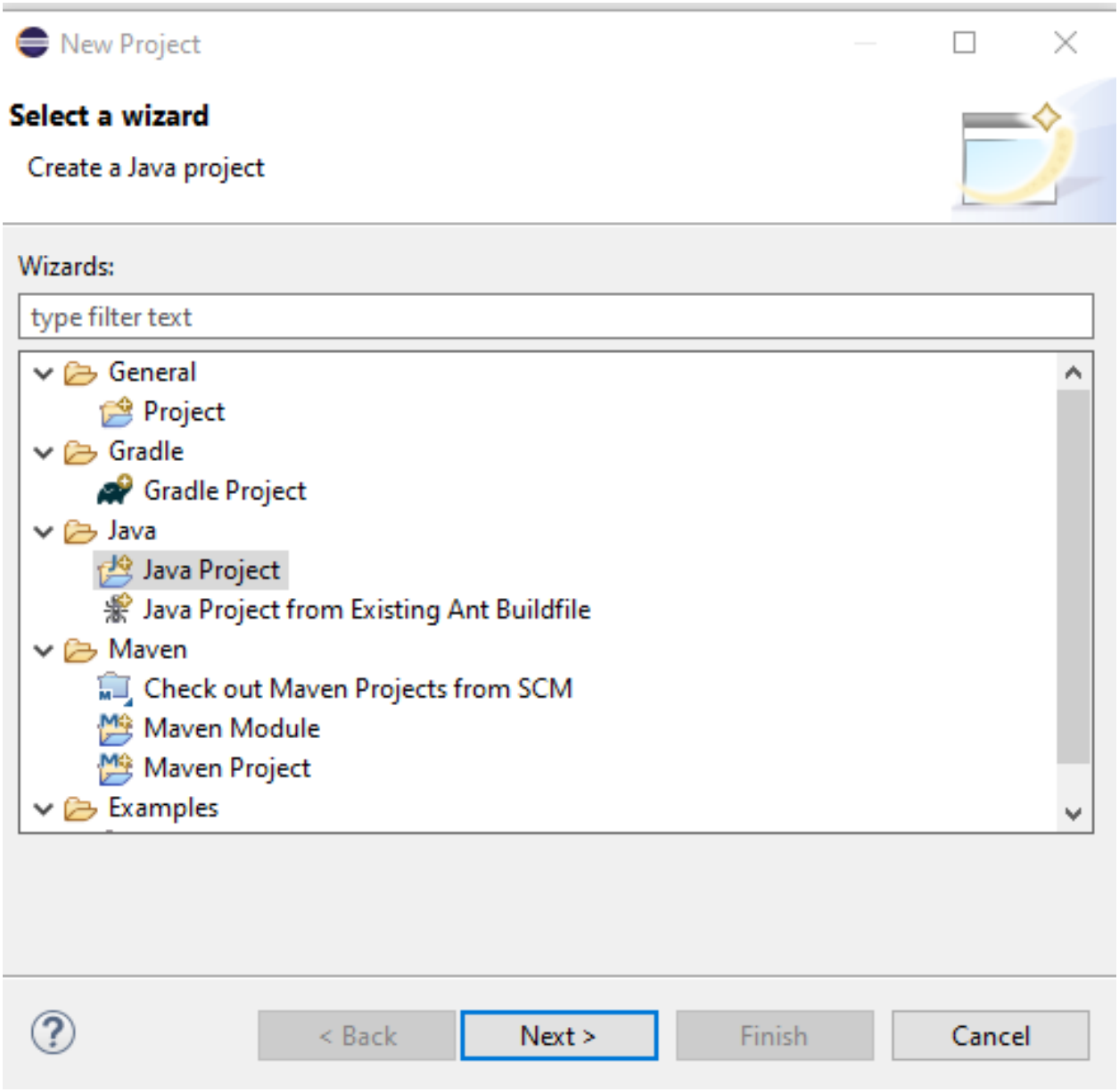
0.103 Shared Library Development - (Eclipse Java Jar Library)-5

- select create a project



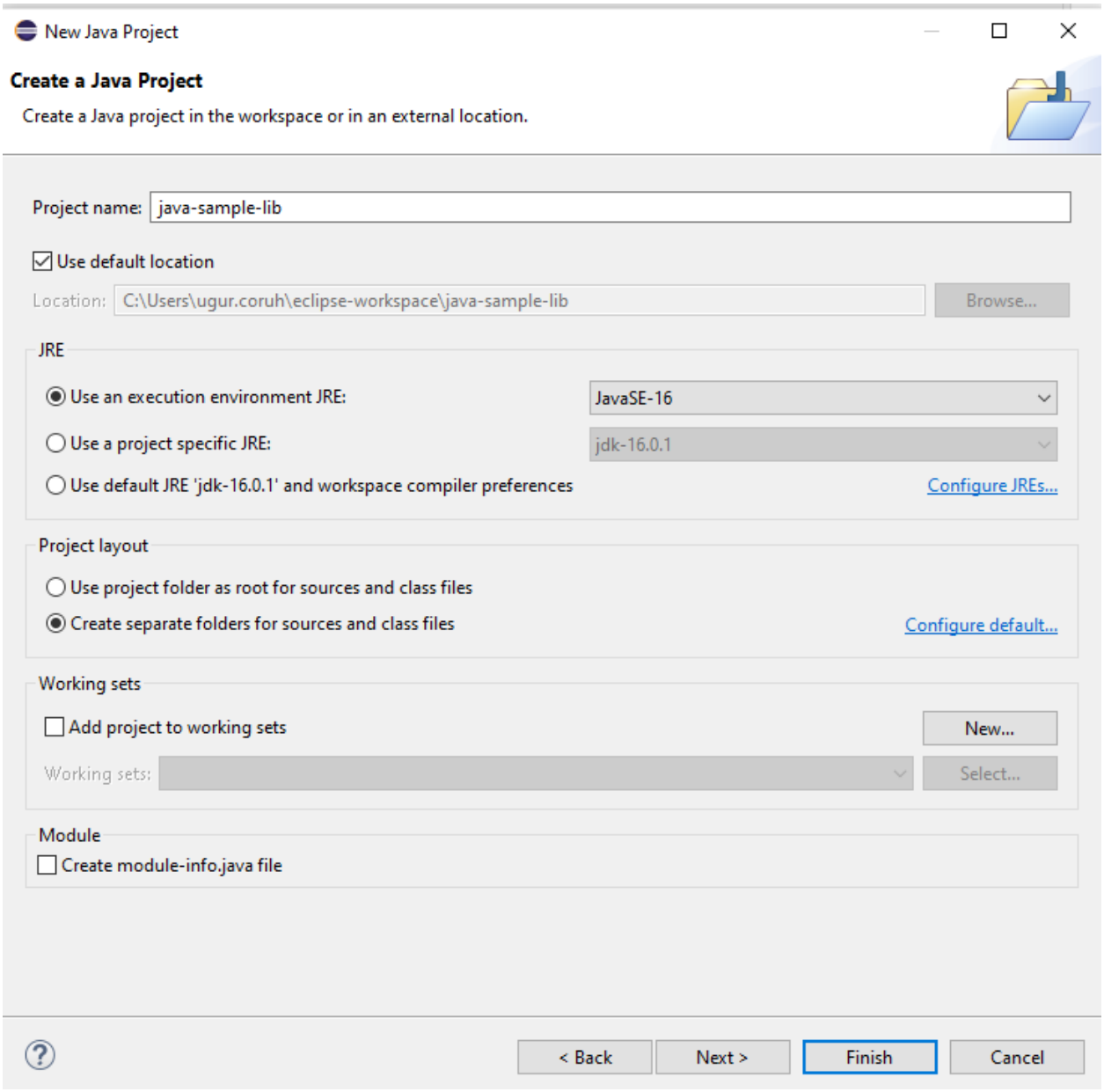
0.104 Shared Library Development - (Eclipse Java Jar Library)-6

select java project



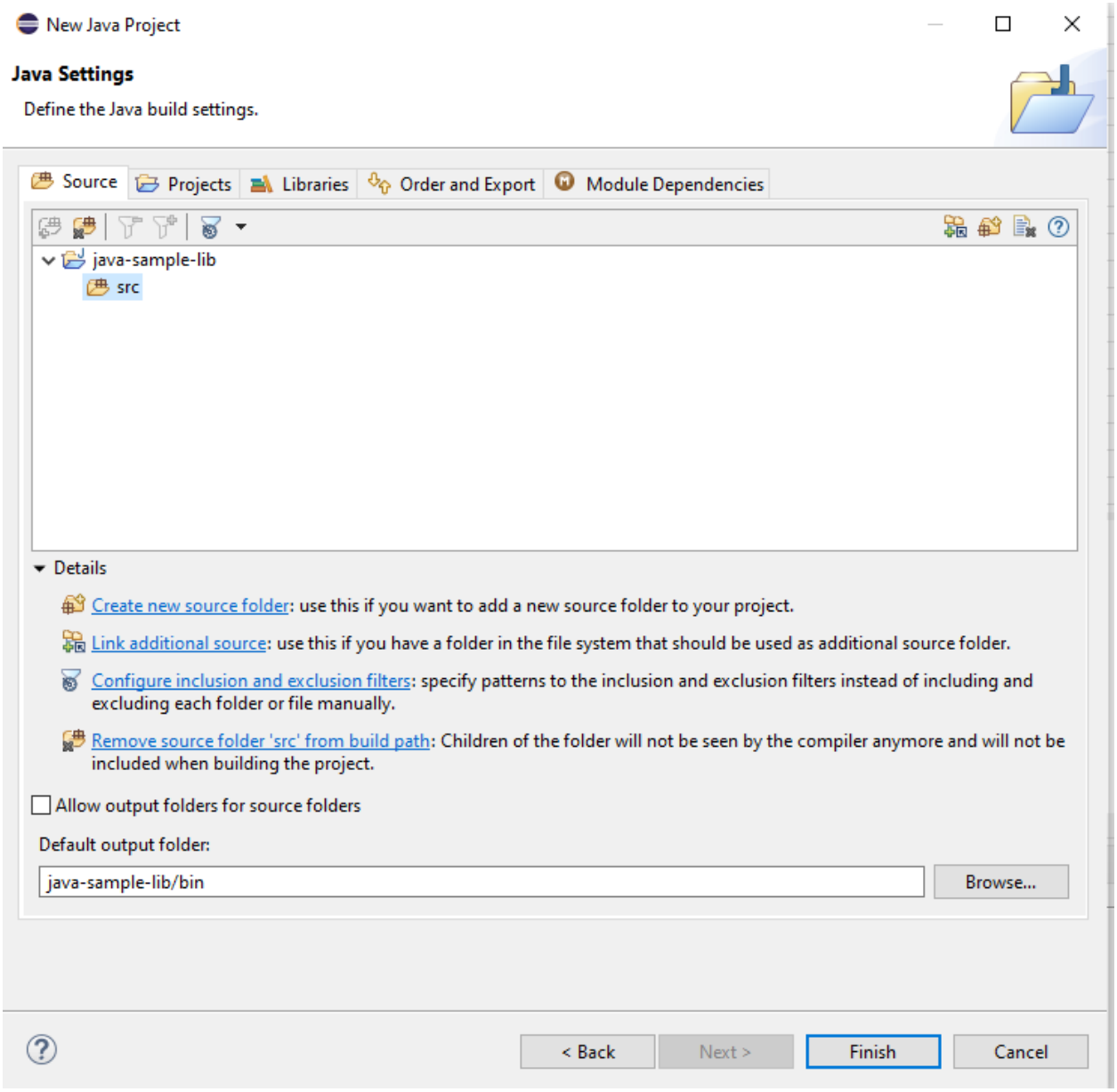
0.105 Shared Library Development - (Eclipse Java Jar Library)-7

- give project name



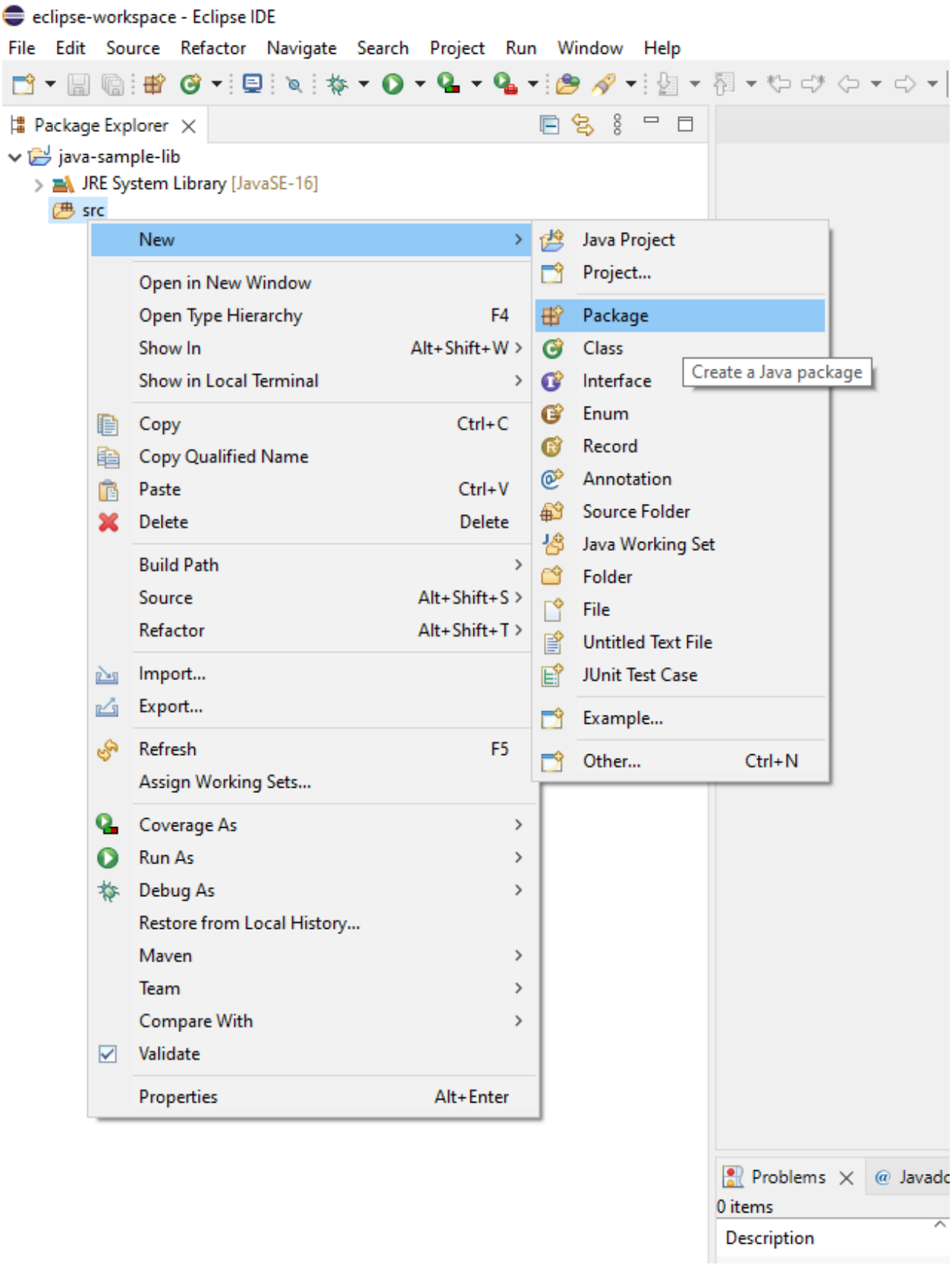
0.106 Shared Library Development - (Eclipse Java Jar Library)-8

- select finish



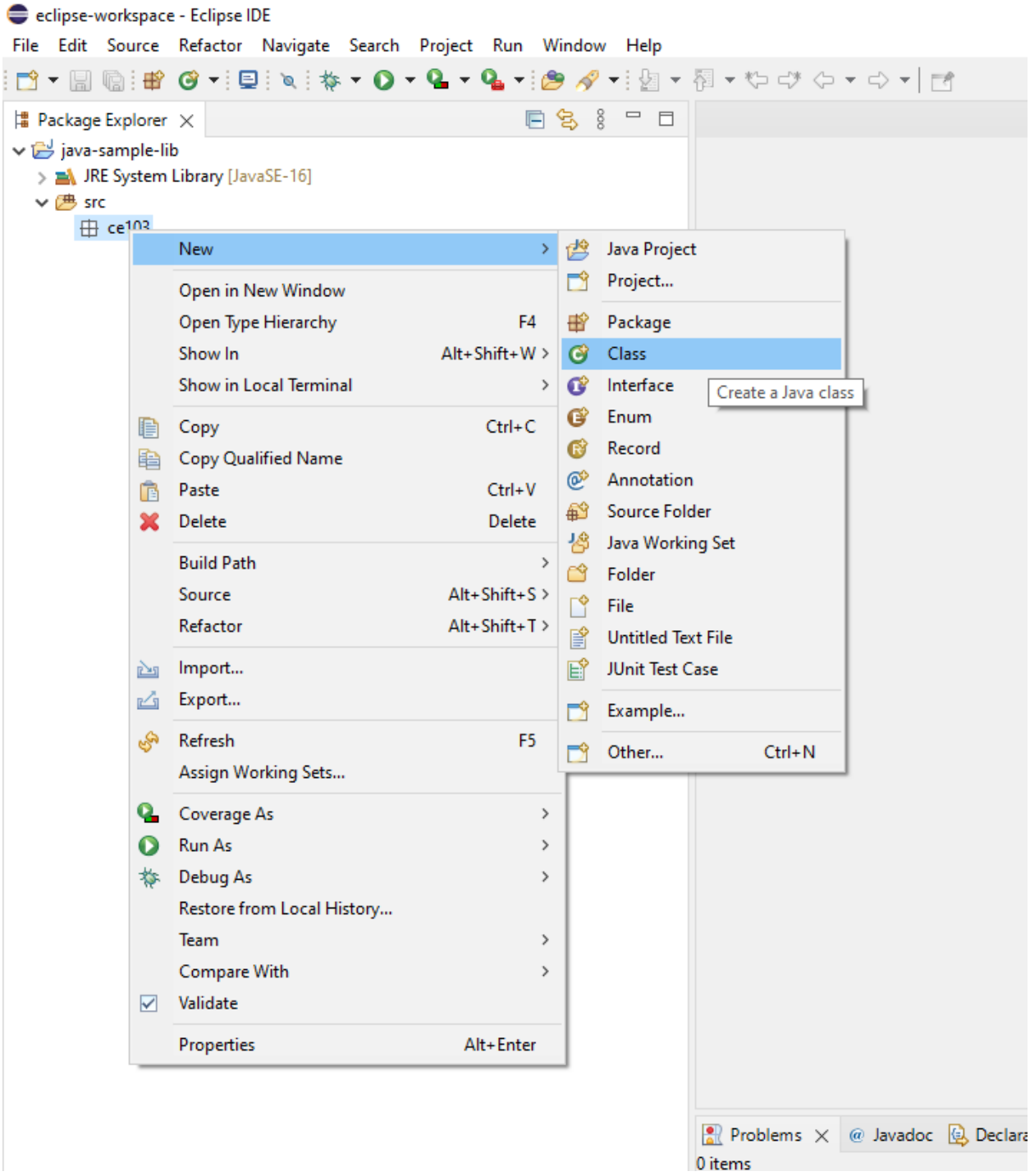
0.107 Shared Library Development - (Eclipse Java Jar Library)-9

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


eclipse-workspace - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
java-sample-lib
JRE System Library [JavaSE-16]
src
New
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Assign Working Sets...
Coverage As
Run As
Debug As
Restore from Local History...
Maven
Team
Compare With
Validate
Properties Alt+Enter
Java Project
Project...
Package
Class
Interface
Enum
Record
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
JUnit Test Case
Example...
Other... Ctrl+N
Create a Java package
Problems
0 items
Description

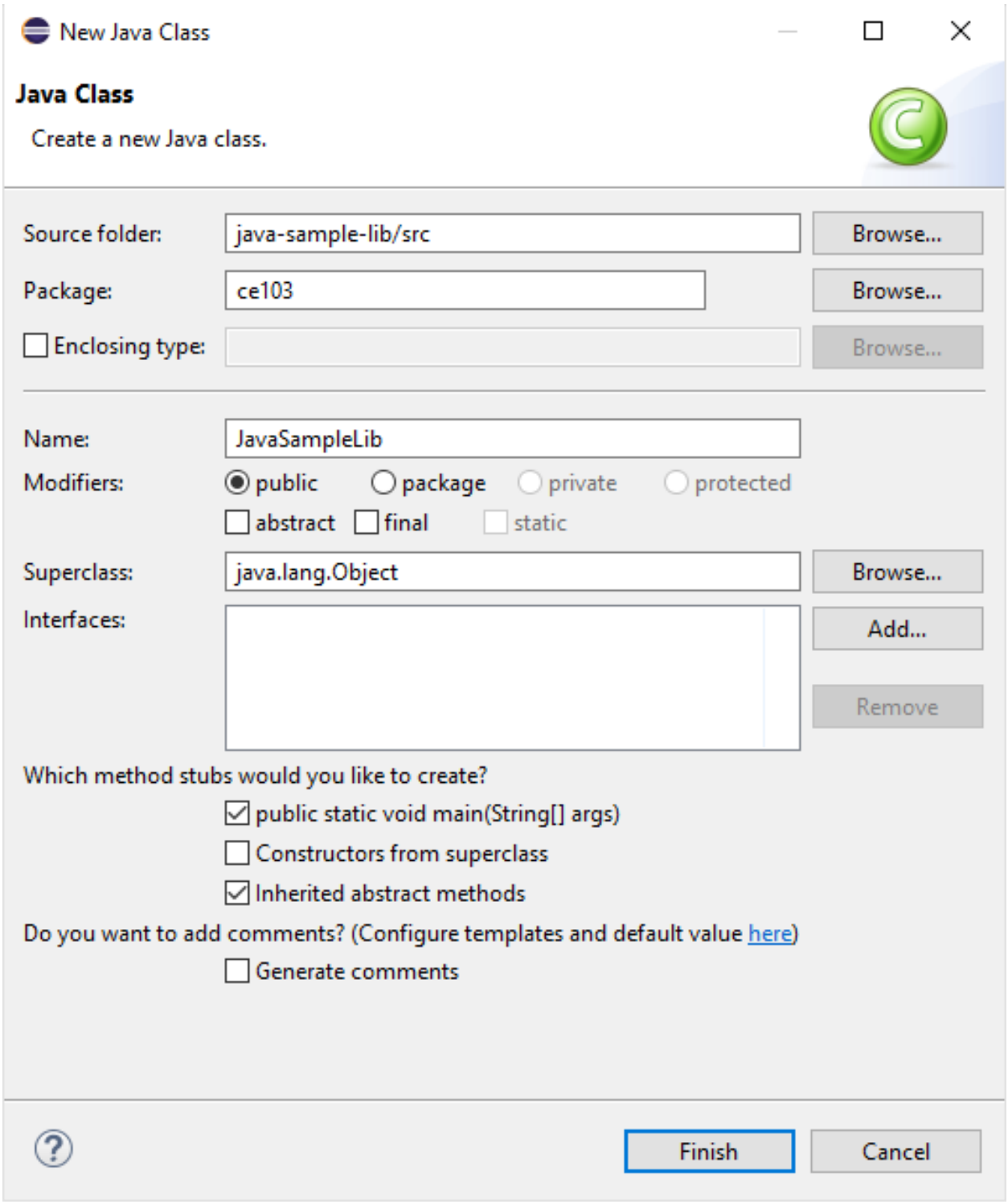
0.108 Shared Library Development - (Eclipse Java Jar Library)-10

- then we can create our class that includes our functions



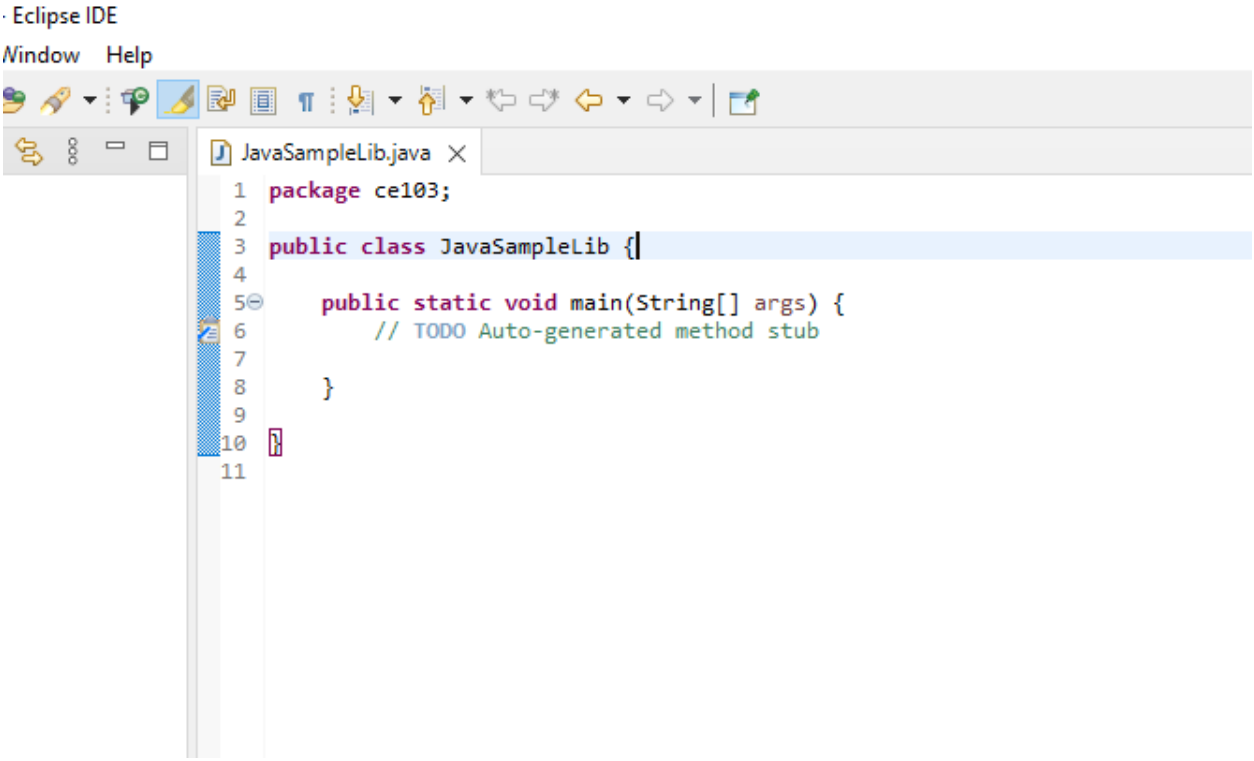
0.109 Shared Library Development - (Eclipse Java Jar Library)-11

- give class a name



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

- you will have following class with main



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

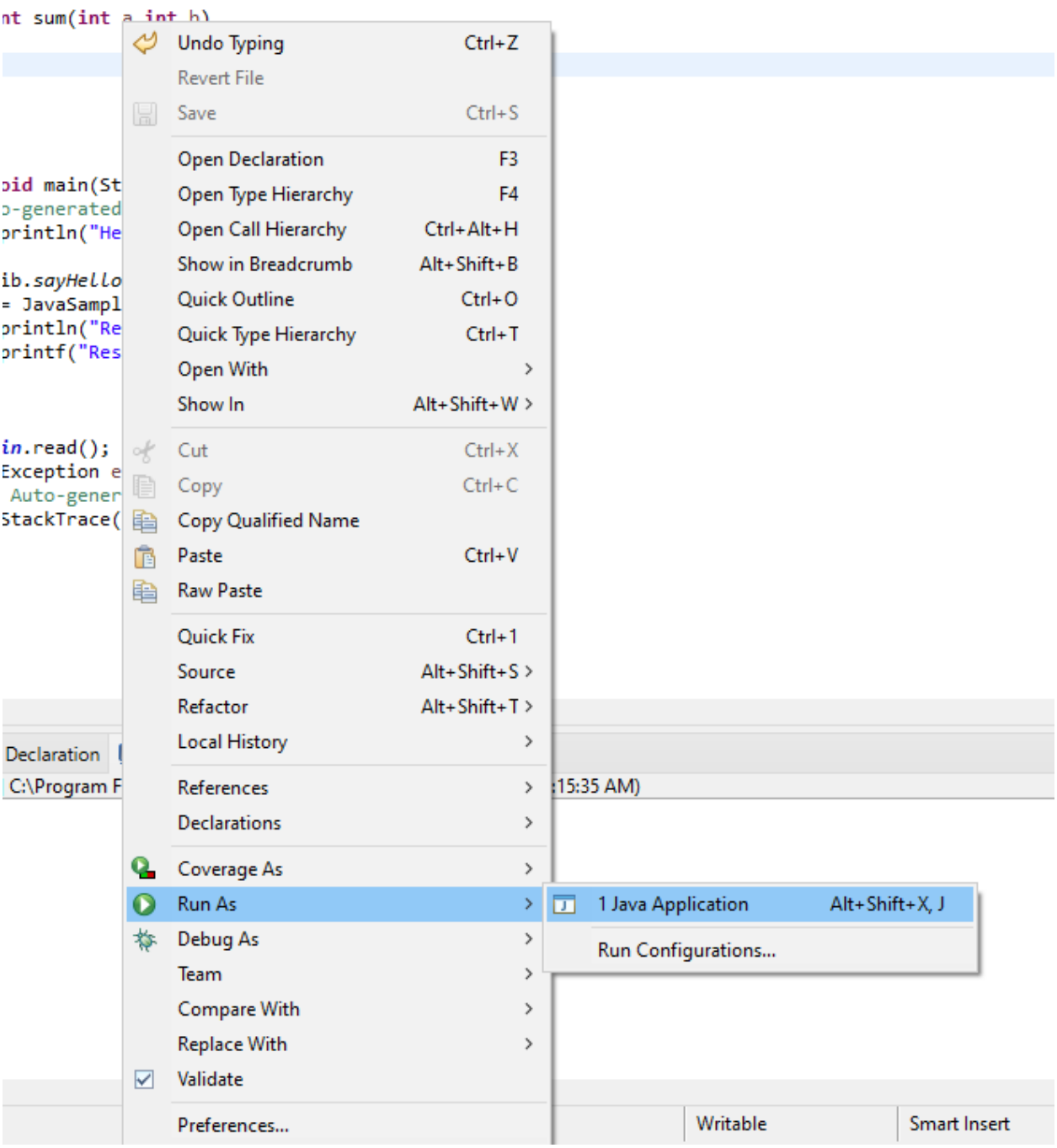
- We will create sample java library with static functions as below.

```
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();
        }

    }

}
```

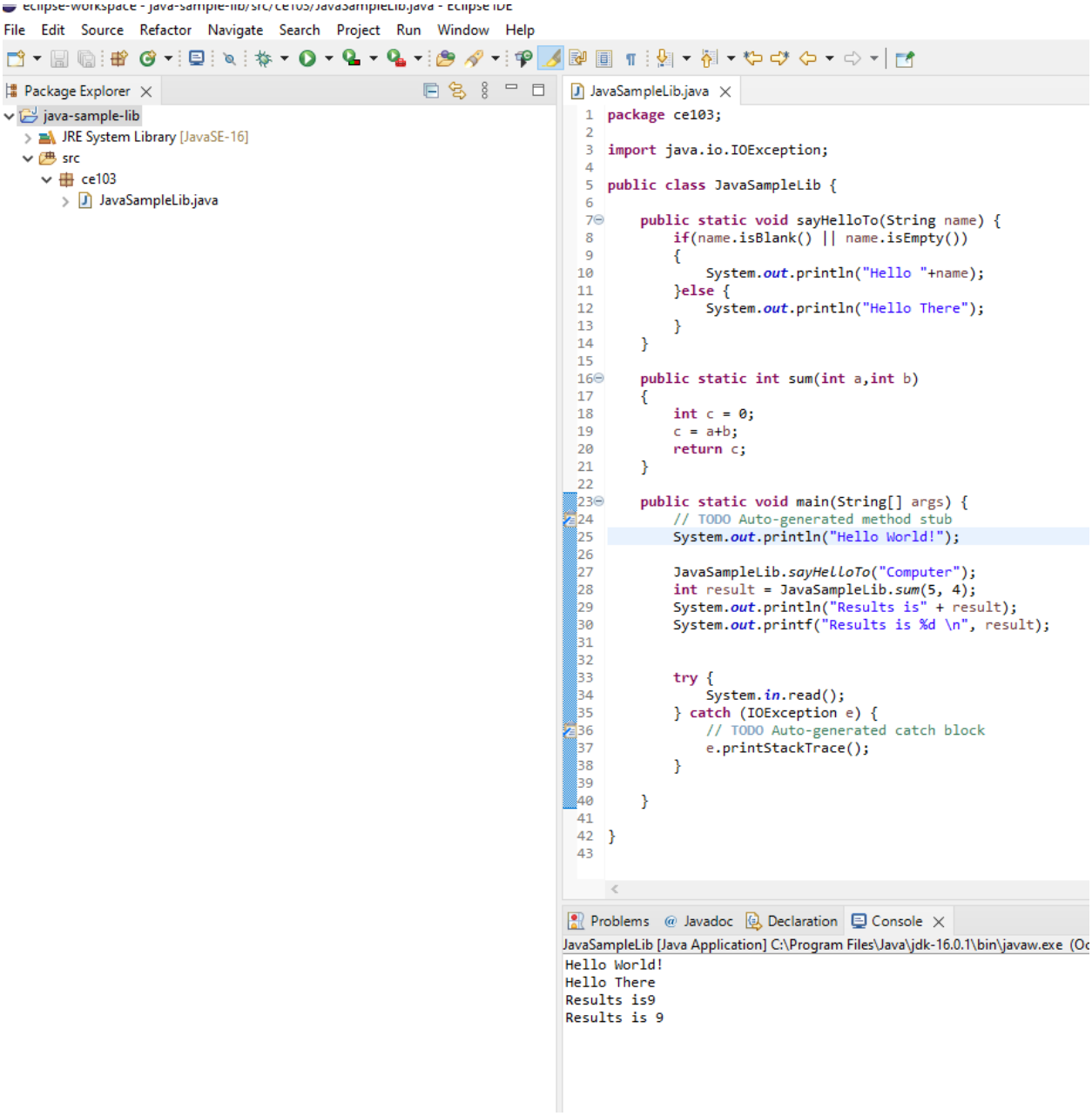
0.112 Shared Library Development - (Eclipse Java Jar Library)-14

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



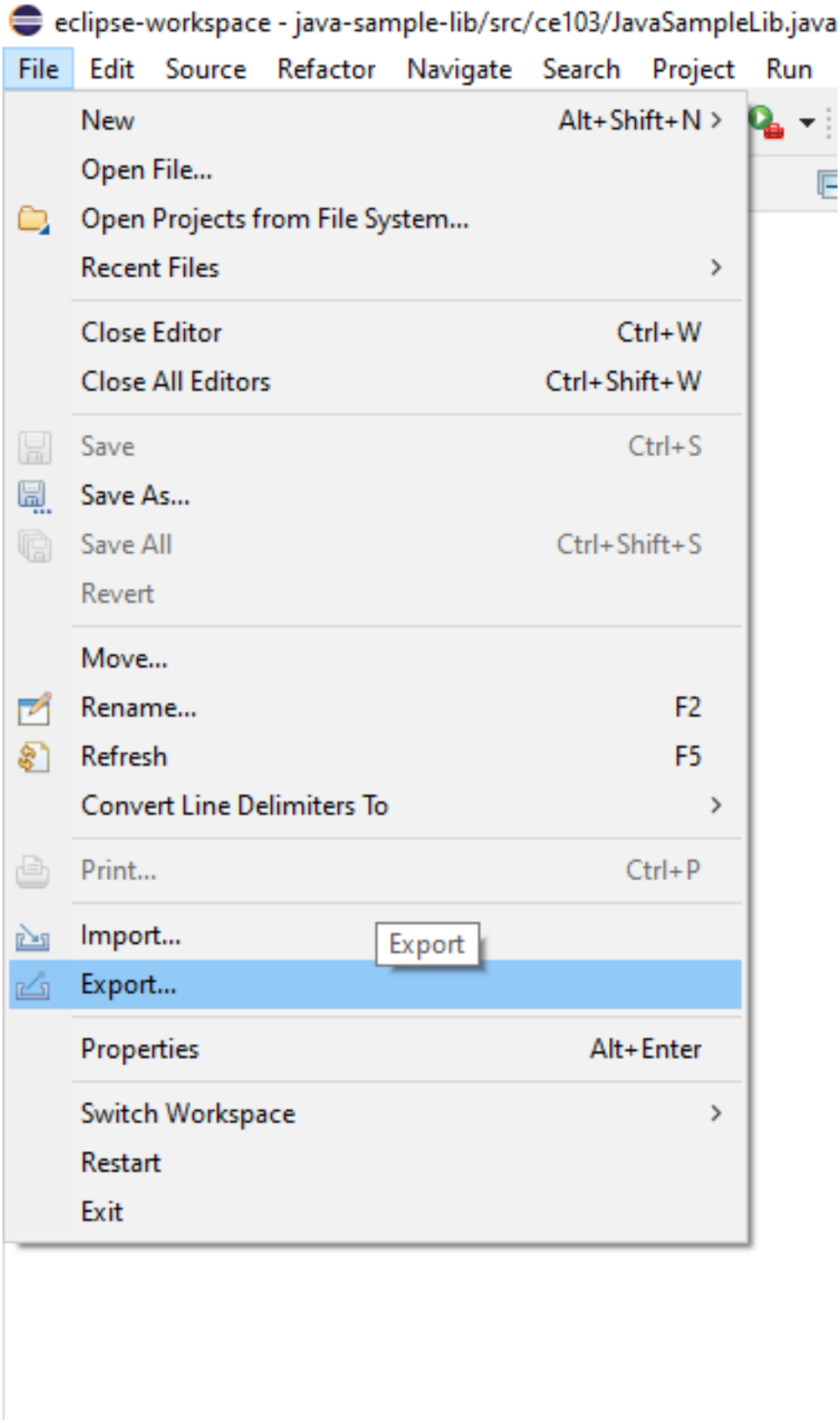
0.113 Shared Library Development - (Eclipse Java Jar Library)-15

- we can see output from console as below



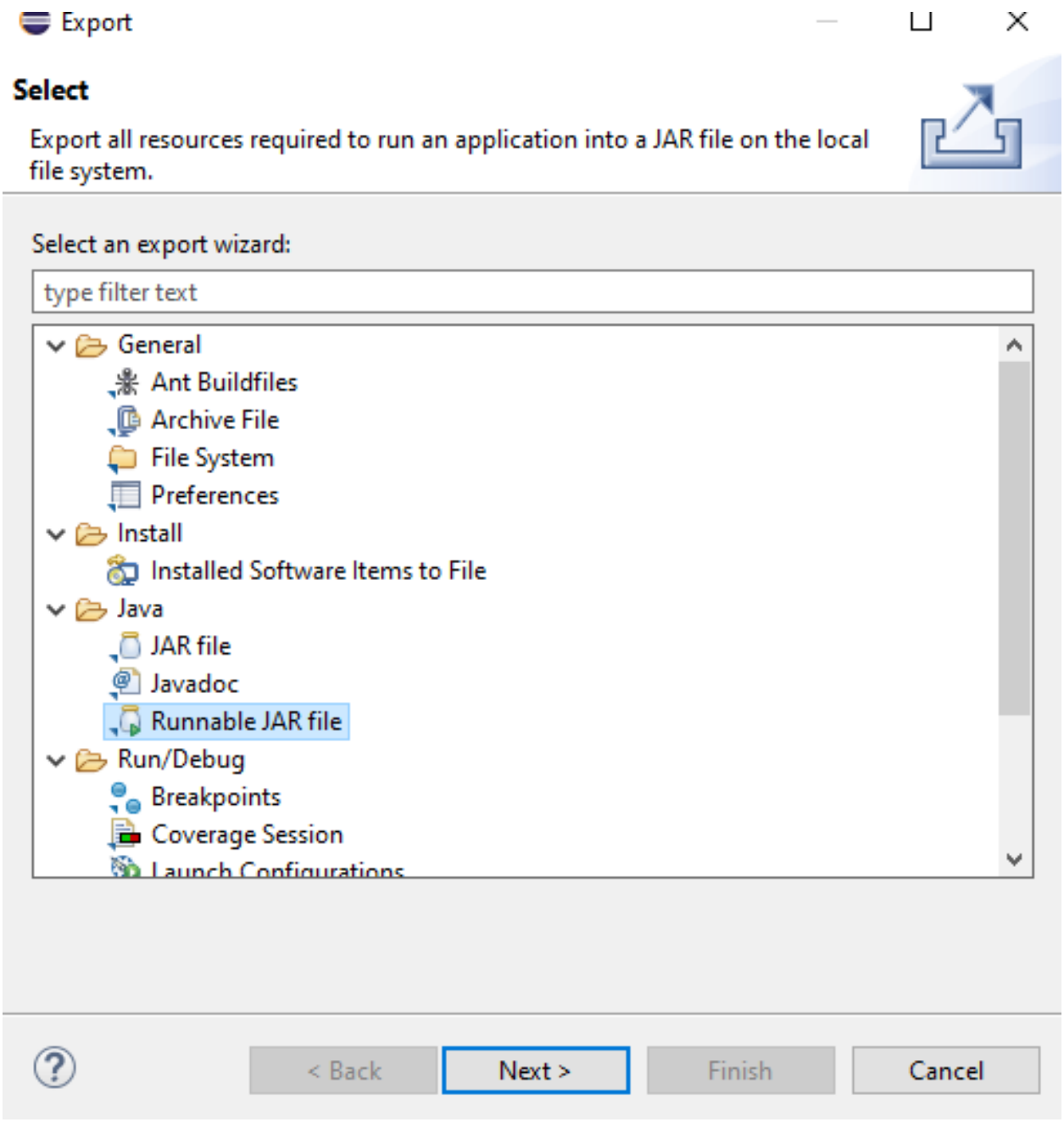
0.114 Shared Library Development - (Eclipse Java Jar Library)-16

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


eclipse-workspace - java-sample-lib/src/ce103/JavaSampleLib.java
File
Edit
Source
Refactor
Navigate
Search
Project
Run
New
Alt+Shift+N
Open File...
Open Projects from File System...
Recent Files
Close Editor
Ctrl+W
Close All Editors
Ctrl+Shift+W
Save
Ctrl+S
Save As...
Save All
Ctrl+Shift+S
Revert
Move...
Rename...
F2
Refresh
F5
Convert Line Delimiters To
Print...
Ctrl+P
Import...
Export
Export...
Properties
Alt+Enter
Switch Workspace
Restart
Exit

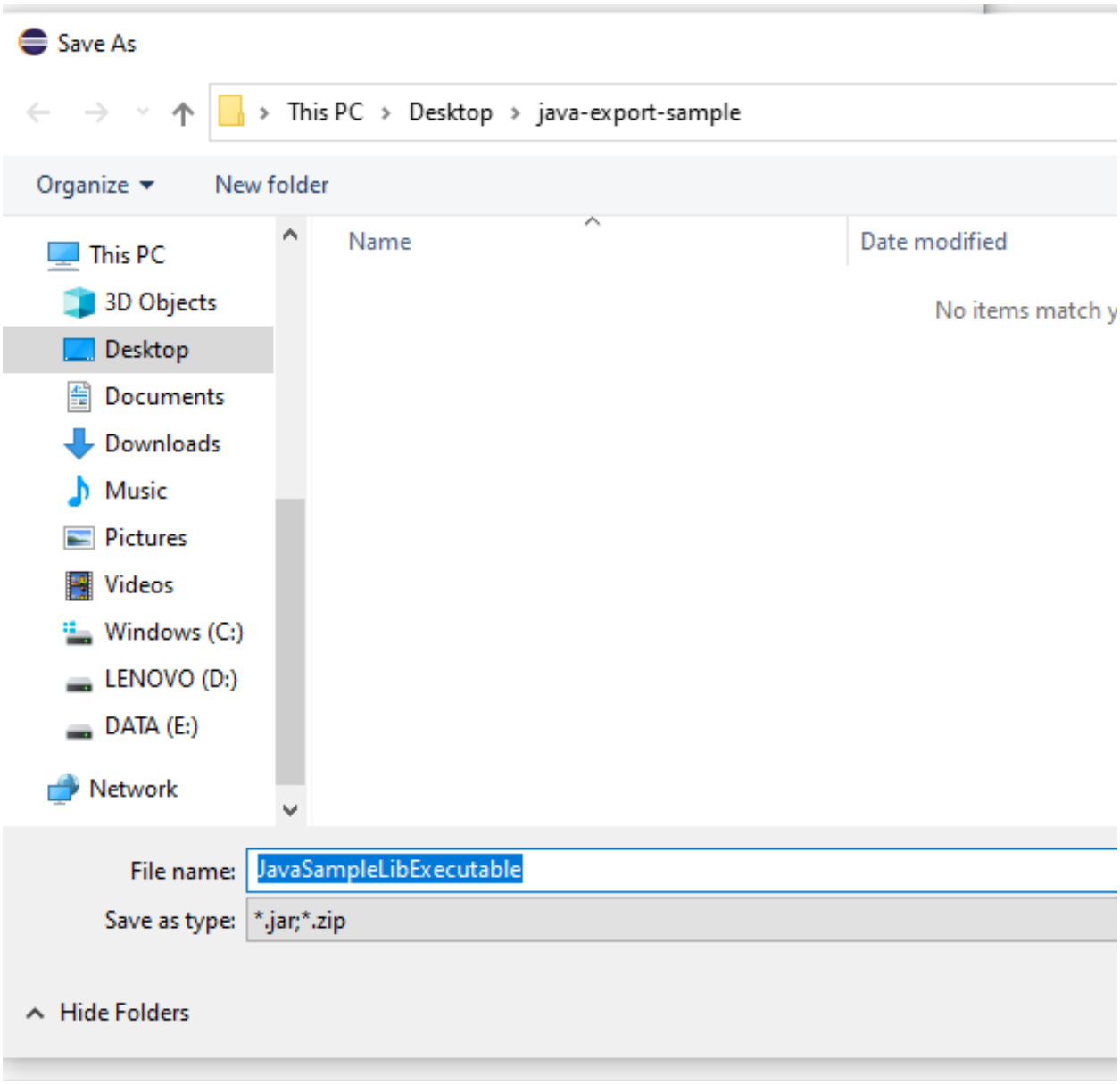
0.115 Shared Library Development - (Eclipse Java Jar Library)-17

- Select Java->Runnable JAR File



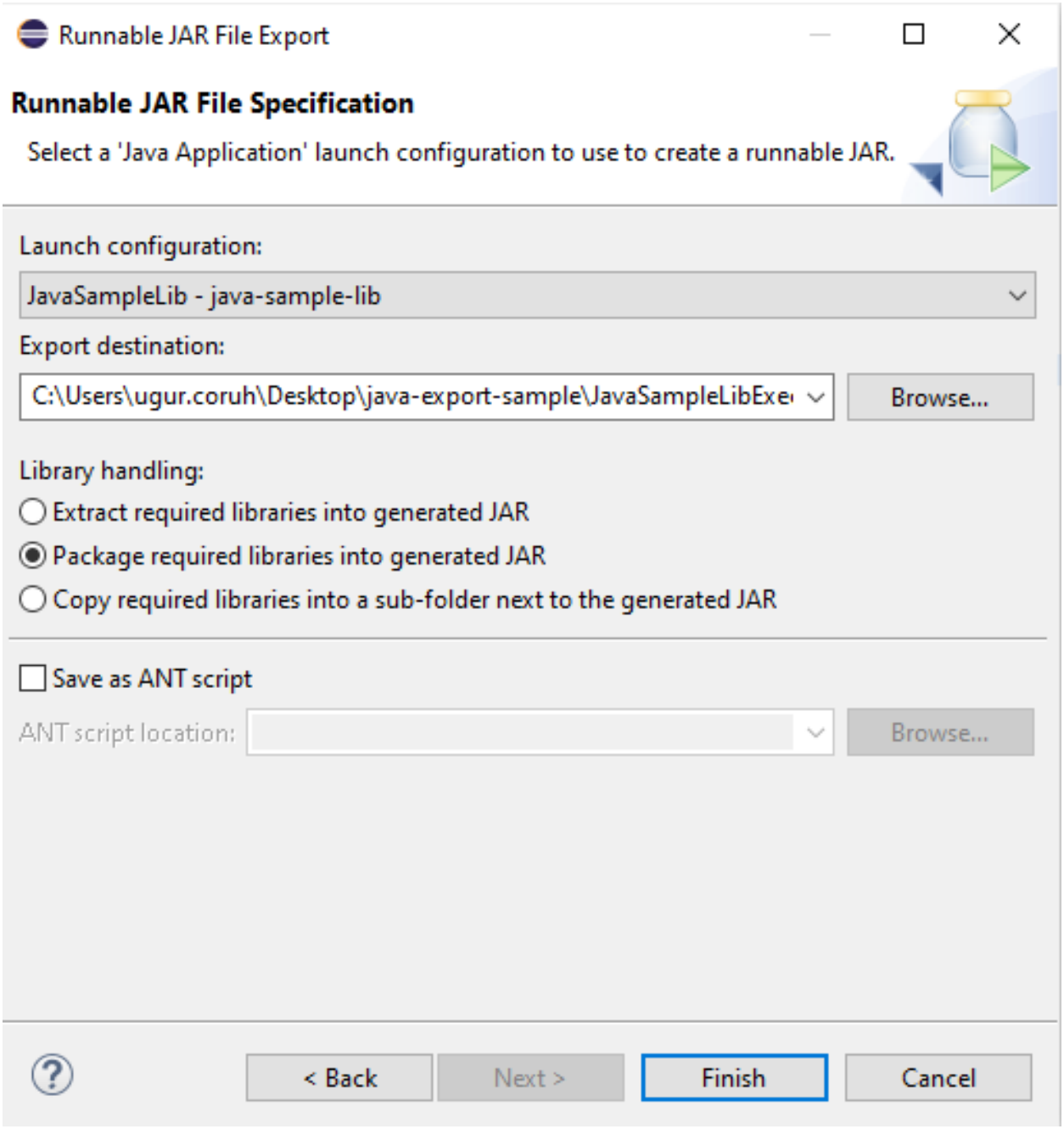
0.116 Shared Library Development - (Eclipse Java Jar Library)-18

click next and set output path for jar file



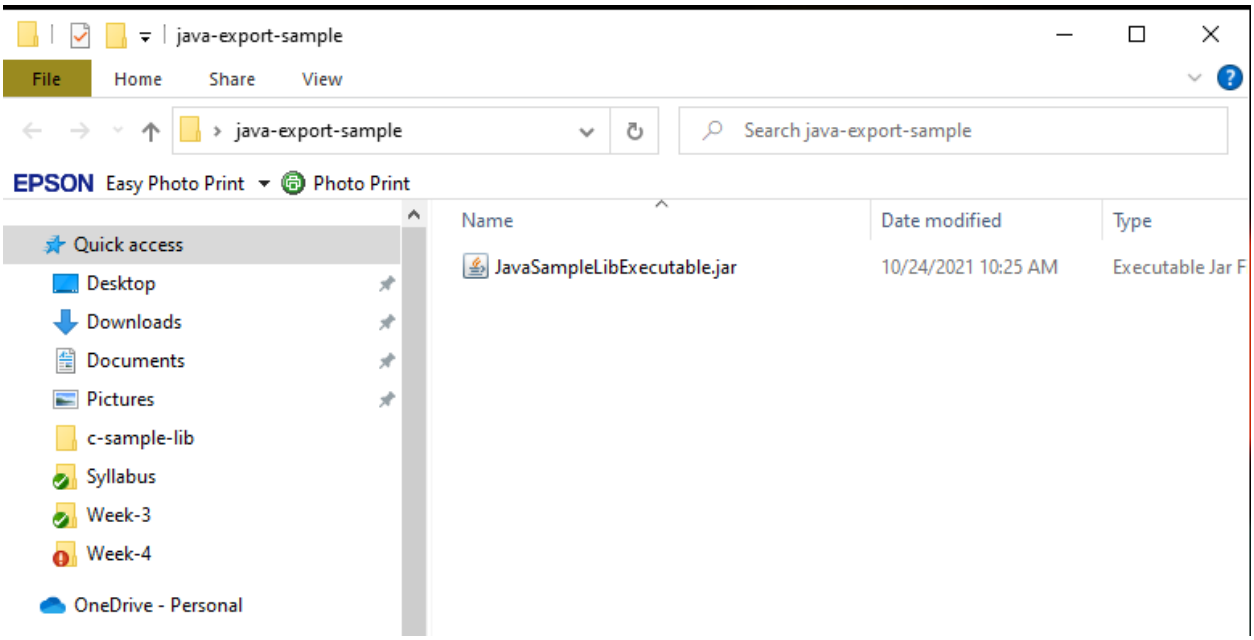
0.117 Shared Library Development - (Eclipse Java Jar Library)-19

- If our project has several external dependecy 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



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

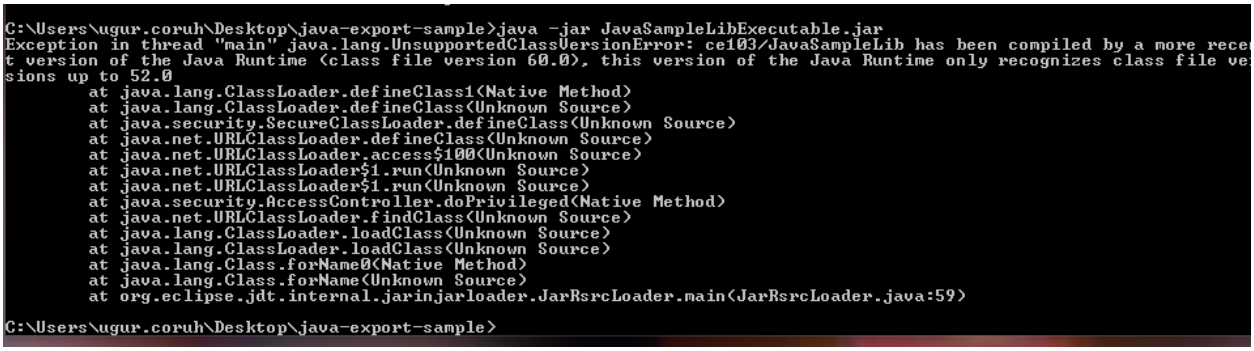
end of this operation we will have the following jar that we can by click



0.119 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
```

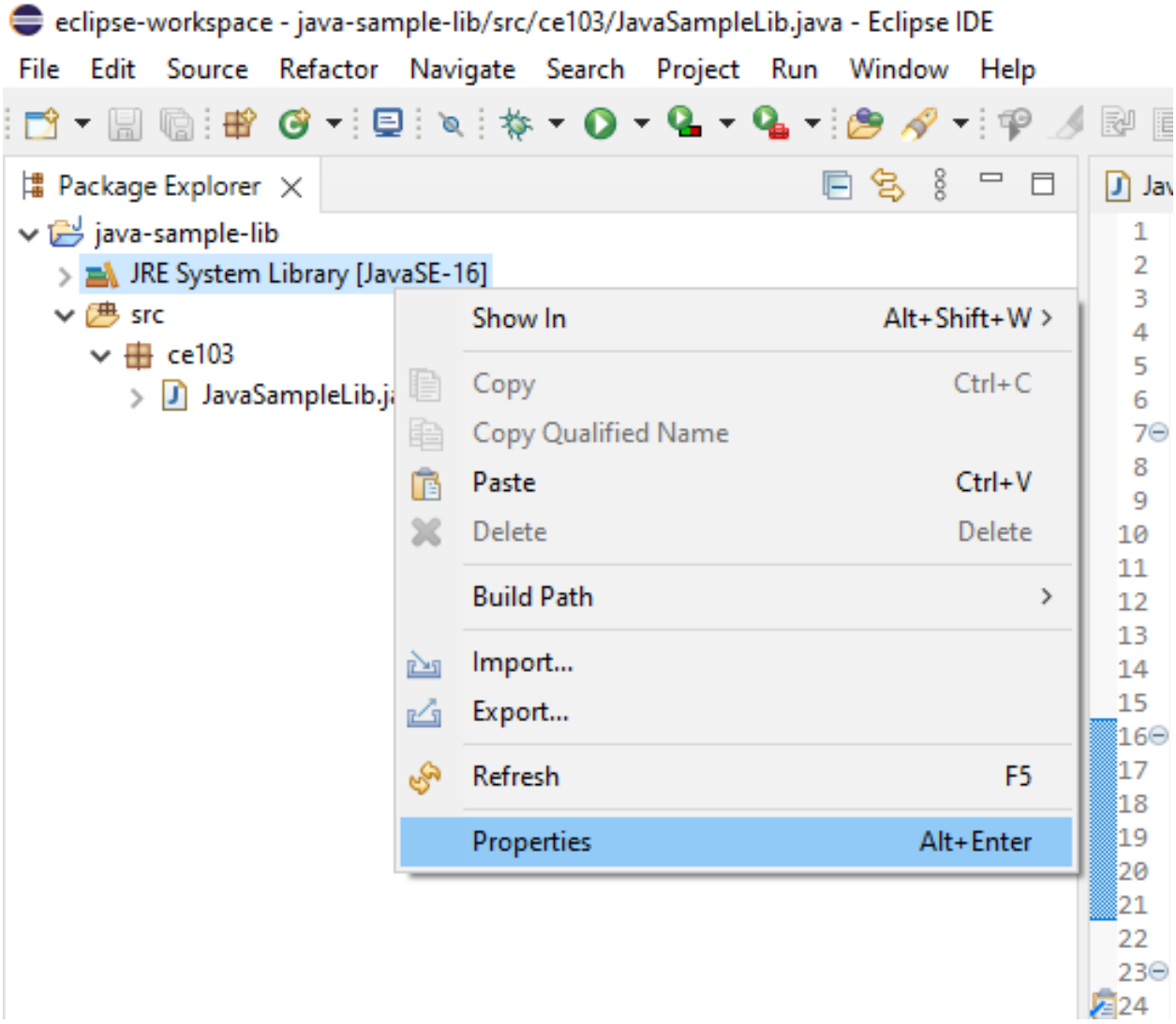


In my case eclipse build JDK is newer than that I installed and set for my OS

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

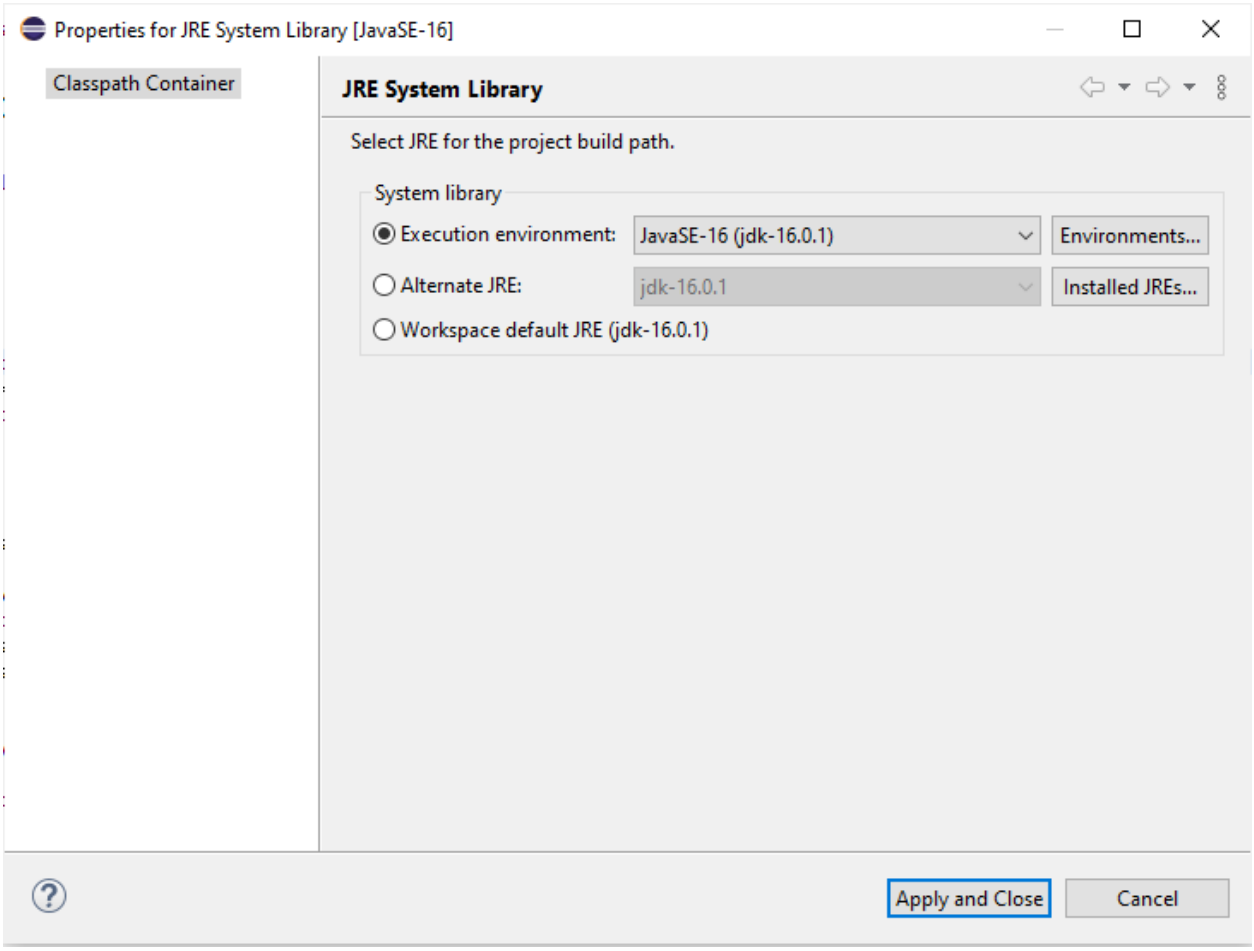
0.120 Shared Library Development - (Eclipse Java Jar Library)-22

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



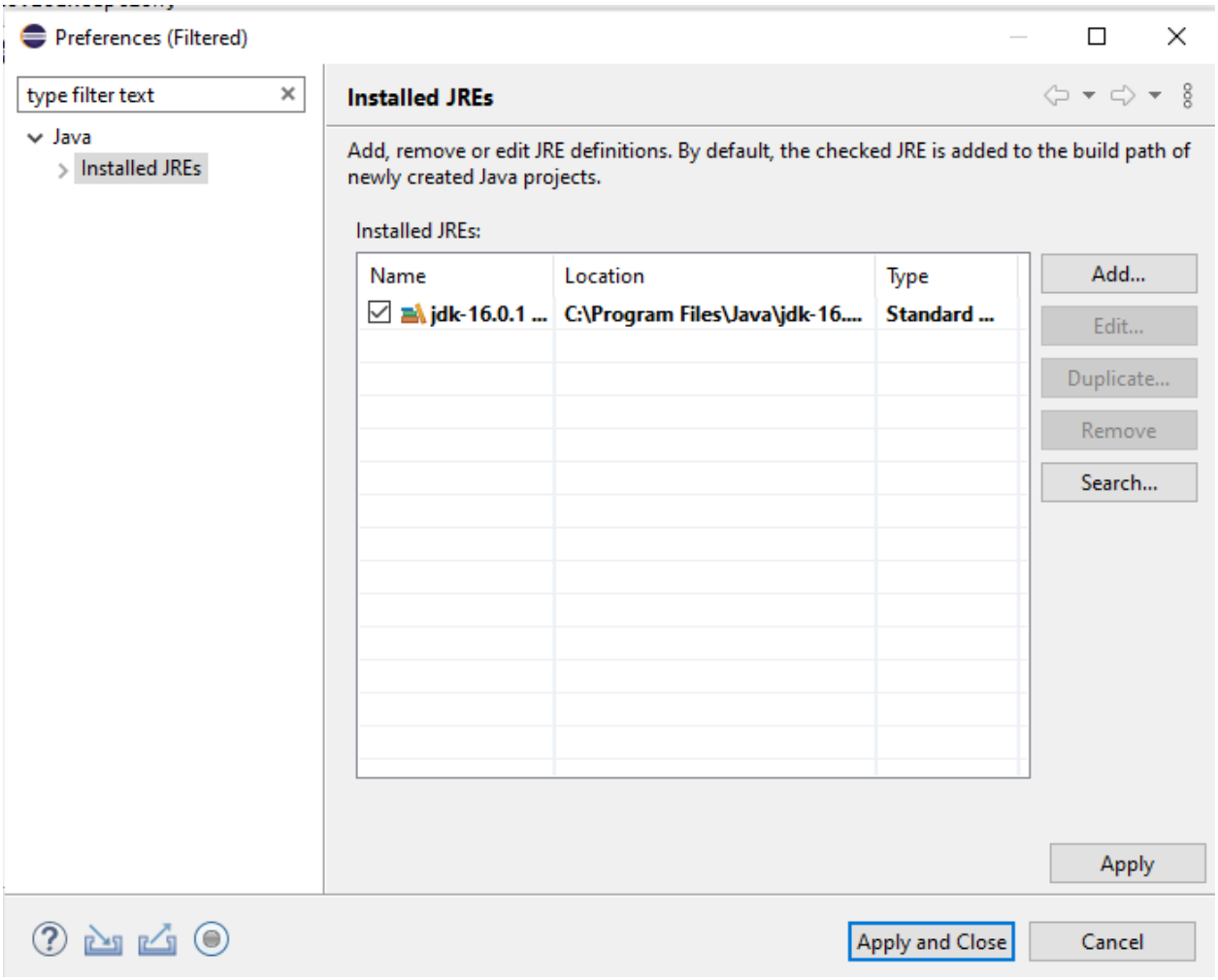
0.121 Shared Library Development - (Eclipse Java Jar Library)-23

- select environments



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

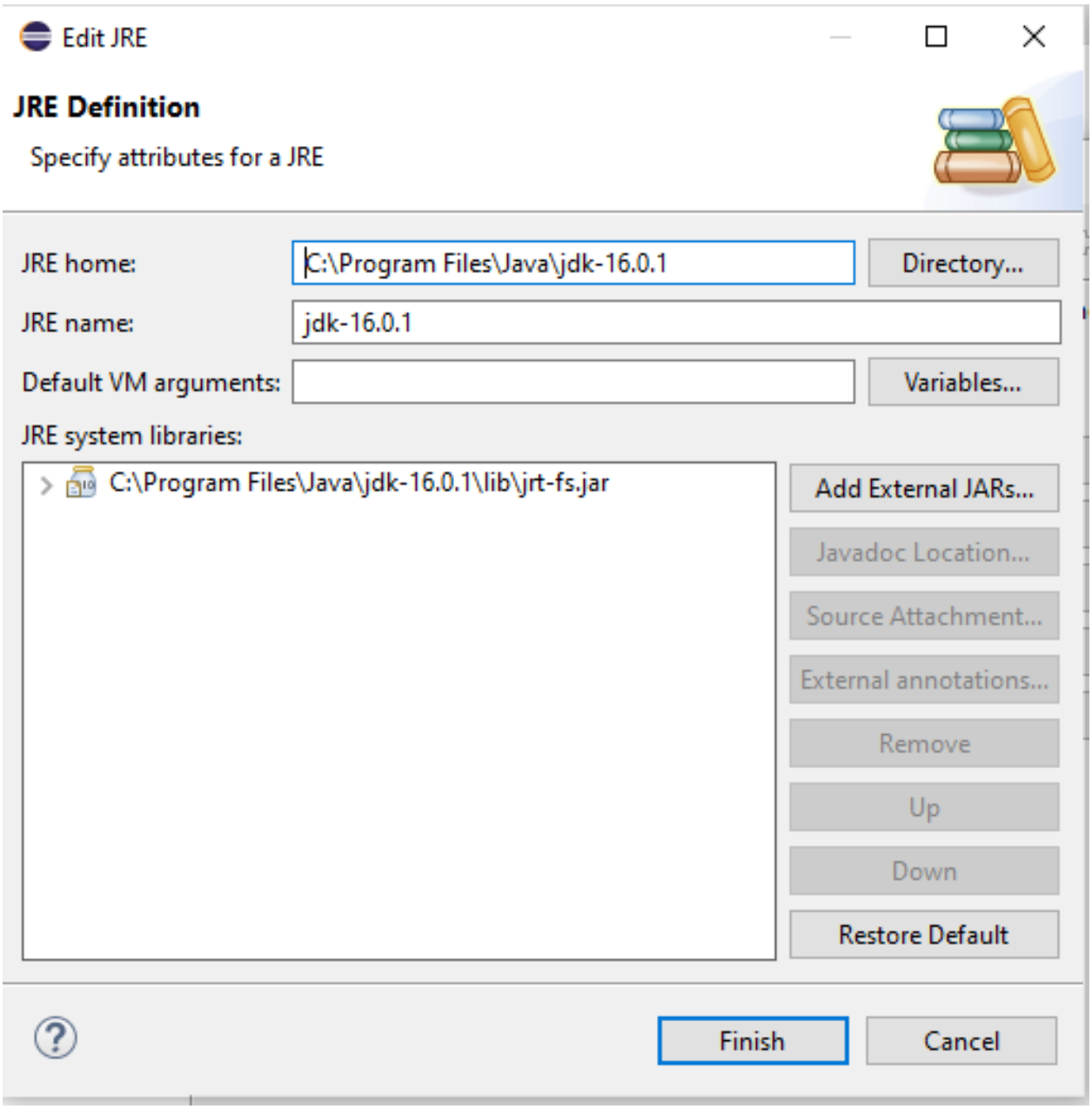
- select installed JRE or JDK



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

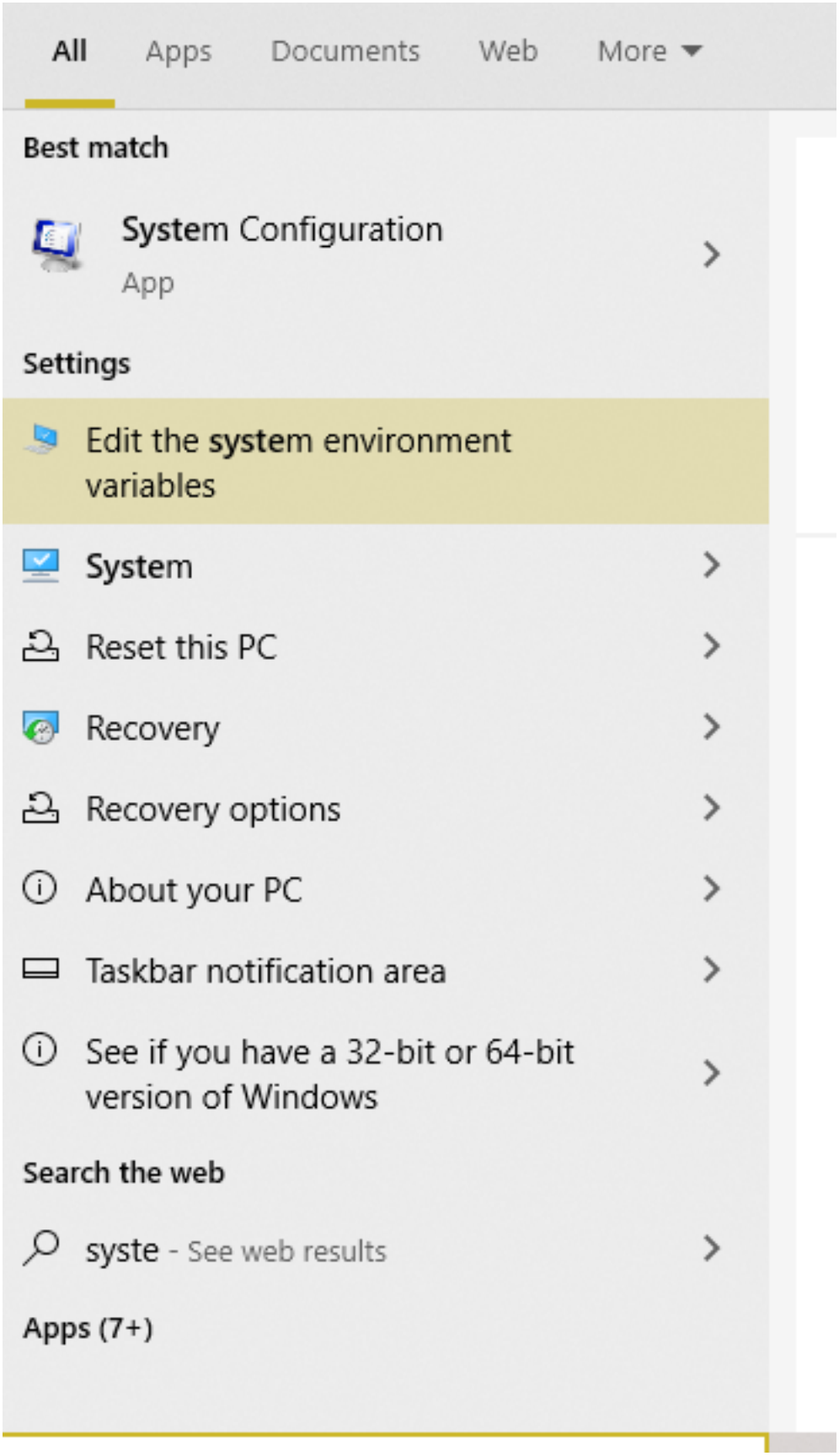
- you can see installed JRE or JDK home

```
C:\Program Files\Java\jdk-16.0.1
```

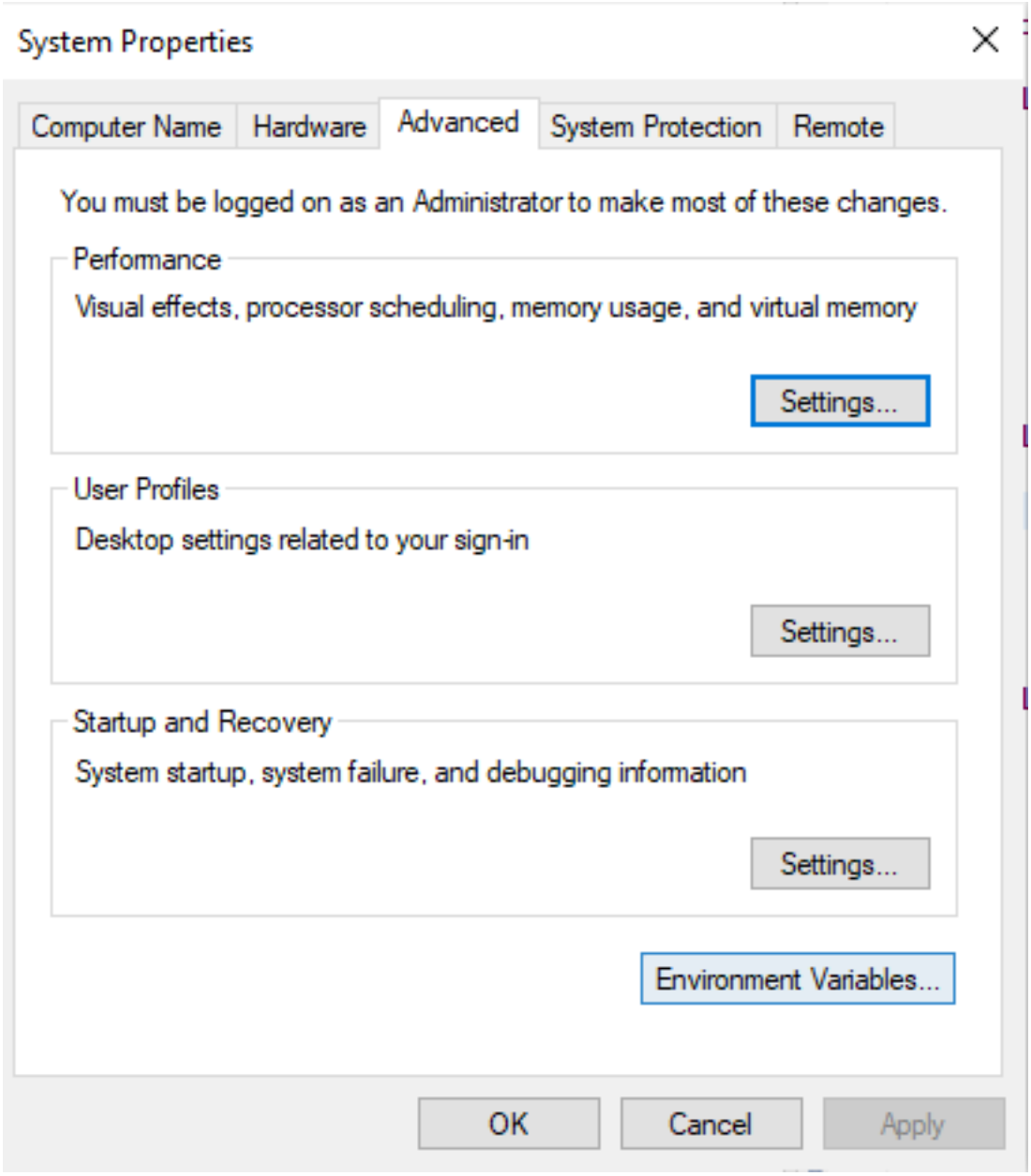


0.124 Shared Library Development - (Eclipse Java Jar Library)-26

- Open system environment to fix this problem

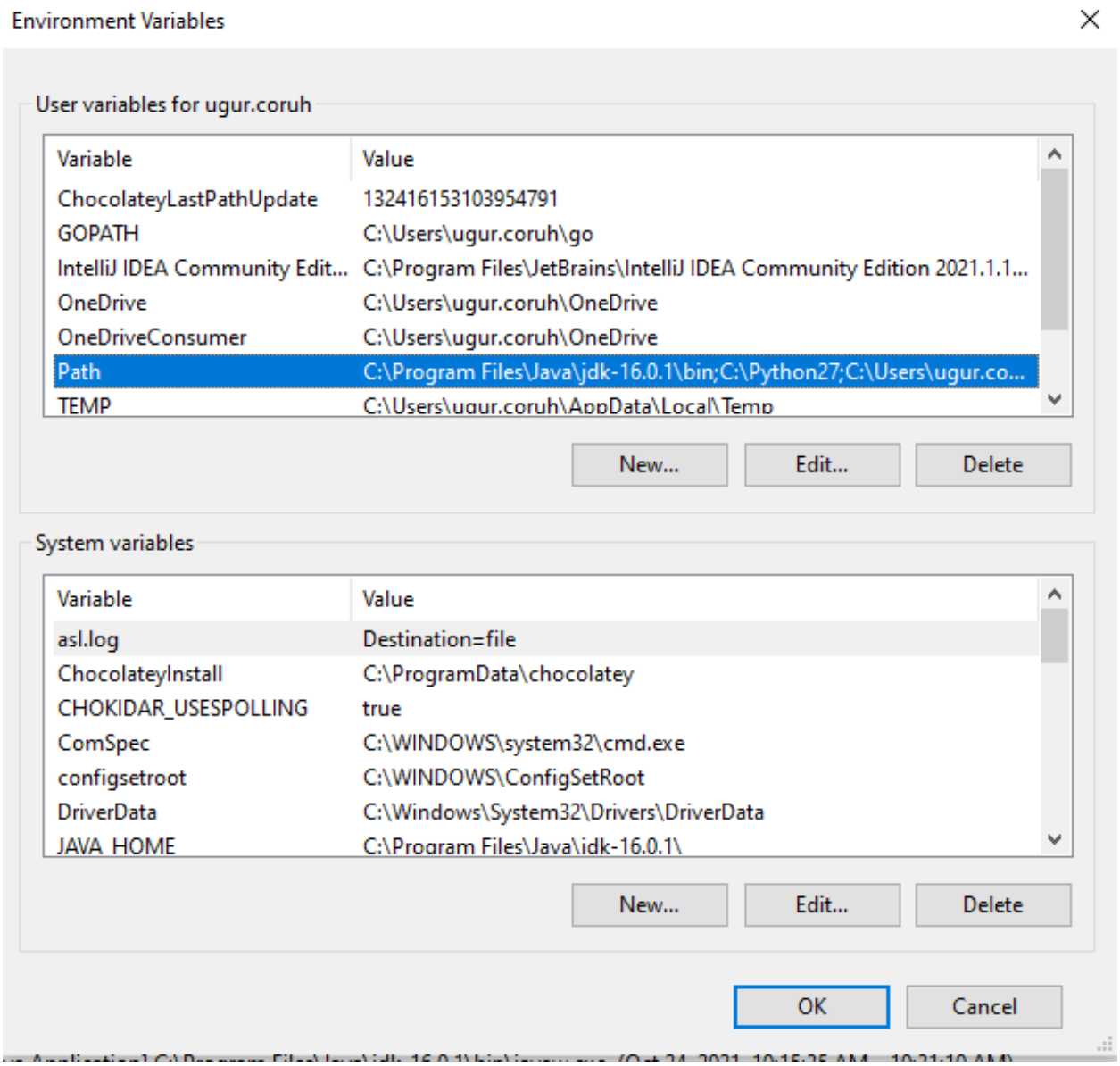

All
Apps
Documents
Web
More
Best match
System Configuration
App
Settings
Edit the system environment variables
System
Reset this PC
Recovery
Recovery options
About your PC
Taskbar notification area
See if you have a 32-bit or 64-bit version of Windows
Search the web
syste - See web results
Apps (7+)

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

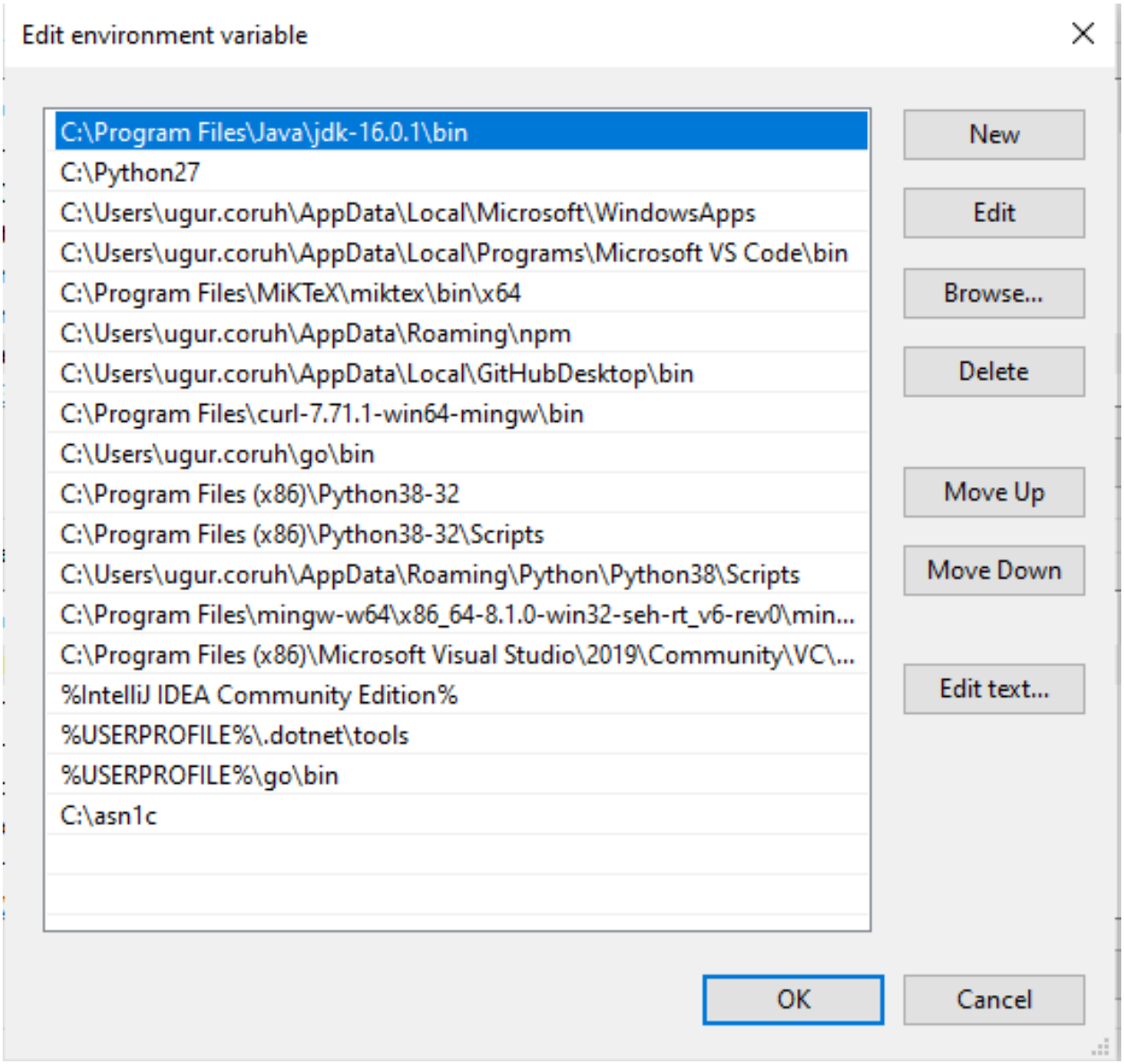


0.126 Shared Library Development - (Eclipse Java Jar Library)-28

- Check user settings first

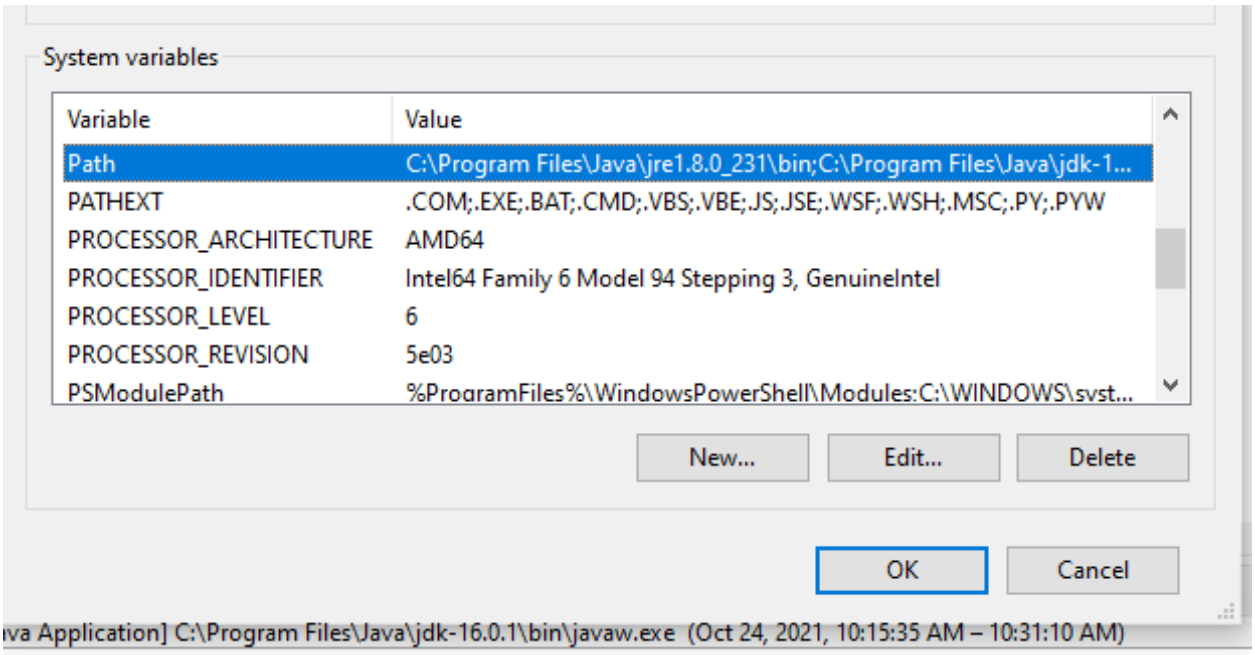

Environment Variables
User variables for ugur.coruh
Variable
Value
ChocolateyLastPathUpdate
132416153103954791
GOPATH
C:\Users\ugur.coruh\go
IntelliJ IDEA Community Edit...
C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.1.1...
OneDrive
C:\Users\ugur.coruh\OneDrive
OneDriveConsumer
C:\Users\ugur.coruh\OneDrive
Path
C:\Program Files\Java\jdk-16.0.1\bin;C:\Python27;C:\Users\ugur.co...
TEMP
C:\Users\ugur.coruh\AppData\Local\Temp
New...
Edit...
Delete
System variables
Variable
Value
asl.log
Destination=file
ChocolateyInstall
C:\ProgramData\chocolatey
CHOKIDAR_USESPOLLING
true
ComSpec
C:\WINDOWS\system32\cmd.exe
configsetroot
C:\WINDOWS\ConfigSetRoot
DriverData
C:\Windows\System32\Drivers\DriverData
JAVA_HOME
C:\Program Files\Java\jdk-16.0.1\
New...
Edit...
Delete
OK
Cancel

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

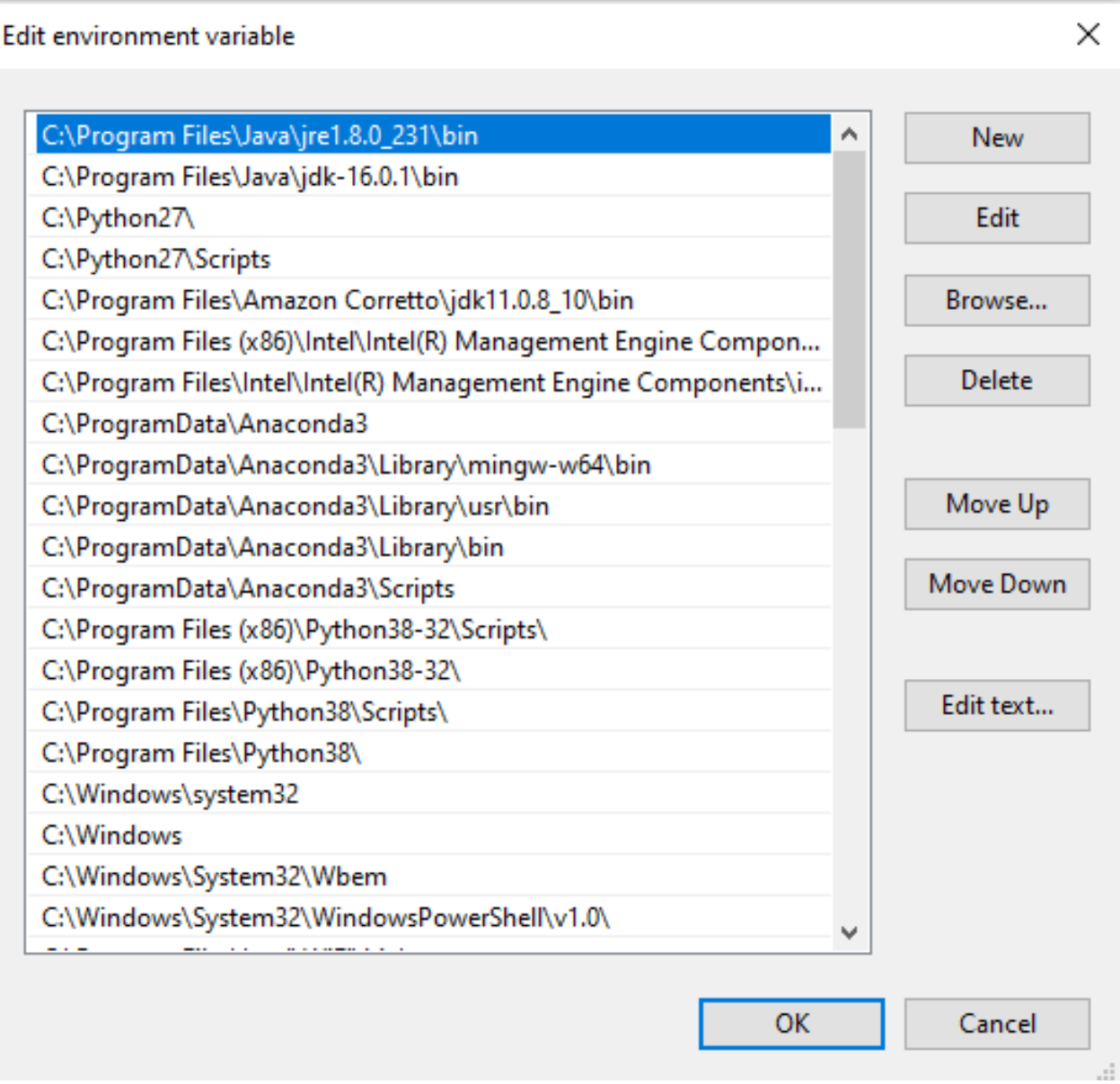


0.128 Shared Library Development - (Eclipse Java Jar Library)-30

- Check system settings

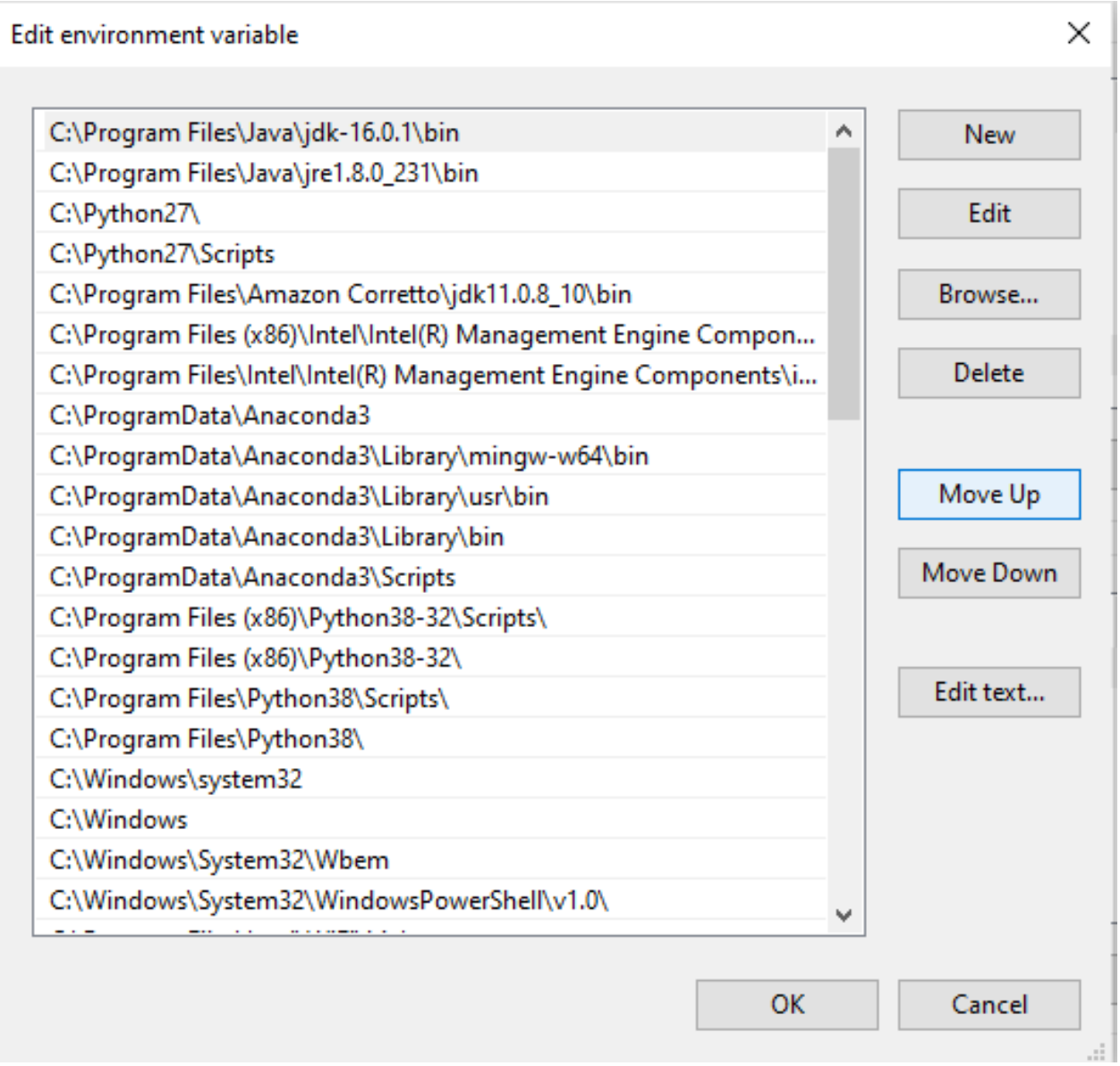

System variables
Variable
Value
Path
C:\Program Files\Java\jre1.8.0_231\bin;C:\Program Files\Java\jdk-1...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW
PROCESSOR_ARCHITECTURE
AMD64
PROCESSOR_IDENTIFIER
Intel64 Family 6 Model 94 Stepping 3, GenuineIntel
PROCESSOR_LEVEL
6
PROCESSOR_REVISION
5e03
PSModulePath
%ProgramFiles%\WindowsPowerShell\Modules;C:\WINDOWS\syst...
New...
Edit...
Delete
OK
Cancel
va Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (Oct 24, 2021, 10:15:35 AM – 10:31:10 AM)

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



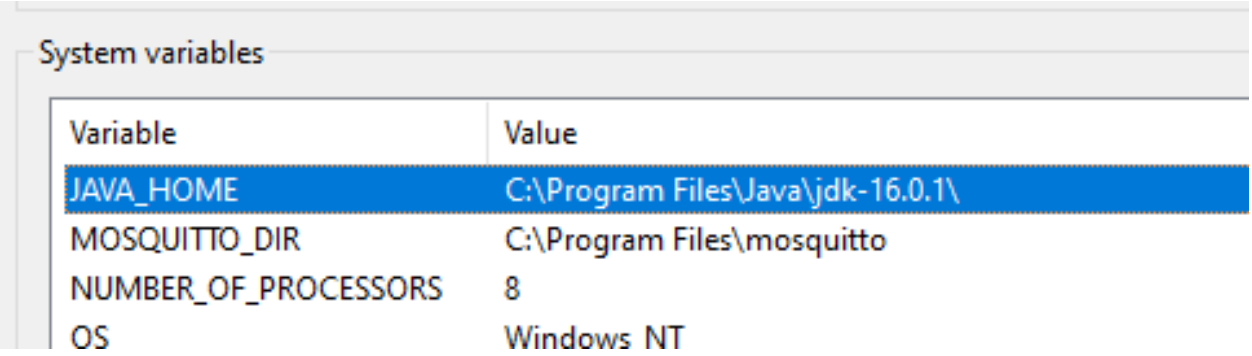
0.130 Shared Library Development - (Eclipse Java Jar Library)-32

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



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

- Also in system setting check JAVA_HOME

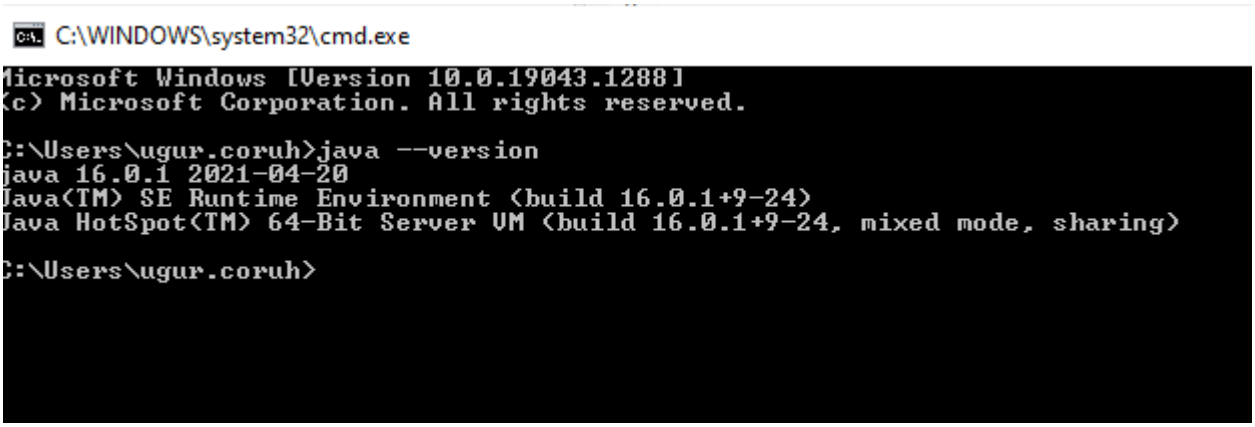


0.132 Shared Library Development - (Eclipse Java Jar Library)-34

- 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



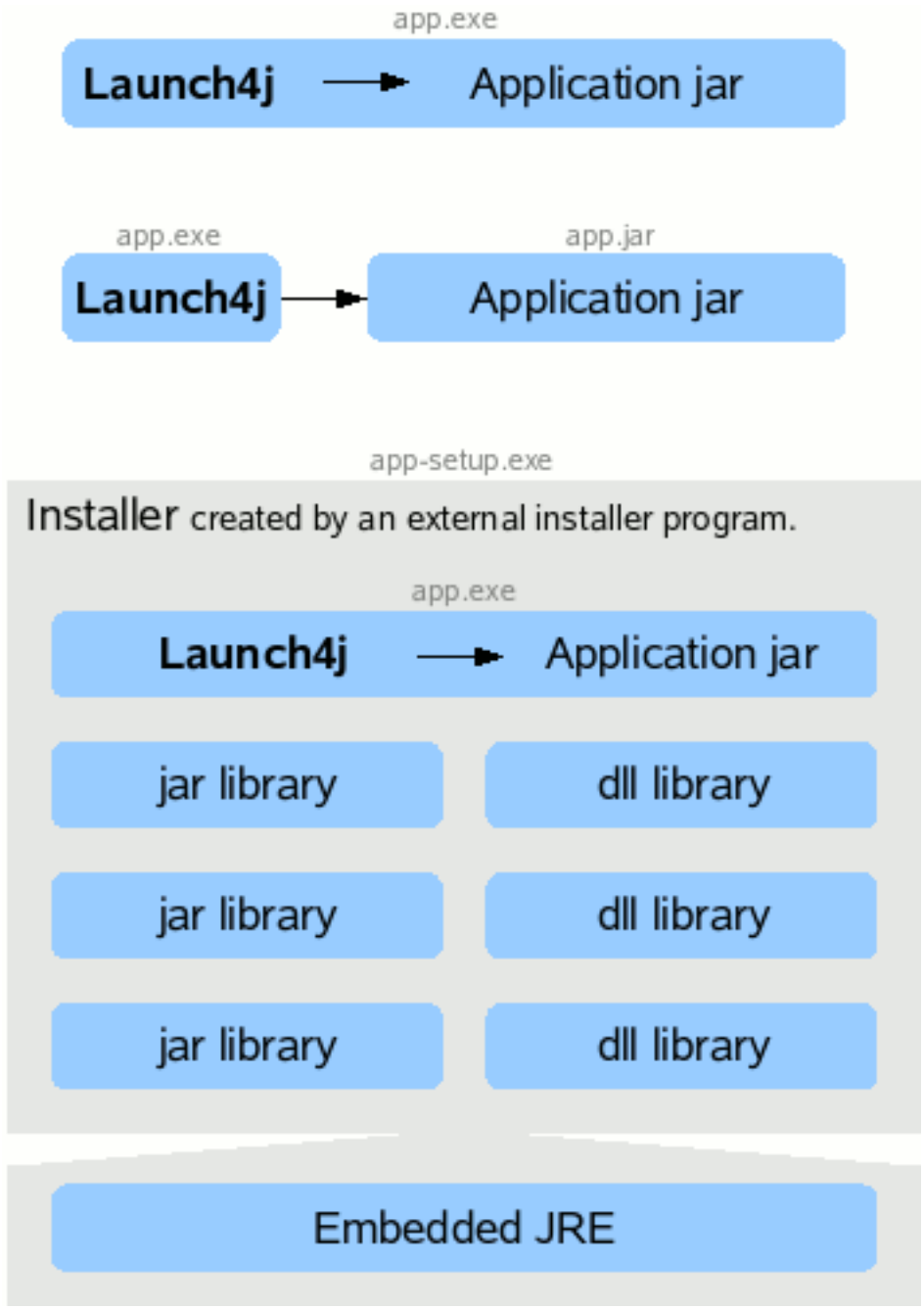
0.133 Shared Library Development - (Eclipse Java Jar Library)-35

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

0.134 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[10]

[10] http://launch4j.sourceforge.net/index.html



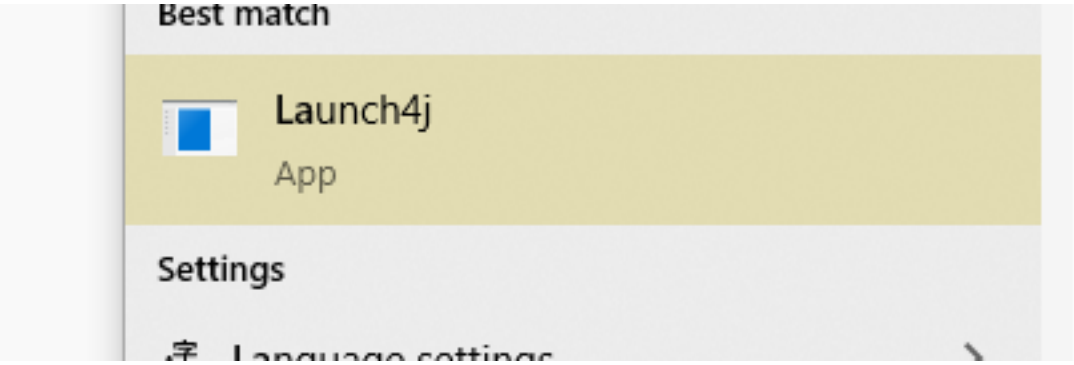
0.135 Shared Library Development - (Eclipse Java Jar Library)-37

- you can watch this tutorial also
  - How to convert jar to exe using Launch4J Full explanation - YouTube[11]

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

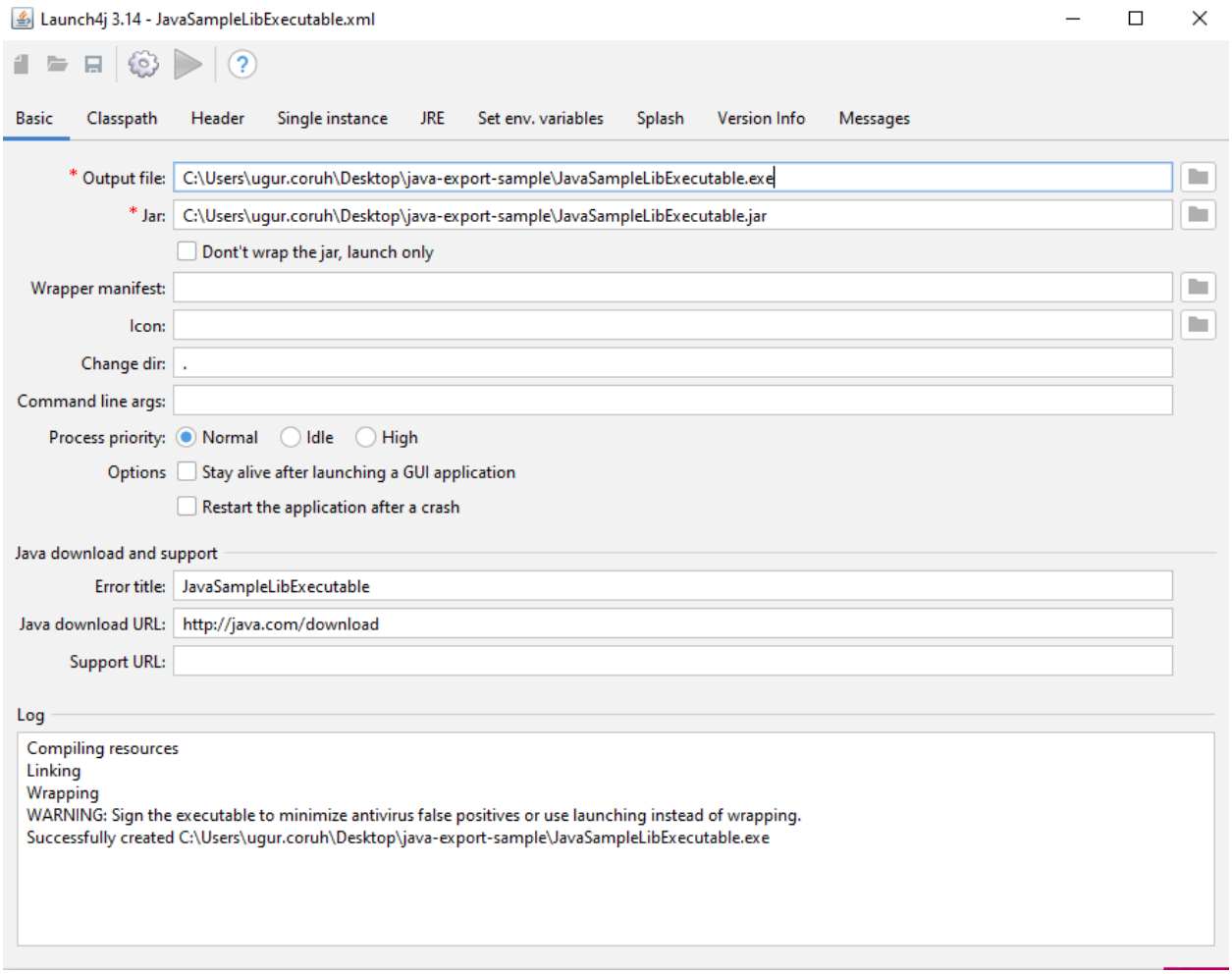
- Download and install launch4j and open application

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



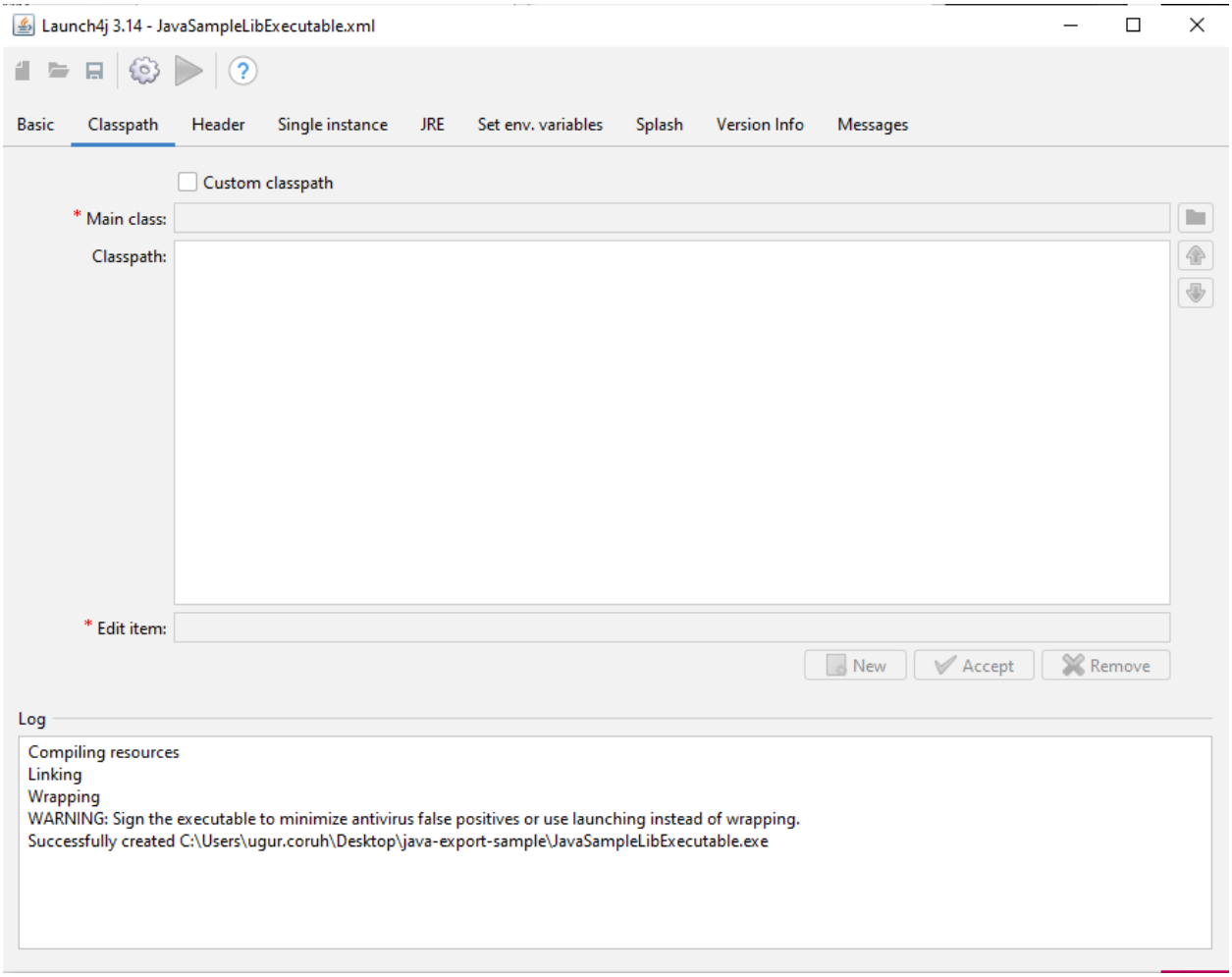
0.137 Shared Library Development - (Eclipse Java Jar Library)-39

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



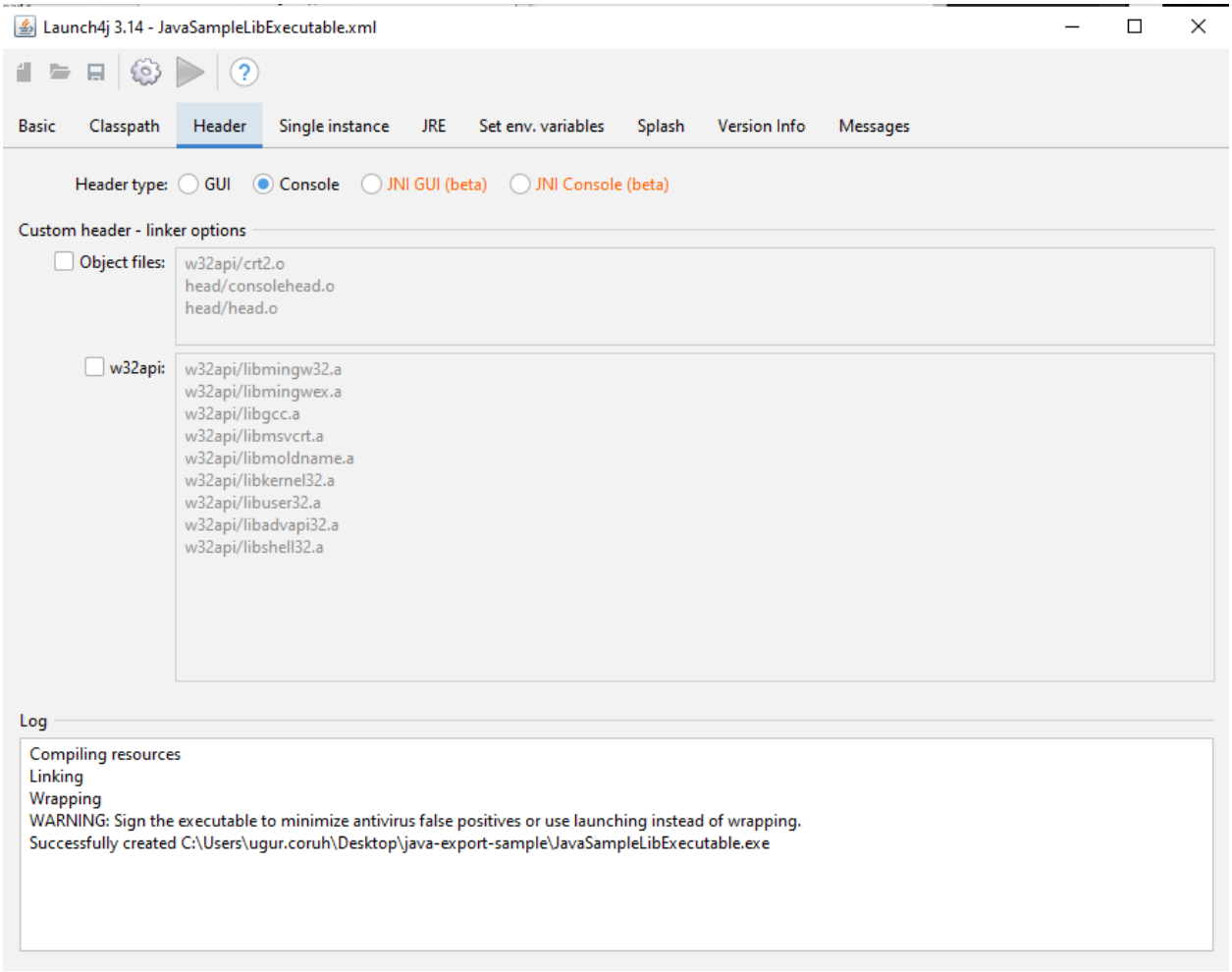
0.138 Shared Library Development - (Eclipse Java Jar Library)-40

- We can customize main class if have multiple main class



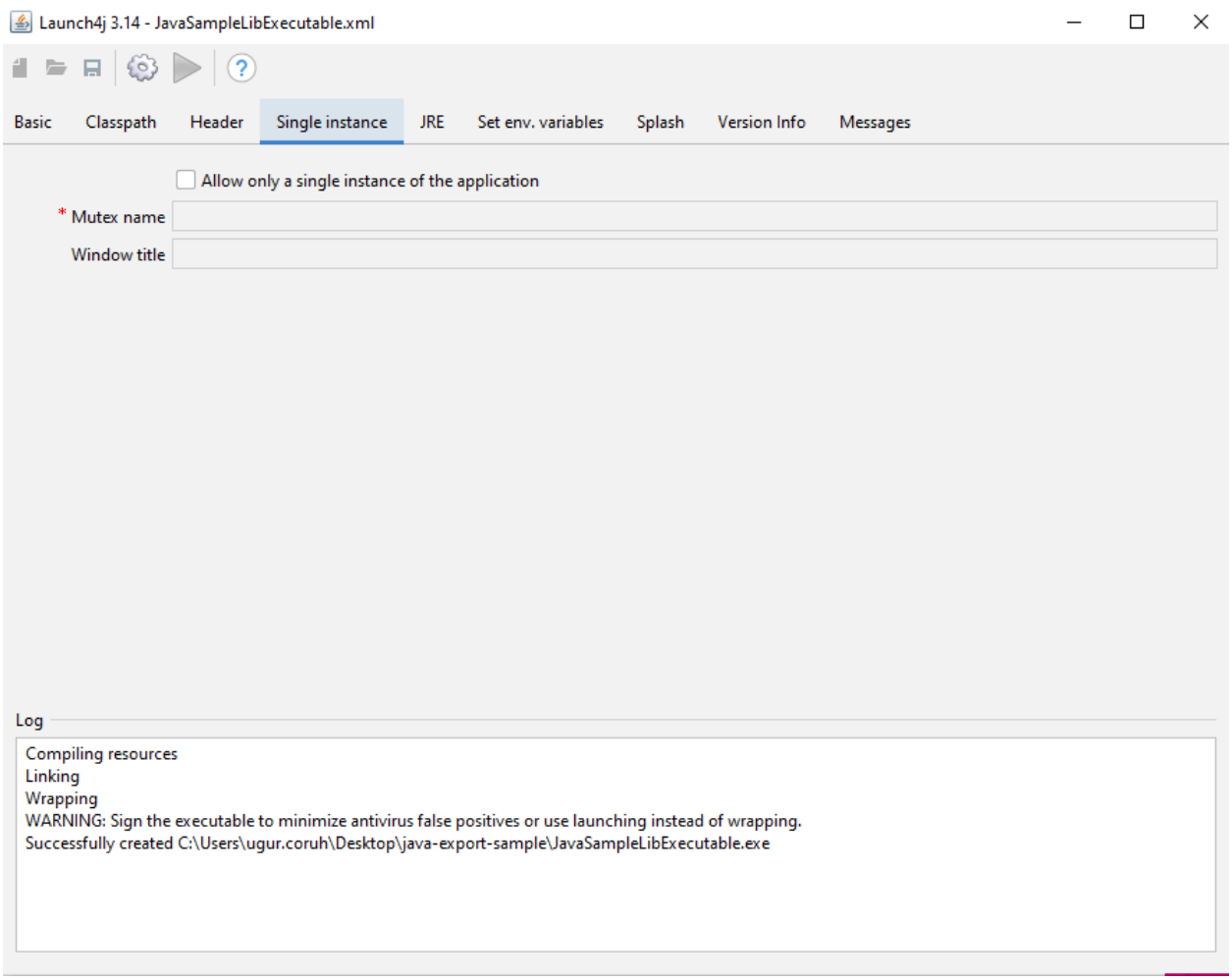
0.139 Shared Library Development - (Eclipse Java Jar Library)-41

select console from setting for this application



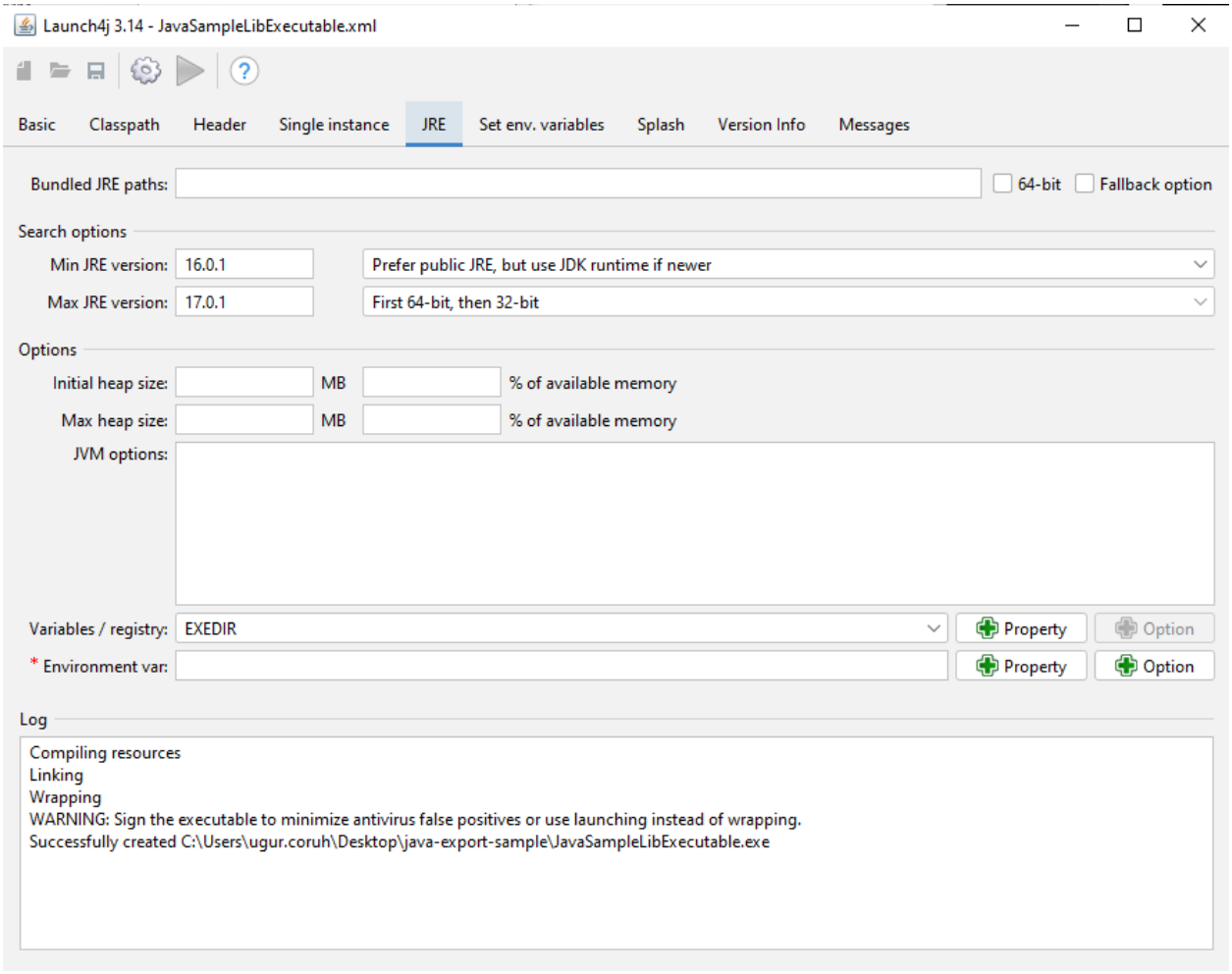
0.140 Shared Library Development - (Eclipse Java Jar Library)-42

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



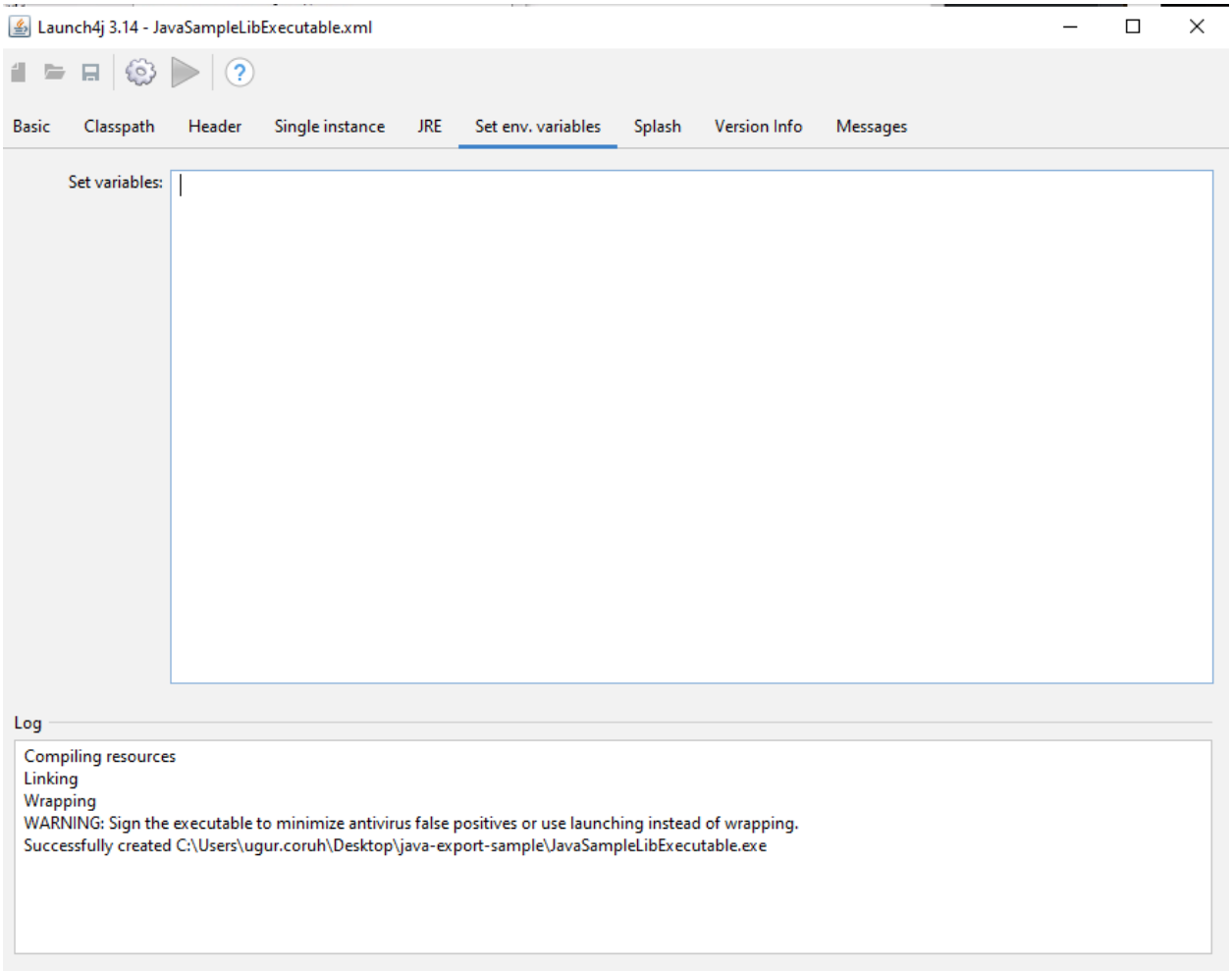
0.141 Shared Library Development - (Eclipse Java Jar Library)-43

- we need to set runtime environment versions



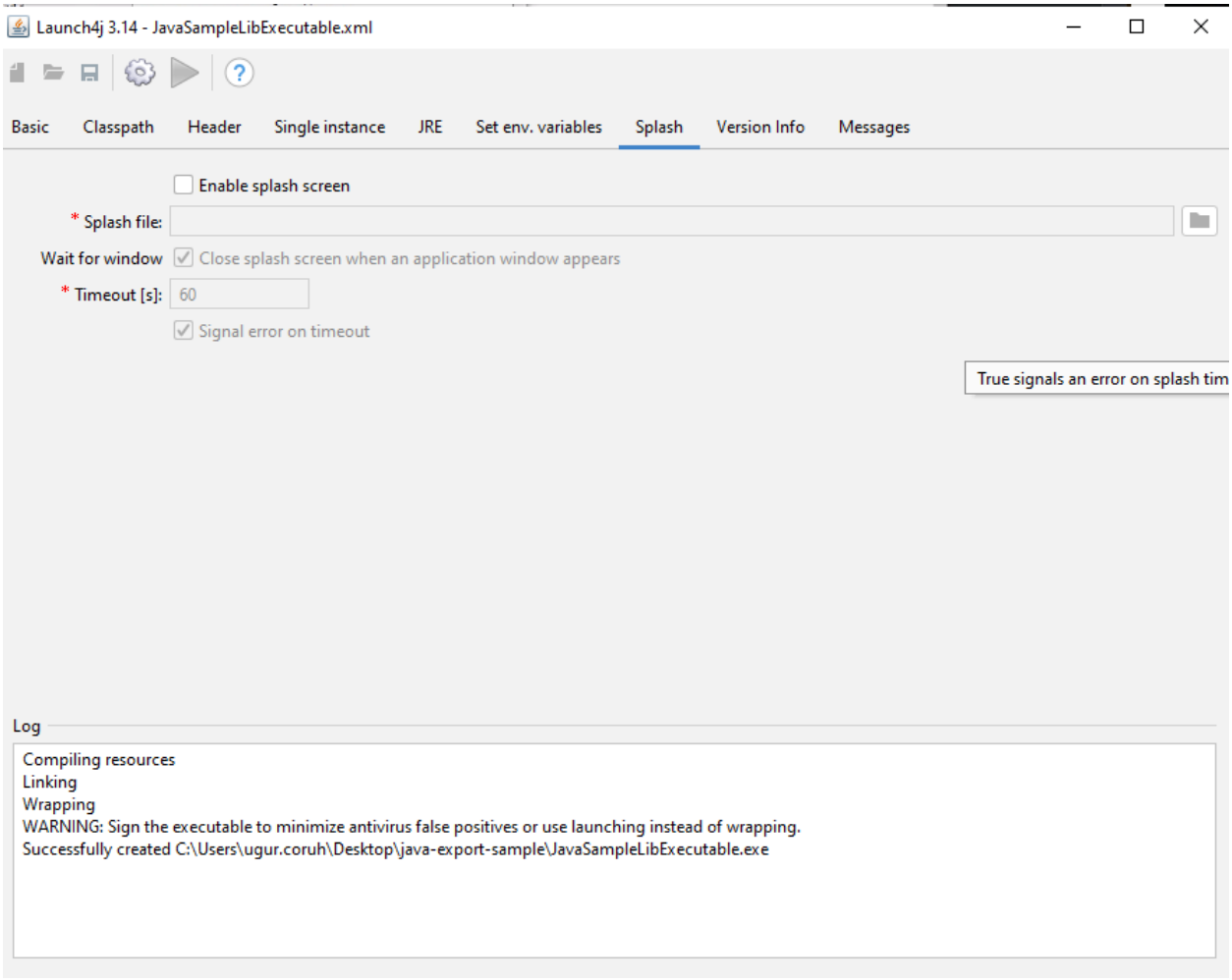
0.142 Shared Library Development - (Eclipse Java Jar Library)-44

you can set system parameters before running application



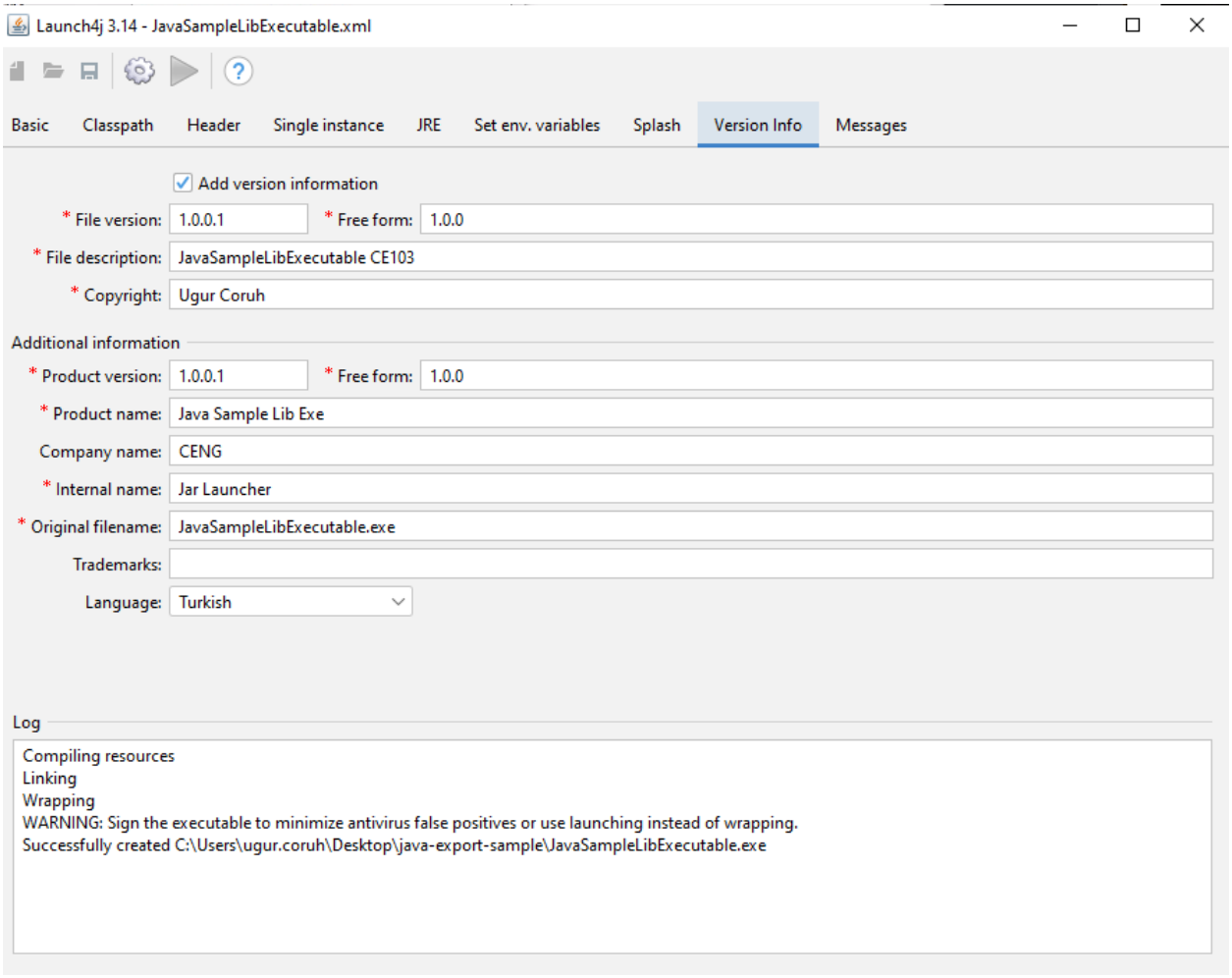
0.143 Shared Library Development - (Eclipse Java Jar Library)-45

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



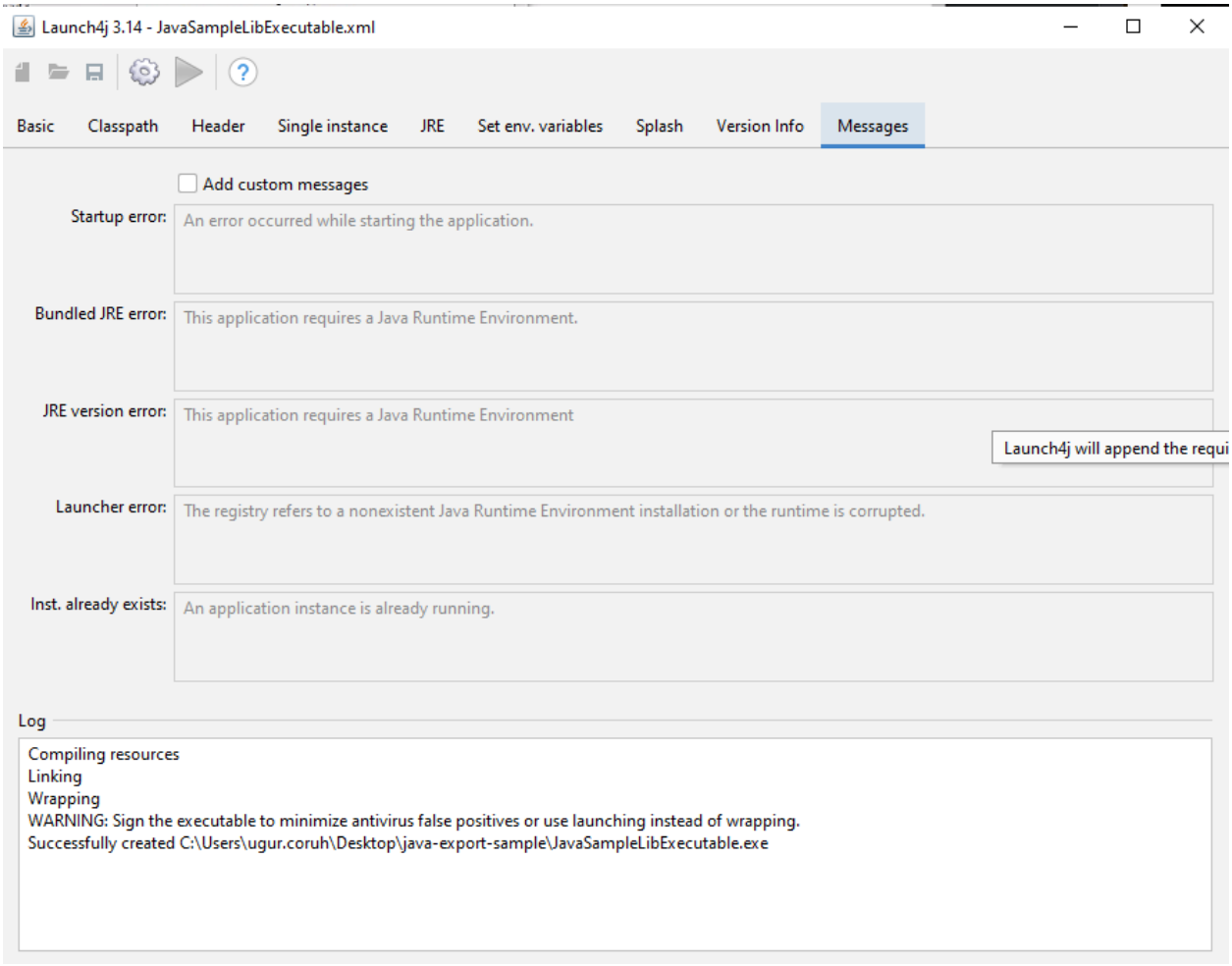
0.144 Shared Library Development - (Eclipse Java Jar Library)-46

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



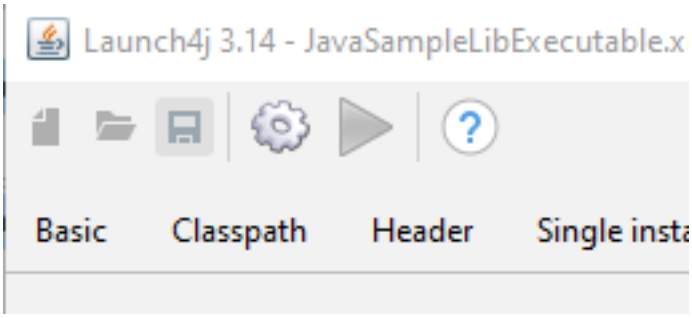
0.145 Shared Library Development - (Eclipse Java Jar Library)-47

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



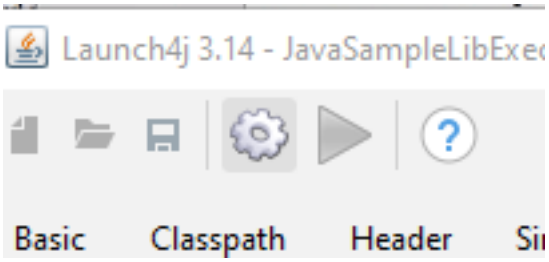
0.146 Shared Library Development - (Eclipse Java Jar Library)-48

- with this options save configuration file xml



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

- and compile settings



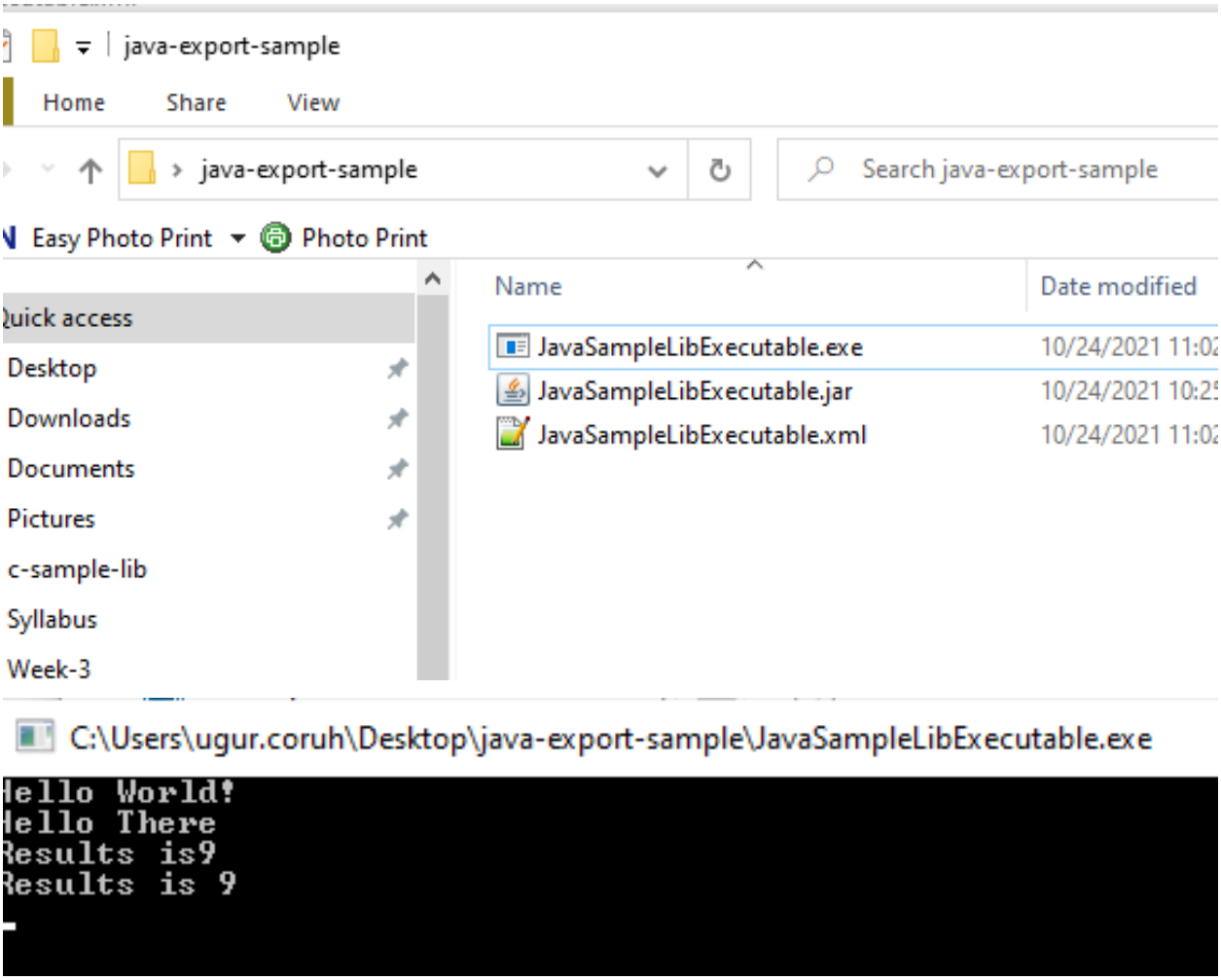
0.148 Shared Library Development - (Eclipse Java Jar Library)-50

- 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
```

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

- now we can run exe by click



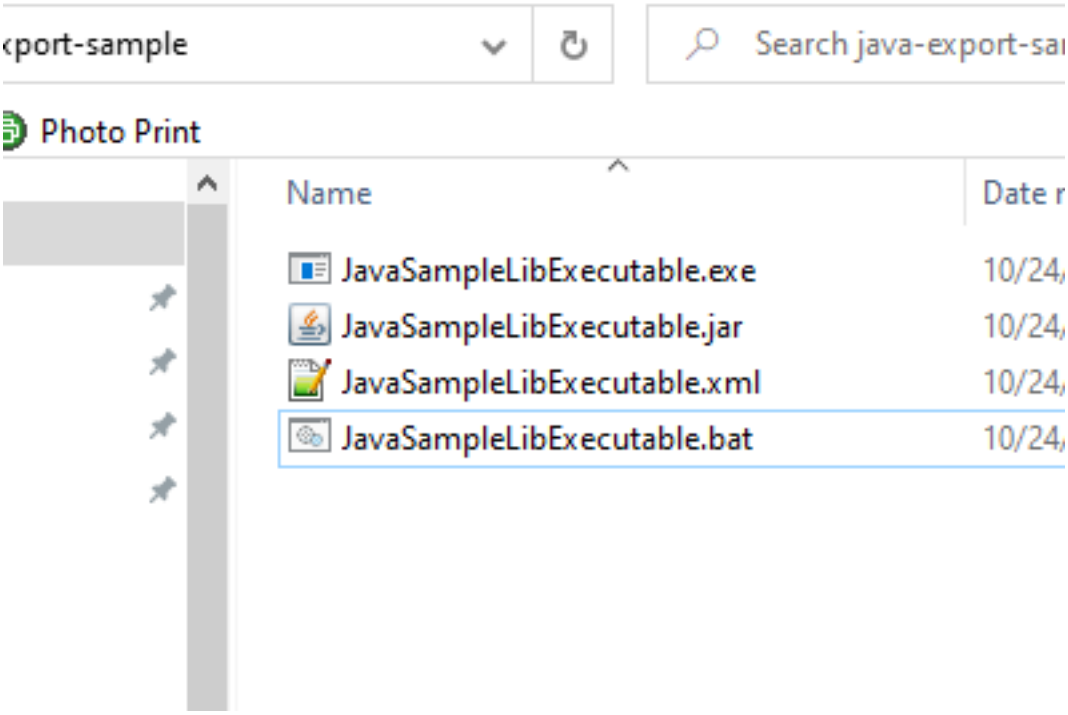
0.150 Shared Library Development - (Eclipse Java Jar Library)-52

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

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

JavaSampleLibExecutable.bat

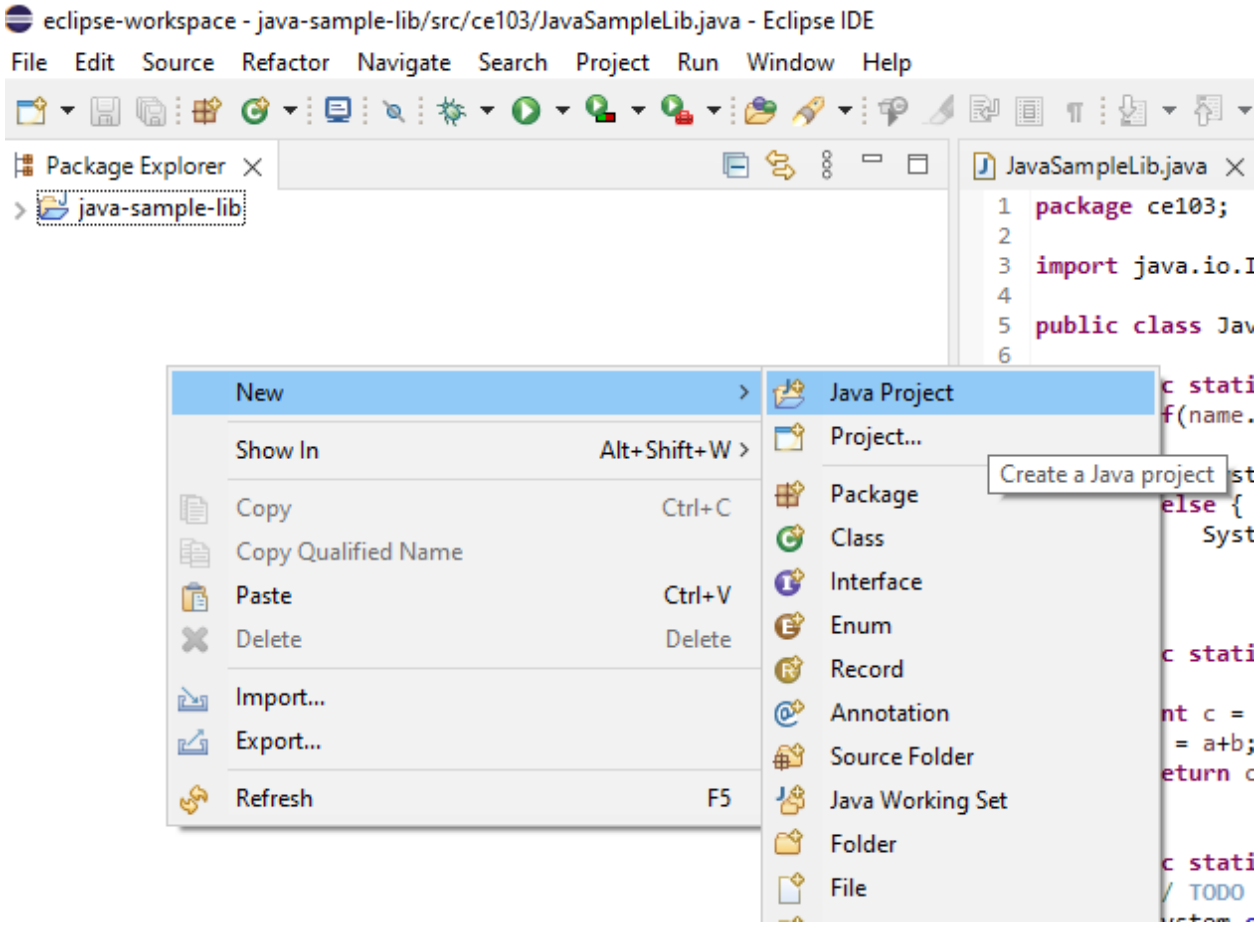
```
java -jar JavaSampleLibExecutable.jar
```



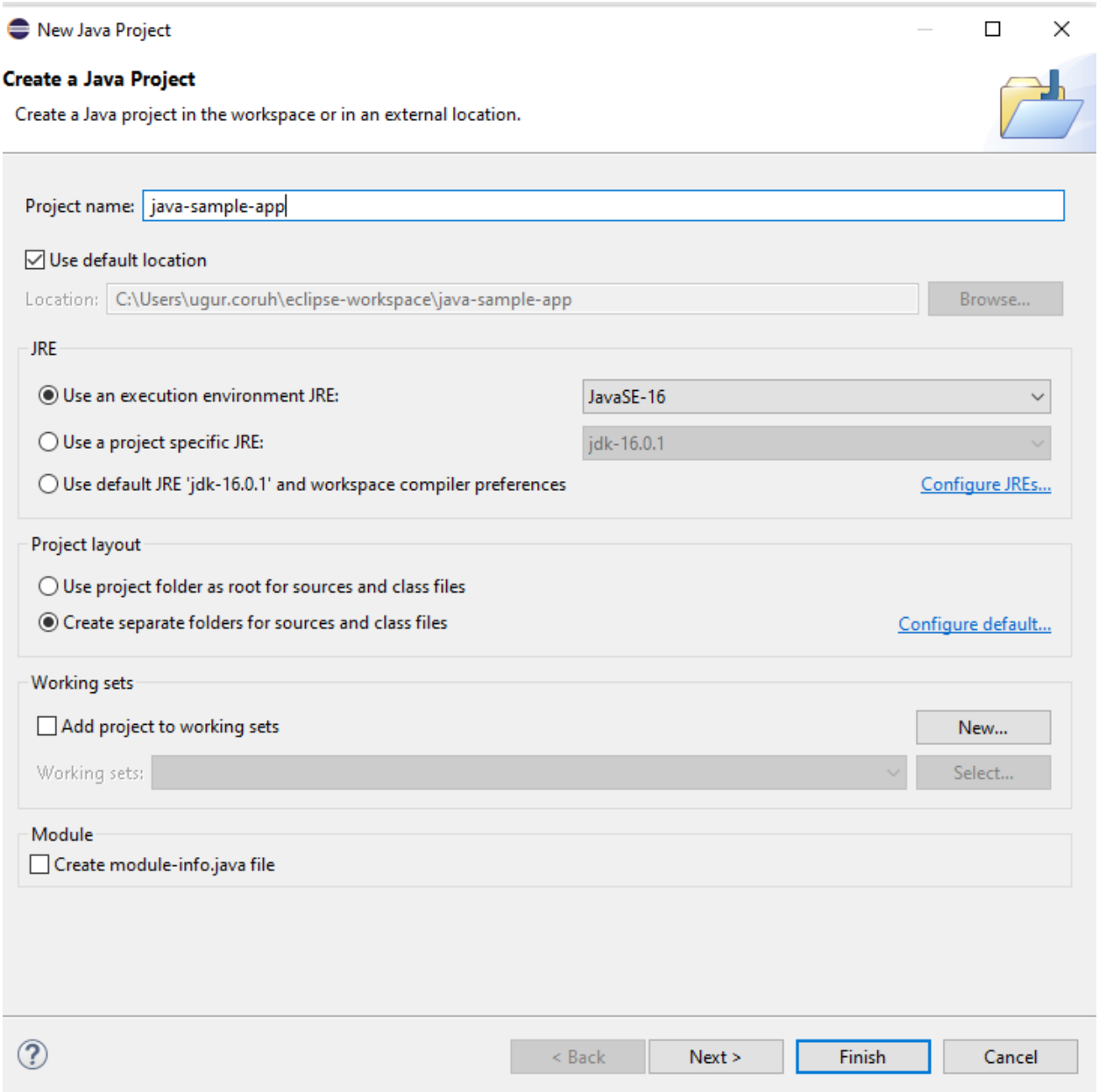
- if we click bat file then we will automate command line task for current jar file

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

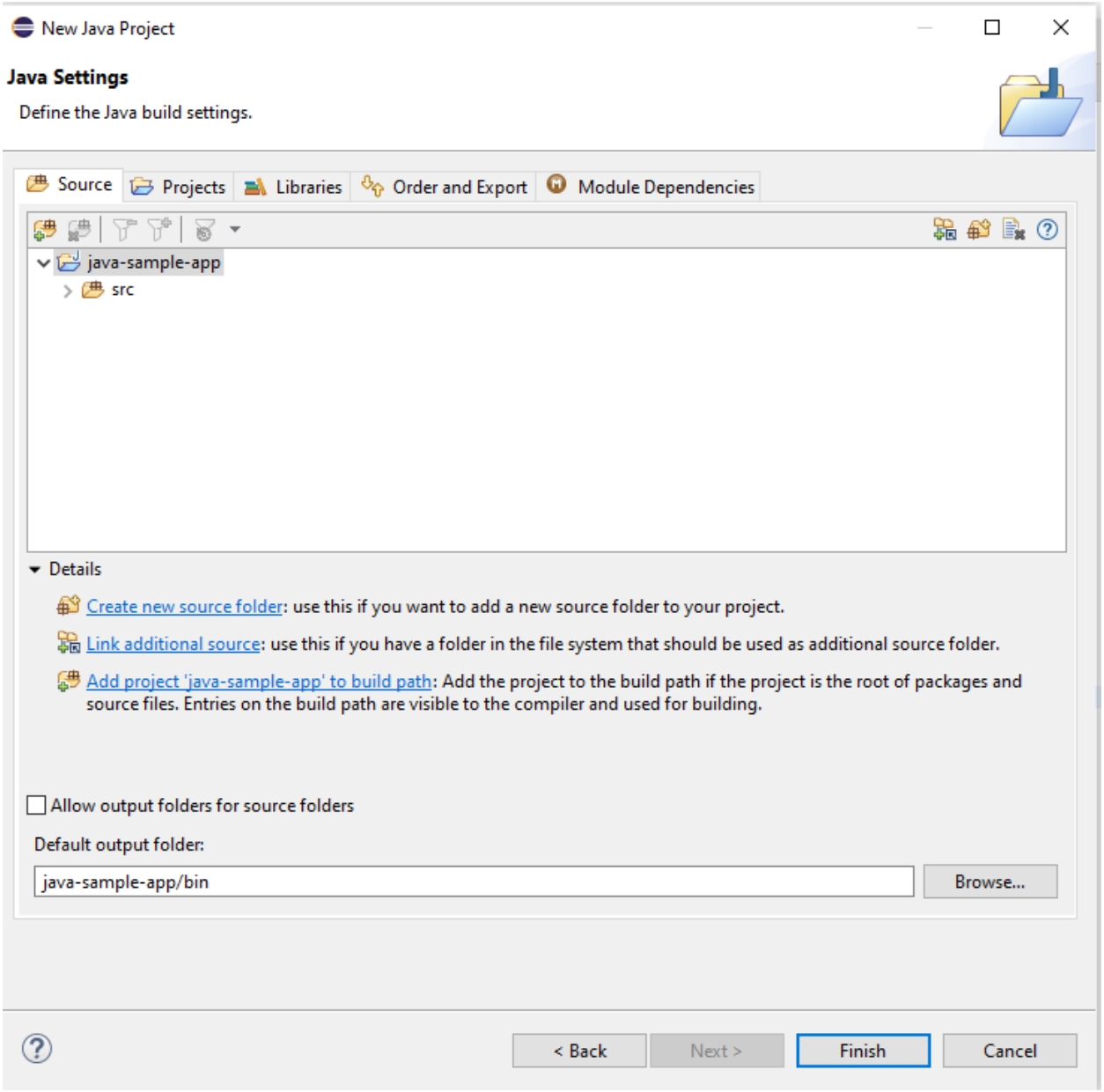
Now return back to our java library and create another console application that use library functions


eclipse-workspace - java-sample-lib/src/ce103/JavaSampleLib.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
java-sample-lib
JavaSampleLib.java
1 package ce103;
3 import java.io.I
5 public class Jav
New
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Import...
Export...
Refresh F5
Java Project
Project...
Package
Class
Interface
Enum
Record
Annotation
Source Folder
Java Working Set
Folder
File
Create a Java project

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

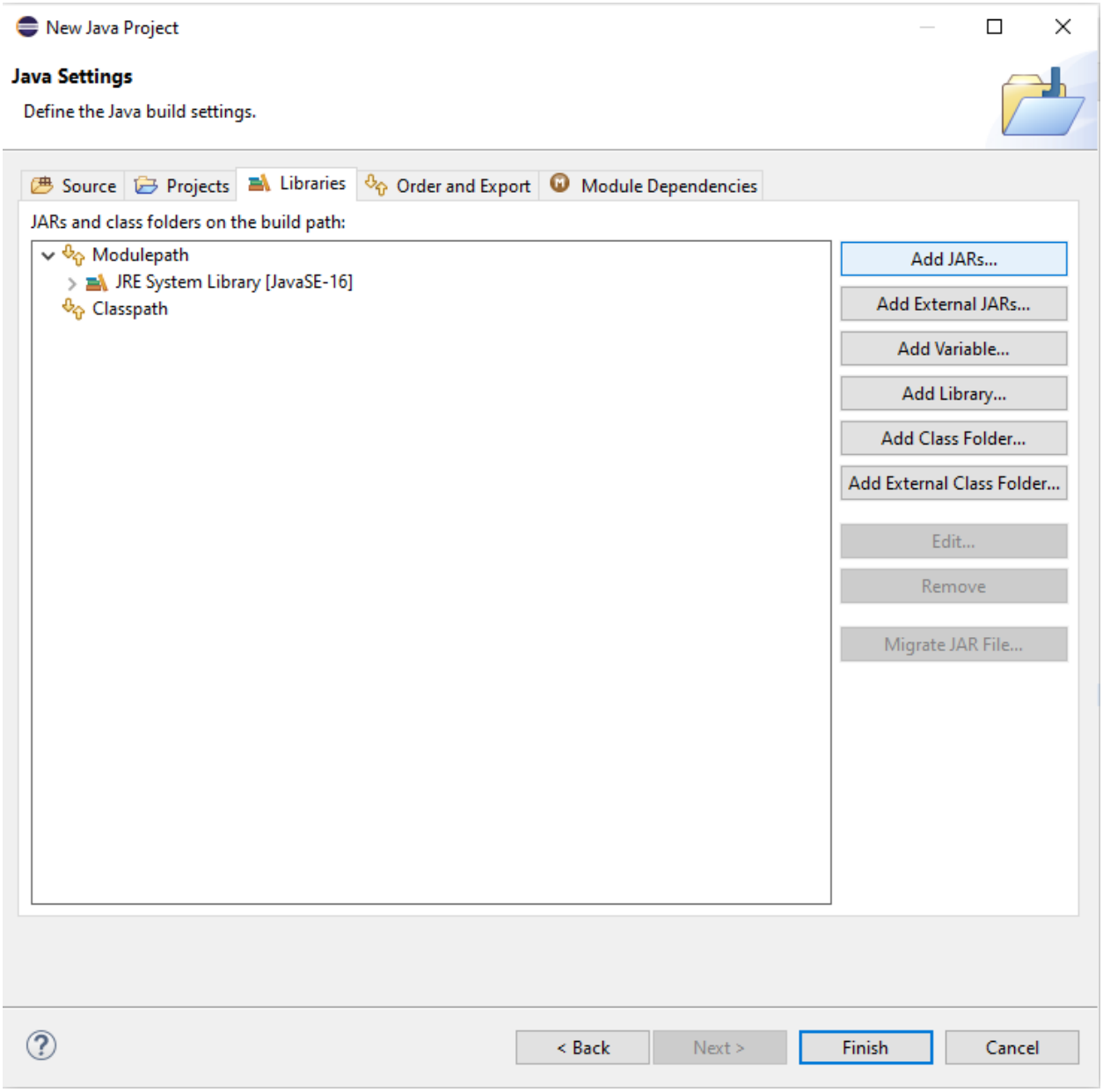


0.154 Shared Library Development - (Eclipse Java Jar Library)-56



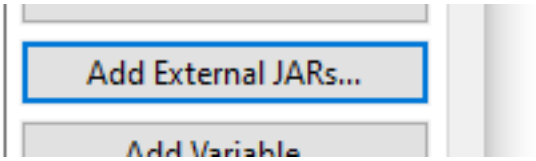
0.155 Shared Library Development - (Eclipse Java Jar Library)-57

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



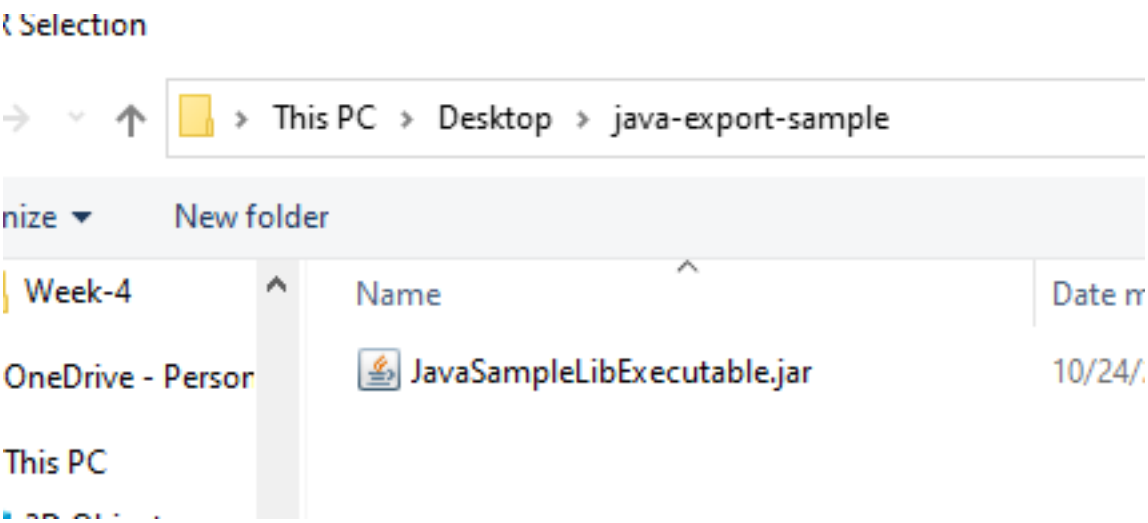
0.156 Shared Library Development - (Eclipse Java Jar Library)-58

- Select Add External JARs...



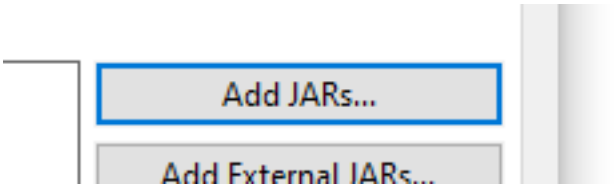
0.157 Shared Library Development - (Eclipse Java Jar Library)-59

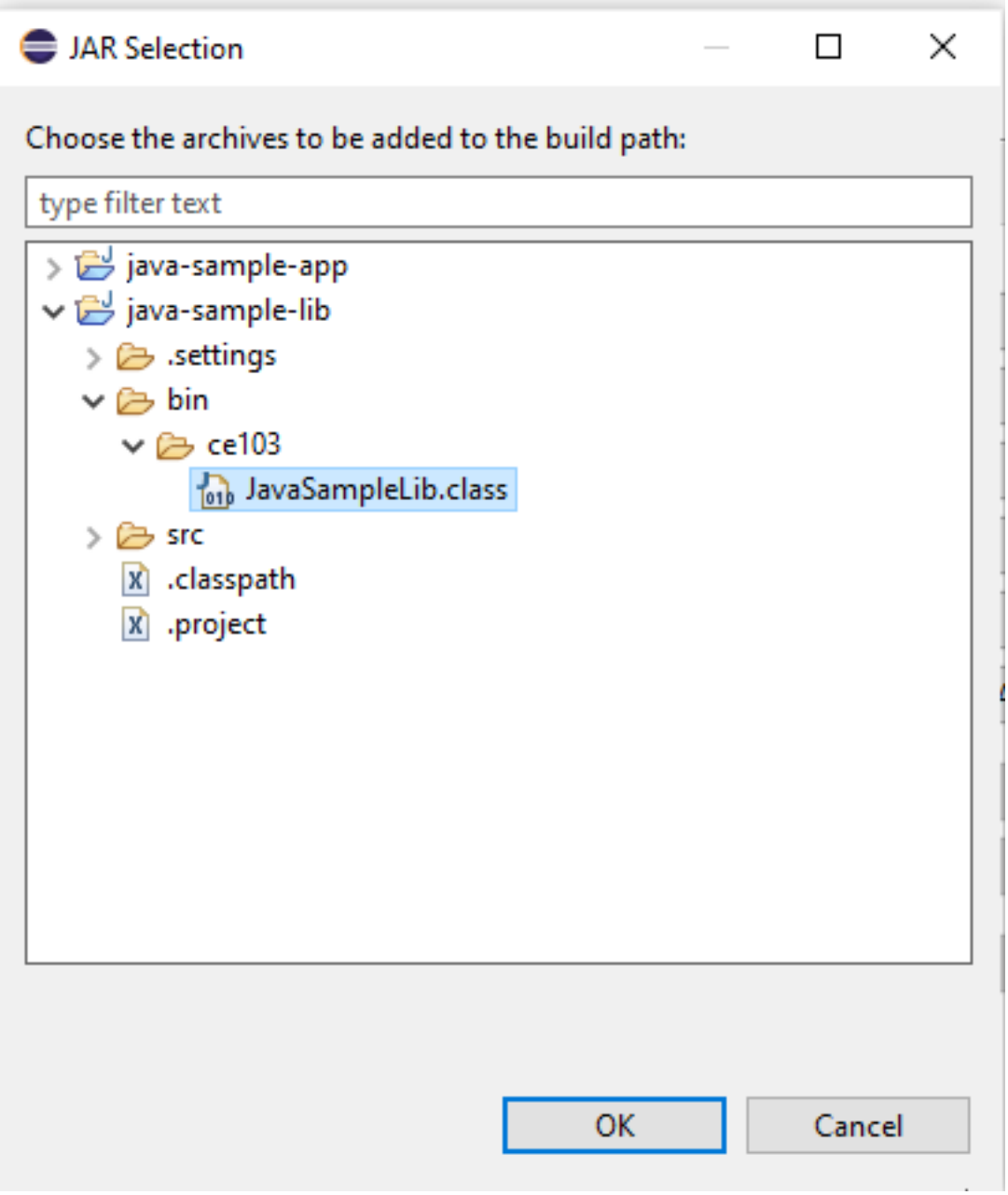
- Open Exported jar folder and select



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

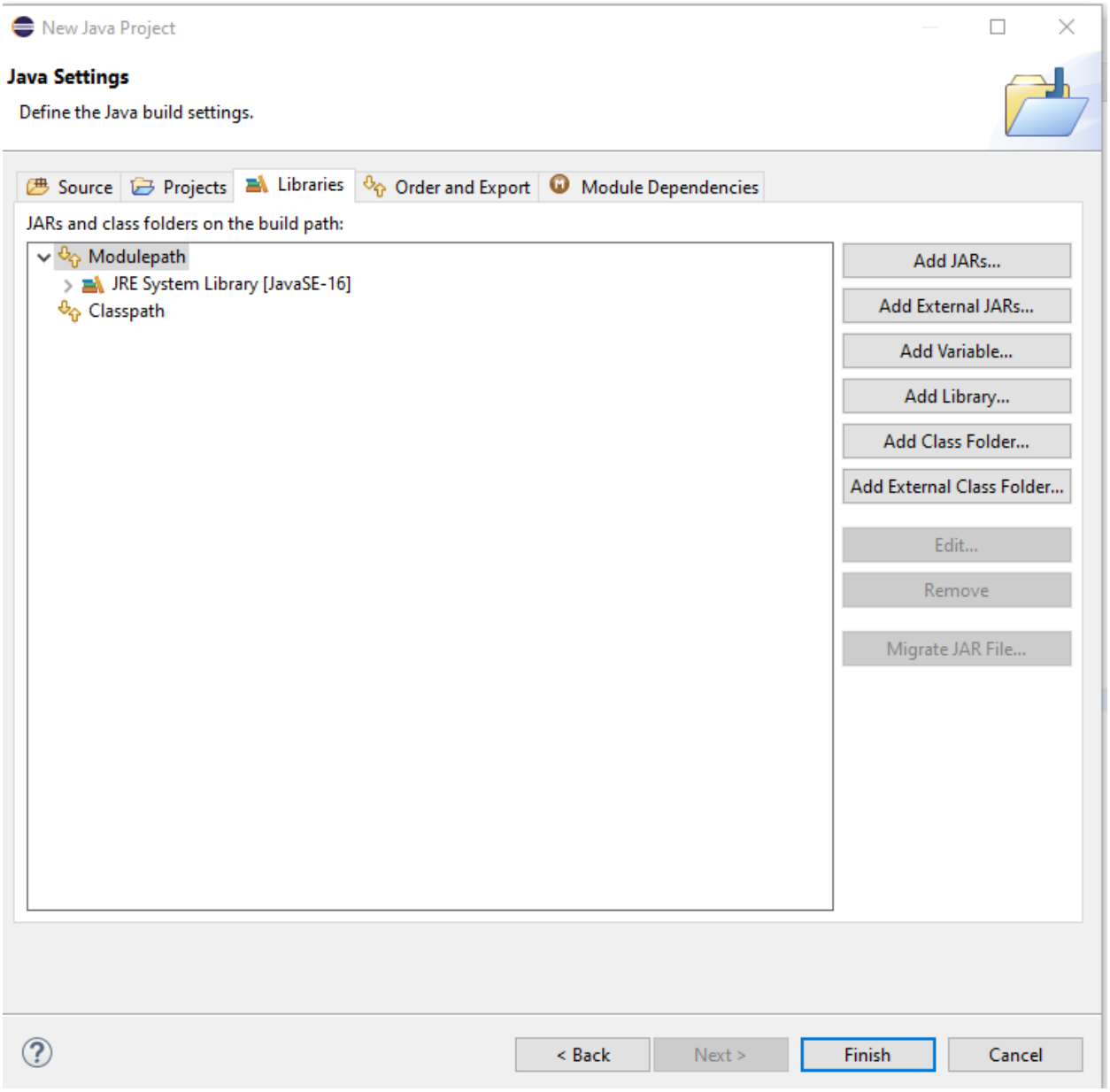
- Or we can select by Add jar from current workspace





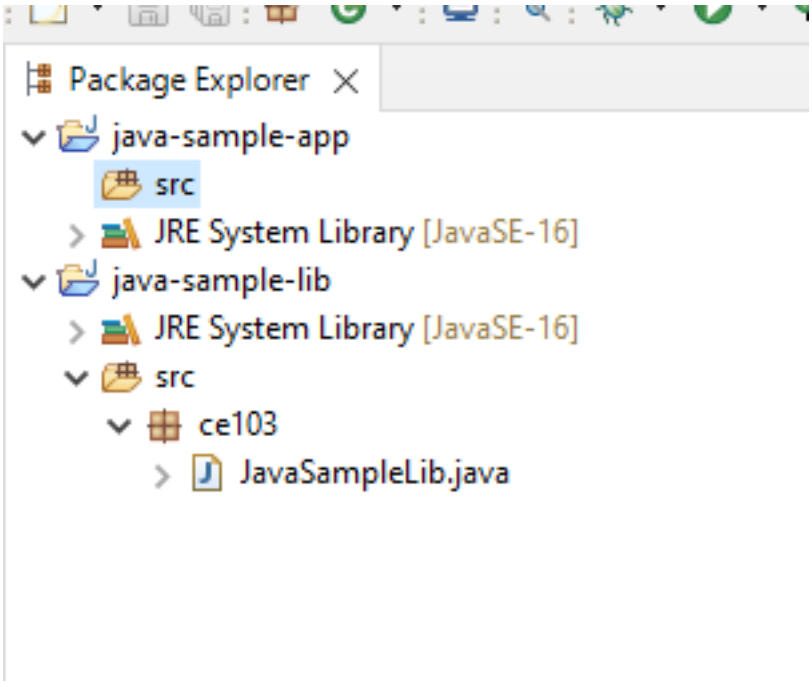
0.159 Shared Library Development - (Eclipse Java Jar Library)-61

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



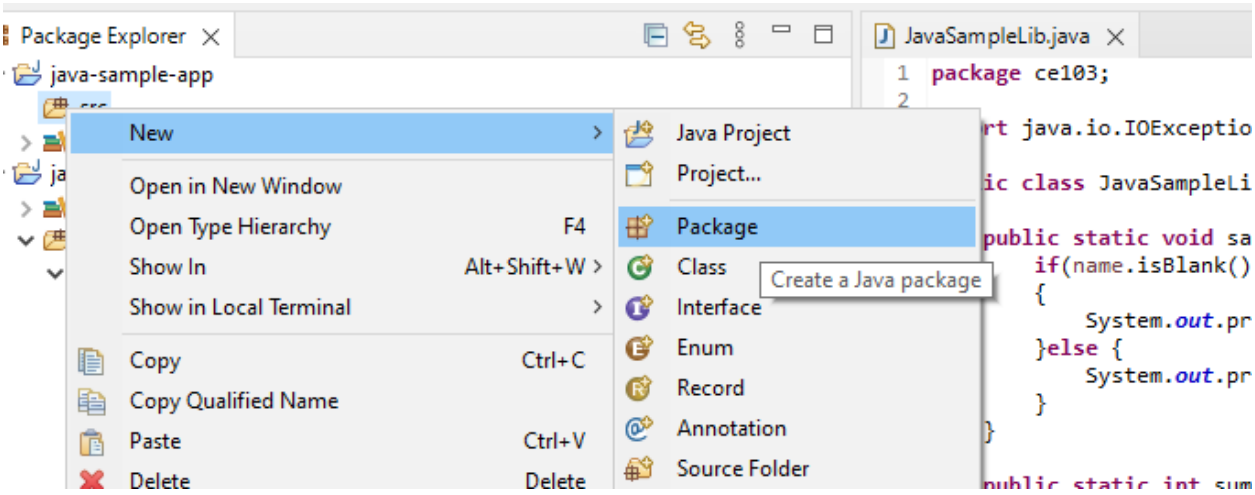
0.160 Shared Library Development - (Eclipse Java Jar Library)-62

- we will have the following project

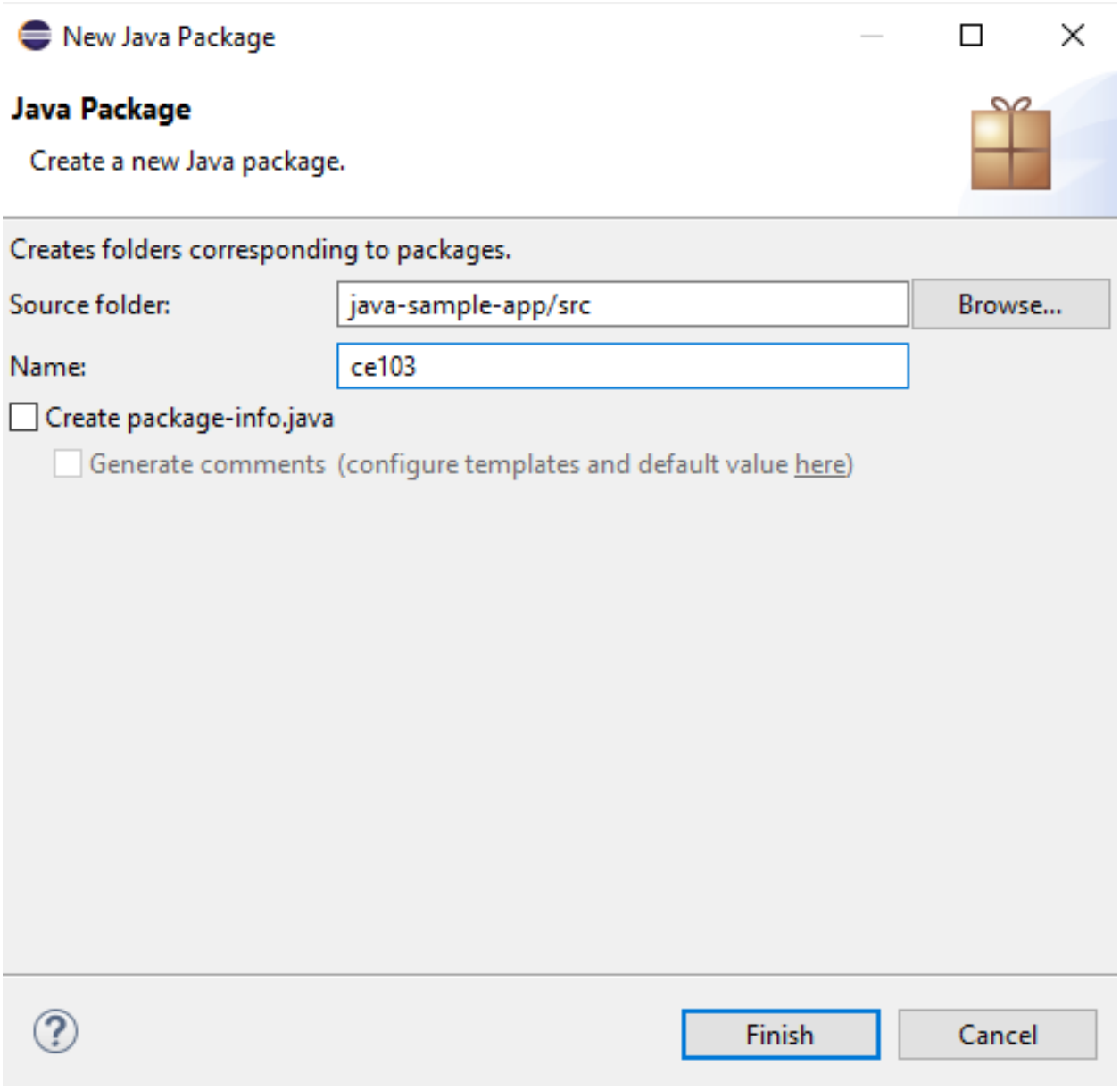


0.161 Shared Library Development - (Eclipse Java Jar Library)-63

- lets create a package

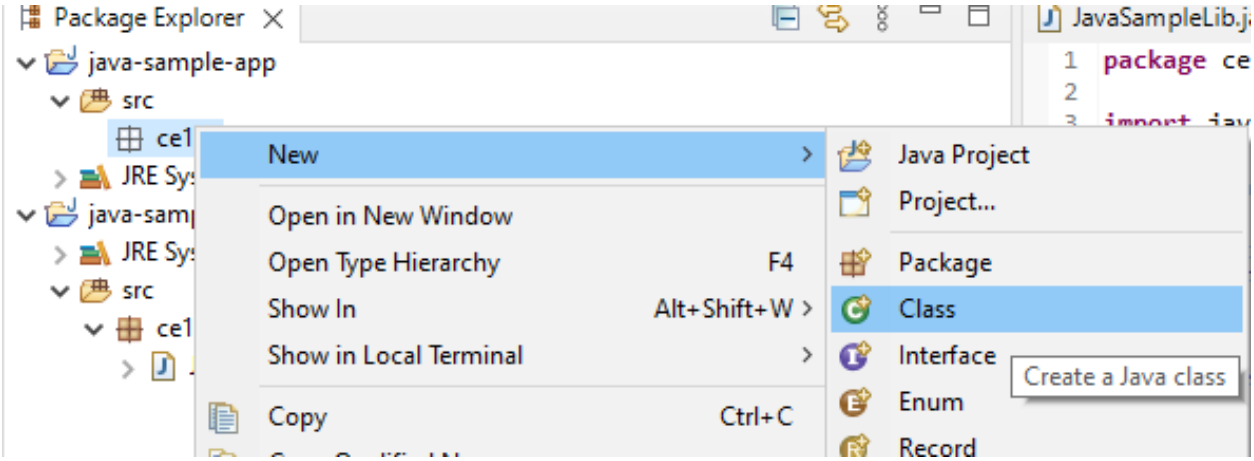


0.162 Shared Library Development - (Eclipse Java Jar Library)-64



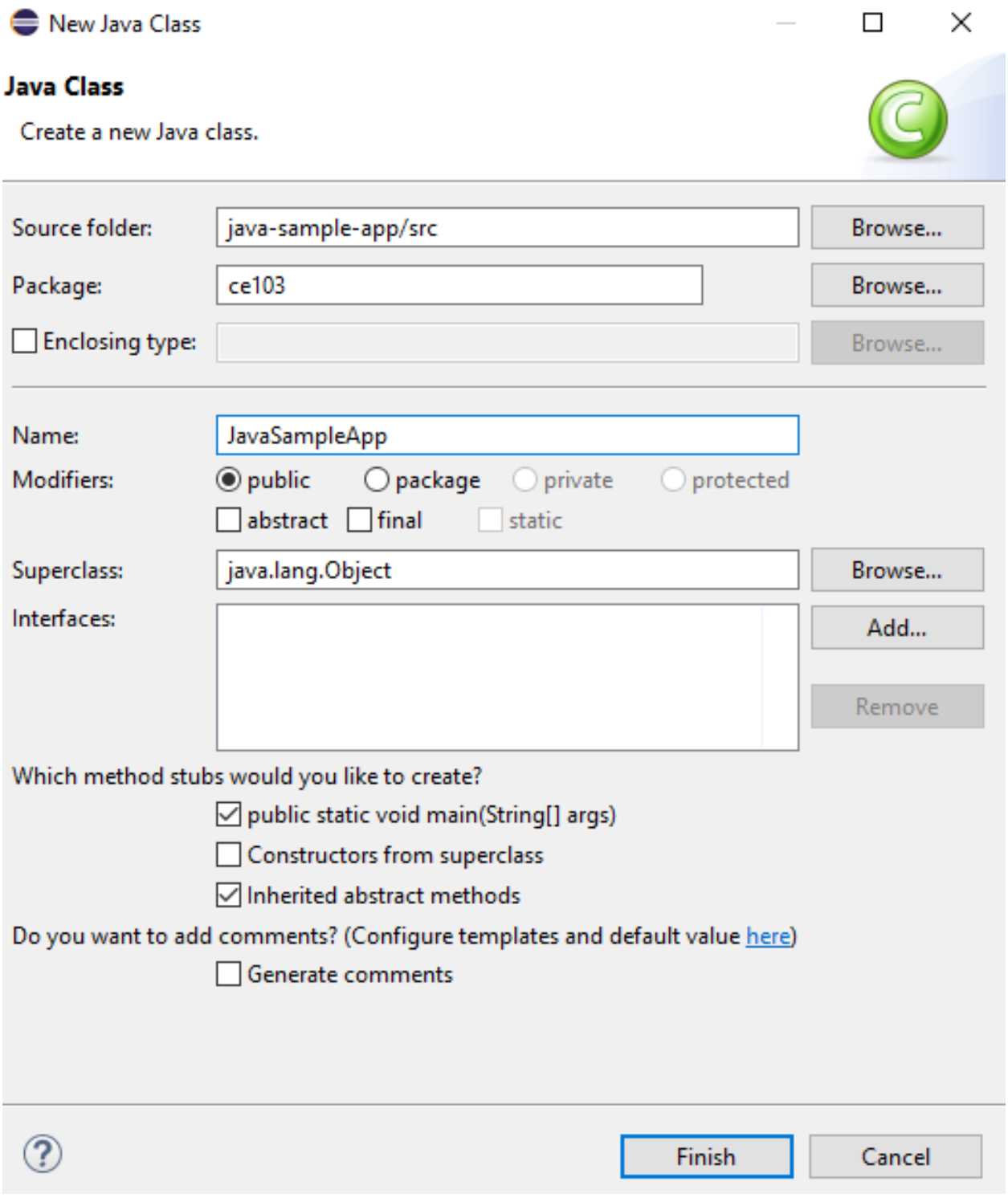
0.163 Shared Library Development - (Eclipse Java Jar Library)-65

- and lets create a main class for our application

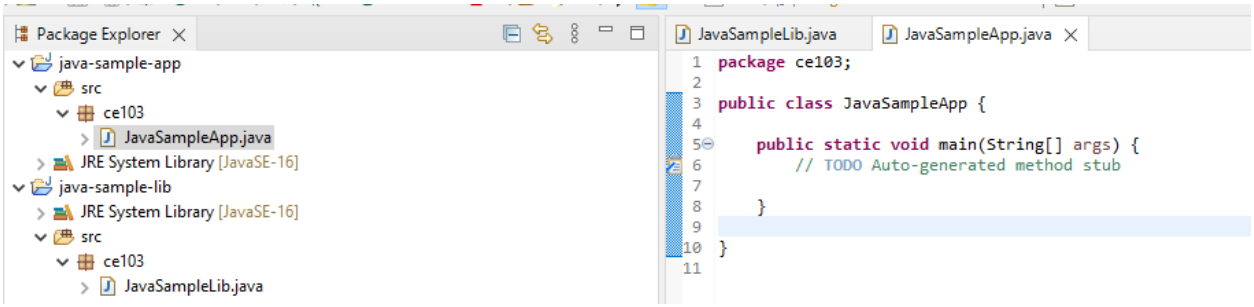


0.164 Shared Library Development - (Eclipse Java Jar Library)-66

- check create main function

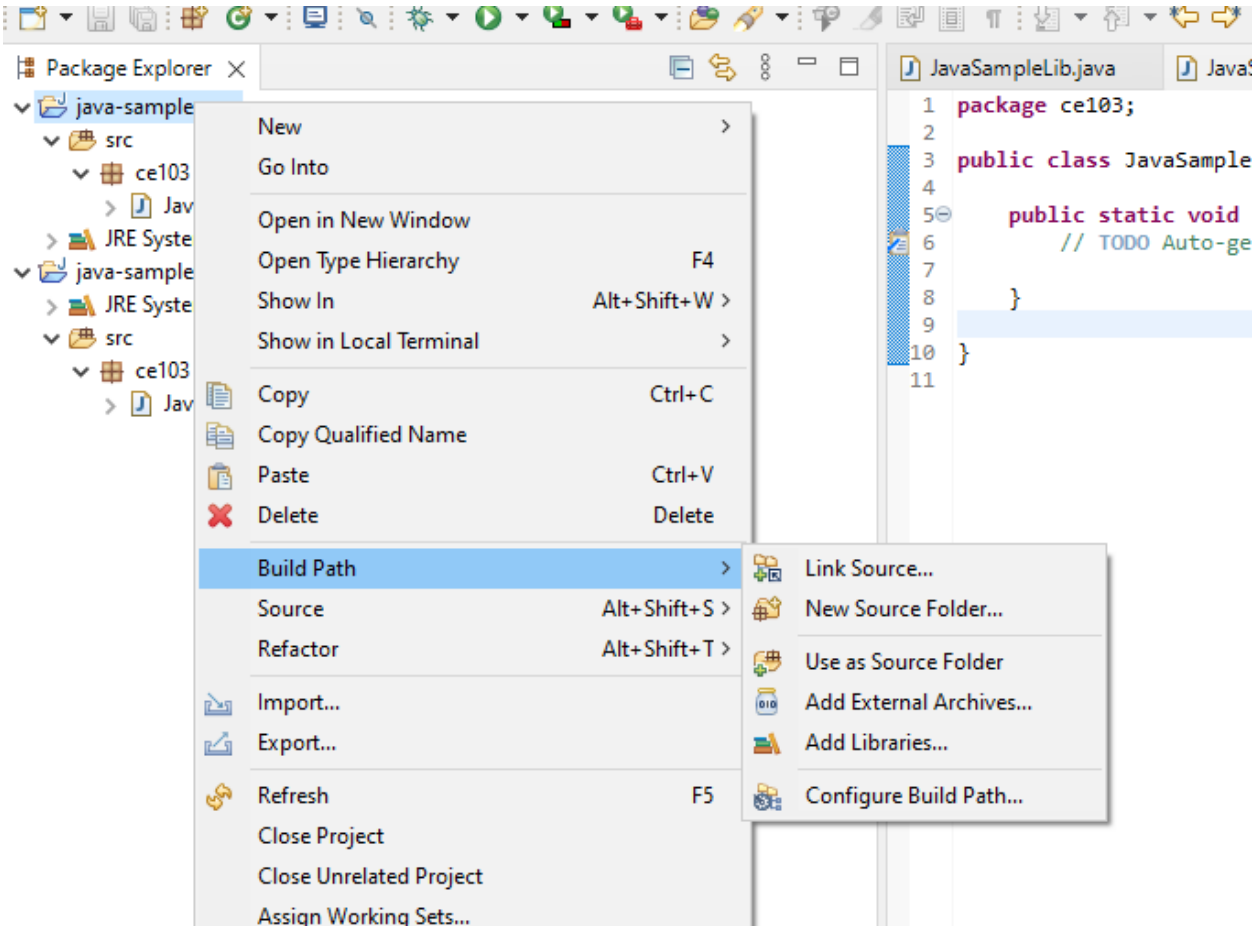

New Java Class
Java Class
Create a new Java class.
Source folder:
java-sample-app/src
Browse...
Package:
ce103
Browse...
Enclosing type:
Browse...
Name:
JavaSampleApp
Modifiers:
public
package
private
protected
abstract
final
static
Superclass:
java.lang.Object
Browse...
Interfaces:
Add...
Remove
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Finish
Cancel

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



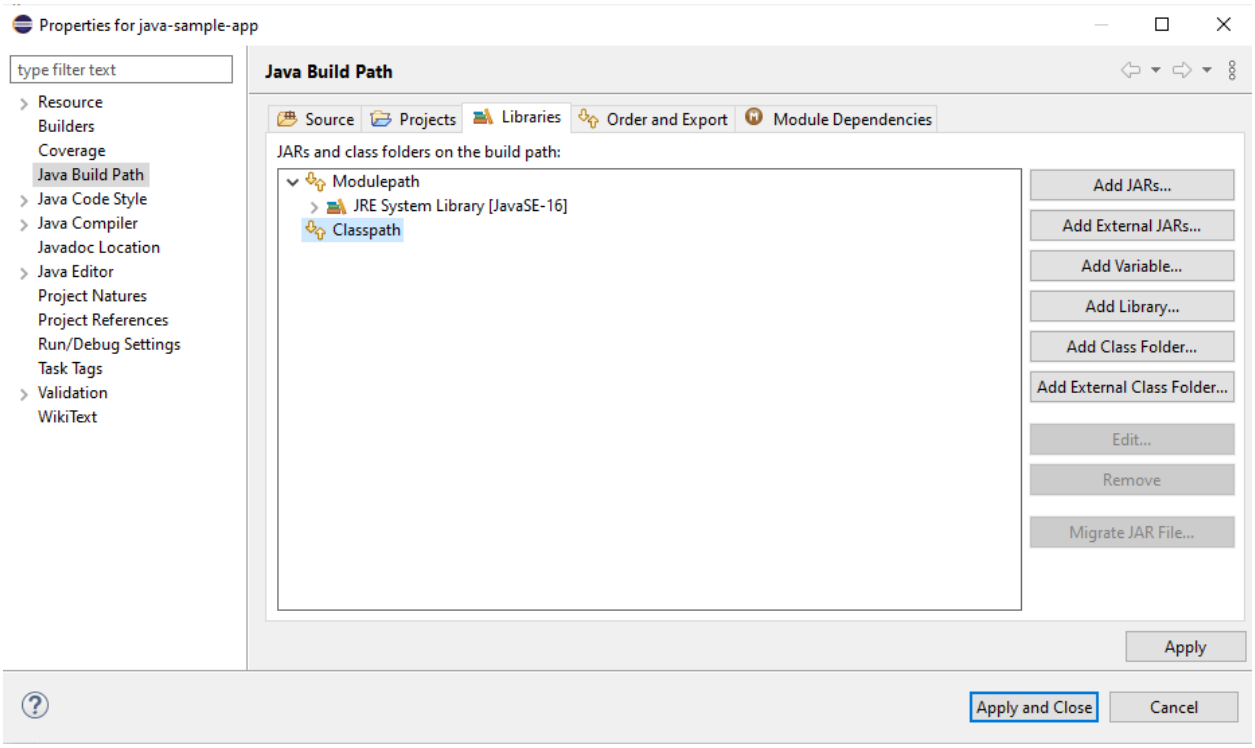
0.166 Shared Library Development - (Eclipse Java Jar Library)-68

- right click to project and add reference



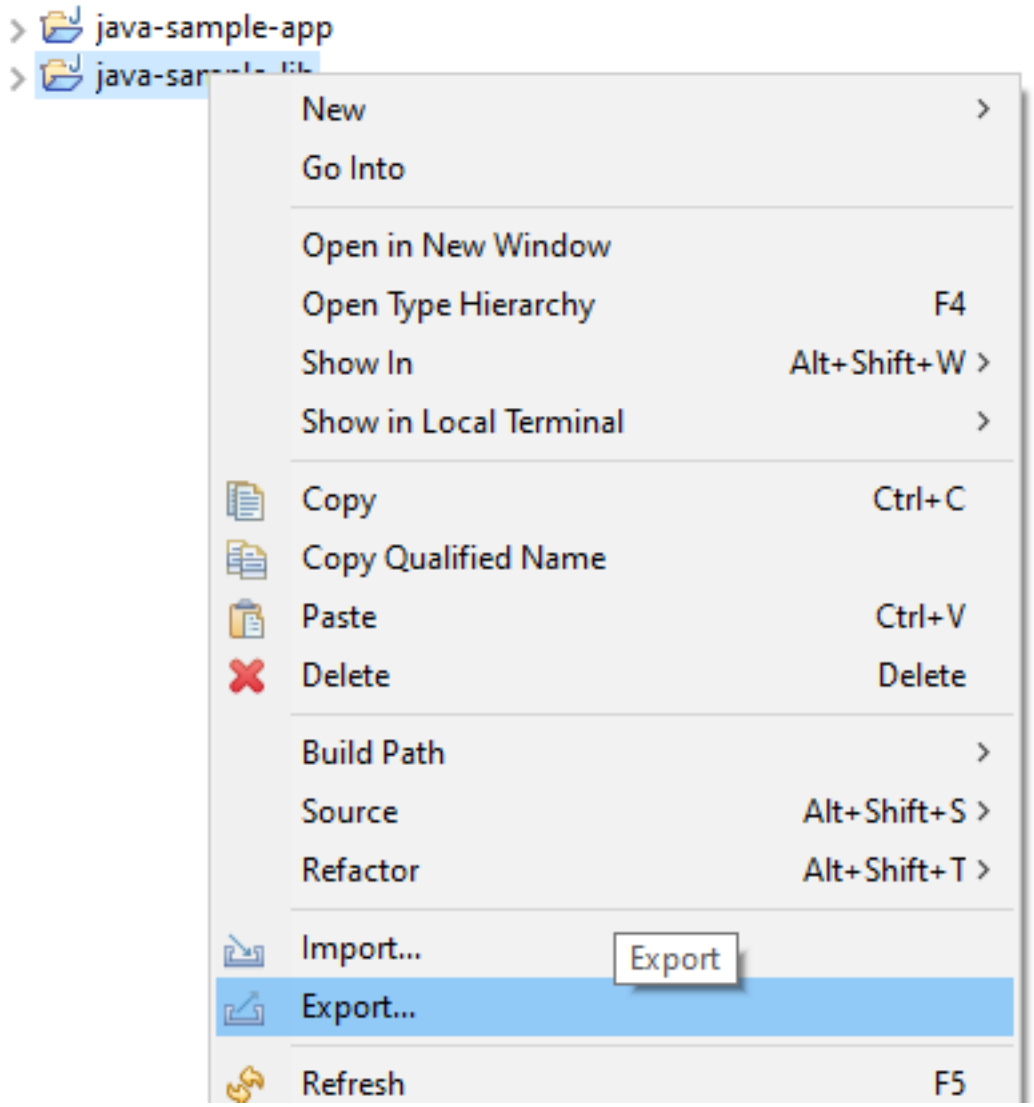
0.167 Shared Library Development - (Eclipse Java Jar Library)-69

- you can enter same configurations from project properties



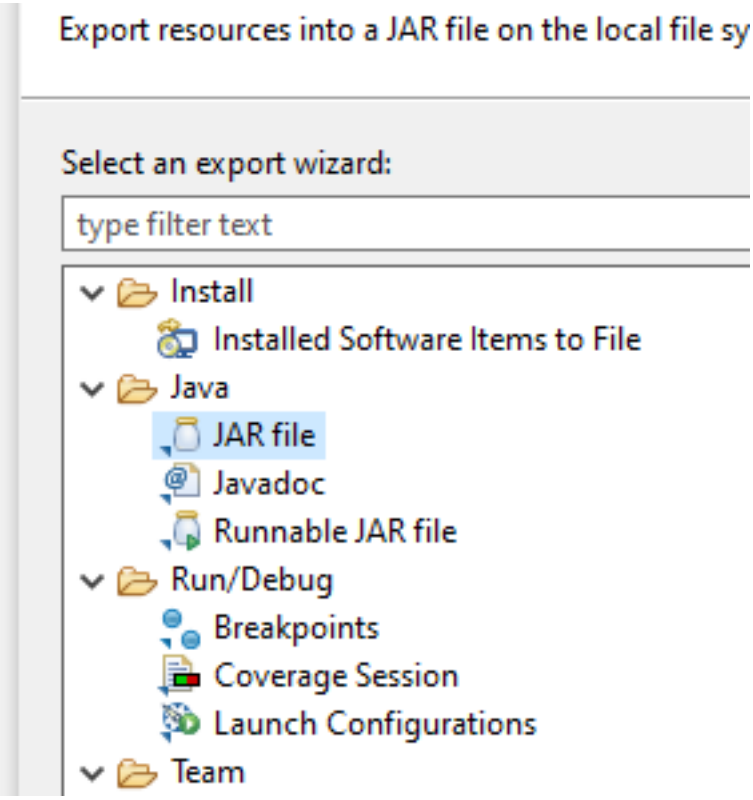
0.168 Shared Library Development - (Eclipse Java Jar Library)-70

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



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

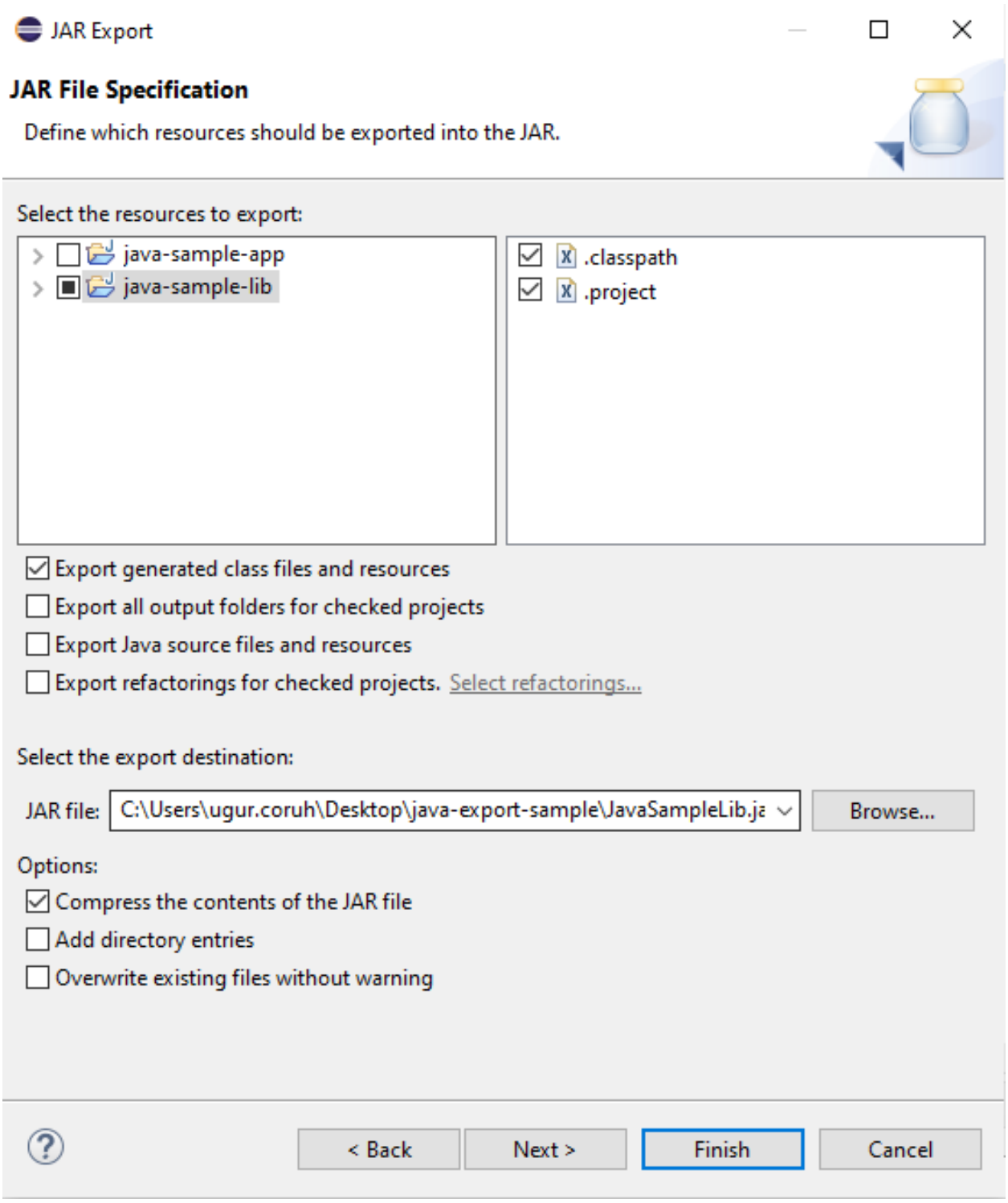
Select JAR file



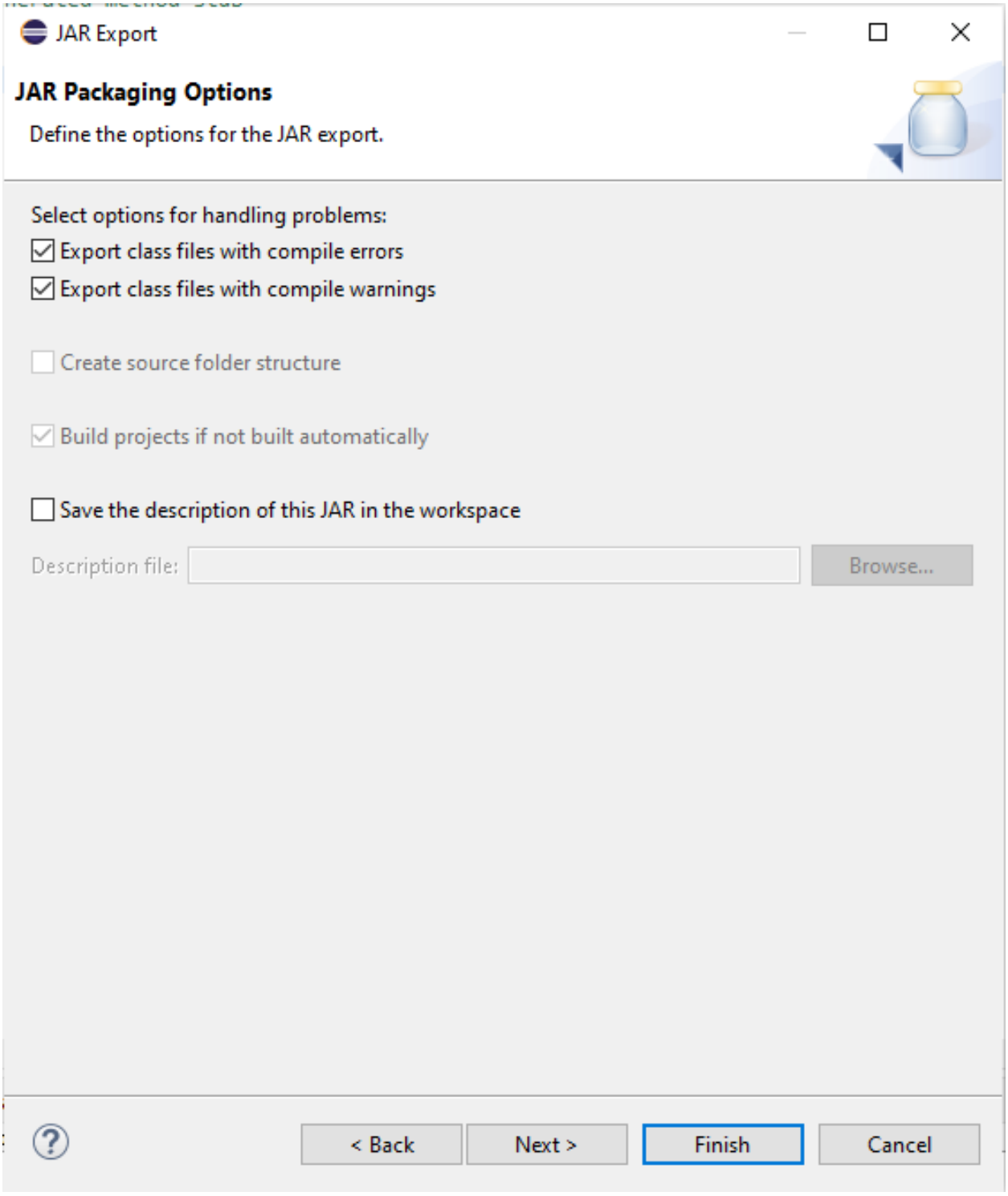
0.170 Shared Library Development - (Eclipse Java Jar Library)-72

we configured output as

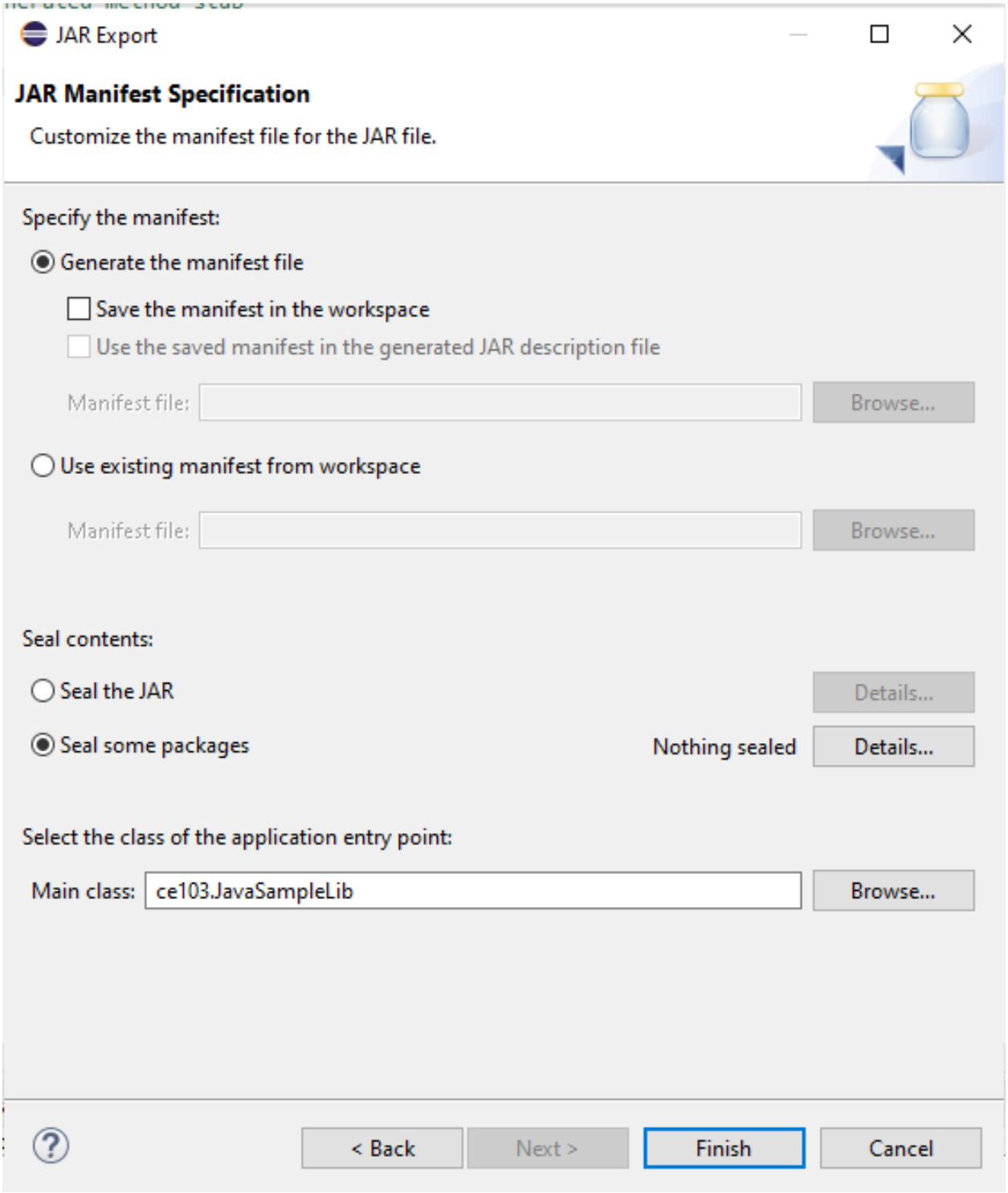
`C:\Users\ugur.coruh\Desktop\java-export-sample\JavaSampleLib.jar`


JAR Export
JAR File Specification
Define which resources should be exported into the JAR.
Select the resources to export:
java-sample-app
java-sample-lib
.classpath
.project
Export generated class files and resources
Export all output folders for checked projects
Export Java source files and resources
Export refactorings for checked projects. Select refactorings...
Select the export destination:
JAR file: C:\Users\ugur.coruh\Desktop\java-export-sample\JavaSampleLib.ja
Browse...
Options:
Compress the contents of the JAR file
Add directory entries
Overwrite existing files without warning
< Back
Next >
Finish
Cancel

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

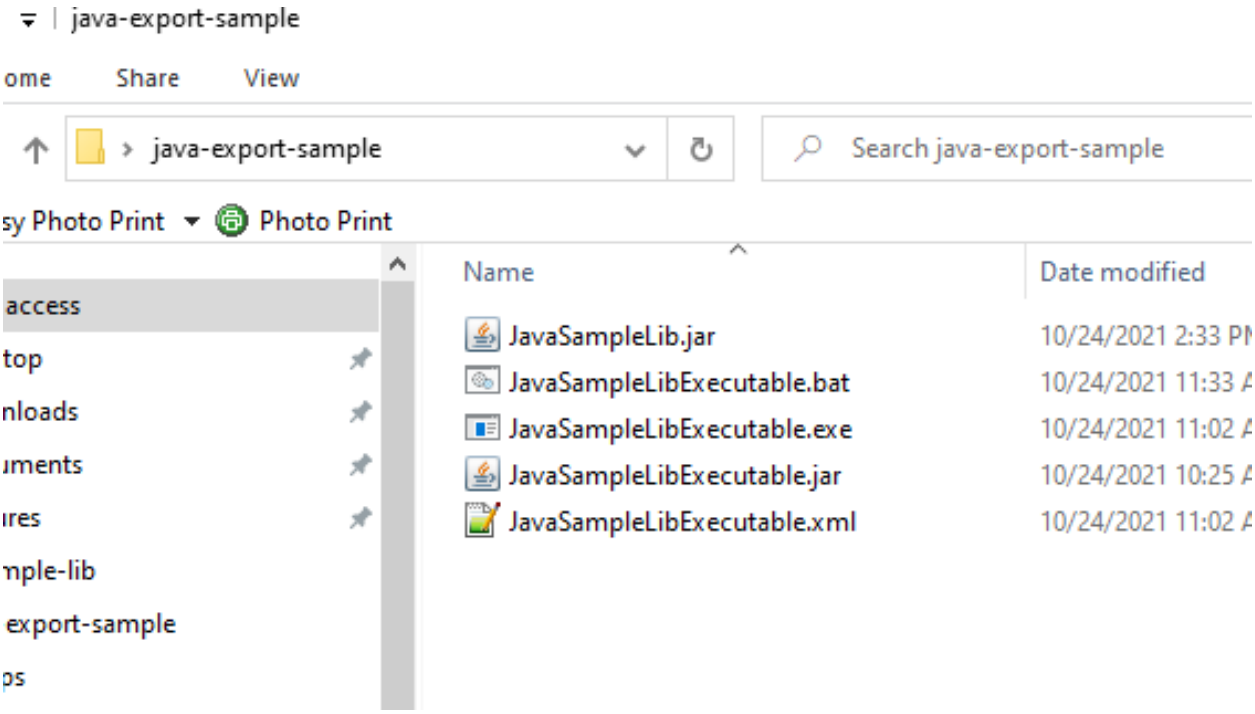


0.172 Shared Library Development - (Eclipse Java Jar Library)-74



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

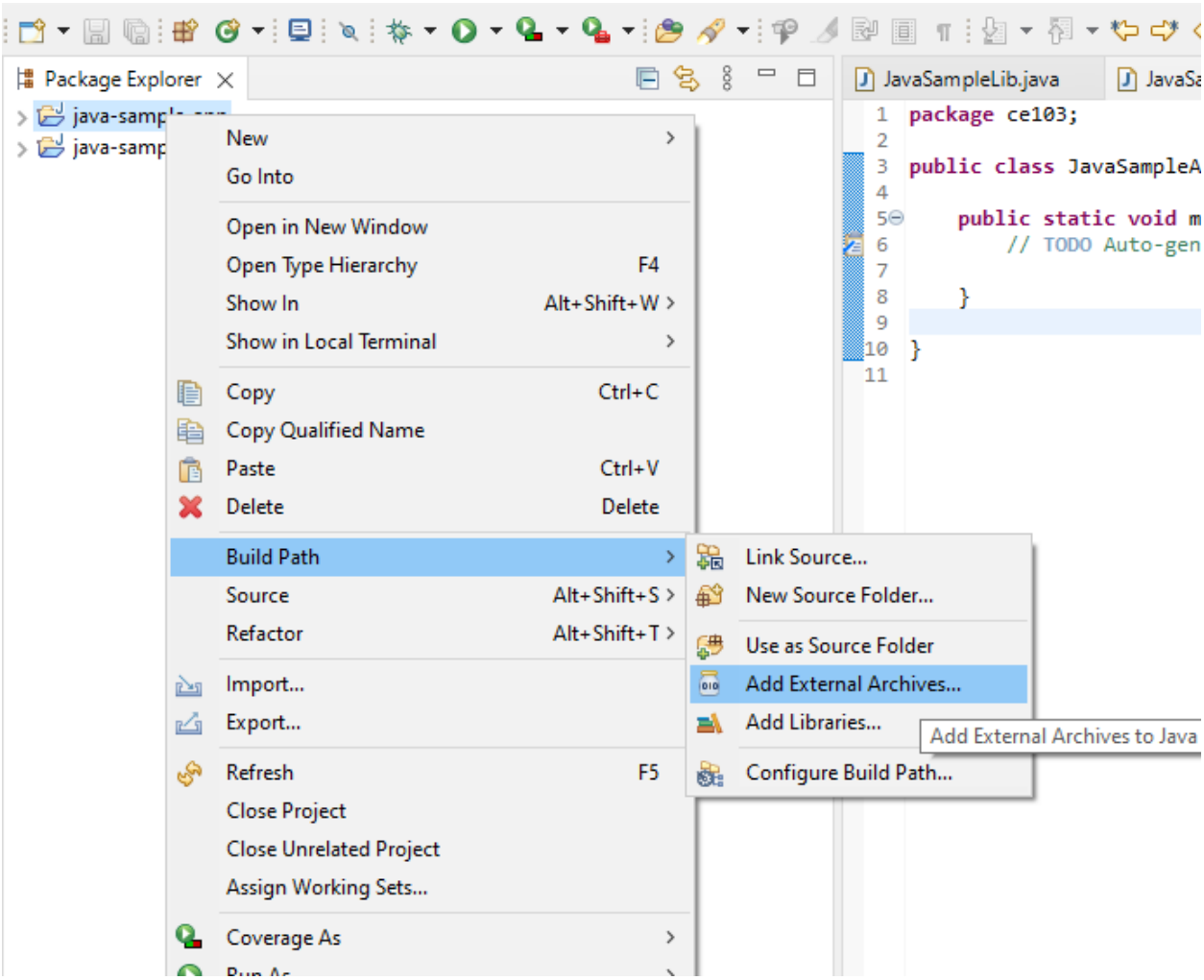
In the same export folder now we have JavaSampleLib.jar



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

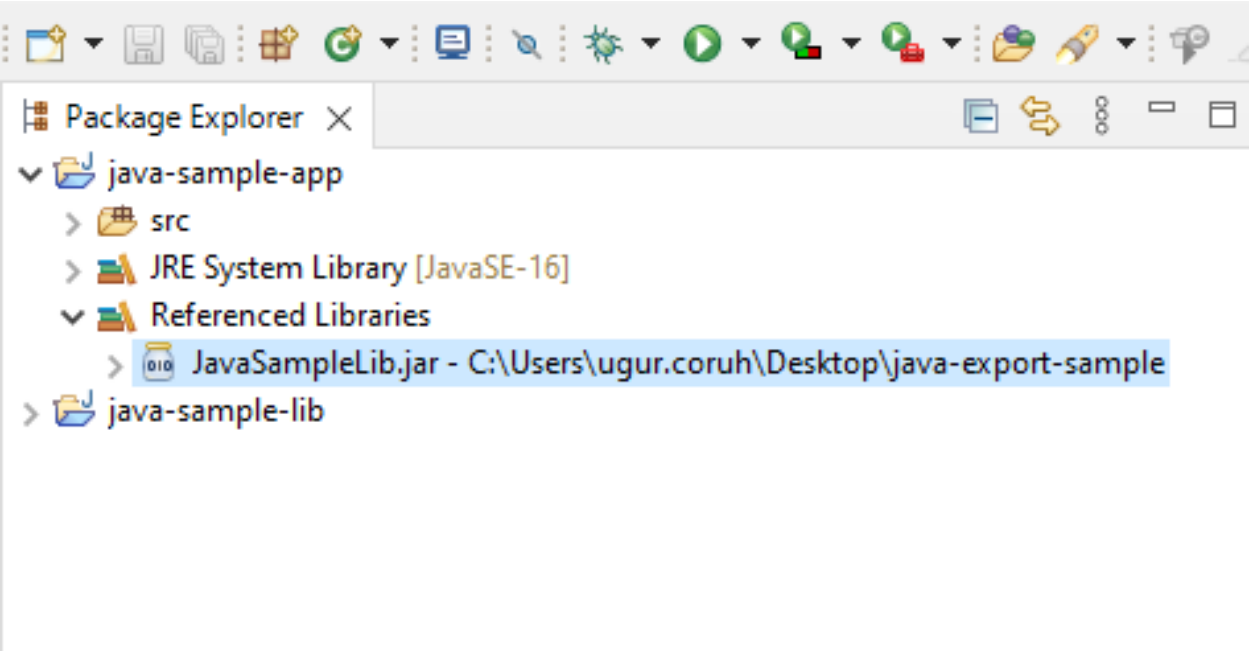
return back to java-sample-app and then add this jar file to our project

Build Path->Add External Archives



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

you will see its added to reference libraries



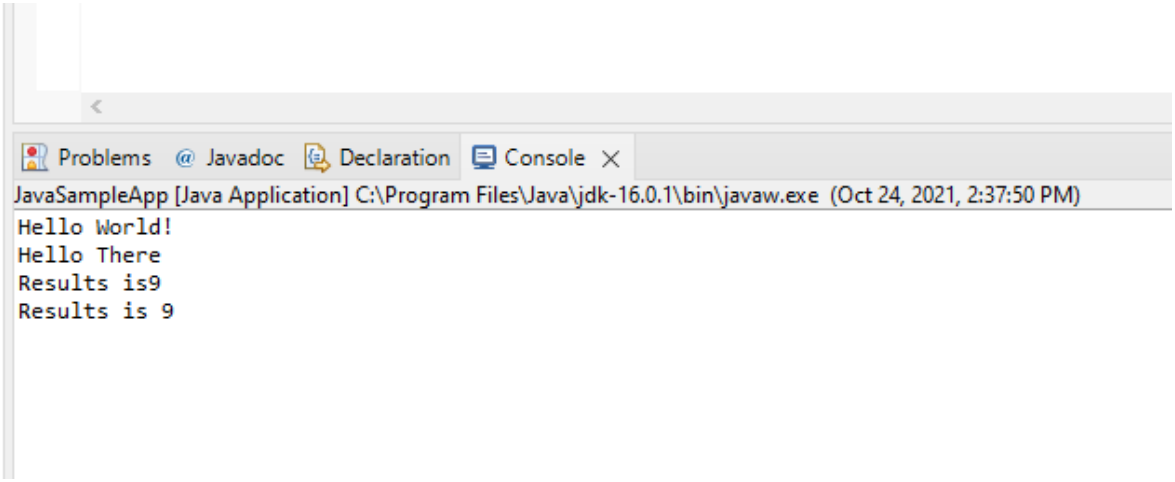
0.176 Shared Library Development - (Eclipse Java Jar Library)-78

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();
        }

    }

}
```

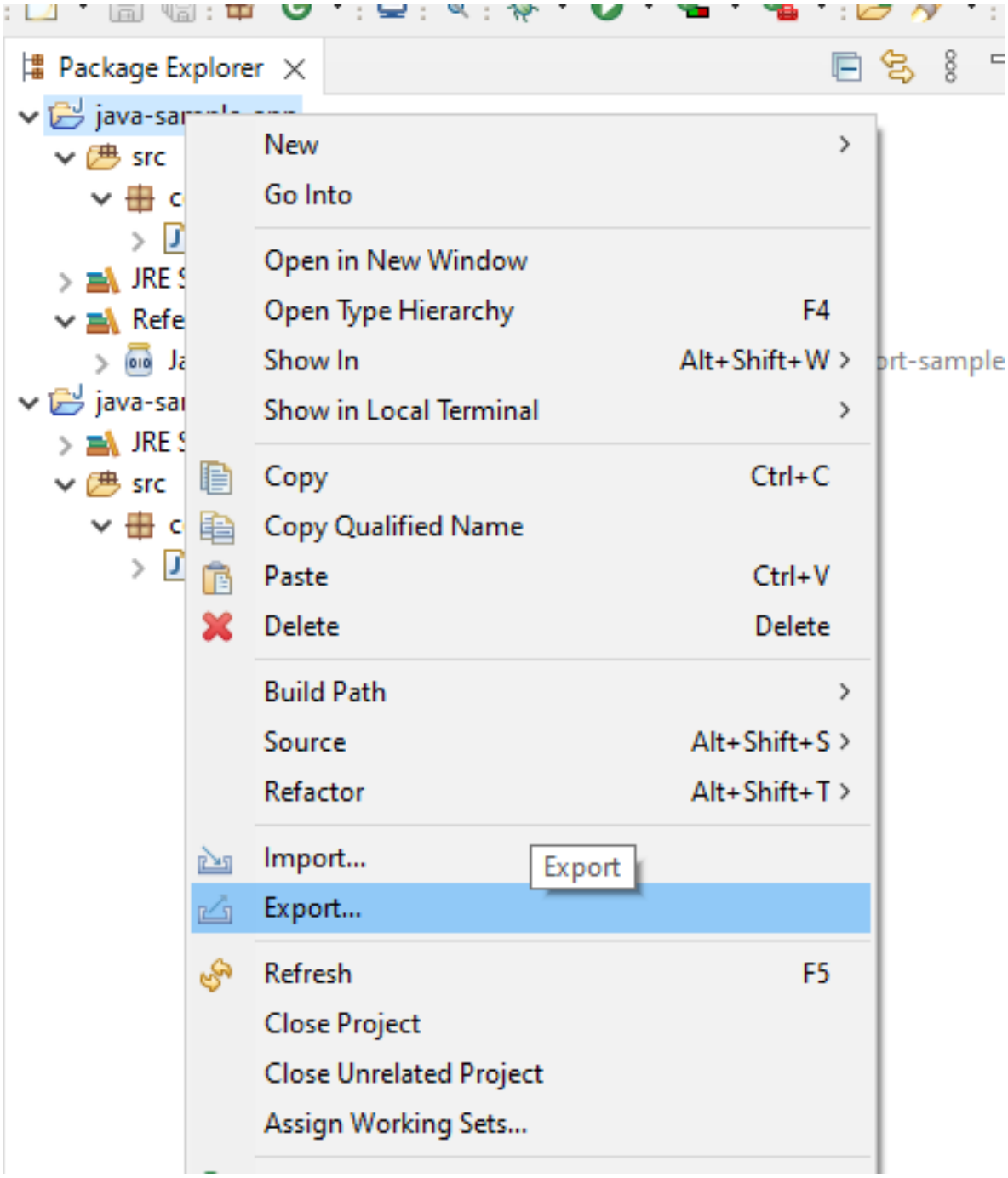
0.177 Shared Library Development - (Eclipse Java Jar Library)-79

When we run application we will see similar output



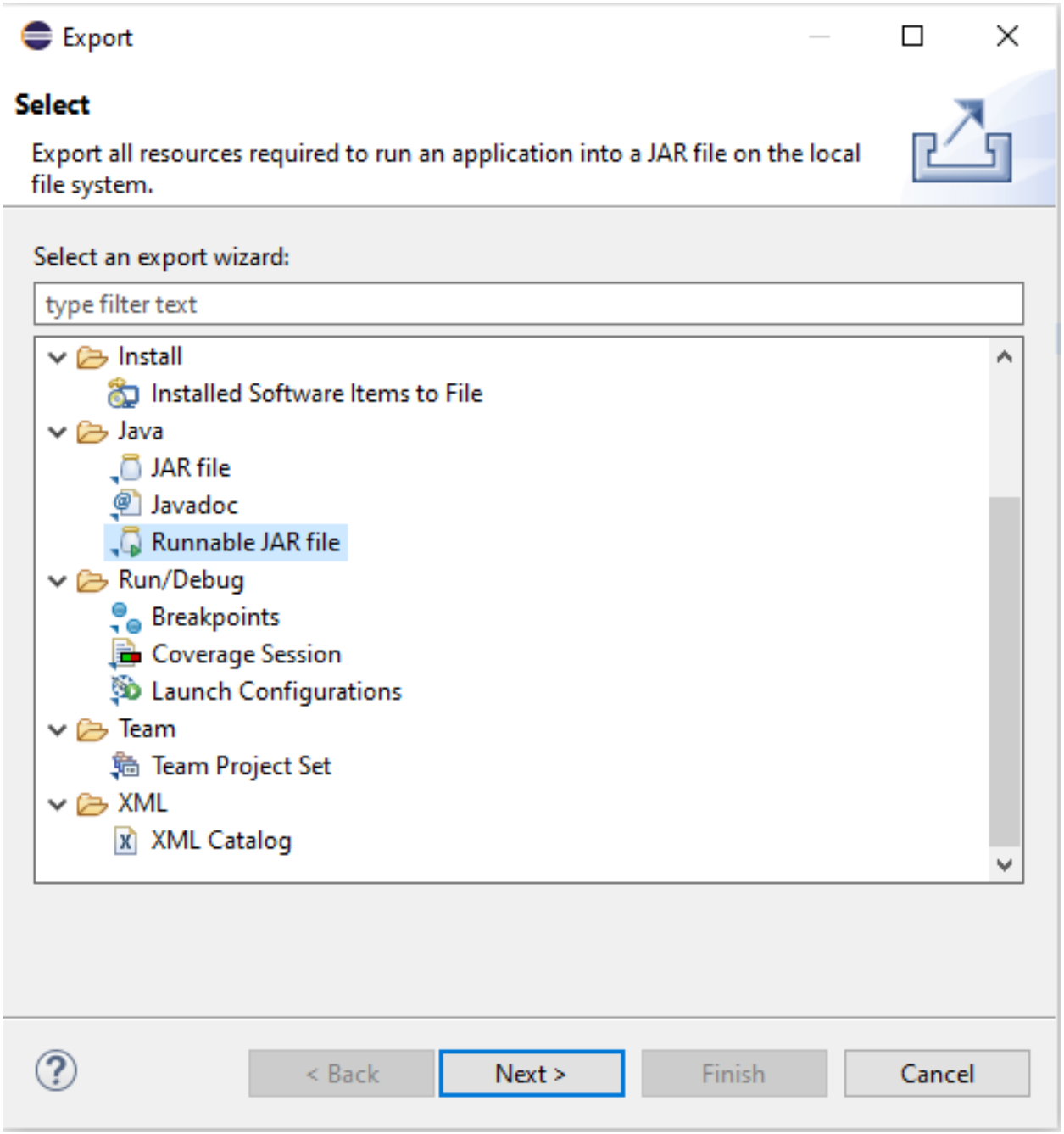
0.178 Shared Library Development - (Eclipse Java Jar Library)-80

Lets export this application with its dependent library



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

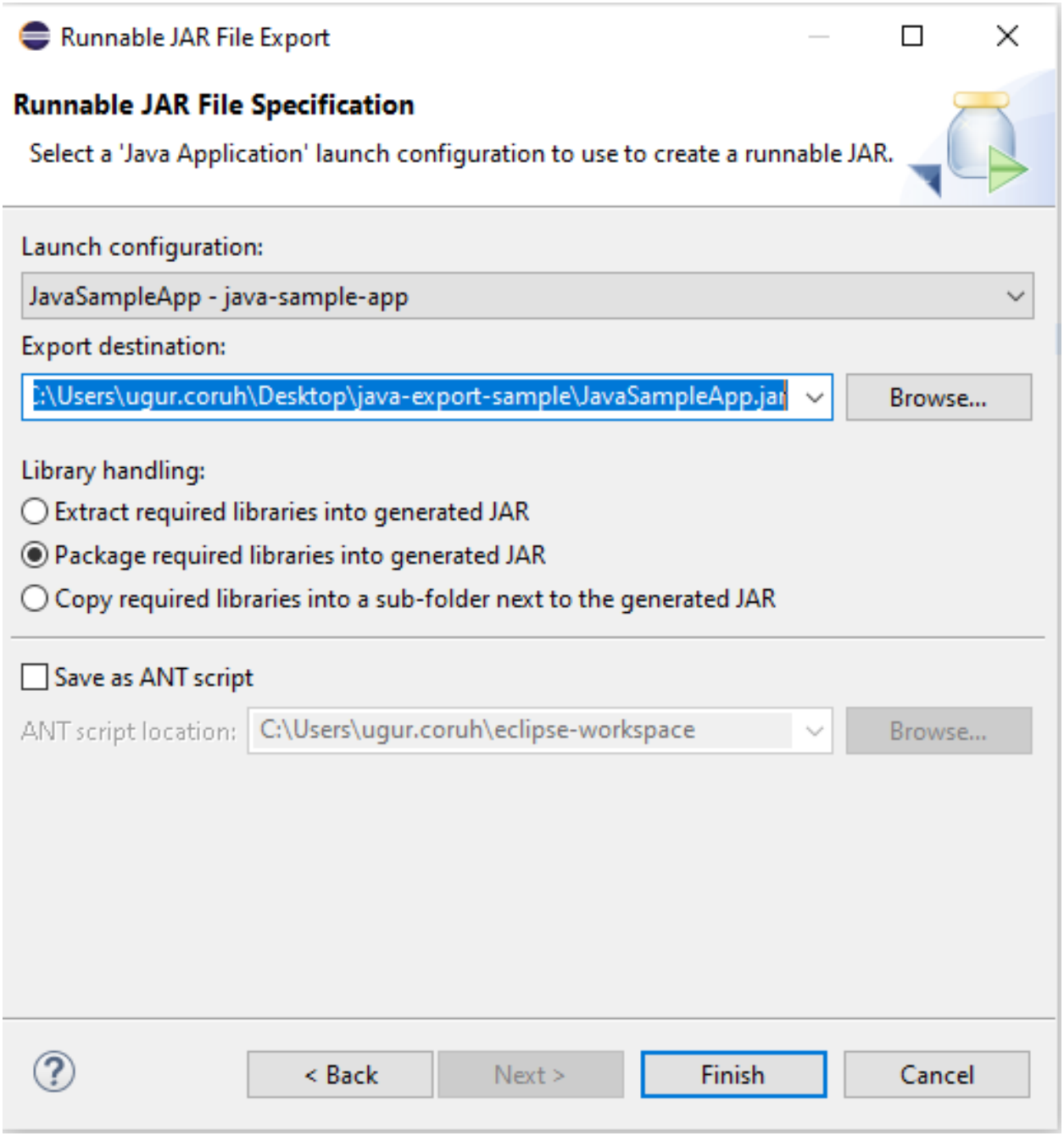
Select runnable jar



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

Set Launch configuration and Export destination

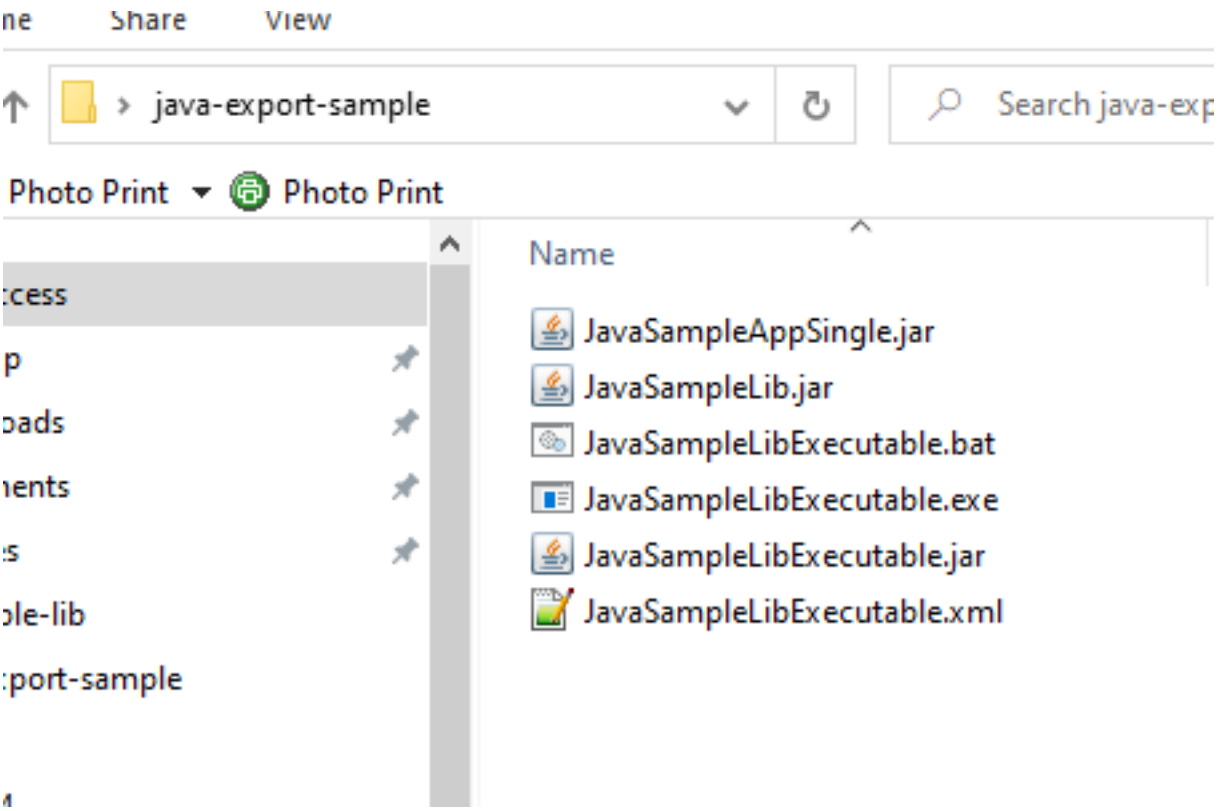
`C:\Users\ugur.coruh\Desktop\java-export-sample\JavaSampleAppSingle.jar`



0.181 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

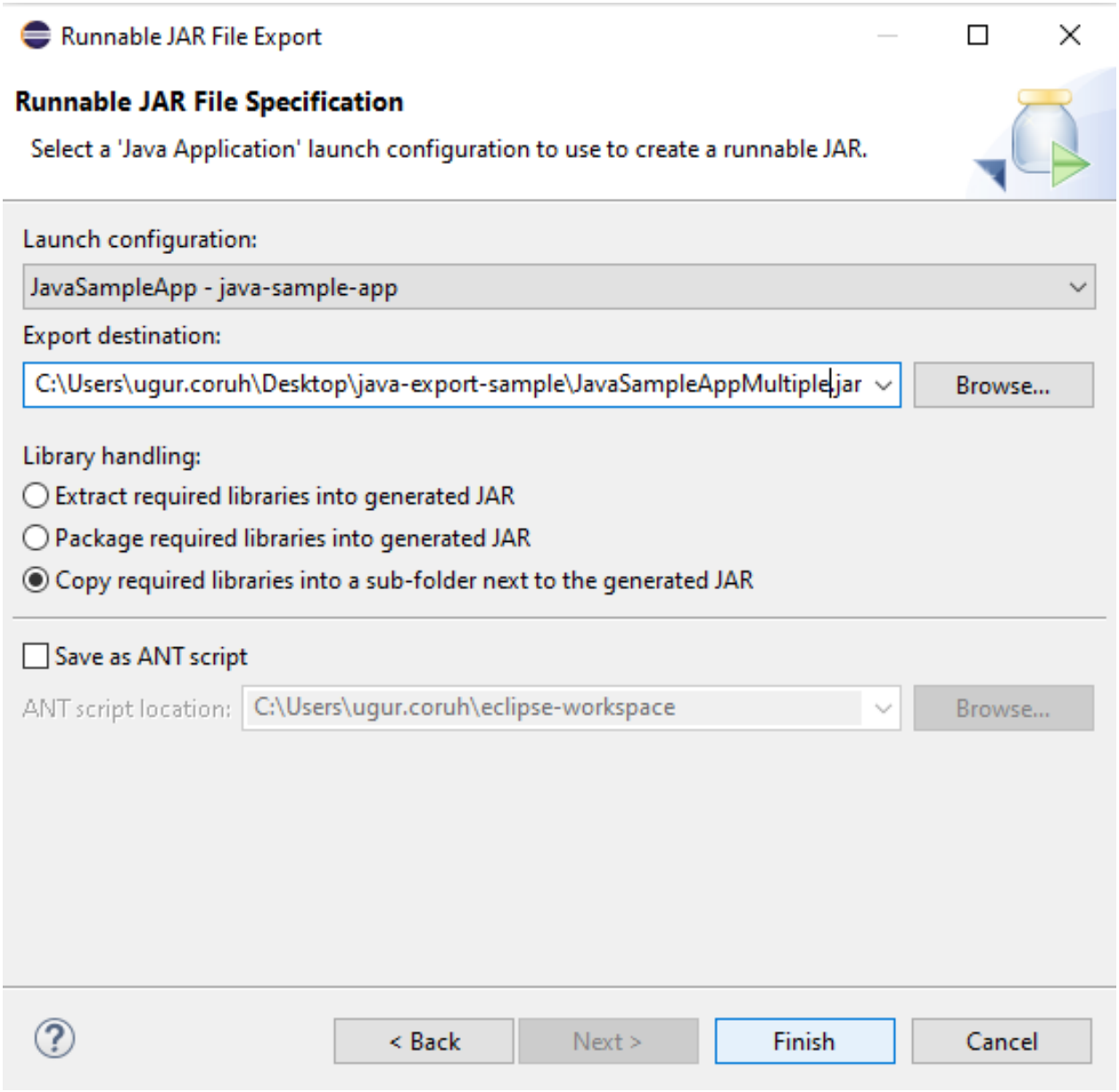
```
C:\Users\ugur.coruh\Desktop\java-export-sample>java -jar JavaSampleAppSingle.jar
Hello World!
Hello There
Results is9
Results is 9
```

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

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

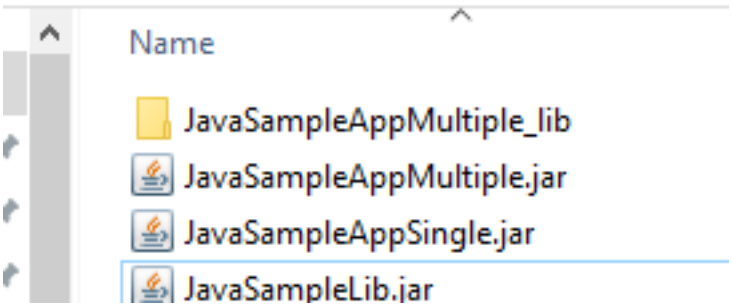
only change copy required libraries setting and then give a new name for new jar file and export

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



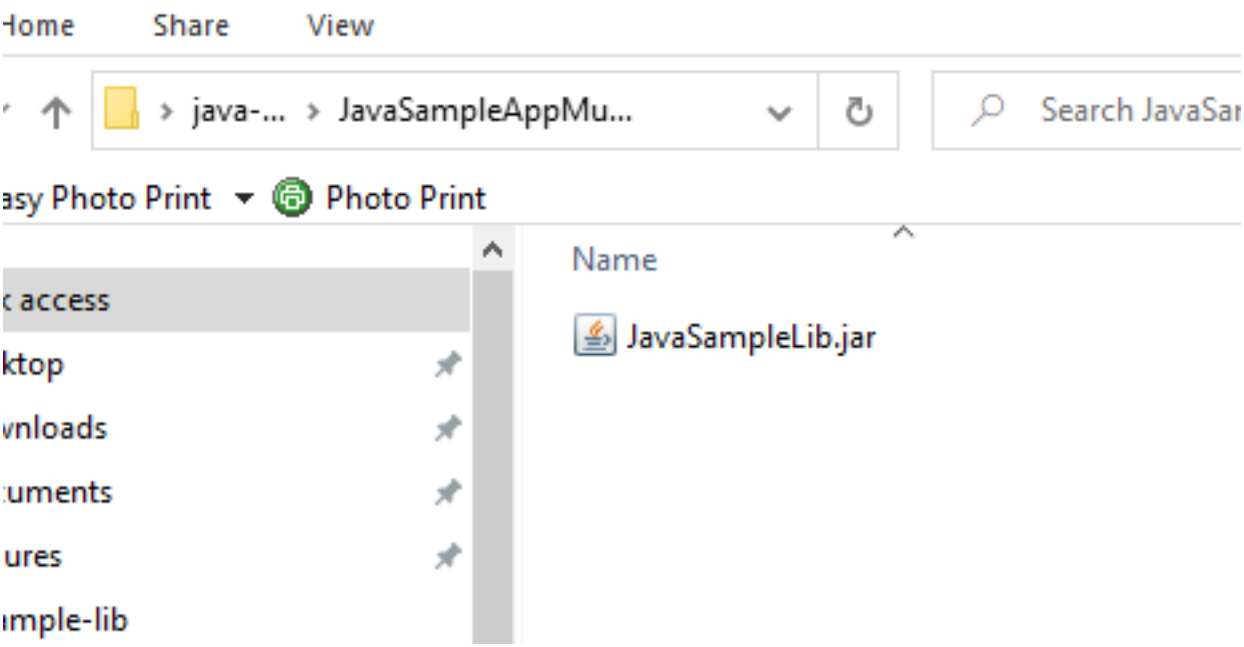
0.183 Shared Library Development - (Eclipse Java Jar Library)-85

now we have a folder that contains our libraries referenced



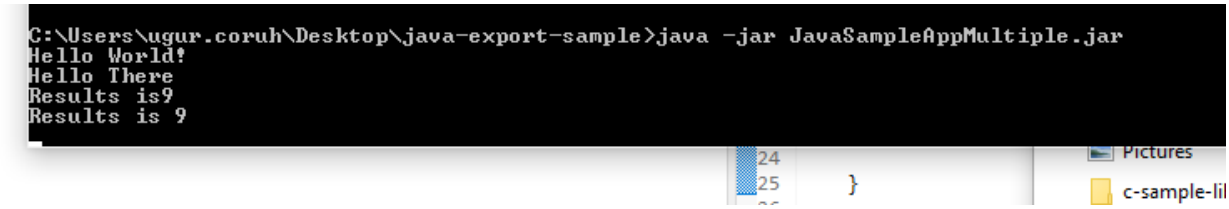
0.184 Shared Library Development - (Eclipse Java Jar Library)-86

in this file we can find our library



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

if we test our application we will see it will work



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

0.186 Application Testing

- C
- C++
- C#
- Java

0.187 Unit Test Development

Wikipedia Unit Test Library List for Each Language

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

---

0.187.0.1 Visual Studio Community Edition

0.187.0.1.1 C Unit Tests

0.187.0.2 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

0.187.0.3 Visual Studio Community Edition

0.187.0.3.1 C++ Unit Tests

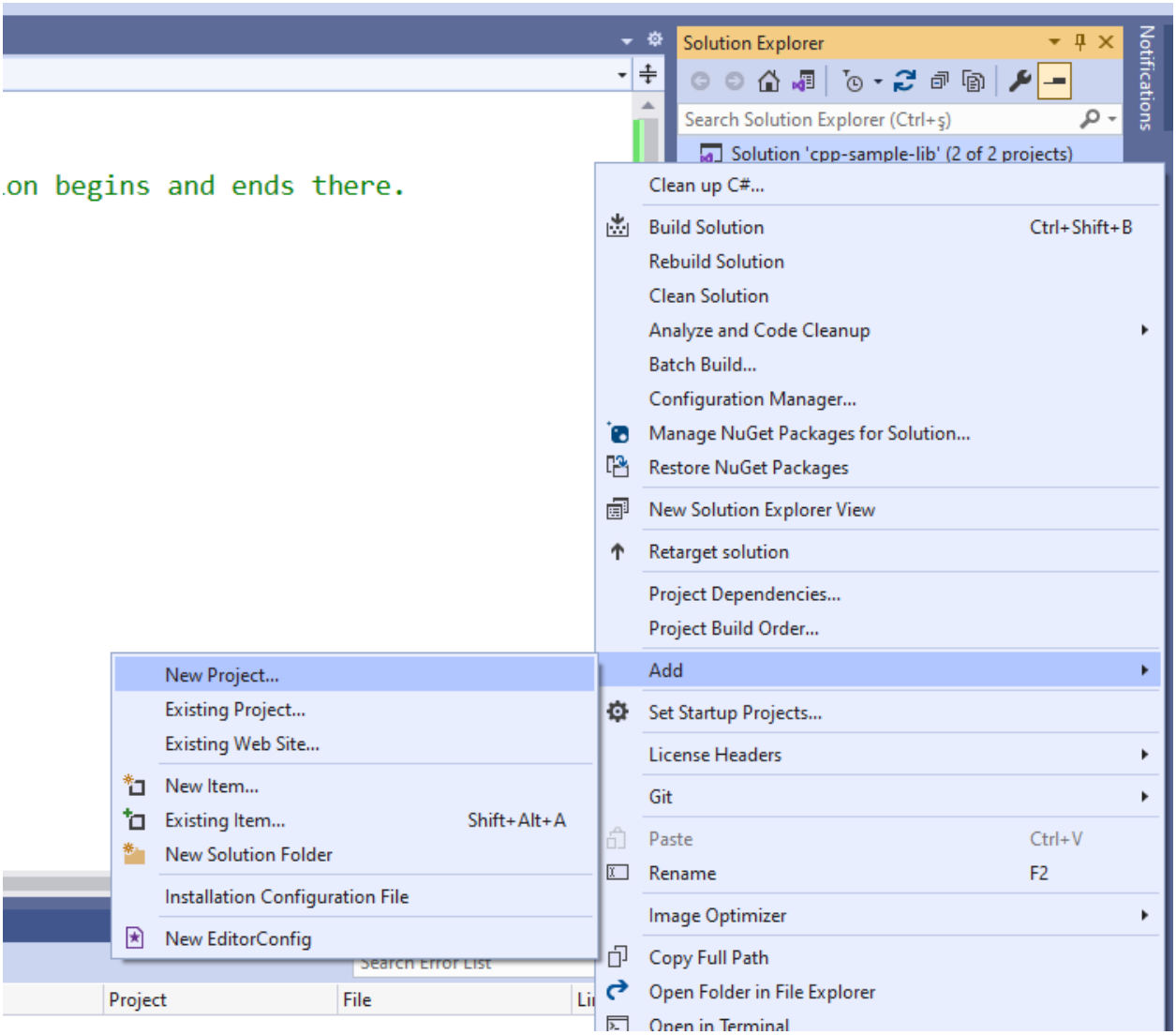
0.187.0.4 Visual Studio Community Edition - C++ Unit Tests-1

- C/C++ için birim testleri yazma - Visual Studio (Windows) | Microsoft Docs[12]

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

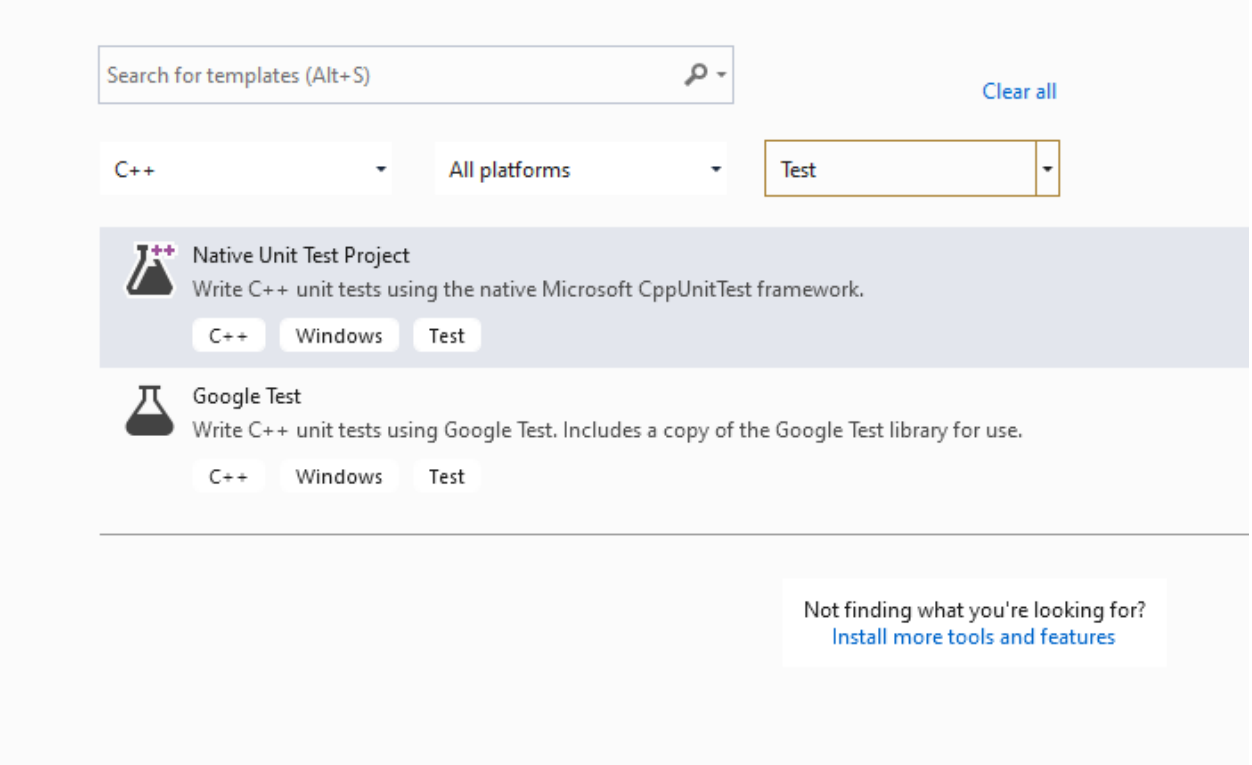
- Use cpp-sample-lib project and add

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



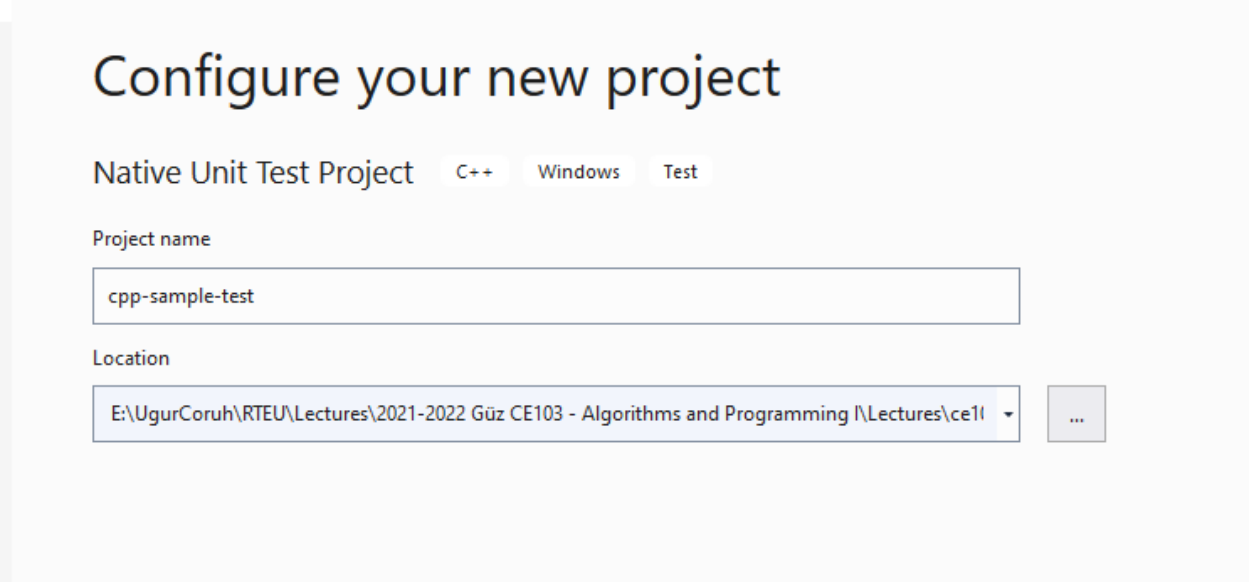
0.187.0.6 Visual Studio Community Edition - C++ Unit Tests-3

- Select Native Unit Test



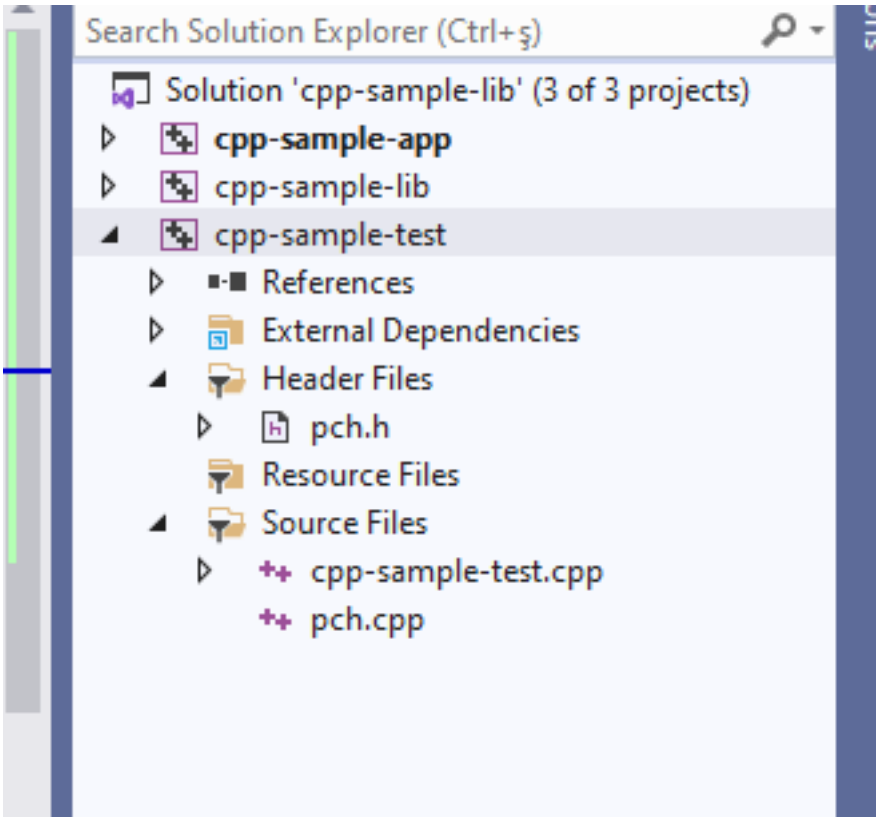
0.187.0.7 Visual Studio Community Edition - C++ Unit Tests-4

- Set project path and name



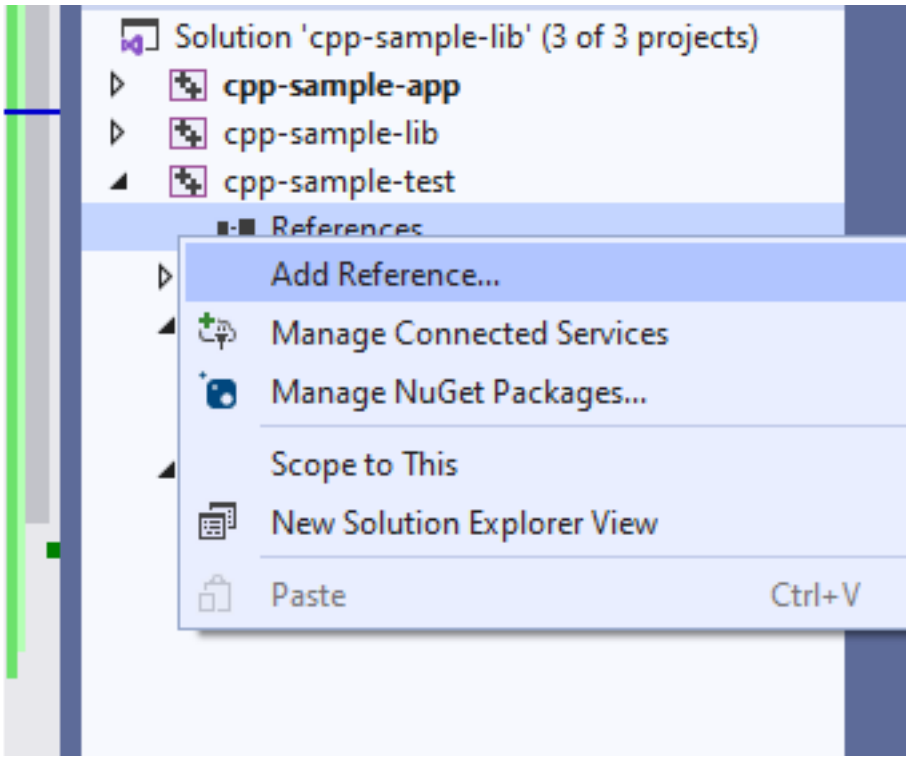
0.187.0.8 Visual Studio Community Edition - C++ Unit Tests-5

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



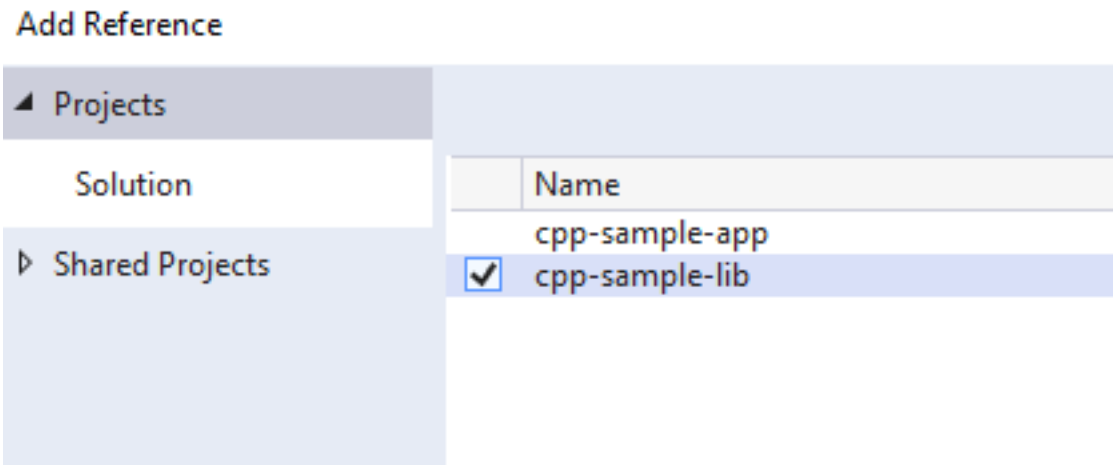
0.187.0.9 Visual Studio Community Edition - C++ Unit Tests-6

- Add library project from references



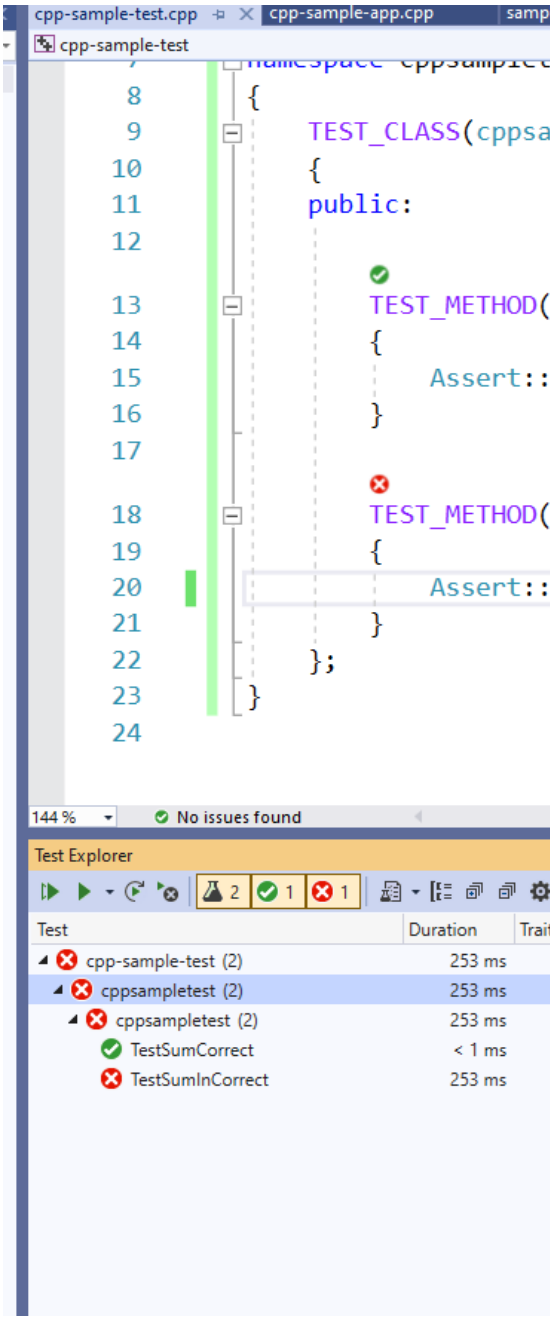
0.187.0.10 Visual Studio Community Edition - C++ Unit Tests-7

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



0.187.0.11 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));
        }
    };
}
```



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

0.187.0.13 Visual Studio Community Edition

0.187.0.13.1 C# Unit Tests

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

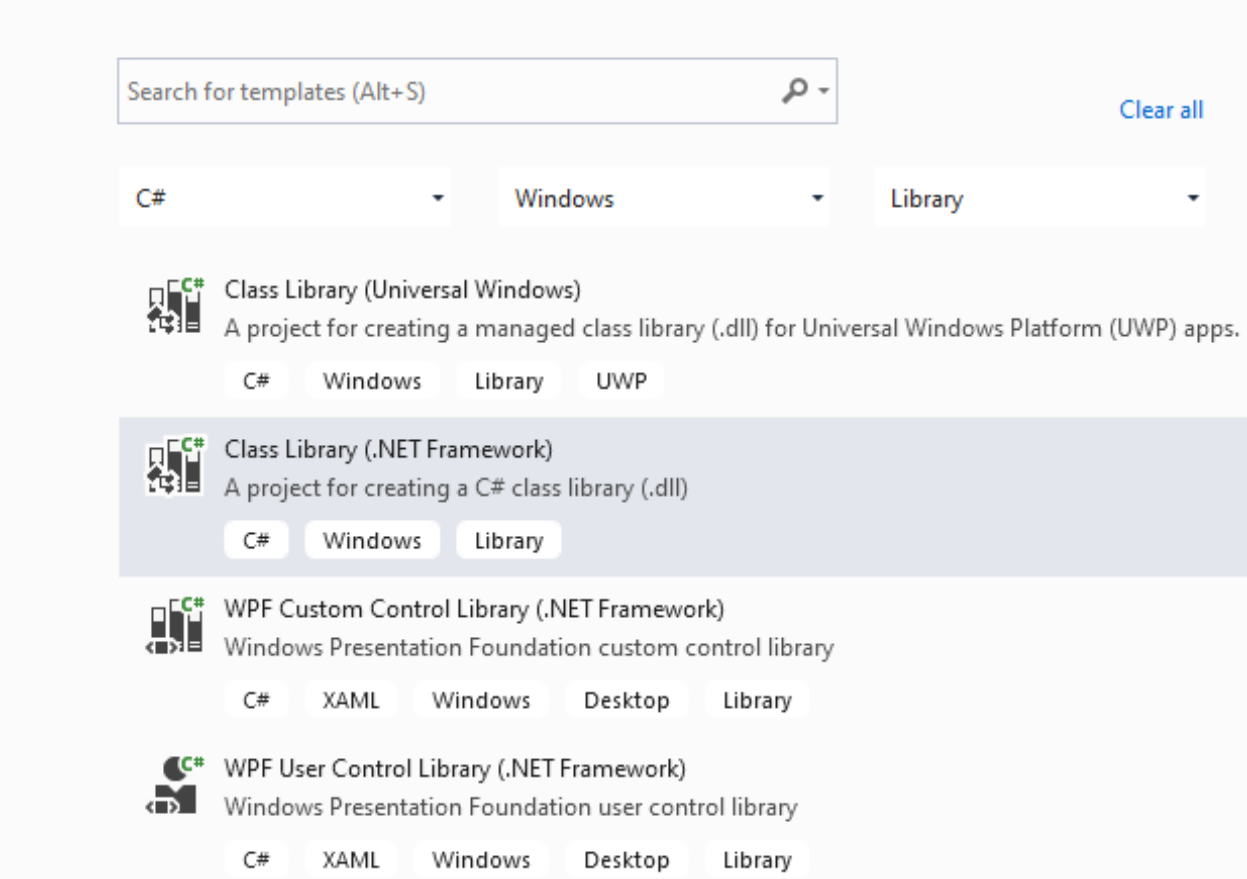
0.187.1 Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-1

- Install extension fine code coverage

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

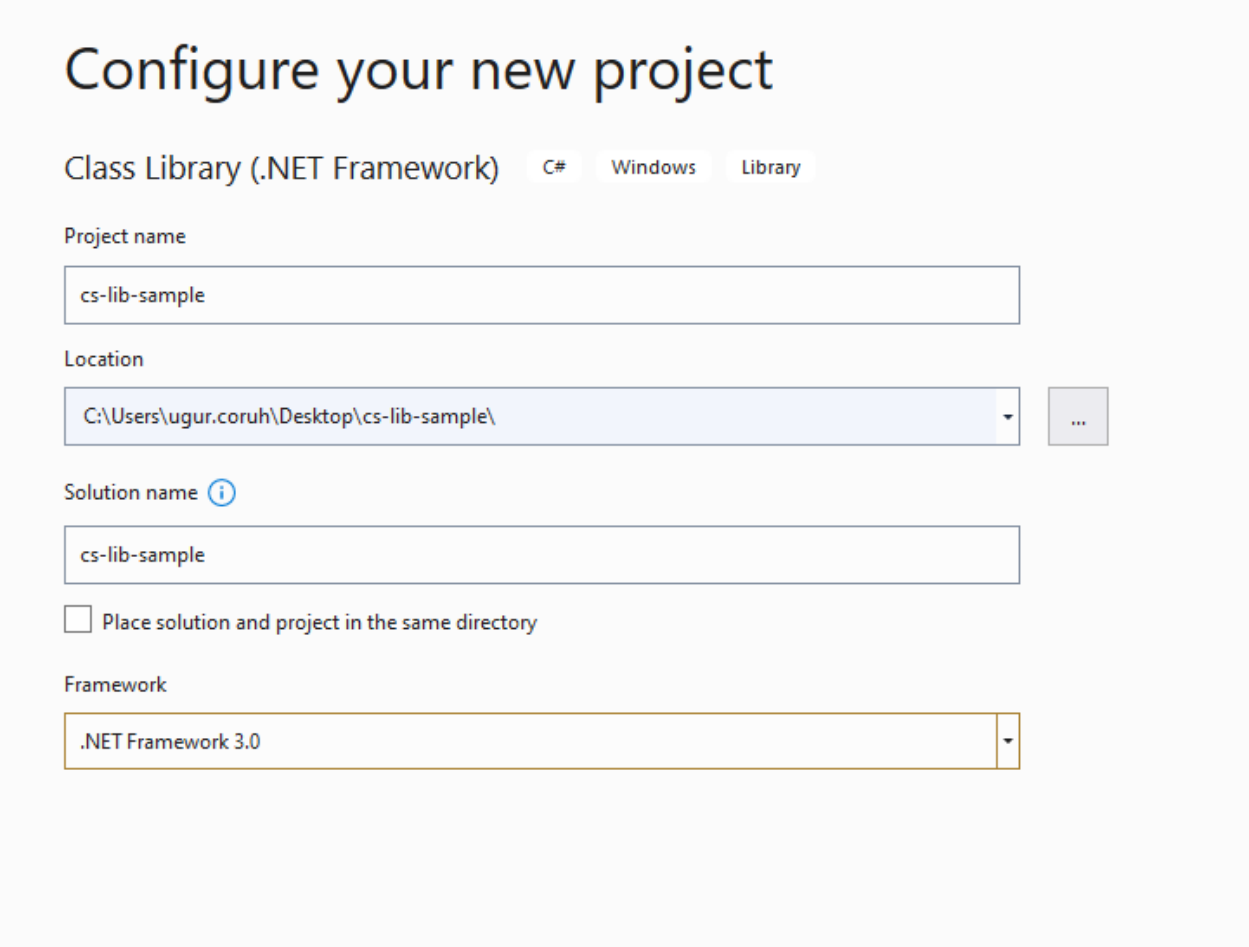
0.187.2 Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-2

- Create a .Net Framework Library



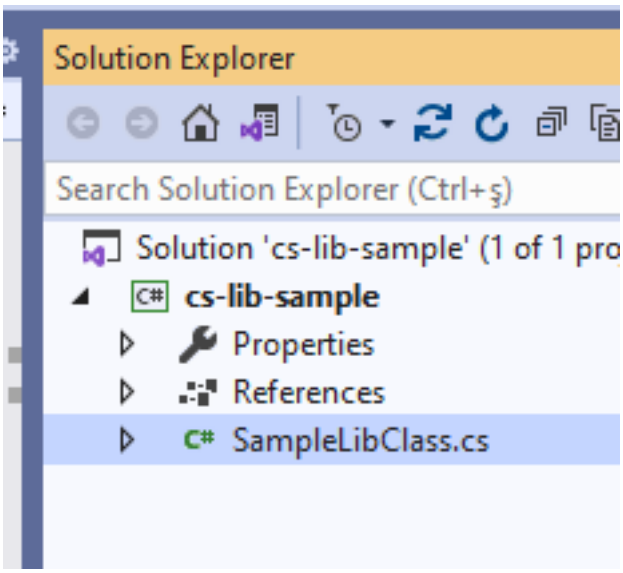
0.187.3 Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-3

- Set project framework and path



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

- Create library functions

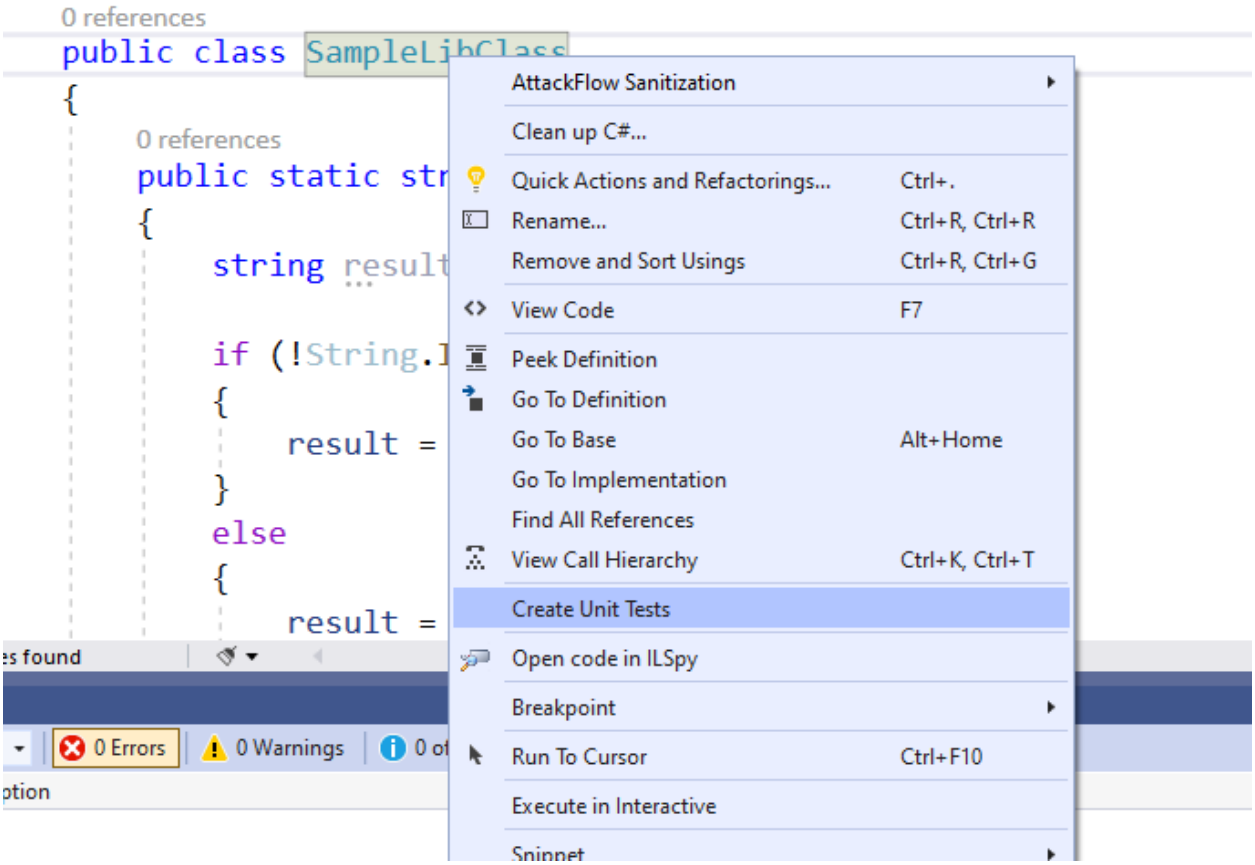


0.187.5 Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-5

```
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;
        }
    }
}
```

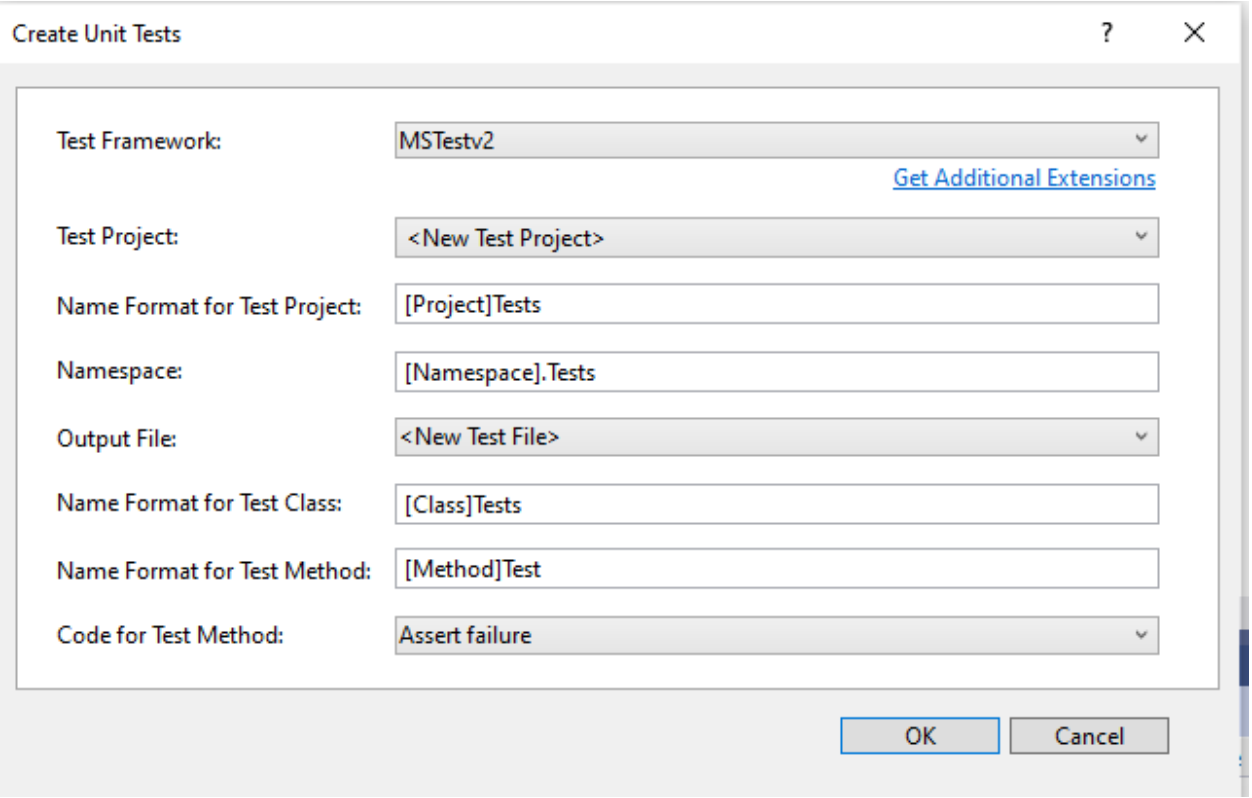
0.187.6 Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-6

- Right click and then create unit test project



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

- Press OK



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

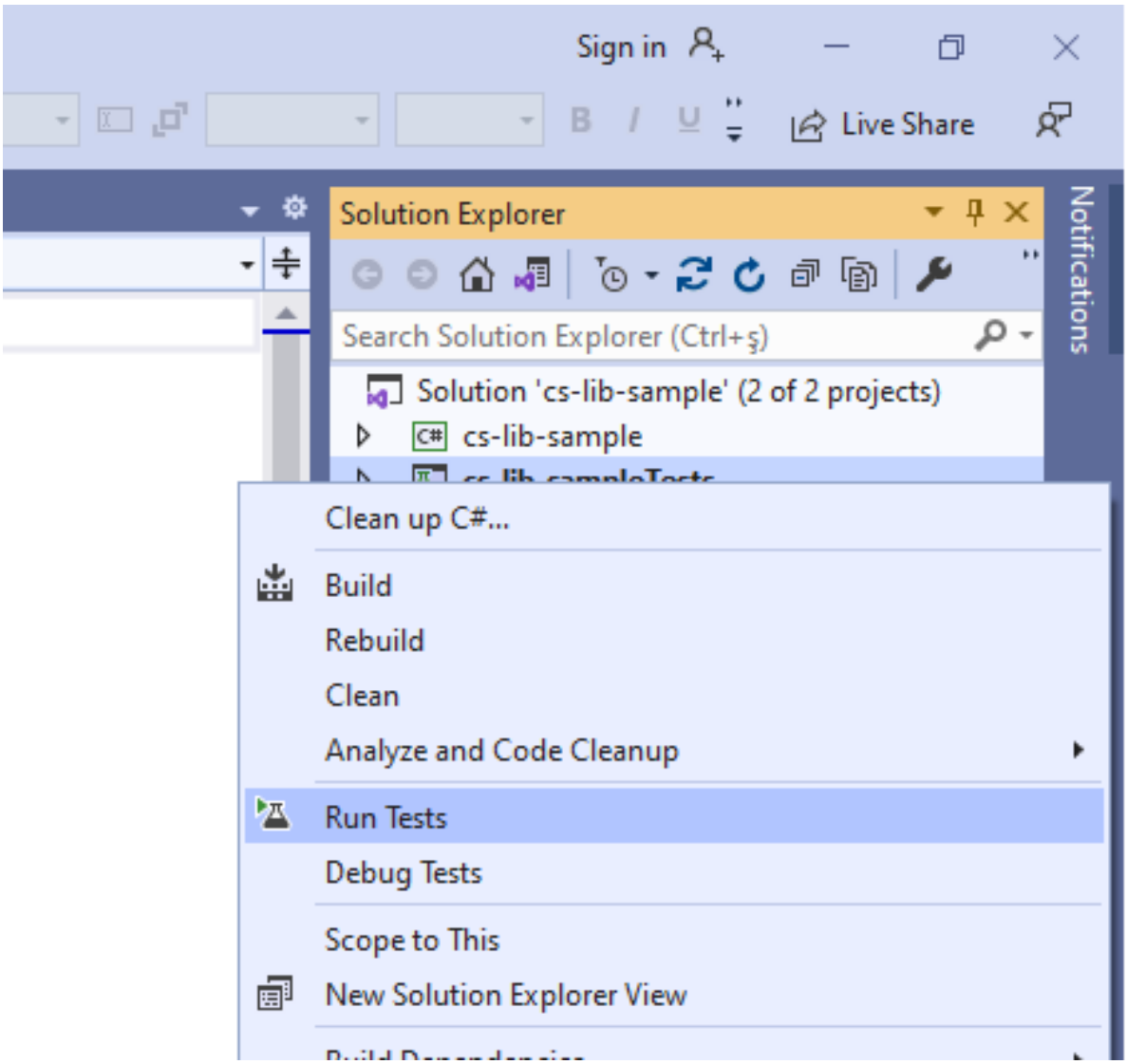
- Enter test code

```
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
        }
        [TestMethod()]
        public void testSayHelloToWrong()
        {
            Assert.AreEqual("Hello All", SampleLibClass.sayHelloTo("Computer"), "Regular say hello won'
        }
```

```
        [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");
        }


    }
}
```

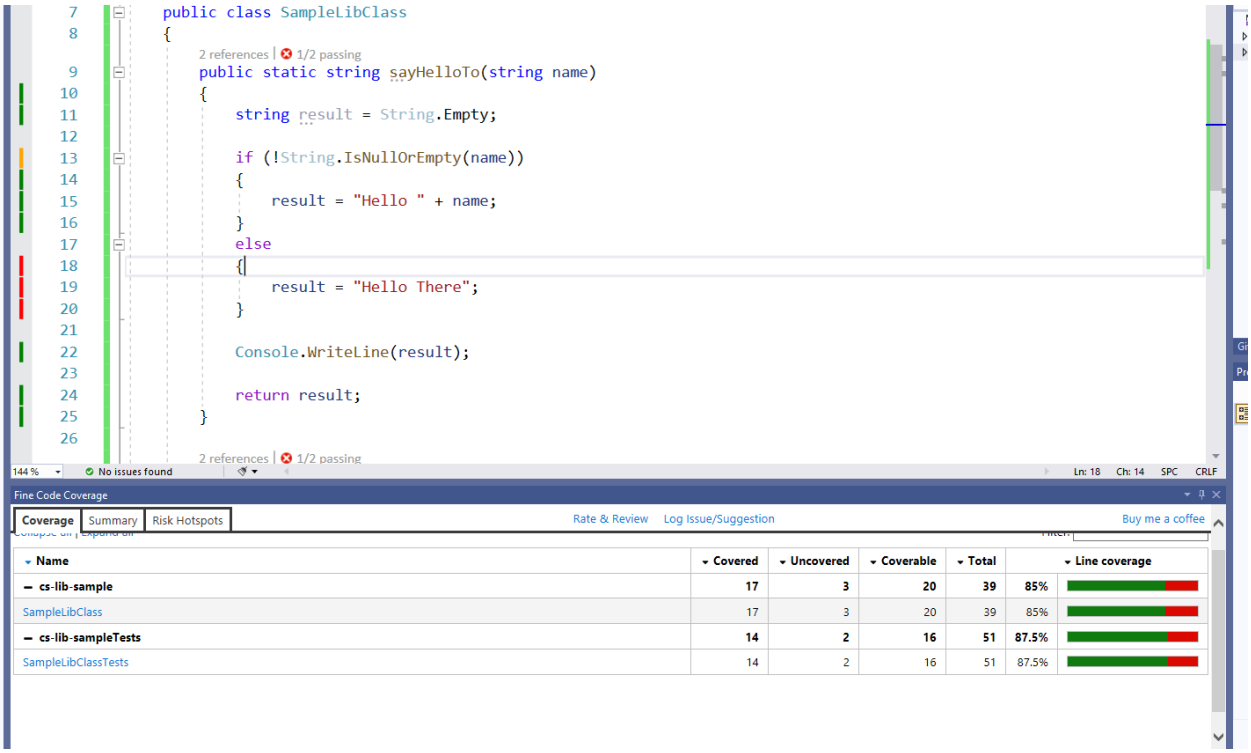
0.187.9 Visual Studio Community Edition (C# Unit Test + MSTestV2+.Net)-9

- Run tests



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

you will code coverage and entered or passed branches

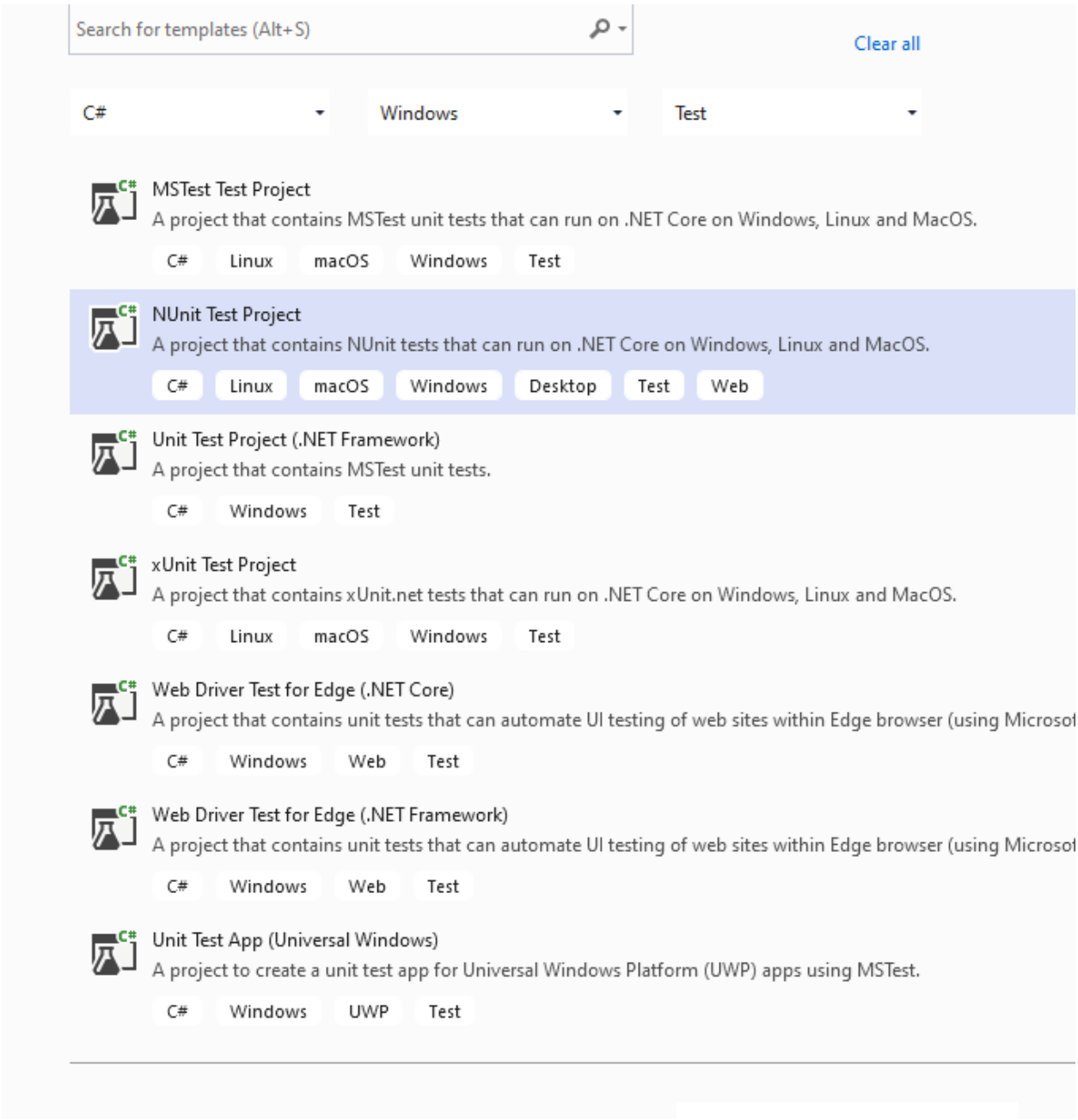


0.187.10.1 Visual Studio Community Edition

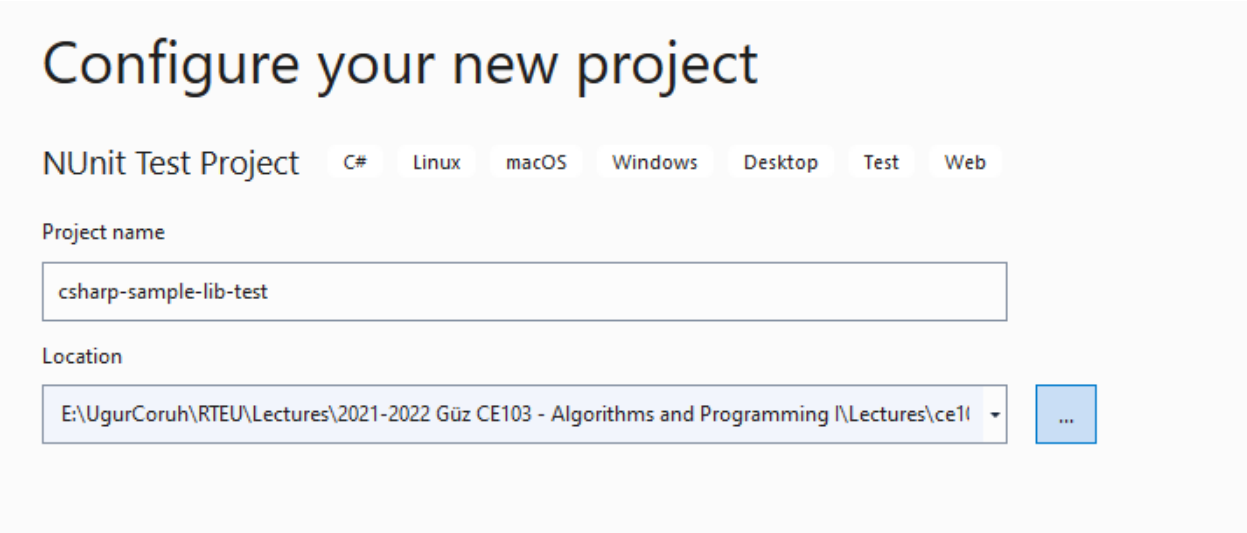
0.187.10.1.1 C# Unit Test + NUnit + .NETCore

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

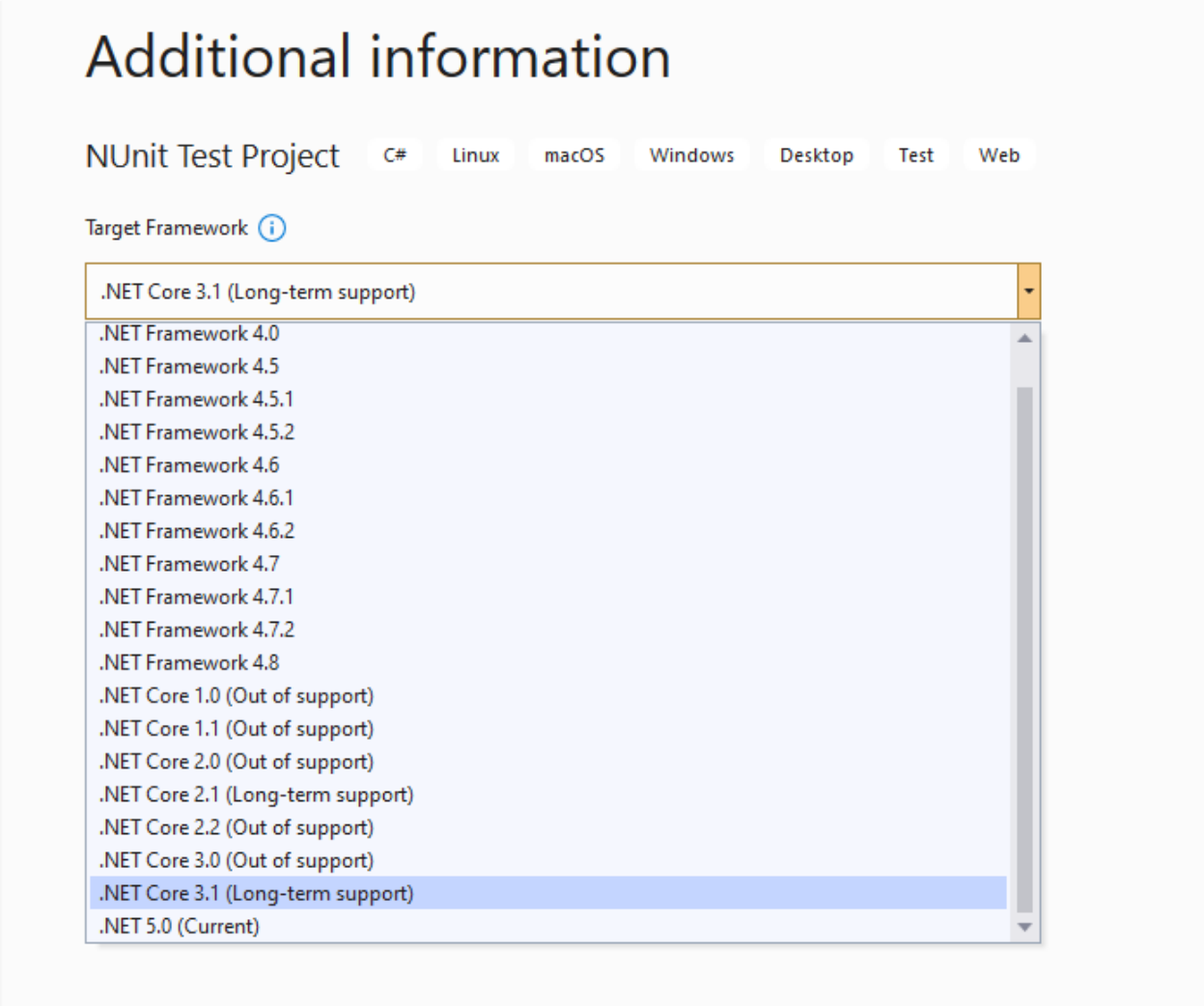
- Use `cshar-sample-lib` for this example
- Create and add a unit test project to solution


Search for templates (Alt+S)
Clear all
C#
Windows
Test
MSTest Test Project
A project that contains MSTest unit tests that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Test
NUnit Test Project
A project that contains NUnit tests that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Desktop Test Web
Unit Test Project (.NET Framework)
A project that contains MSTest unit tests.
C# Windows Test
xUnit Test Project
A project that contains xUnit.net tests that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Test
Web Driver Test for Edge (.NET Core)
A project that contains unit tests that can automate UI testing of web sites within Edge browser (using Microsof
C# Windows Web Test
Web Driver Test for Edge (.NET Framework)
A project that contains unit tests that can automate UI testing of web sites within Edge browser (using Microsof
C# Windows Web Test
Unit Test App (Universal Windows)
A project to create a unit test app for Universal Windows Platform (UWP) apps using MSTest.
C# Windows UWP Test

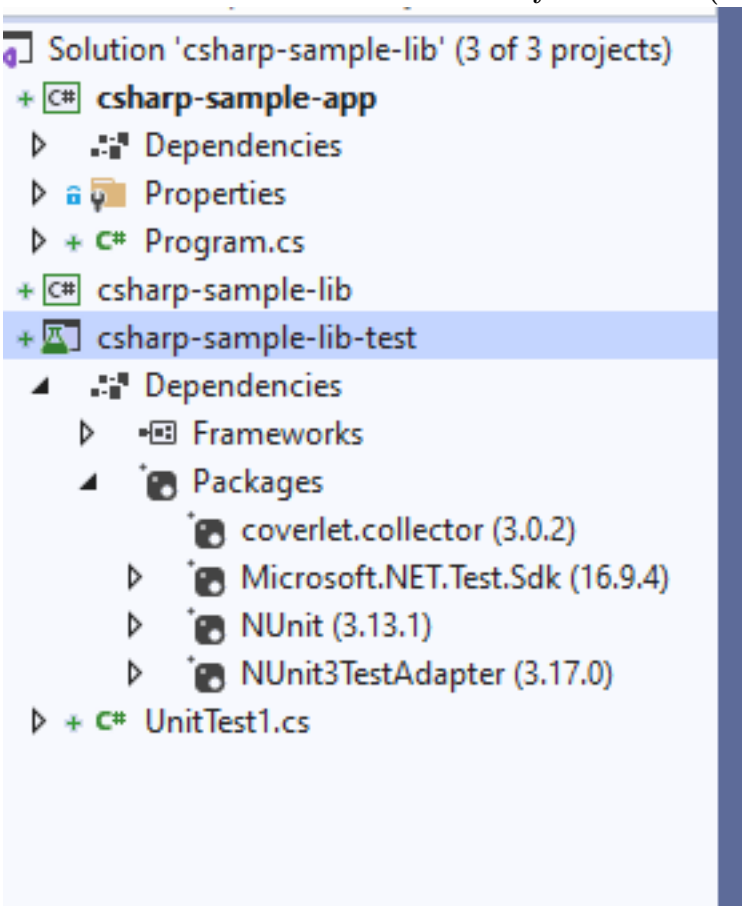
0.187.10.3 Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-2



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

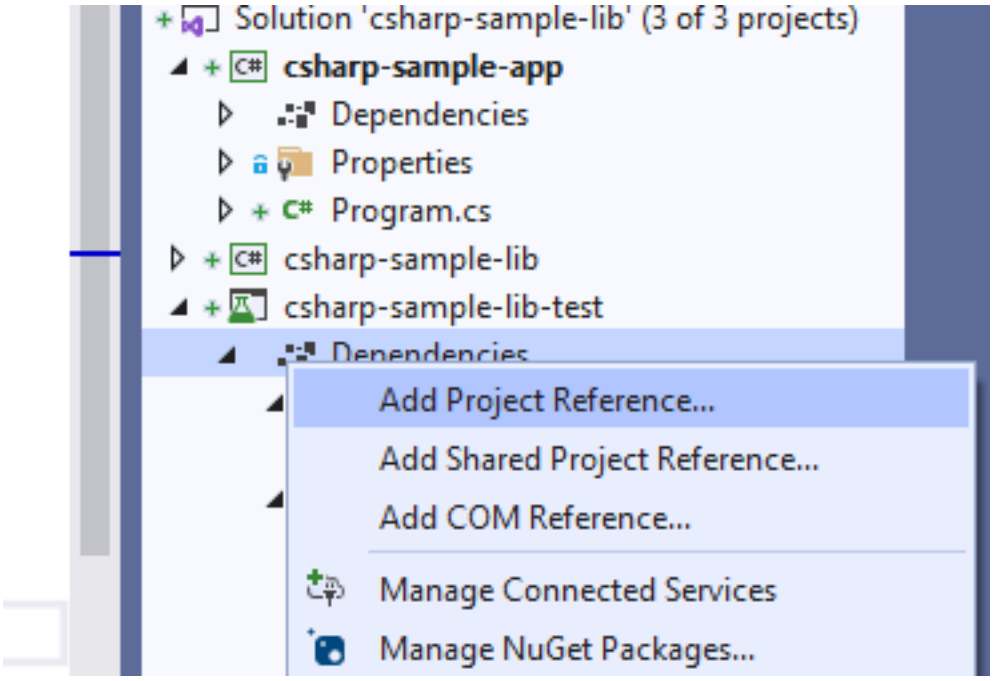


0.187.10.5 Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-4

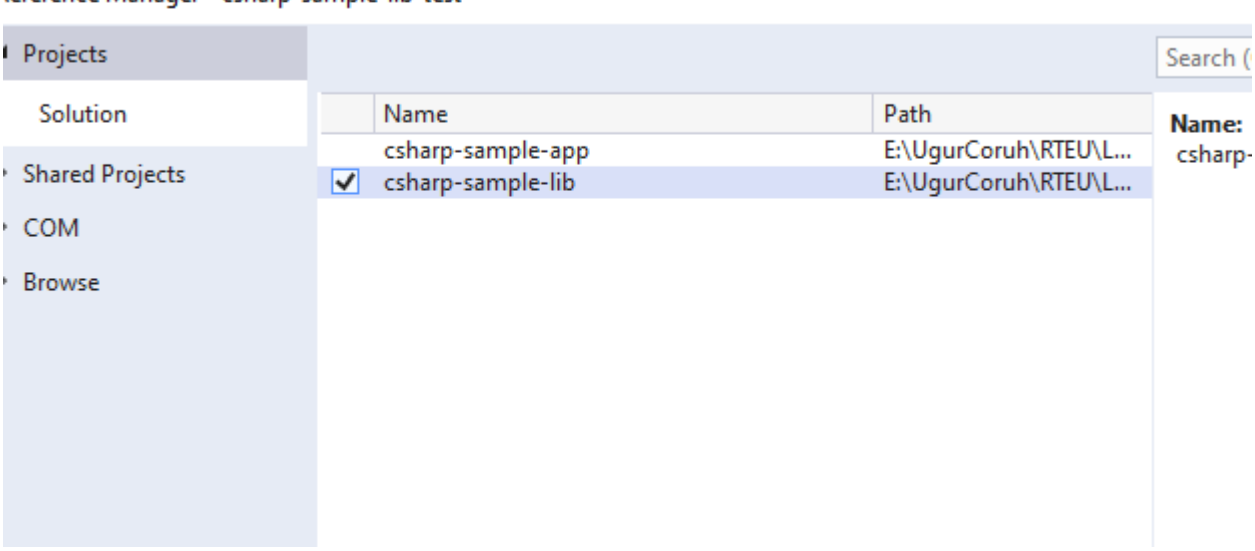


0.187.10.6 Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-5

- Add project reference



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



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

```
using csharp_sample_lib;
using NUnit.Framework;

namespace csharp_sample_lib_test
{
    public class SampleLibraryTestClass
    {
        sampleLibClass sampleLib;

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

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

        [Test]
        public void testSayHelloToWrong()
        {
            Assert.AreEqual("Hello All", sampleLibClass.sayHelloTo("Computer"), "Regular say hello won'

        [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");
        }
    }
}
```

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

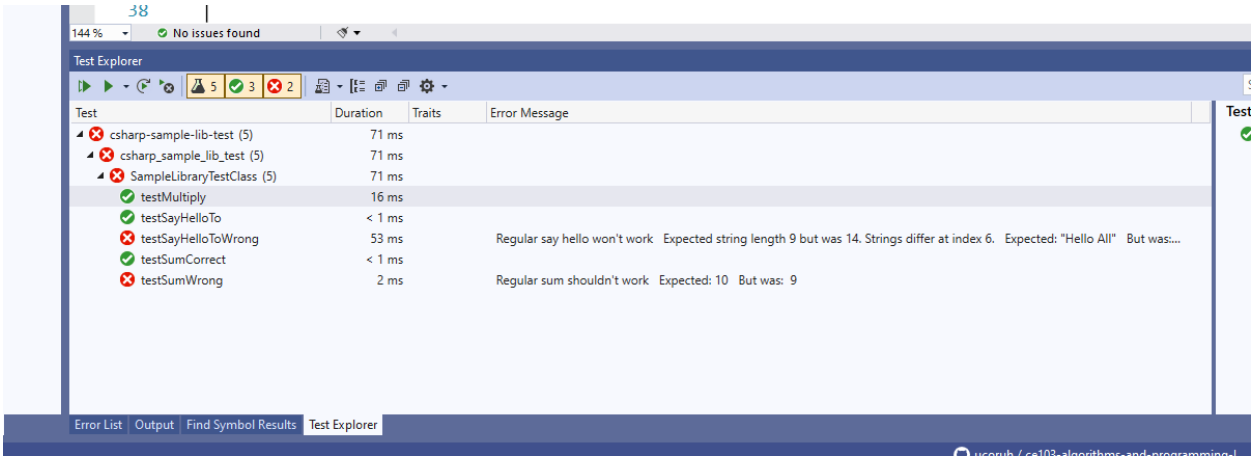
- Sample class library

```
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;
```

```
        }
    }
}
```

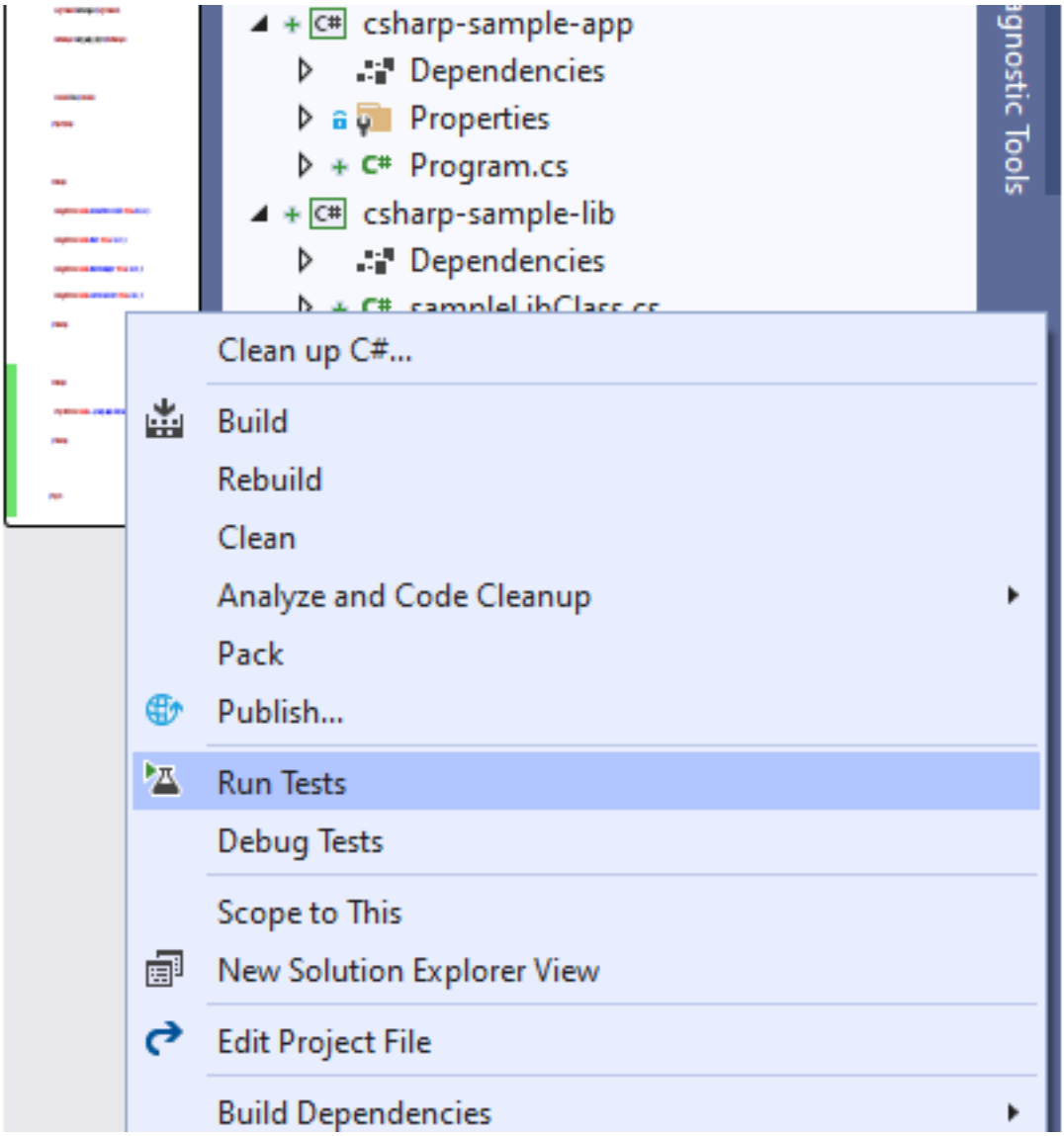
0.187.10.10 Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-9

- Open test explorer and run tests



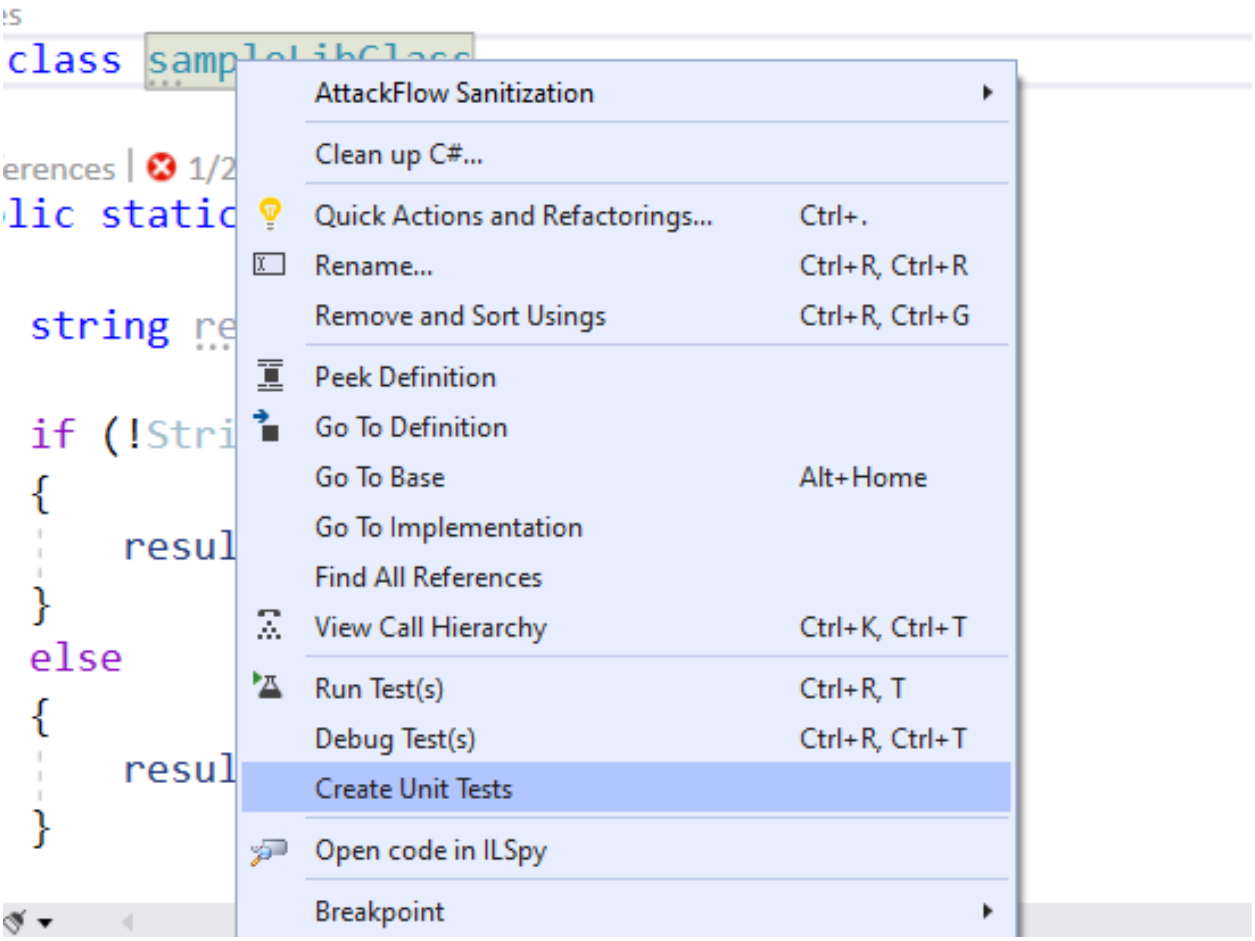
0.187.10.11 Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-10

- or you can run from project

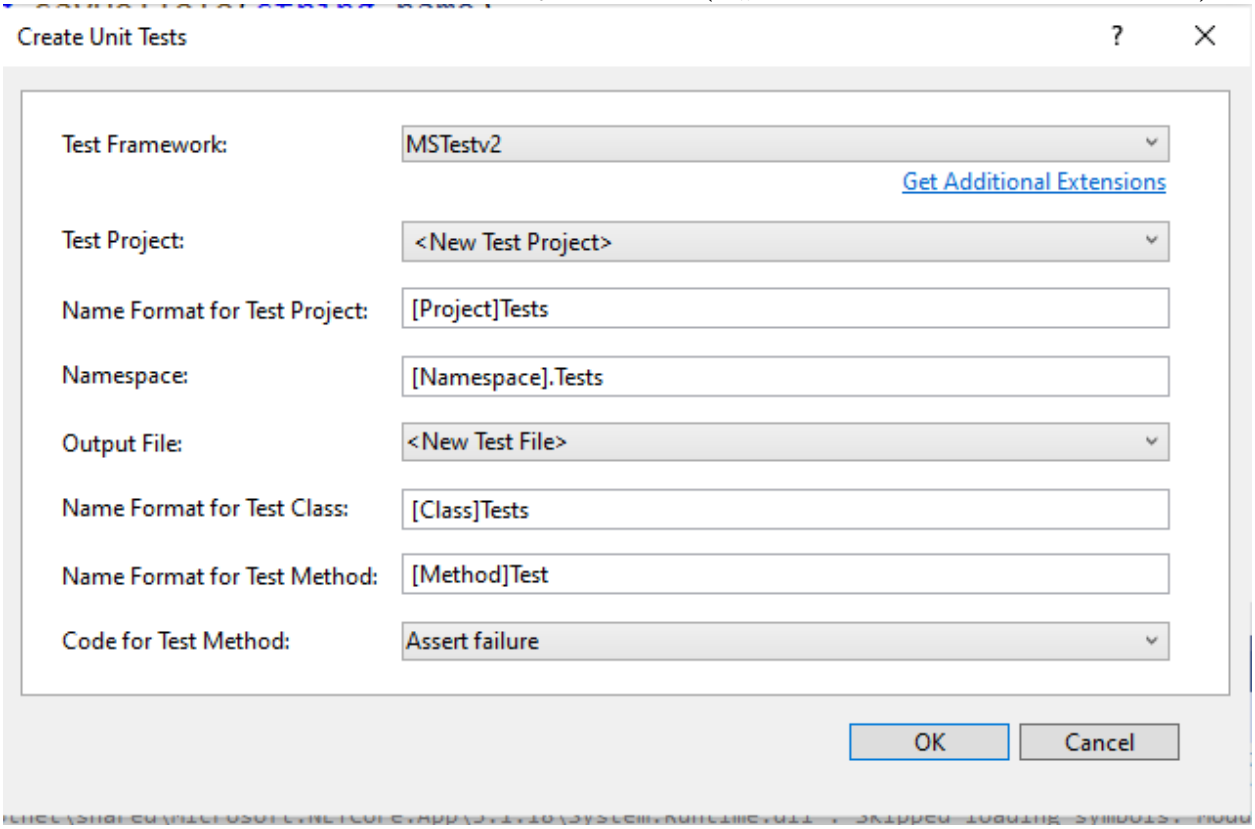


0.187.10.12 Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-11

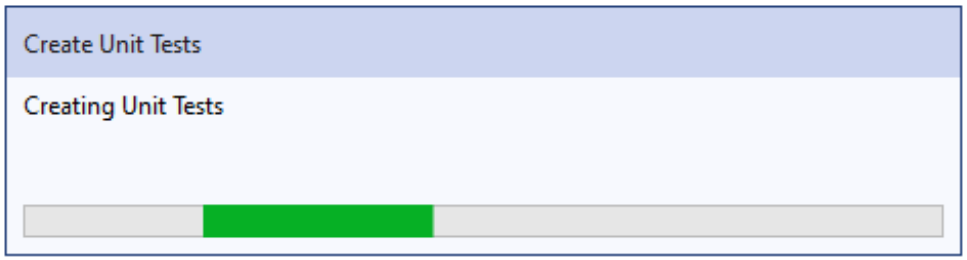
- Also we can create unit test from library class,
- Right click the sampleLibClass and select create unit tests but this option do not provide nunit tests.


class sampleLibClass
erences | 1/2
lic static
string re
if (!Stri
{
resul
}
else
{
resul
}
AttackFlow Sanitization
Clean up C#...
Quick Actions and Refactorings... Ctrl+.
Rename... Ctrl+R, Ctrl+R
Remove and Sort Usings Ctrl+R, Ctrl+G
Peek Definition
Go To Definition
Go To Base Alt+Home
Go To Implementation
Find All References
View Call Hierarchy Ctrl+K, Ctrl+T
Run Test(s) Ctrl+R, T
Debug Test(s) Ctrl+R, Ctrl+T
Create Unit Tests
Open code in ILSpy
Breakpoint

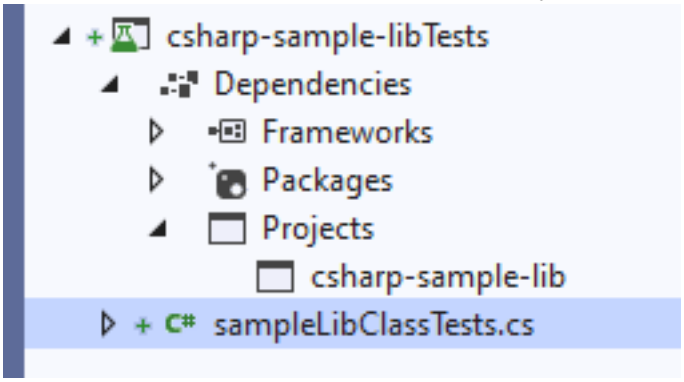
0.187.10.13 Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-12



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



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

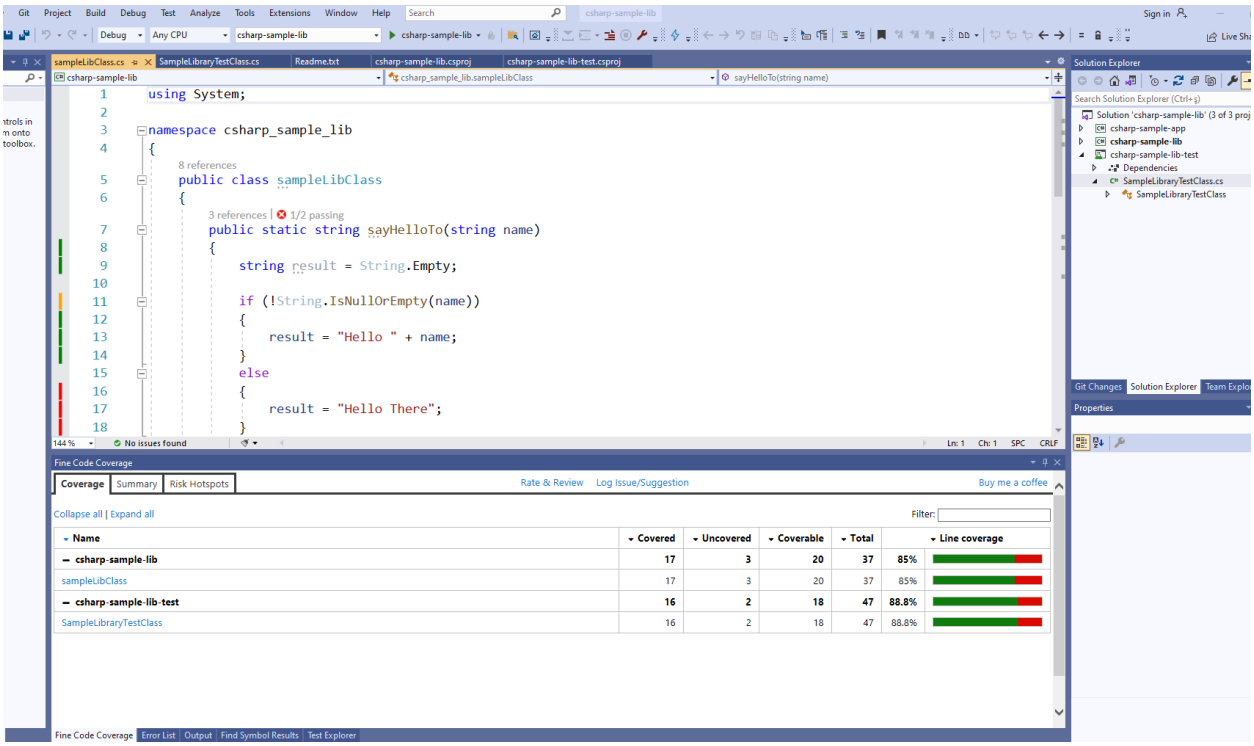


0.187.10.16 Visual Studio Community Edition (C# Unit Test+NUnit+.NETCore)-15

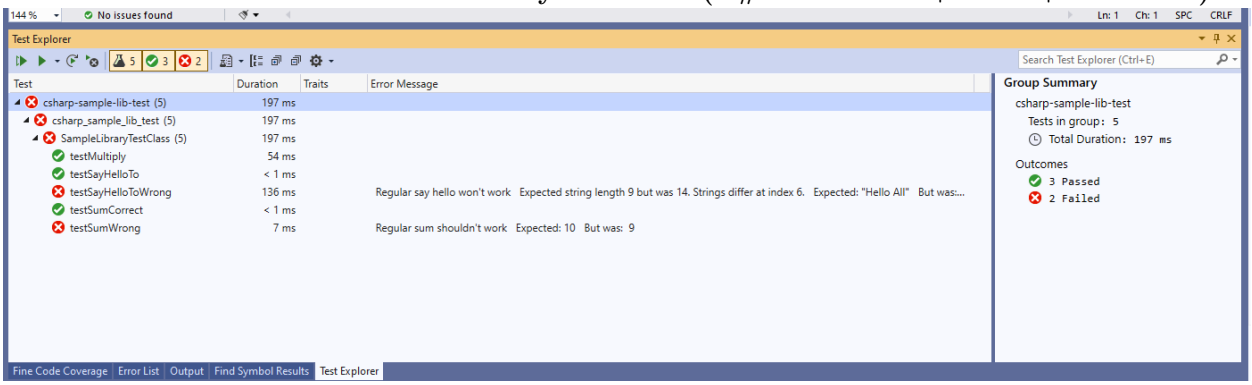
```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using csharp_sample_lib;
using System;
using System.Collections.Generic;
using System.Text;

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();
        }
    }
}
```

0.187.10.17 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



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



- Inline code highlight is part of enterprise visual studio edition
  - Analyzing code coverage in Visual Studio - DEV Community[13]

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

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

0.187.10.20.1 TL;DR

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

- 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[14]

- Follow the instructions on the link
  - CMD OpenCover · sukhoi1/Useful-Notes Wiki · GitHub[15]
- Install OpenCover, ReportGenerator, Nunit,Runners packages then use the package installation folder to get tools that you need

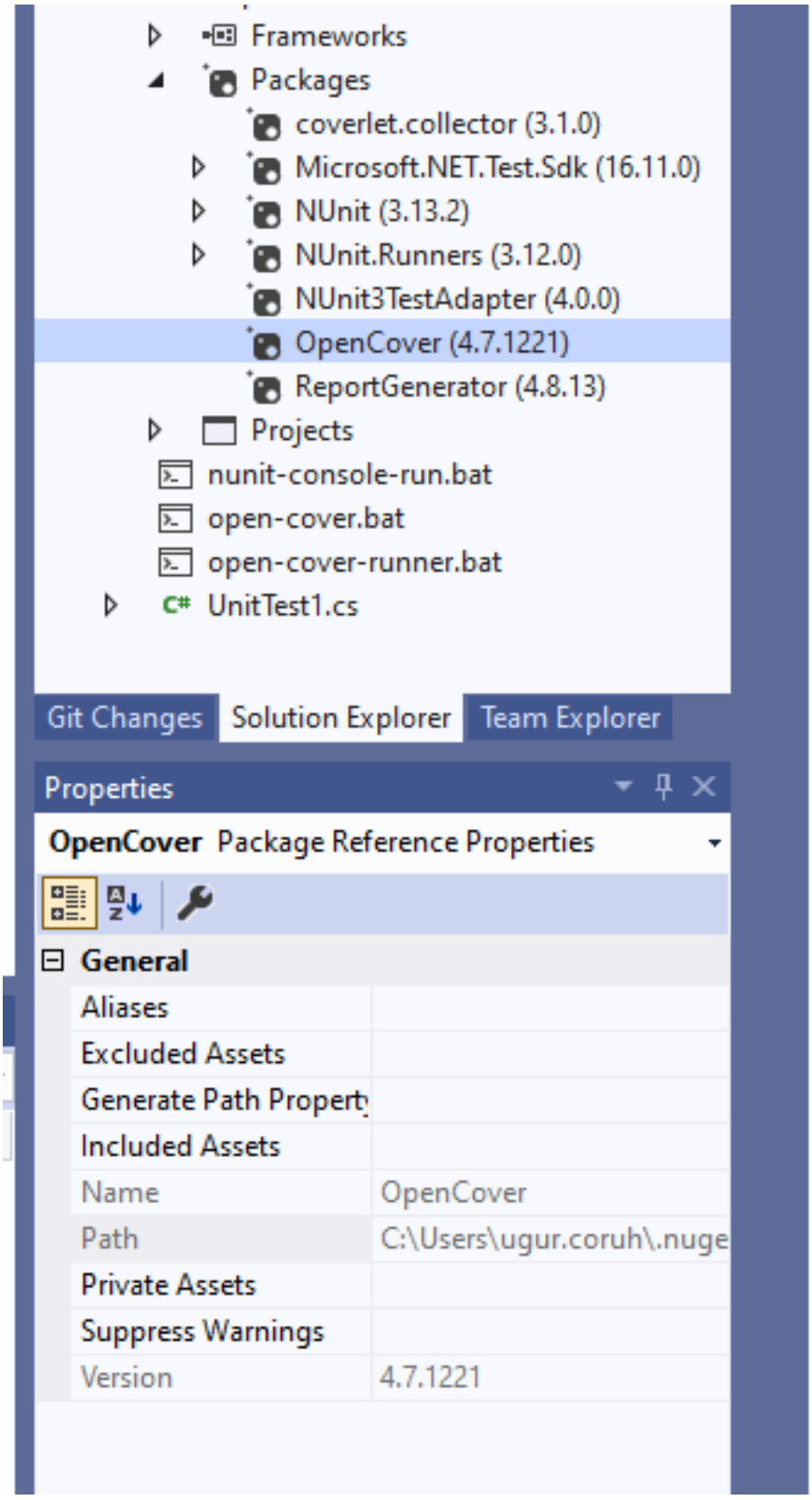
---

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

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

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

[15] https://github.com/sukhoi1/Useful-Notes/wiki/CMD-OpenCover



0.187.10.22 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`

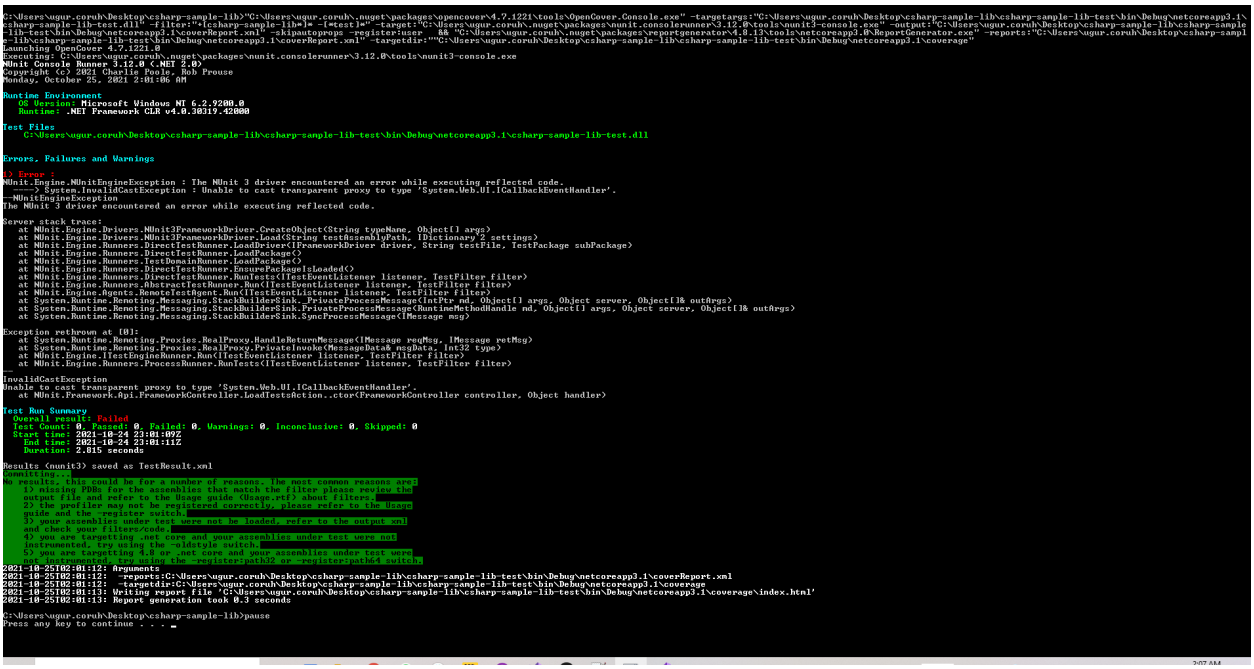
- You need to setup some batch similar with following

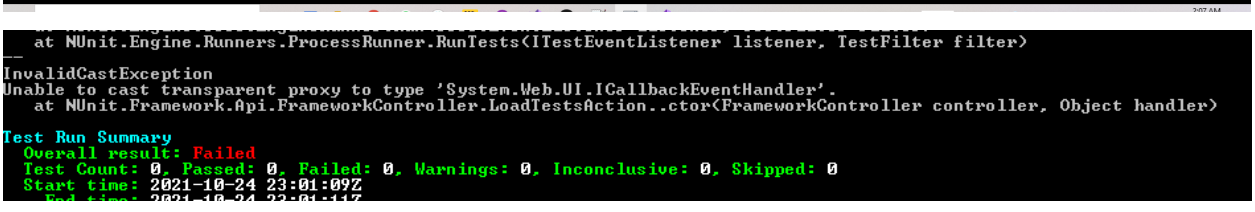
run-test-coverage.bat

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

"%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" -ta
pause
```

0.187.10.23 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-4

- but `nunit3-console.exe` gives error





0.187.10.24 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[16]
  - Code coverage of .NET Core unit tests with OpenCover – Automation Rhapsody[17]
- Sample OpenCover report
  - Summary - Coverage Report[18]

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

0.187.10.26 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,

---

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

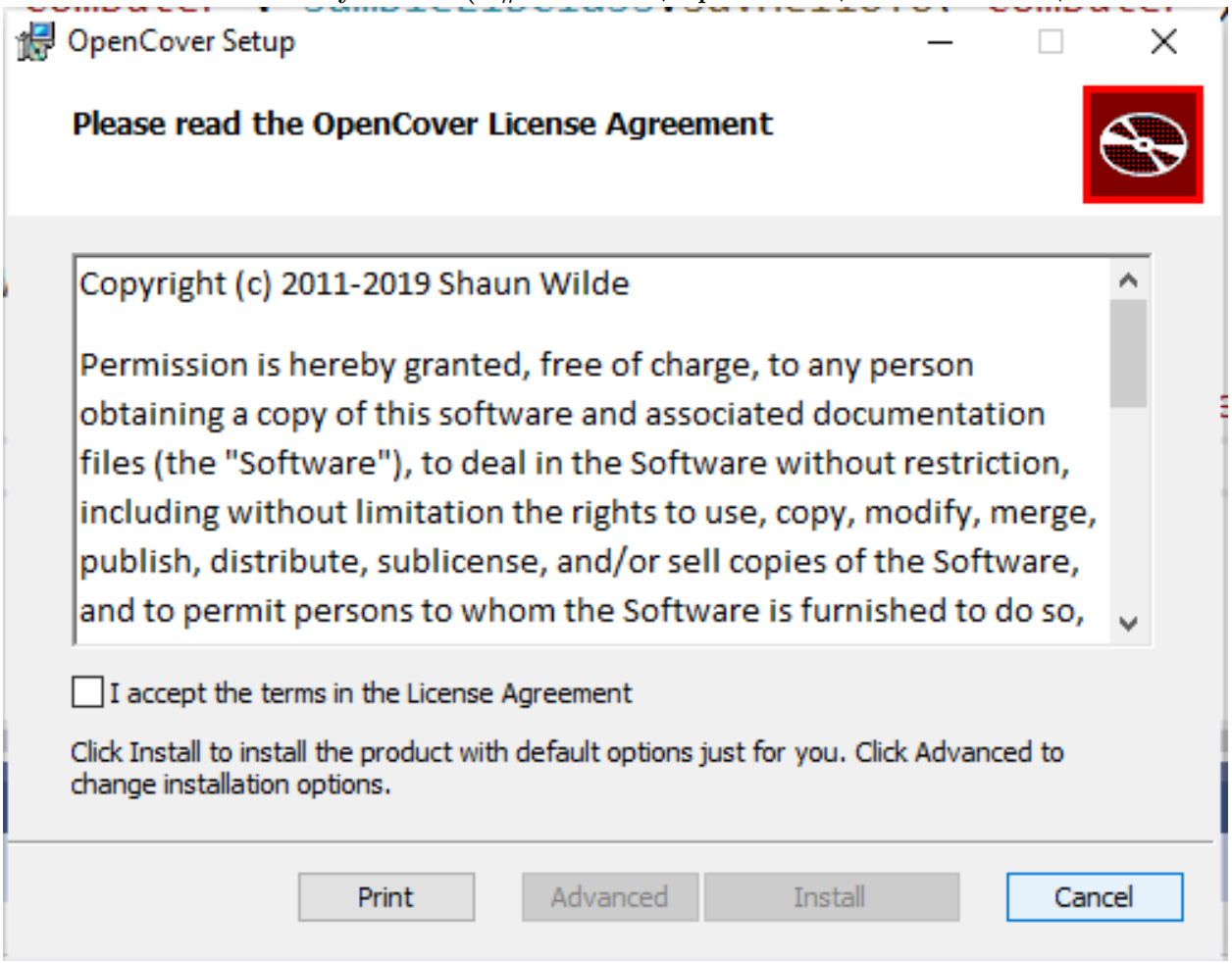
0.187.10.28 OpenCover

- Releases · OpenCover/opencover · GitHub[19]

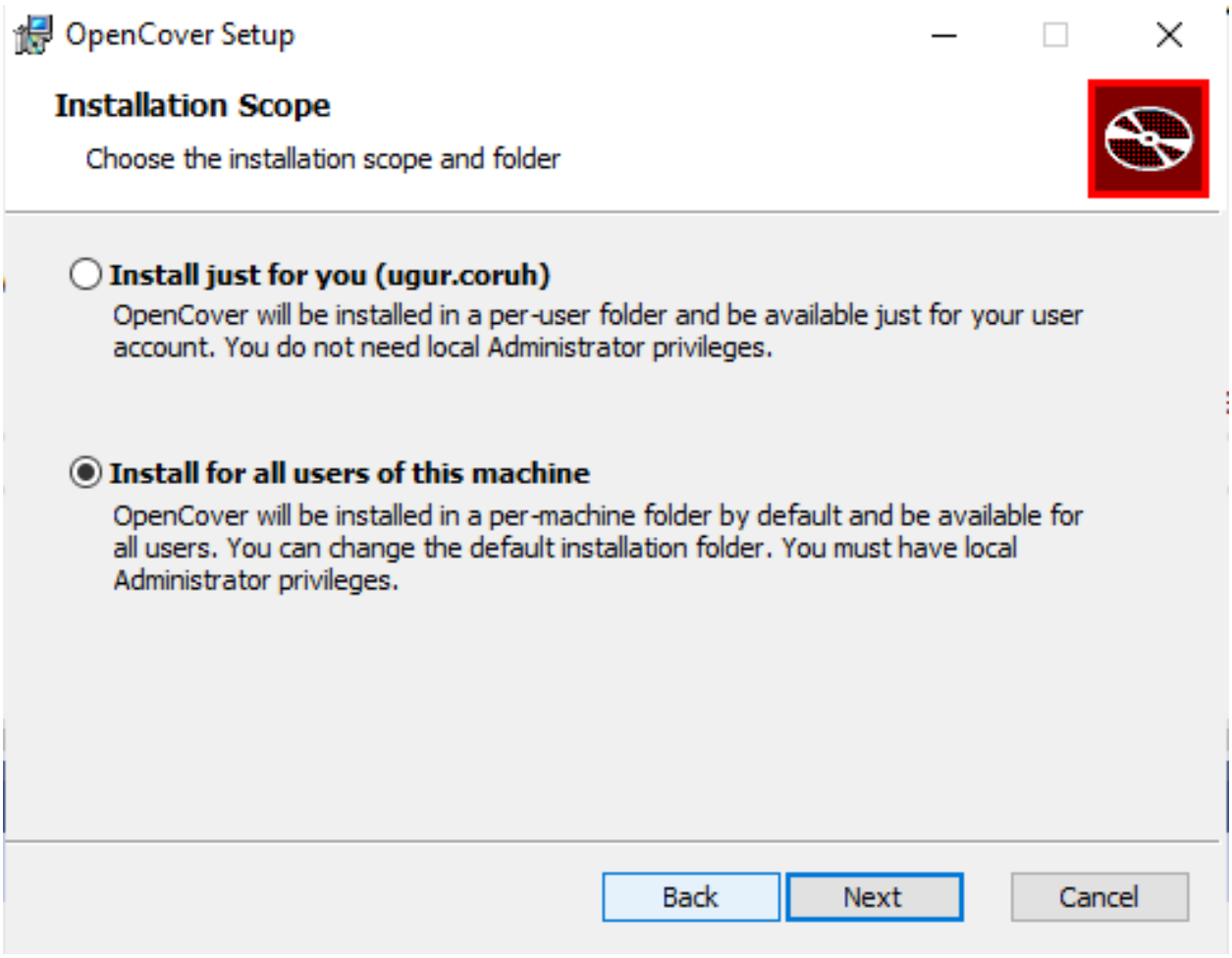
---

[16] https://automationrhapsody.com/code-coverage-manual-automated-tests-opencover-net-applications/
[17] https://automationrhapsody.com/code-coverage-net-core-unit-tests-opencover/
[18] https://automationrhapsody.com/examples/OpenCover-report/
[19] https://github.com/OpenCover/opencover/releases

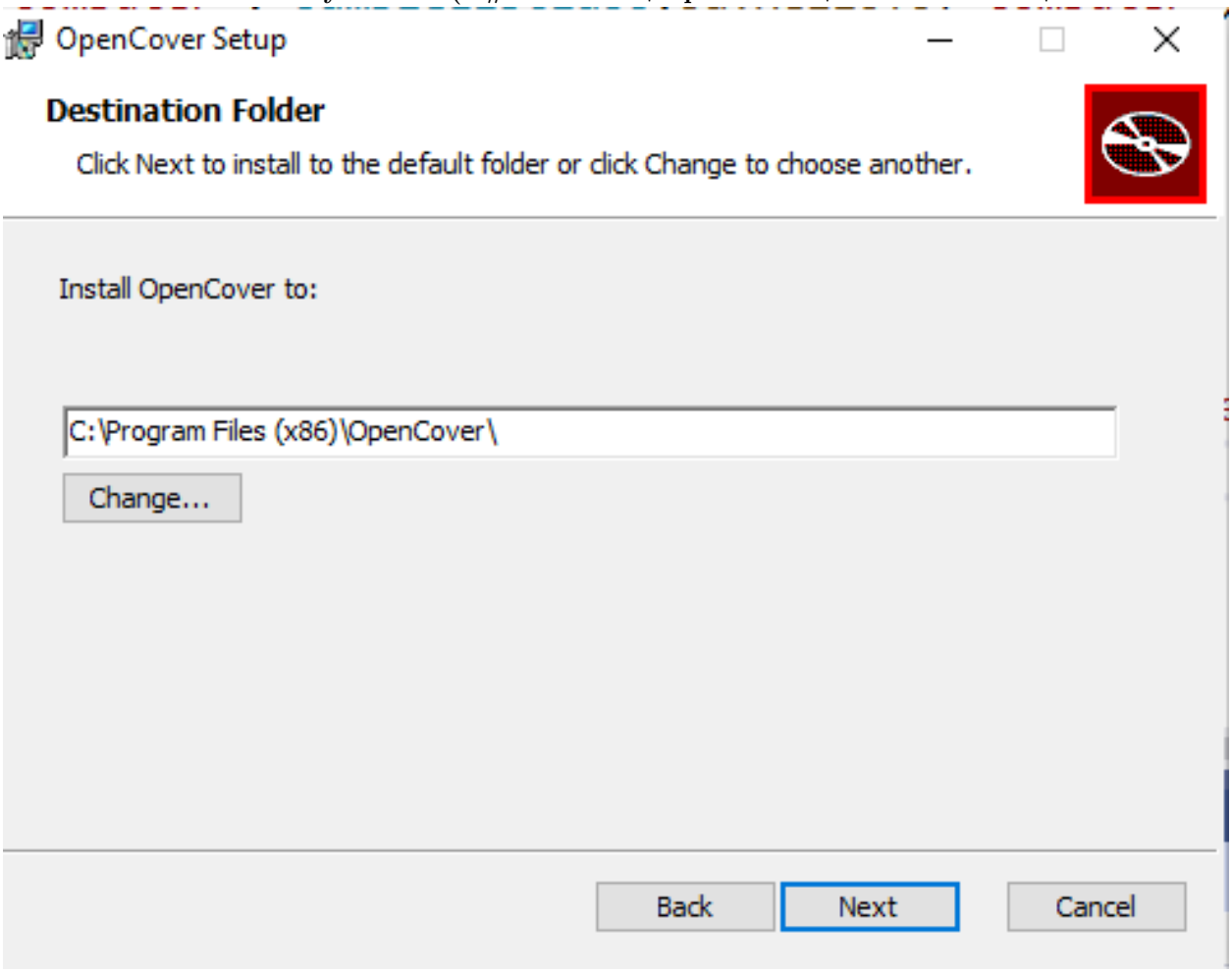
0.187.10.29 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-8



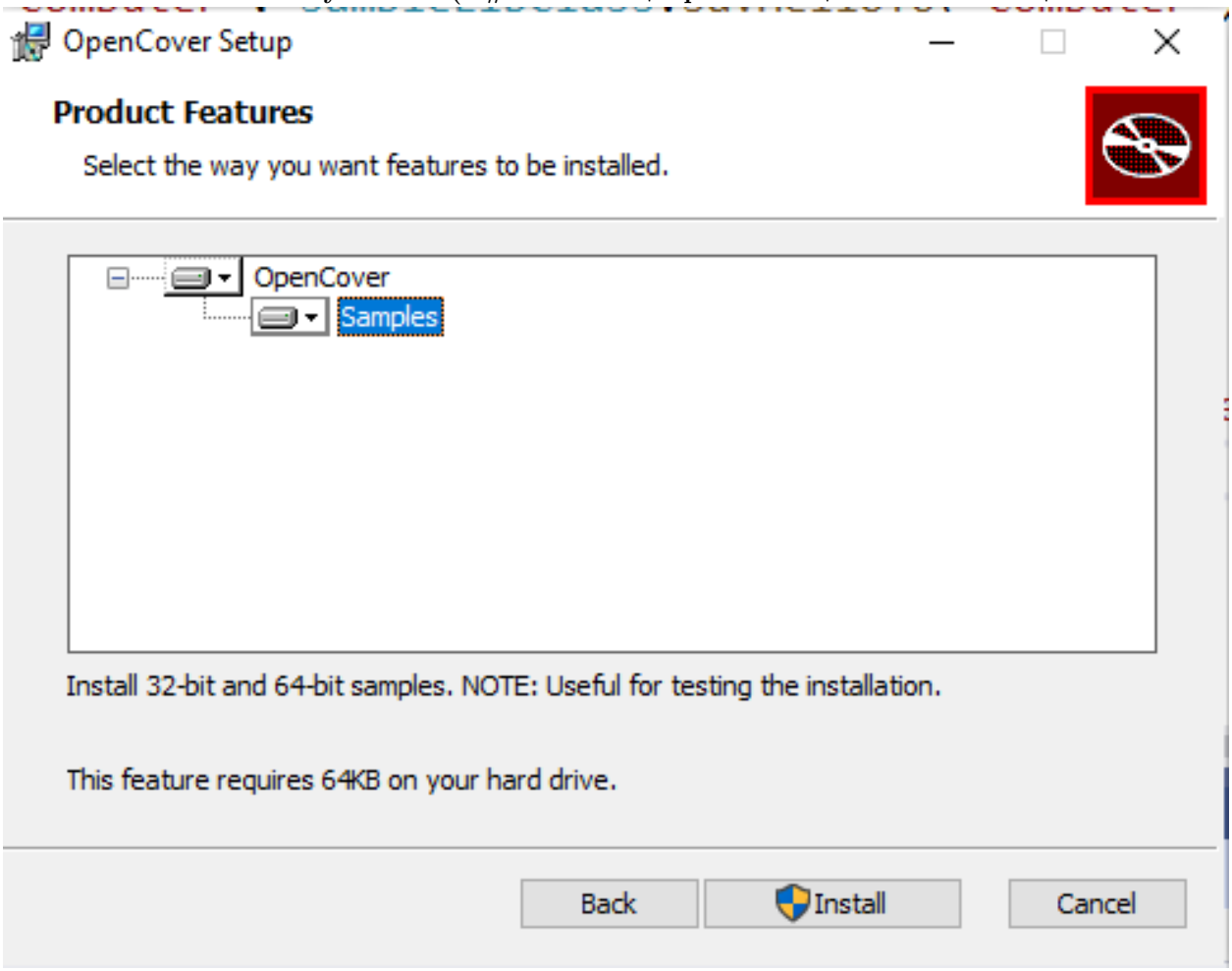
0.187.10.30 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-9 Select advanced and then install for all users


OpenCover Setup
Installation Scope
Choose the installation scope and folder
Install just for you (ugur.coruh)
OpenCover will be installed in a per-user folder and be available just for your user account. You do not need local Administrator privileges.
Install for all users of this machine
OpenCover will be installed in a per-machine folder by default and be available for all users. You can change the default installation folder. You must have local Administrator privileges.
Back
Next
Cancel

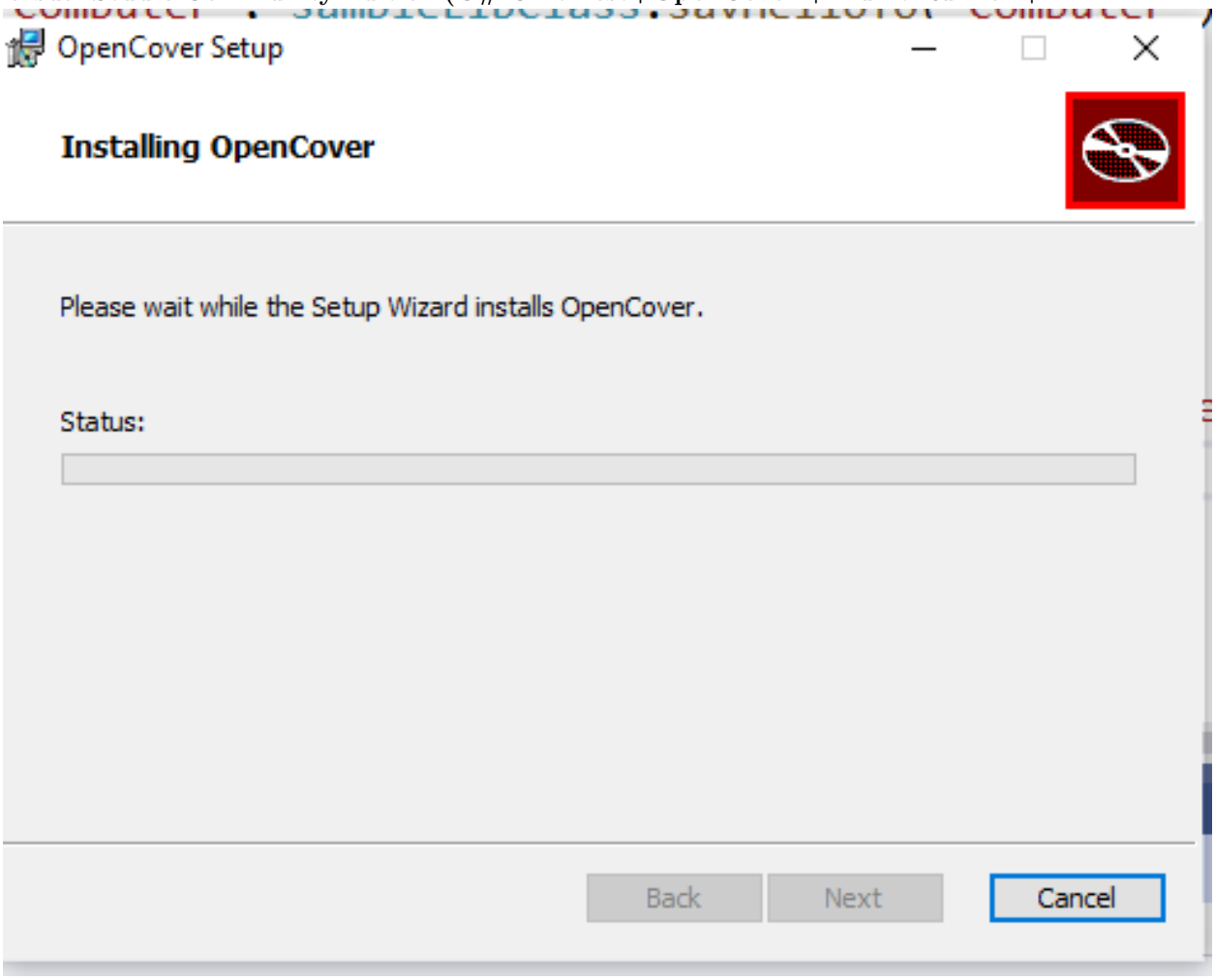
0.187.10.31 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-10



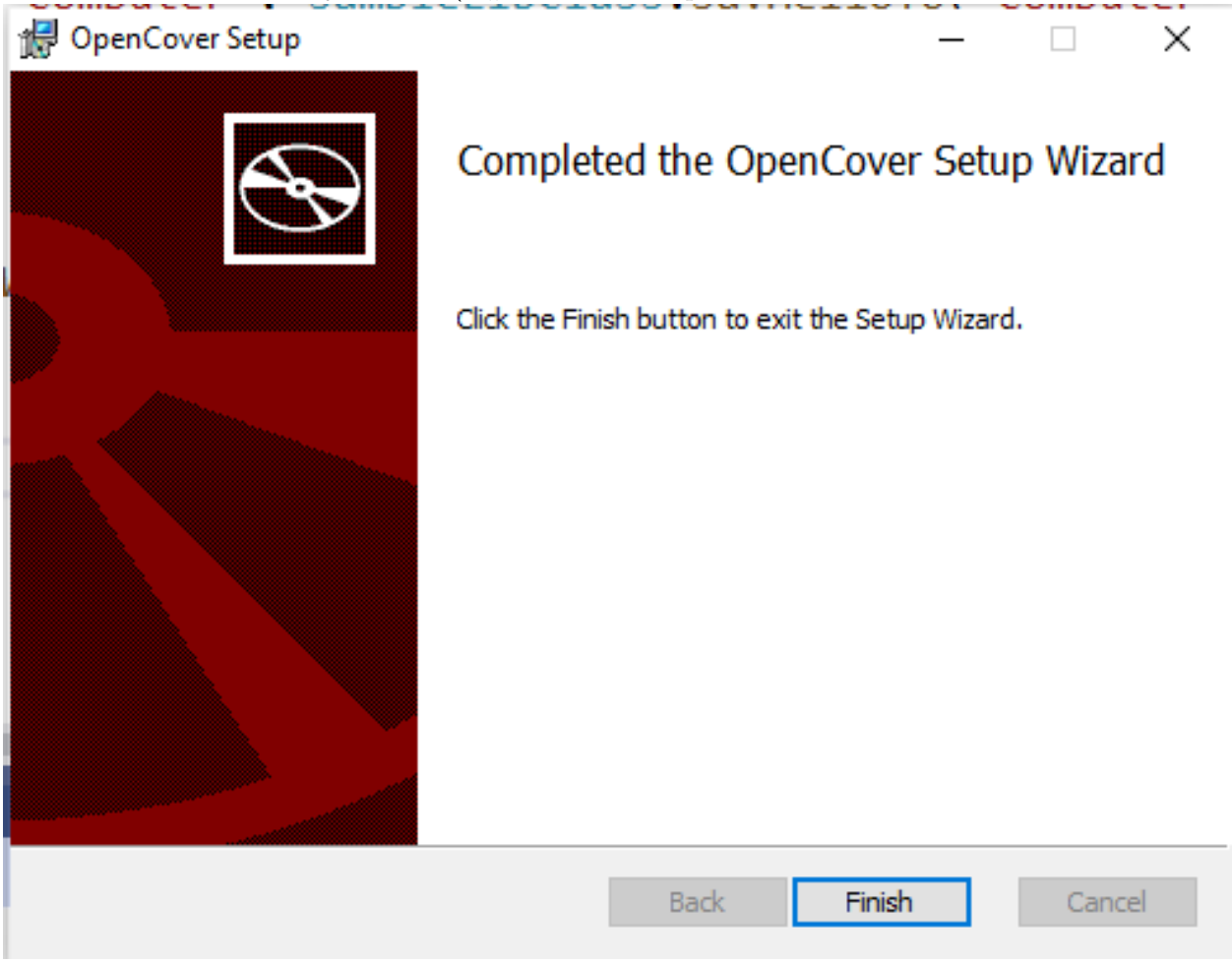
0.187.10.32 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-11



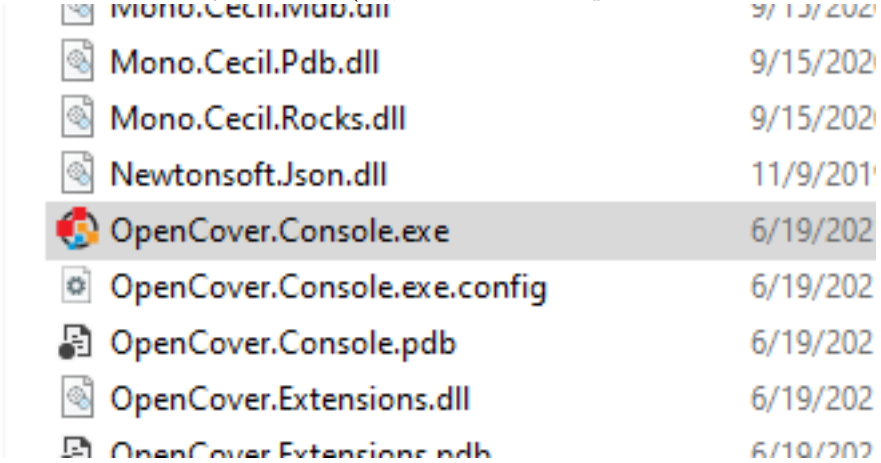
0.187.10.33 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-12



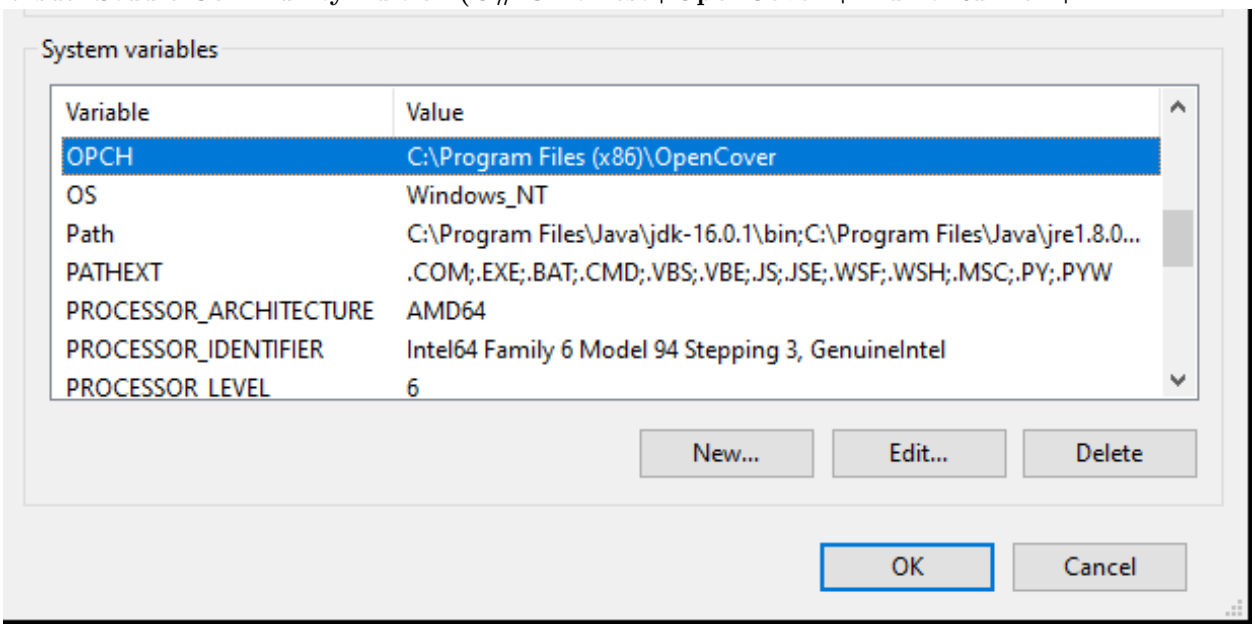
0.187.10.34 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-13

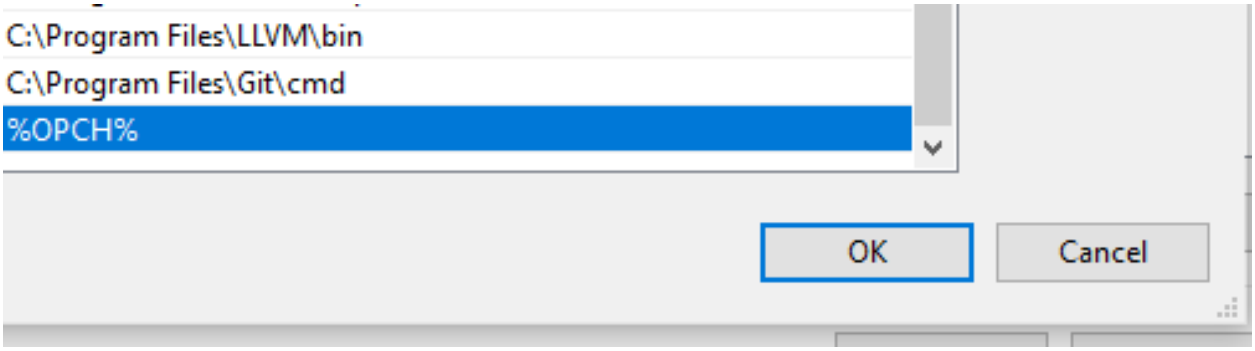


0.187.10.35 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-14

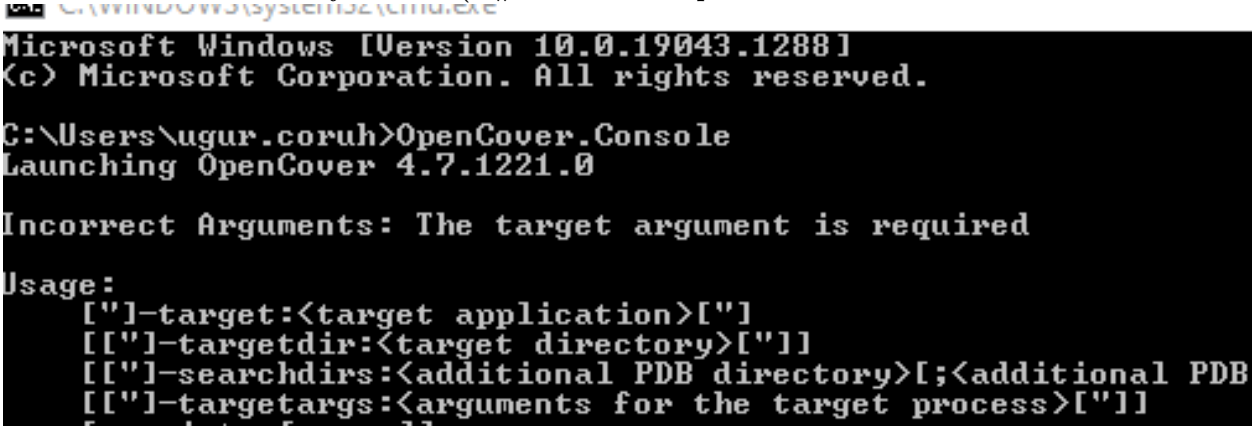


0.187.10.36 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-15





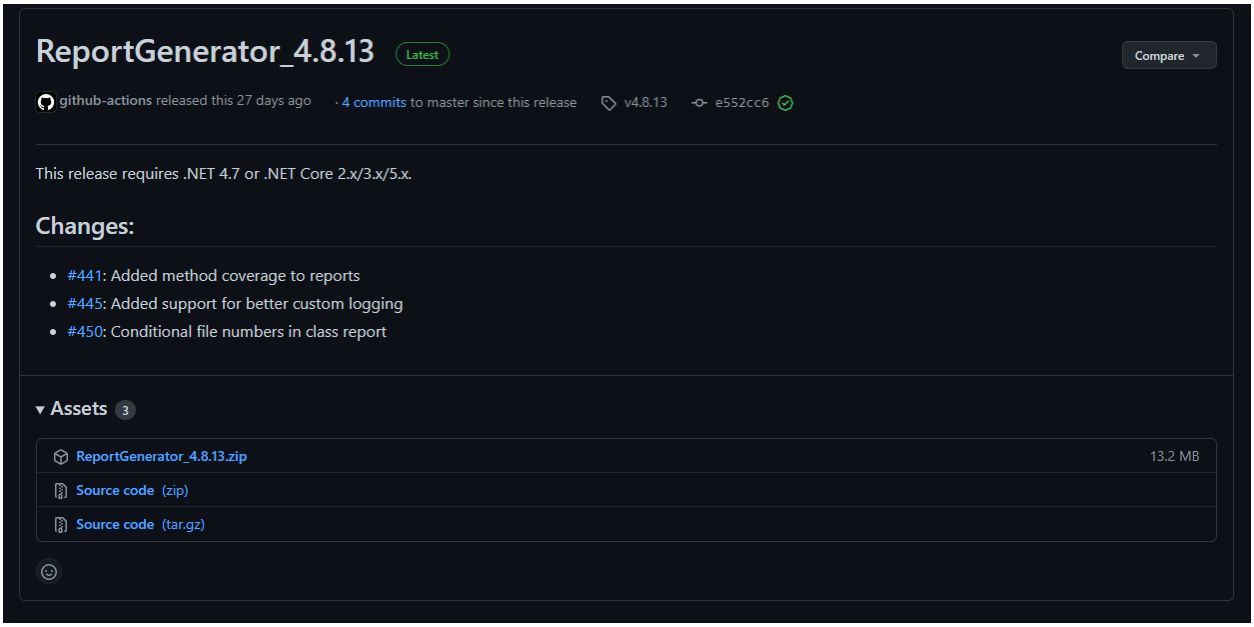
0.187.10.37 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-16



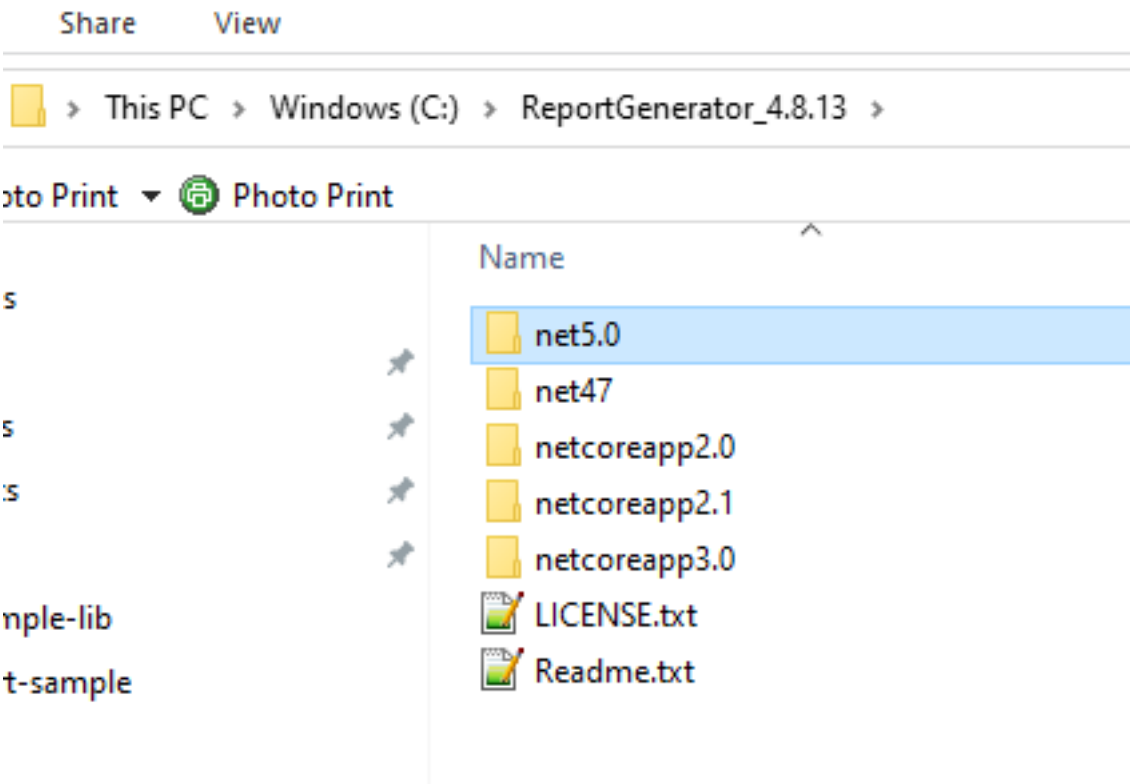
0.187.10.38 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-17

0.187.10.38.1 ReportGenerator

- Release ReportGenerator_4.8.13 · danielpalme/ReportGenerator · GitHub[20]



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

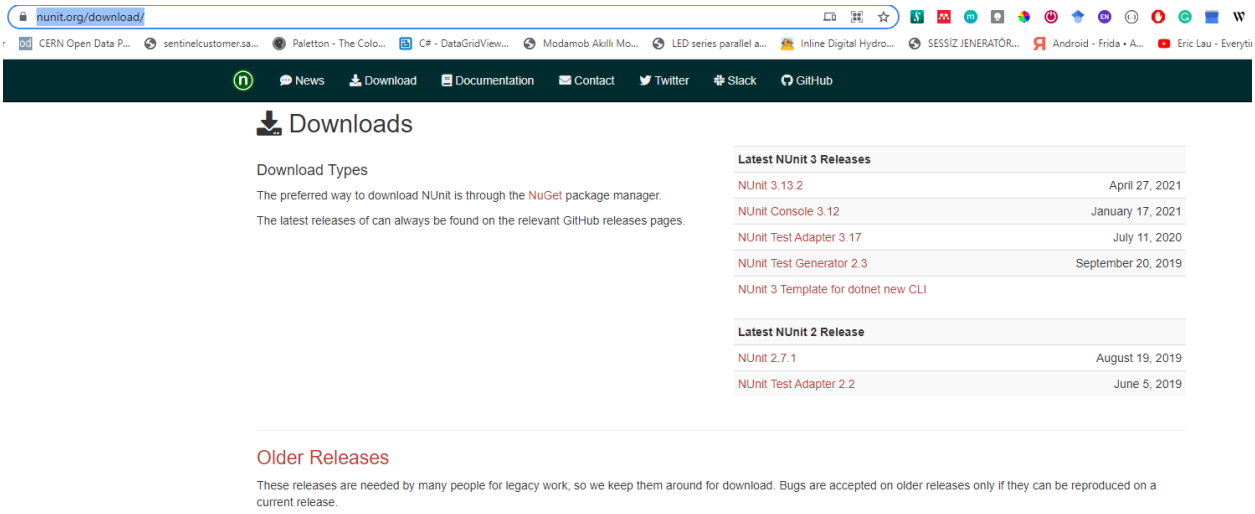


0.187.10.40 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-19

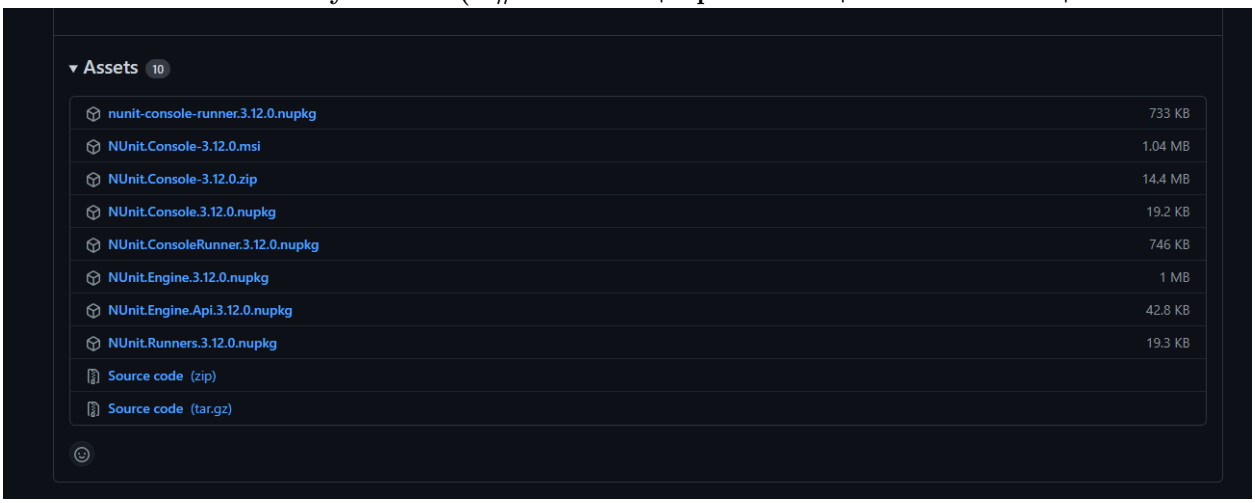
[20] https://github.com/danielpalme/ReportGenerator/releases/tag/v4.8.13

0.187.10.40.1 NUnit Console

- Downloads[21]

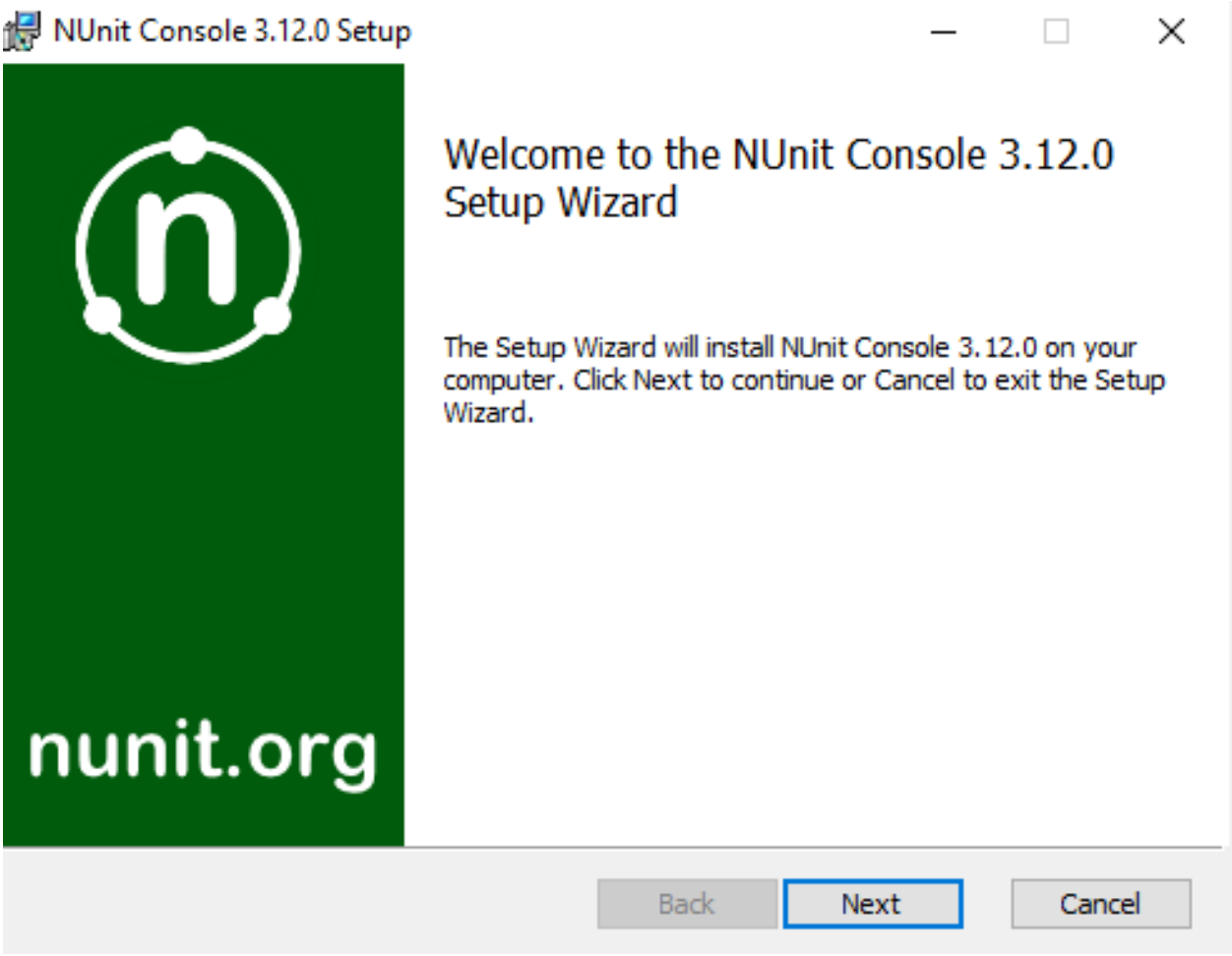


0.187.10.41 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-20



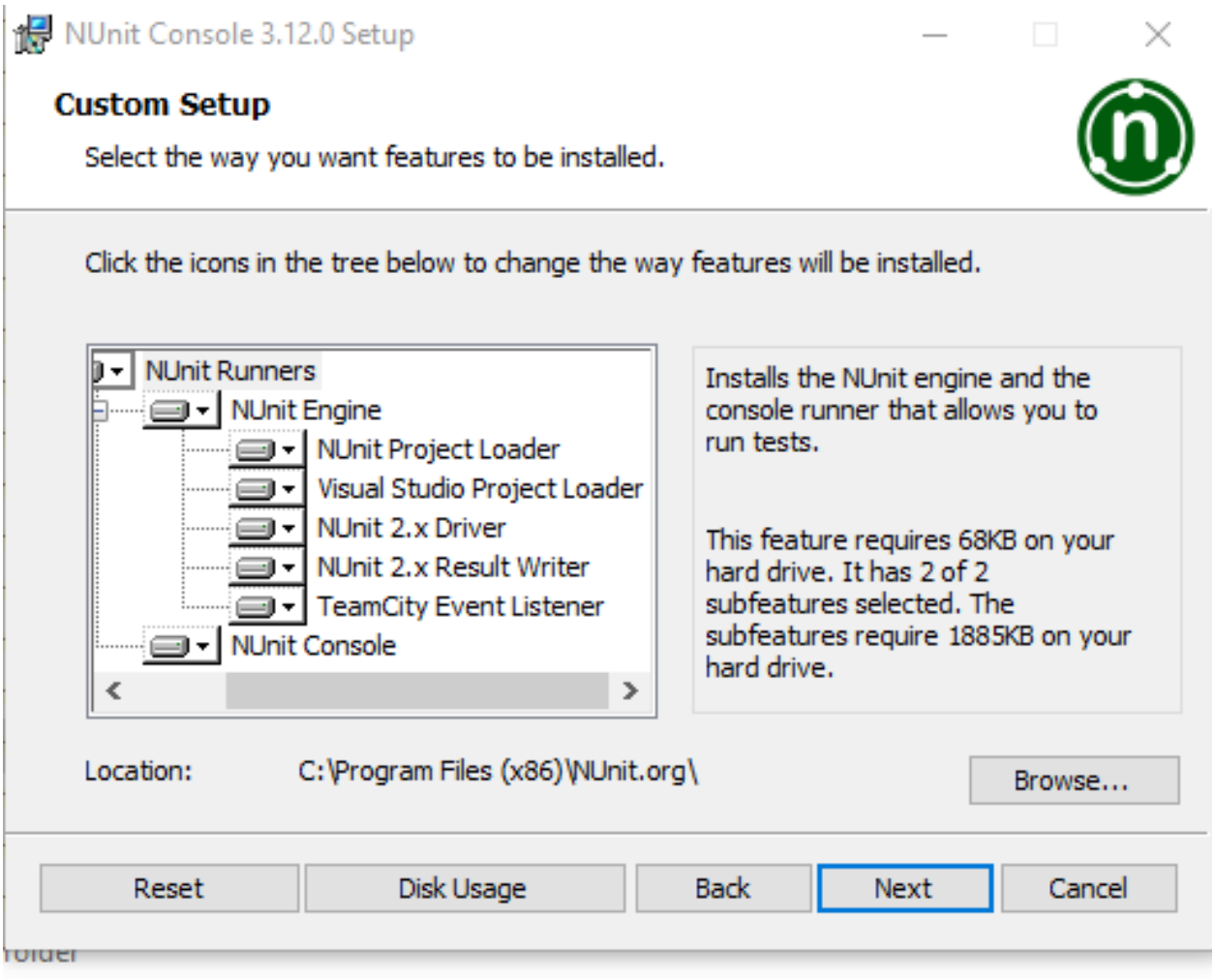
[21] https://nunit.org/download/

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



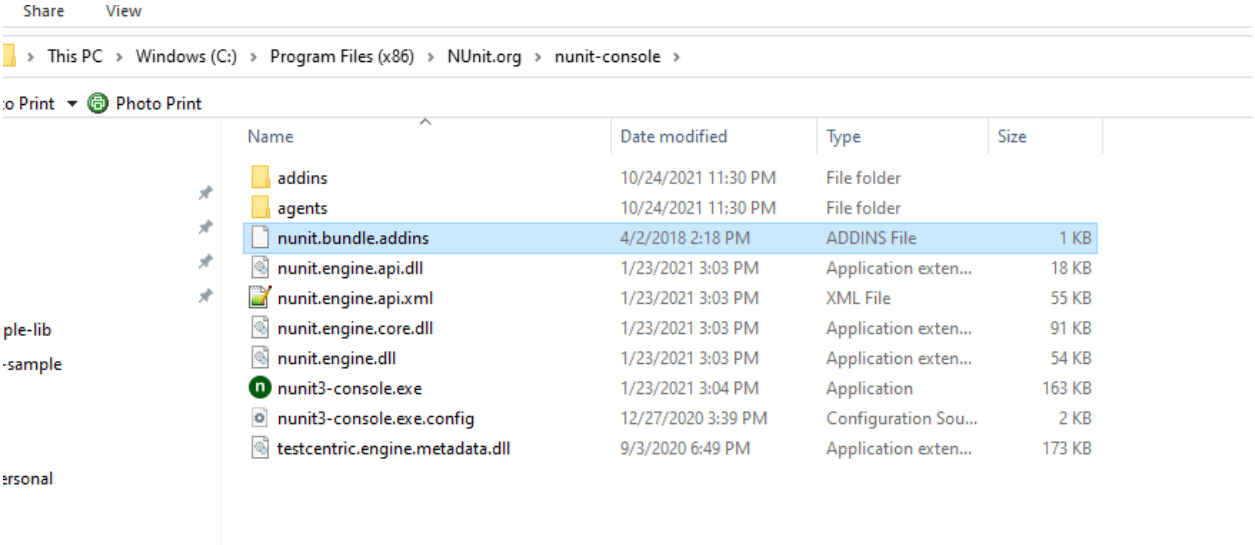
0.187.10.43 Visual Studio Community Edition (C# Unit Test+OpenCover + Nunit Runner + Report)-22

- Download setup



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

- Install setup



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

0.187.11 NUnit + MSTest Batch Report Generation (Not Tested)

- OpenCover and ReportGenerator Unit Test Coverage in Visual Studio 2013 and 2015 – CodeHelper.Net[22]
- OpenCover and ReportGenerator Unit Test Coverage in Visual Studio 2013 and 2015 - CodeProject[23]

0.187.12 Java Unit Tests

0.187.12.1 Eclipse IDE (JUnit4 , JUnit5)

0.187.12.2 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[24]

JUnit 5[25]

JUnit 5 User Guide[26]

https://www.eclemma.org/

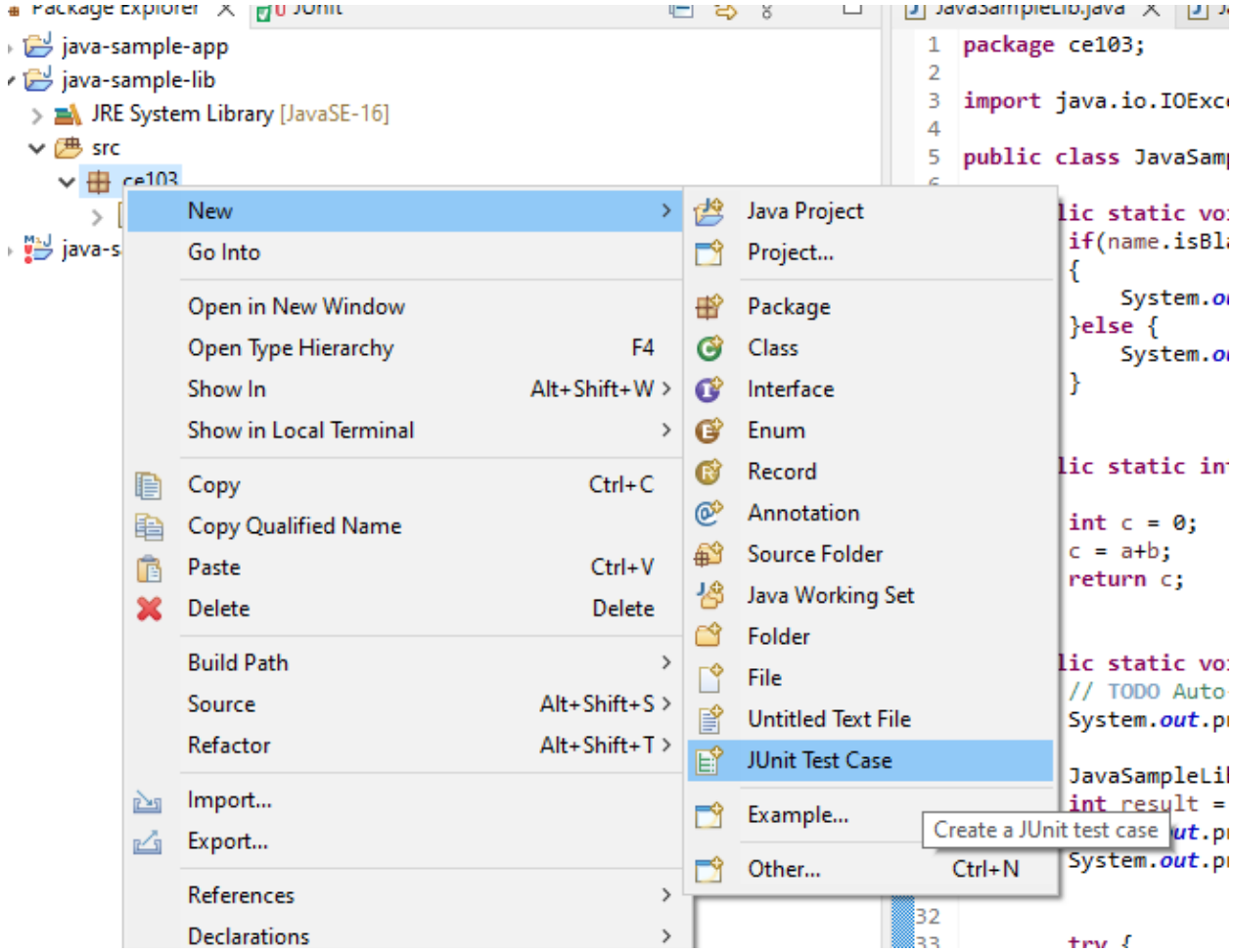
JUnit Hello World Example - Examples Java Code Geeks - 2021[27]

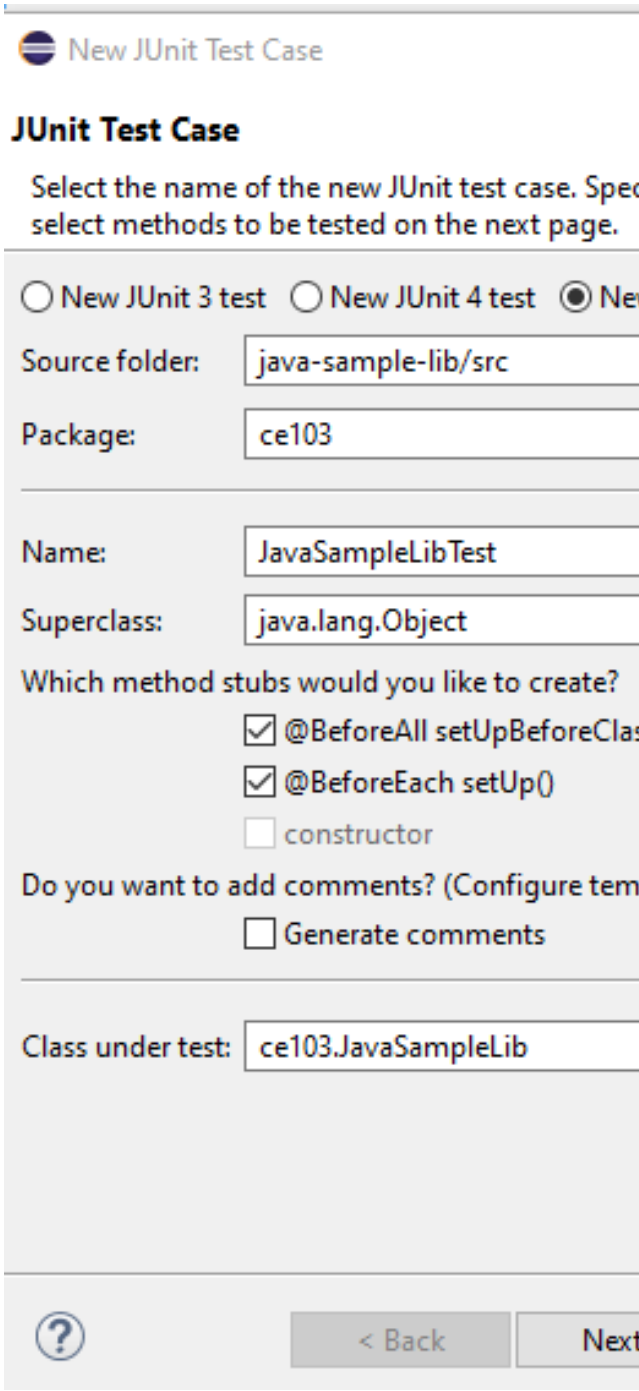
https://yasinmemic.medium.com/java-ile-unit-test-yazmak-birim-test-ca15cf0d024b

---

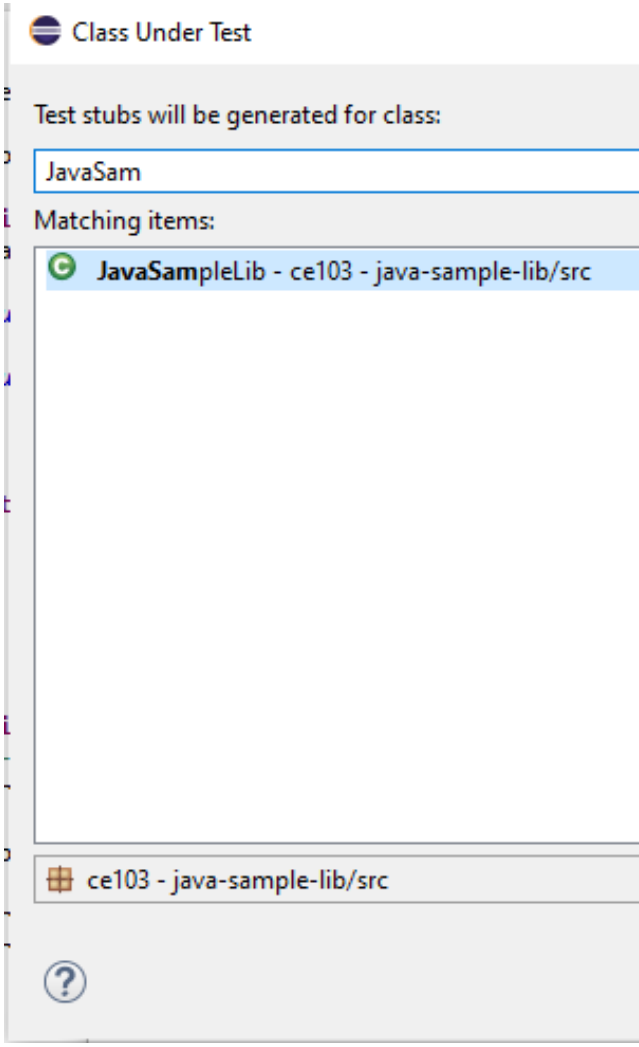
0.187.12.3 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test In normal java application we can right click the project java-sample-lib and add Junit case

[22] http://codehelper.net/unit-testing/opencover-and-reportgenerator-unit-test-coverage-in-visual-studio-2013-and-2015/
[23] https://www.codeproject.com/Articles/1276980/OpenCover-and-ReportGenerator-Unit-Test-Coverage-i
[24] https://www.vogella.com/tutorials/JUnit/article.html
[25] https://junit.org/junit5/
[26] https://junit.org/junit5/docs/current/user-guide/
[27] https://examples.javacodegeeks.com/core-java/junit/junit-hello-world-example/

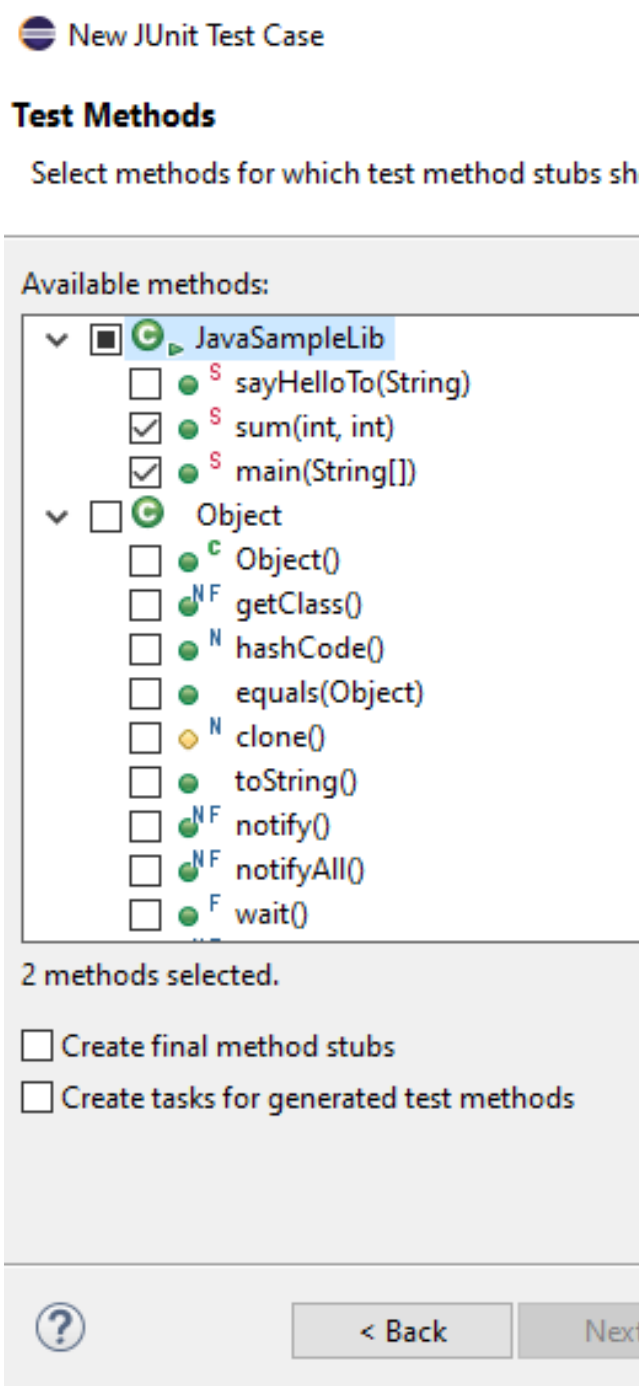

java-sample-app
java-sample-lib
JRE System Library [JavaSE-16]
src
ce103
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
References
Declarations
Java Project
Project...
Package
Class
Interface
Enum
Record
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
JUnit Test Case
Example...
Other... Ctrl+N
Create a JUnit test case
package ce103;
import java.io.IOExc
public class JavaSam
lic static vo
if(name.isBl
{
System.o
}else {
System.o
}
lic static in
int c = 0;
c = a+b;
return c;
lic static vo
// TODO Auto
System.out.p
JavaSampleLi
int result =
System.out.p
32



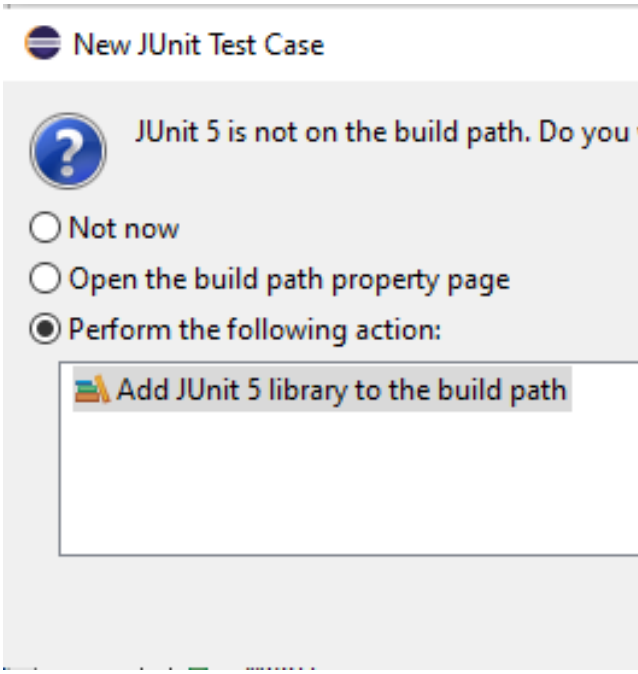
0.187.12.4 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test



0.187.12.5 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

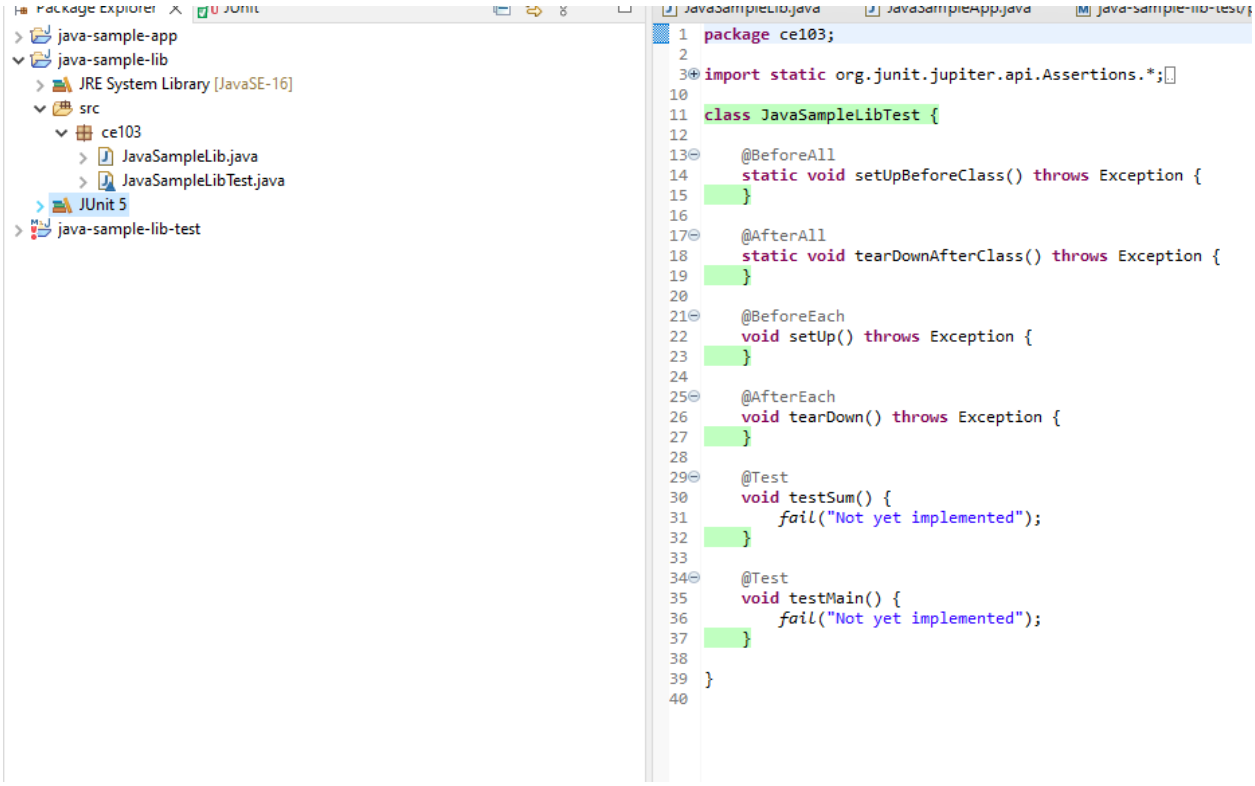


0.187.12.6 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test



0.187.12.7 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

0.187.12.8 Eclipse IDE (JUnit4, JUnit5) + Java Unit Test and you will have the following test class



0.187.12.9 Eclipse IDE (JUnit4, JUnit5) + Java Unit Test Now we will create tests that check our function flowchart and return values

We need to cover all code branches that we coded

I have updated `JavaSampleLib.java` as follows to check outputs

JavaSampleLib.java

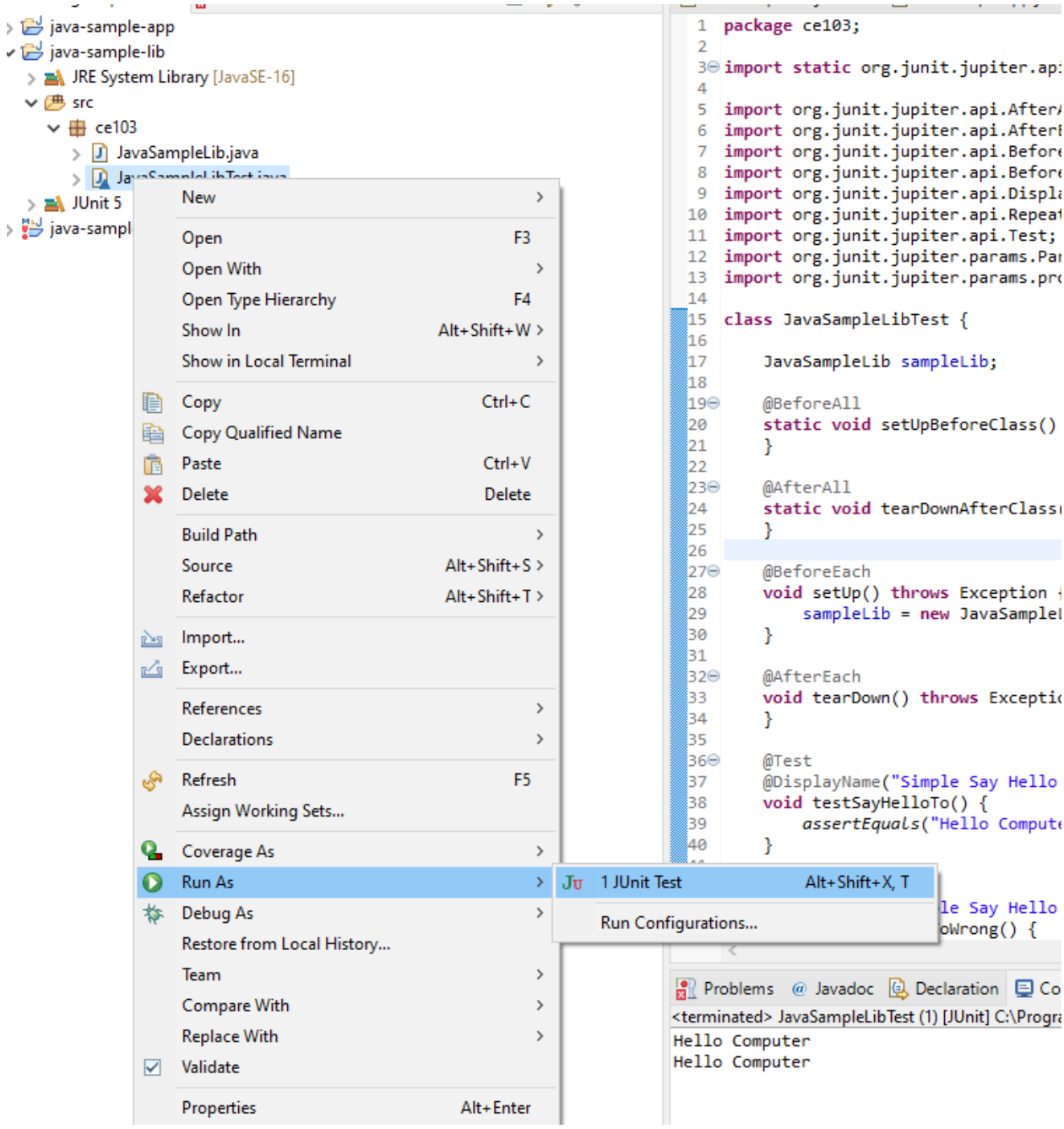
```
package ce103;

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();
//        }
//
//    }

}
```

0.187.12.10 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test and JavaSampleLibTest.java

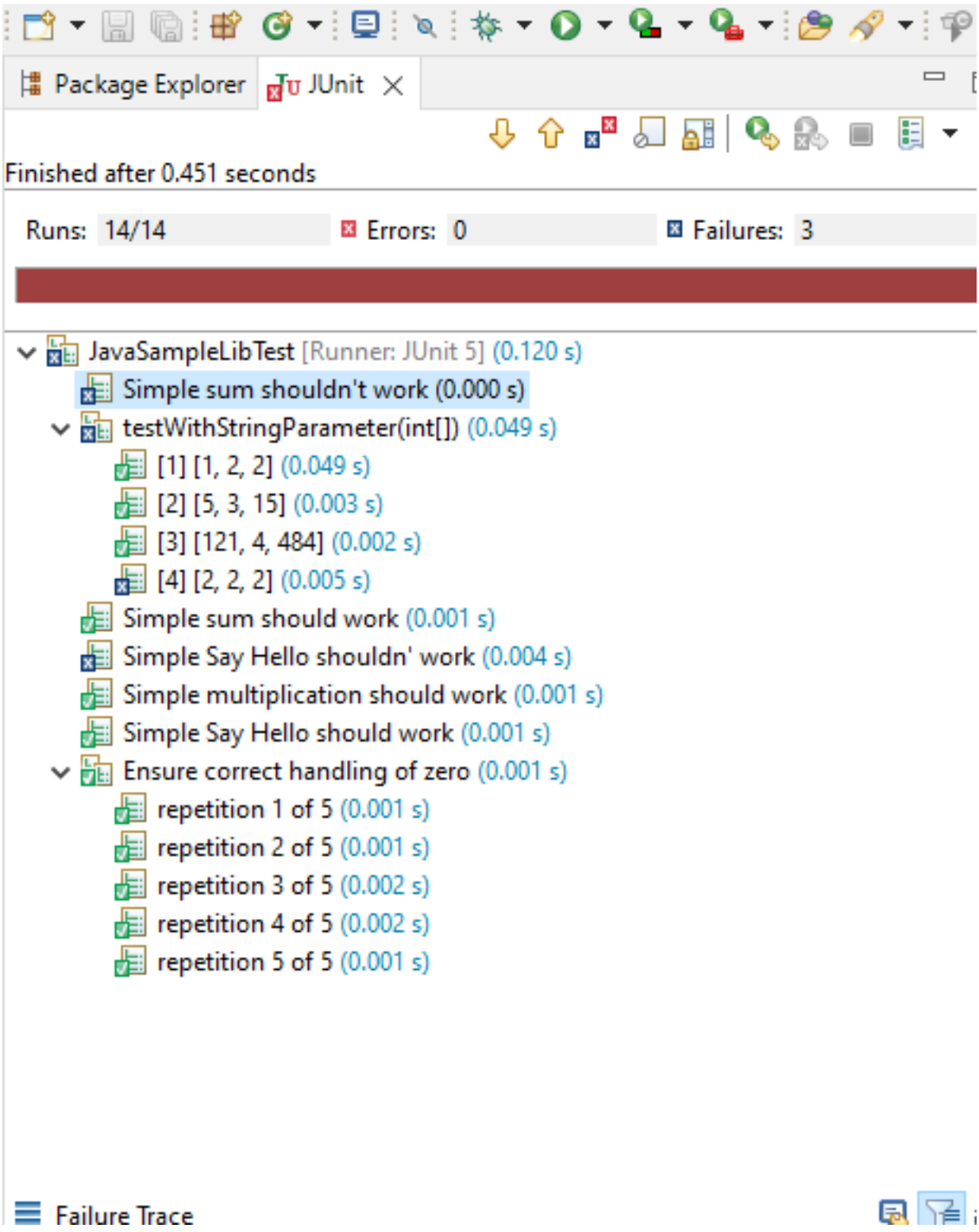
```java
package ce103;

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 w
    }

    @Test
    @DisplayName("Simple Say Hello shouldn' 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));
    }

}
```

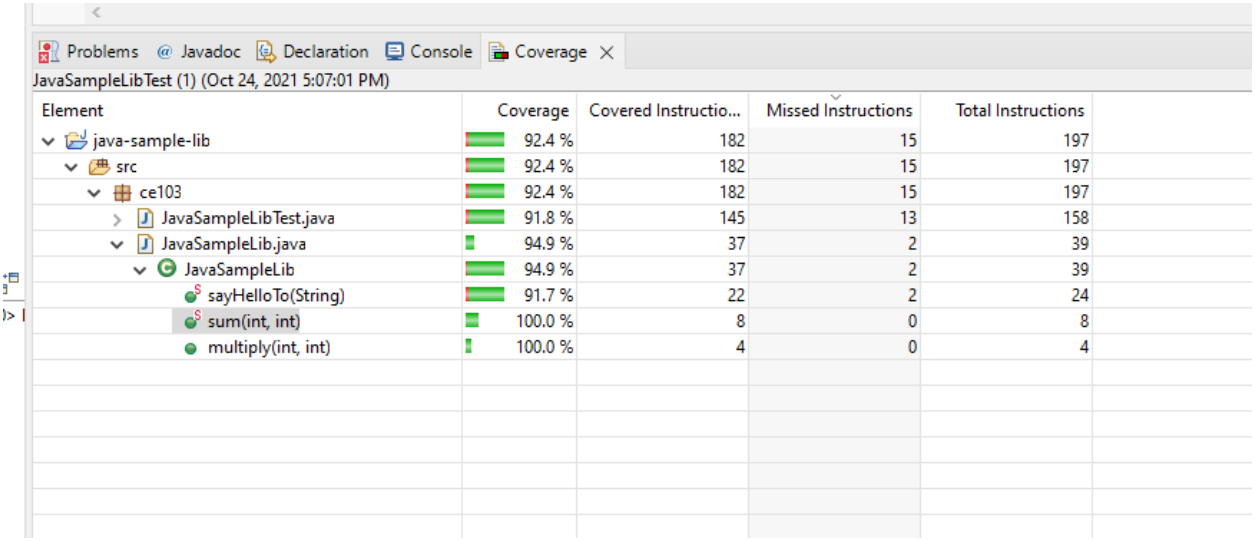
0.187.12.11 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test if we run tests



0.187.12.12 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test we will see all results there



0.187.12.13 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test also we can see the code coverage of tests



0.187.12.14 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

```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;
    }

```

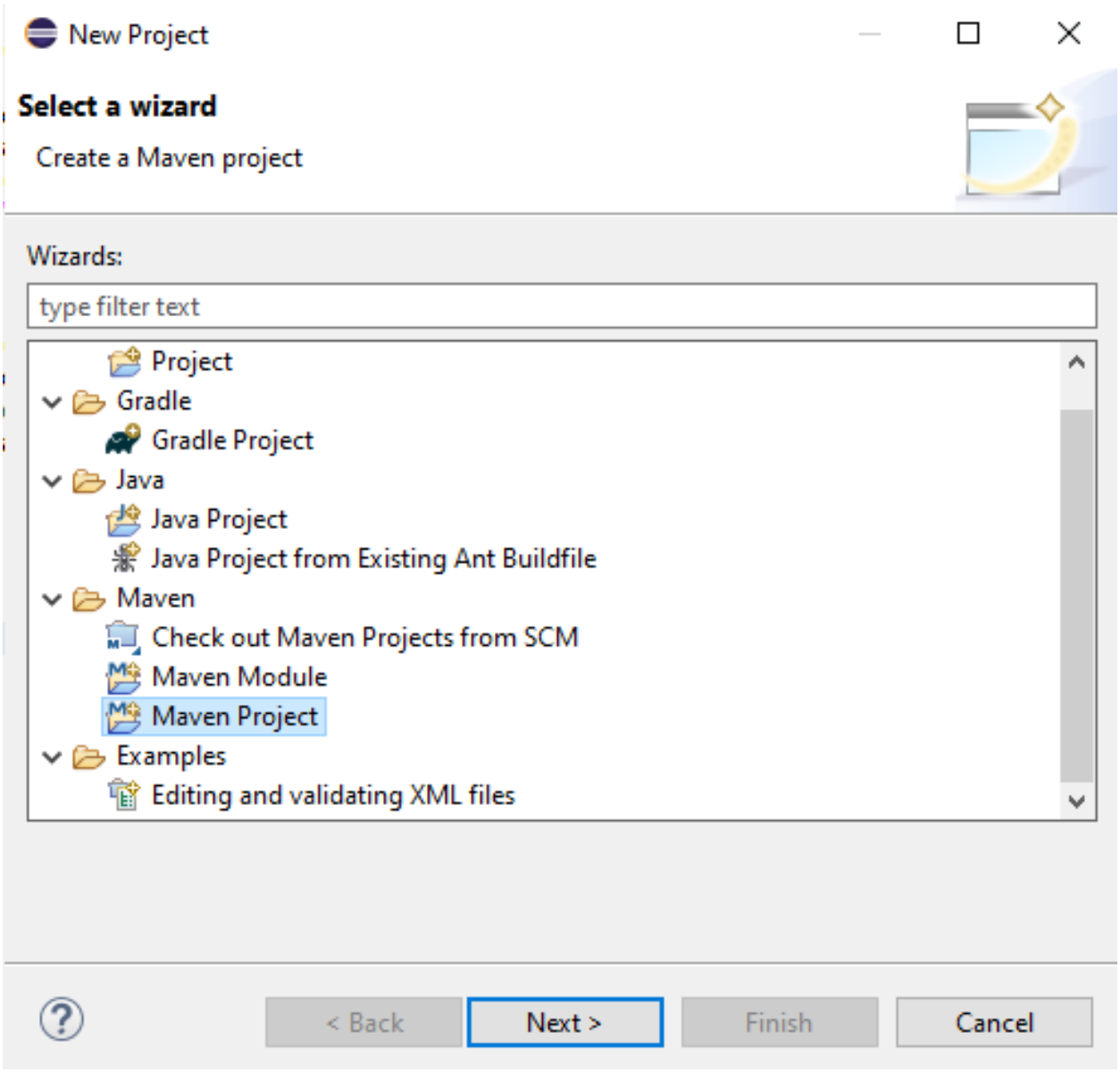
0.187.12.15 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

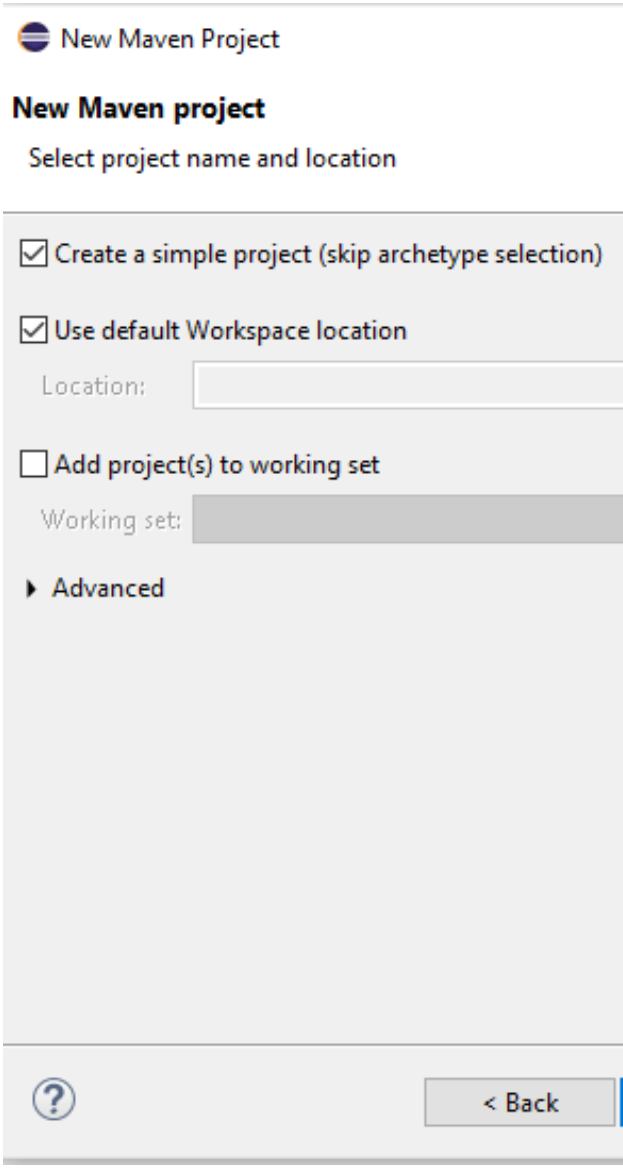
0.187.12.16 Maven Java Application + JUnit

Lets create Maven project with tests

Create a maven project

File -> New -> Maven Project


New Project
Select a wizard
Create a Maven project
Wizards:
type filter text
Project
Gradle
Gradle Project
Java
Java Project
Java Project from Existing Ant Buildfile
Maven
Check out Maven Projects from SCM
Maven Module
Maven Project
Examples
Editing and validating XML files
< Back
Next >
Finish
Cancel



0.187.12.17 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

0.187.12.18 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[28]

Also for intro you can use this

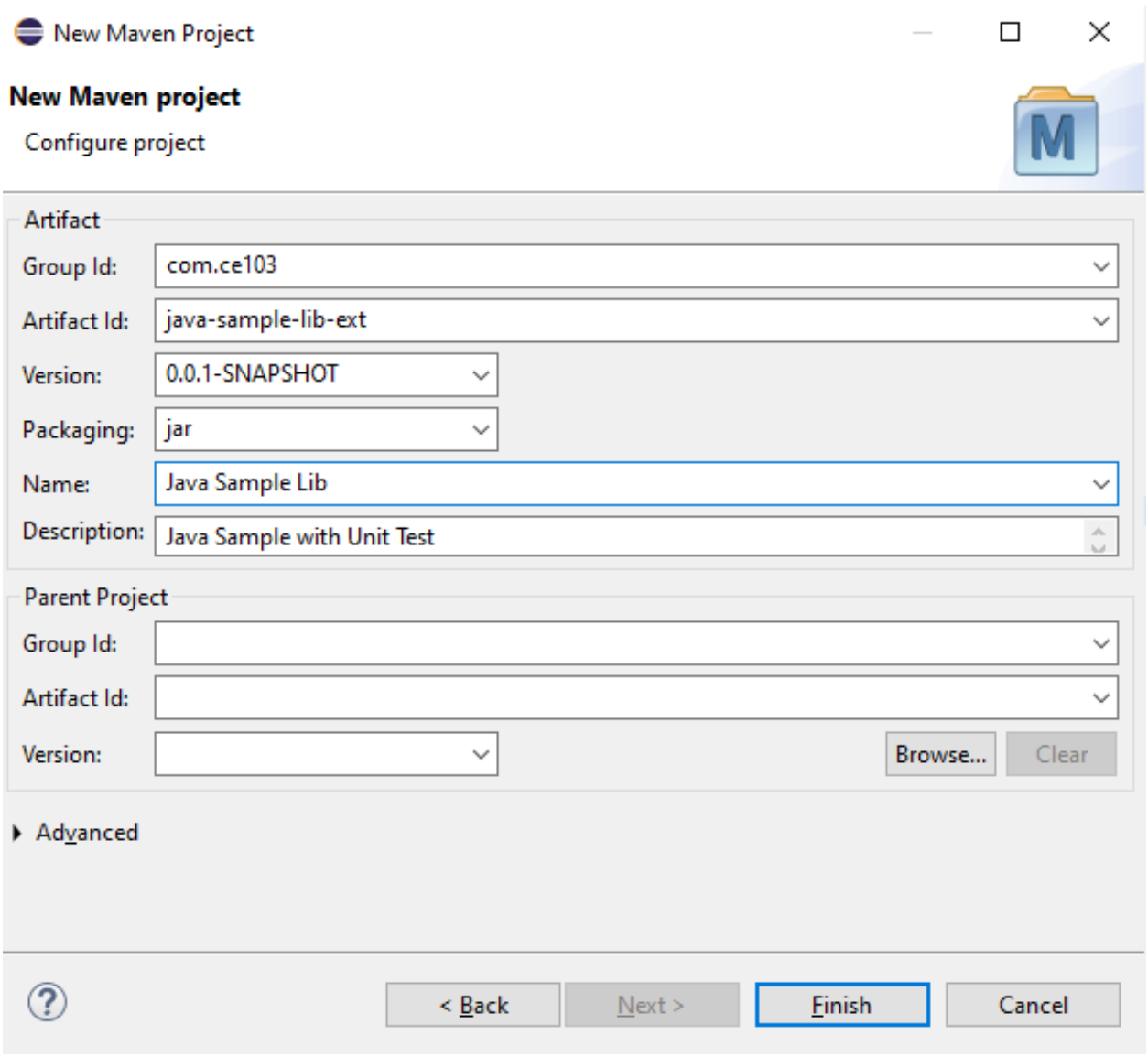
JUnit Hello World Example - Examples Java Code Geeks - 2021[29]

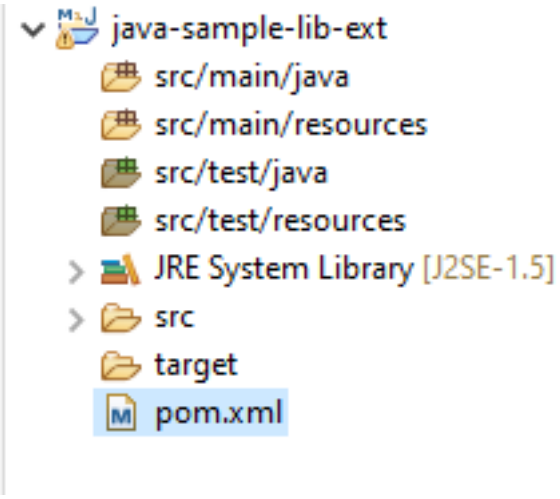
Eclipse
Maven
Java
JUnit 4.12 (pulled by Maven automatically)

0.187.12.19 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

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

[28] http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html
[29] https://examples.javacodegeeks.com/core-java/junit/junit-hello-world-example/





0.187.12.20 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

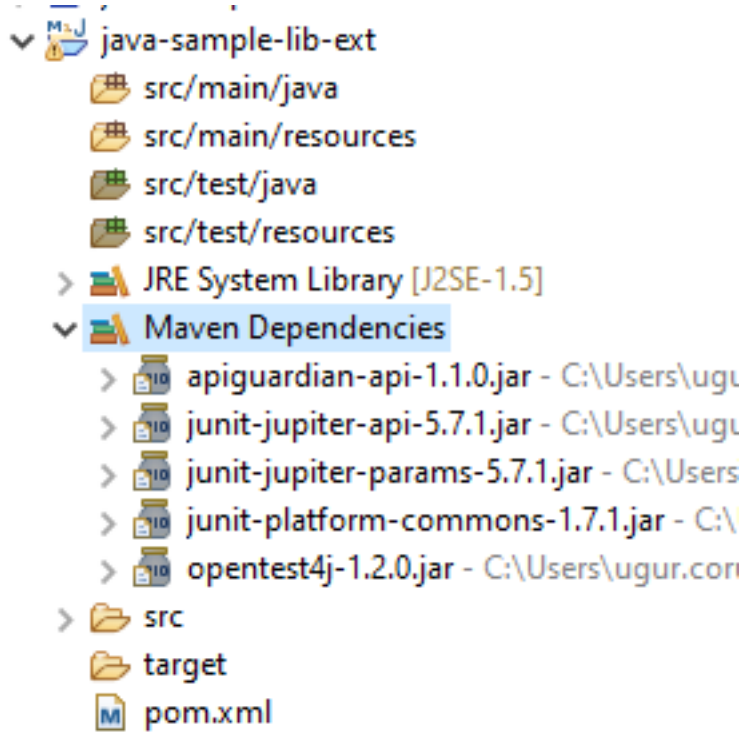
0.187.12.21 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test **pom.xml** file

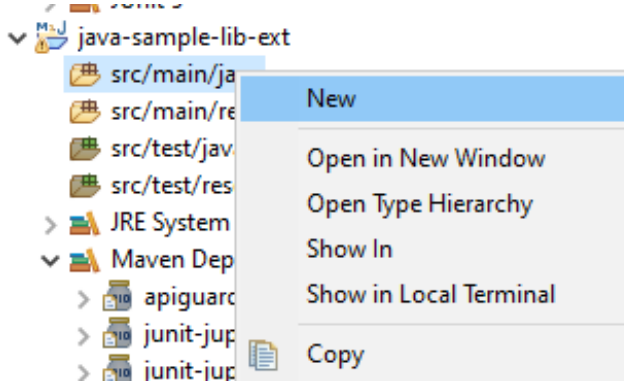
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.ce103</groupId>
  <artifactId>java-sample-lib-ext</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Java Sample Lib</name>
  <description>Java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

0.187.12.22 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>
```

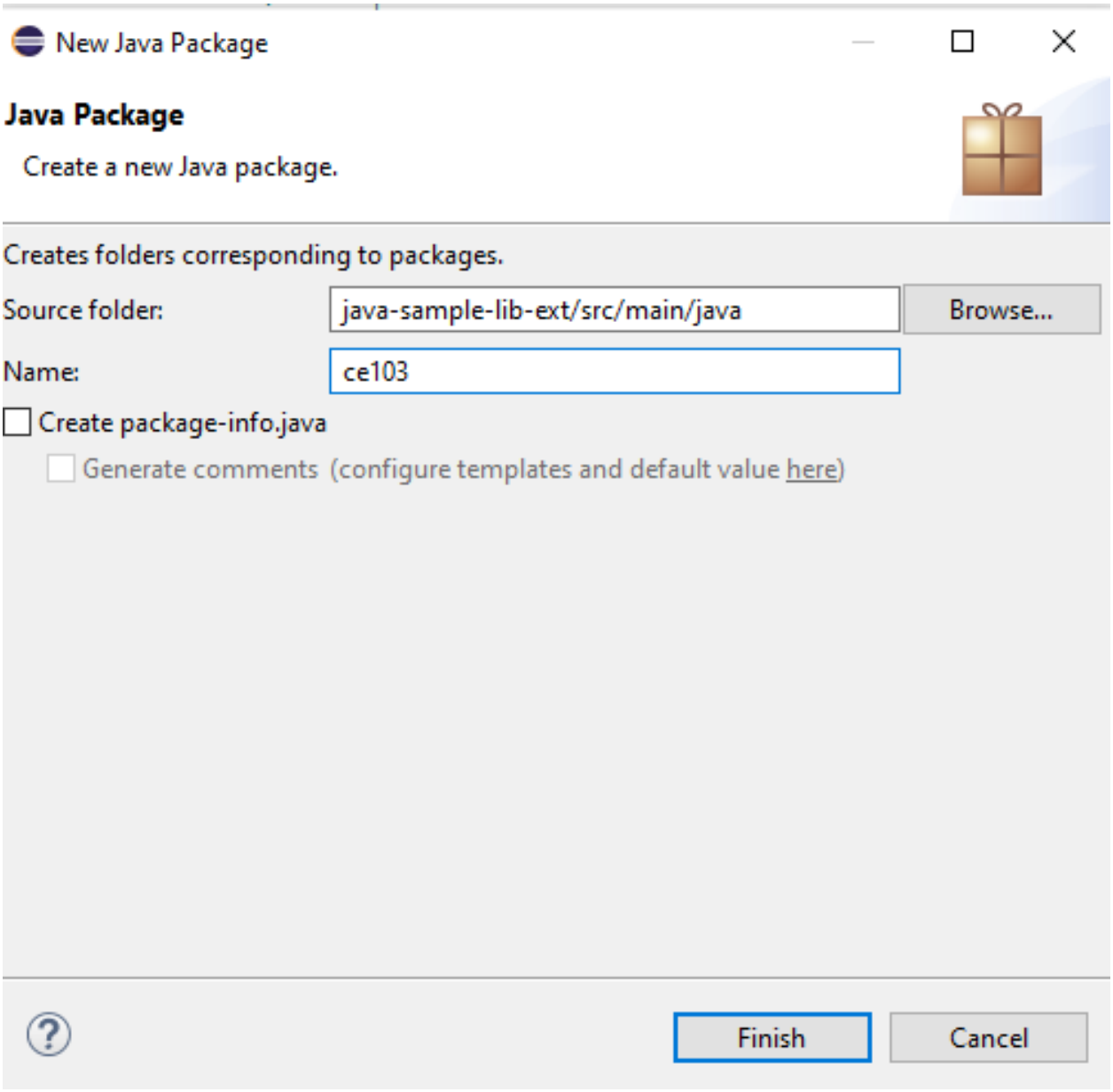
0.187.12.23 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test it will automatically download libraries



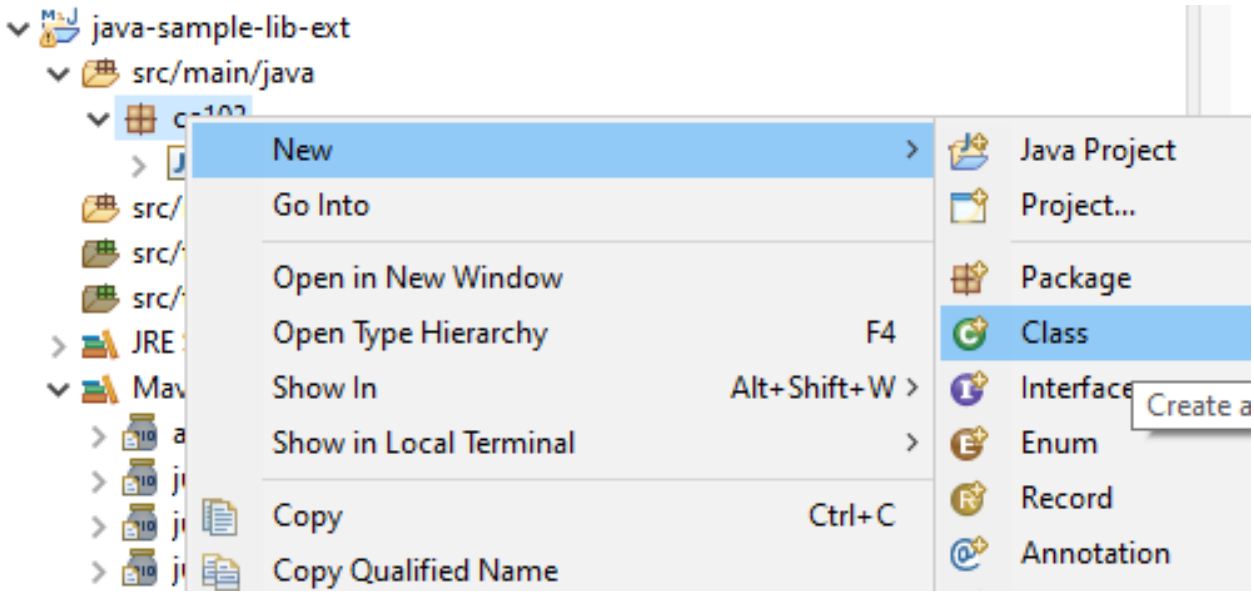


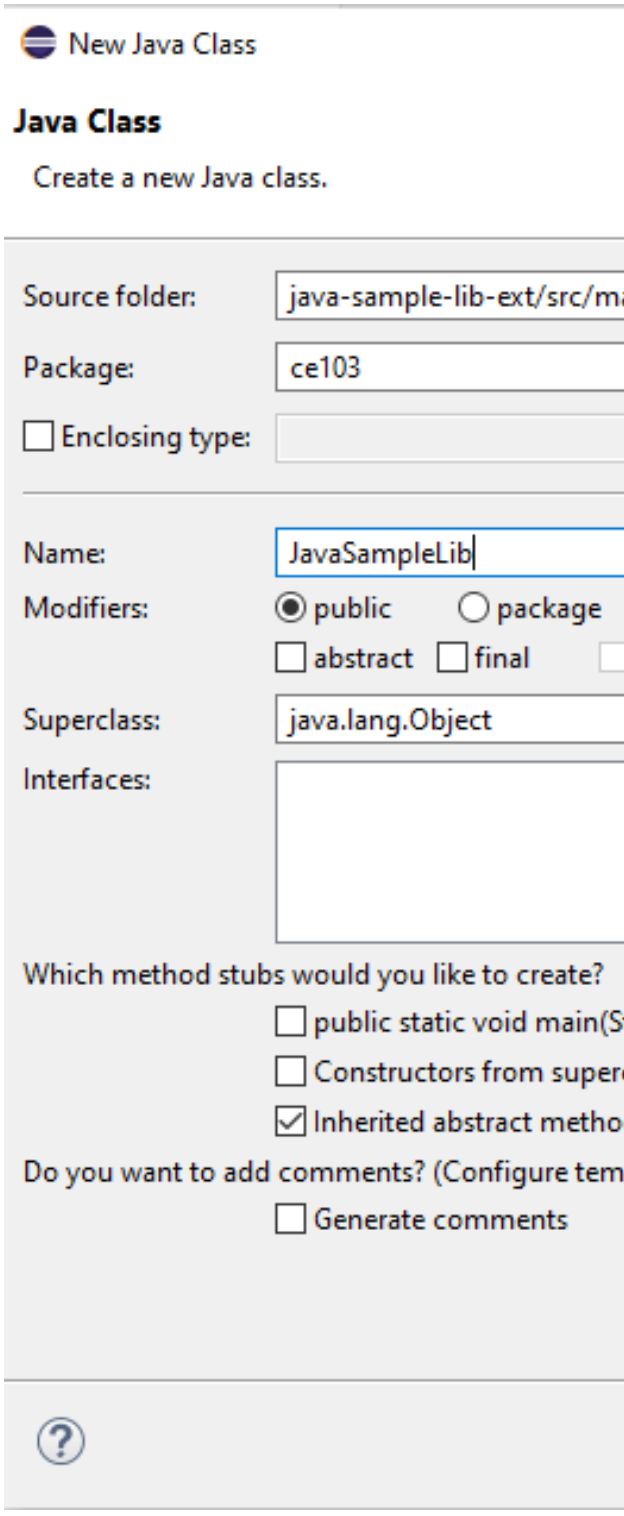
0.187.12.24 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

0.187.12.25 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test Create java sample library in ce103 package, first create java package



0.187.12.26 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test In this package create library class


java-sample-lib-ext
src/main/java
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Copy Qualified Name
Java Project
Project...
Package
Class
Interface
Enum
Record
Annotation



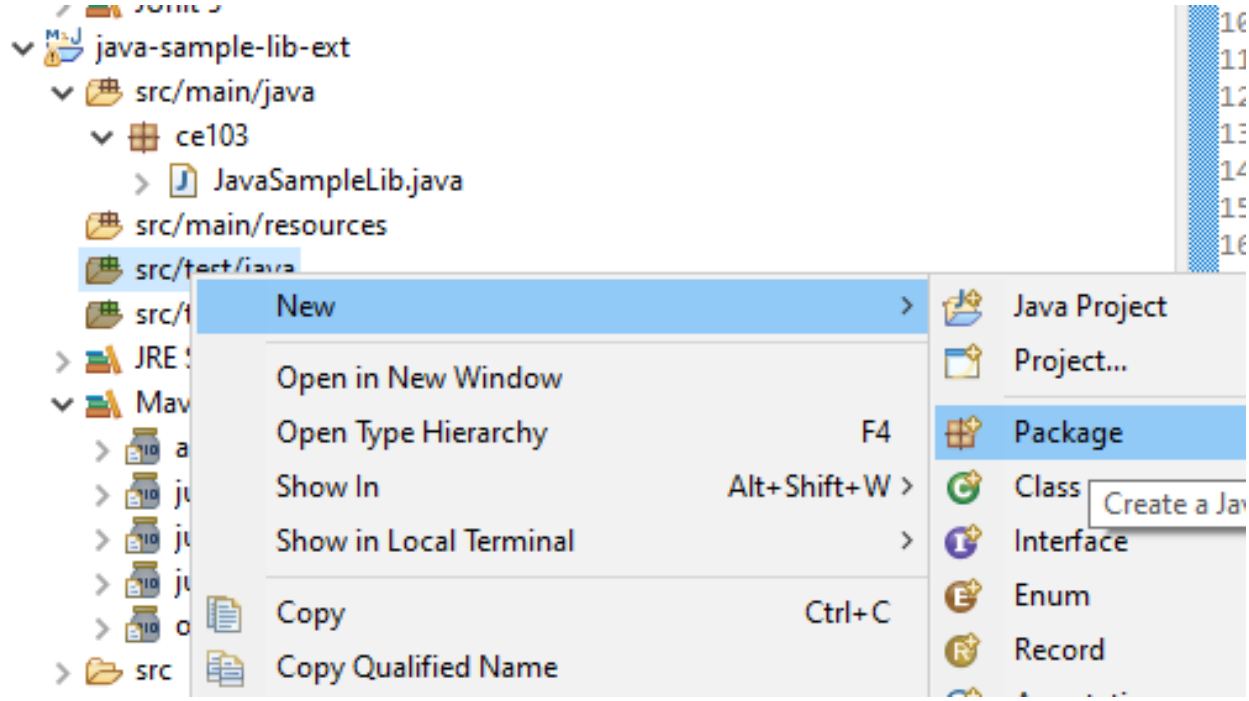
0.187.12.27 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

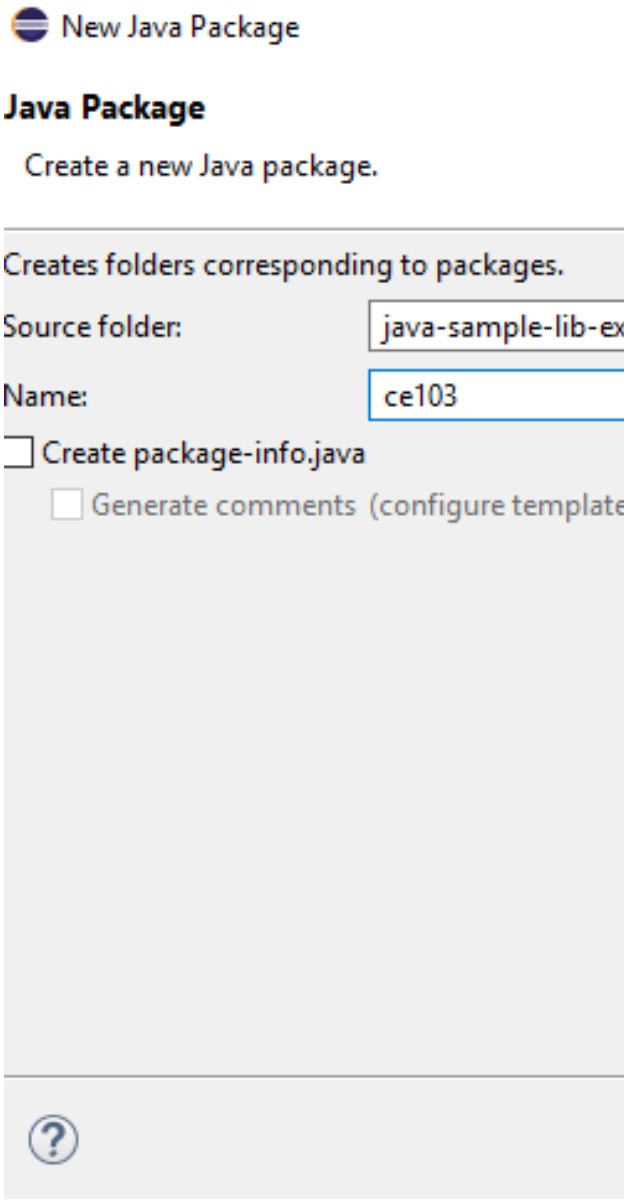
0.187.12.28 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test copy content from other library

```
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;
    }


}
```

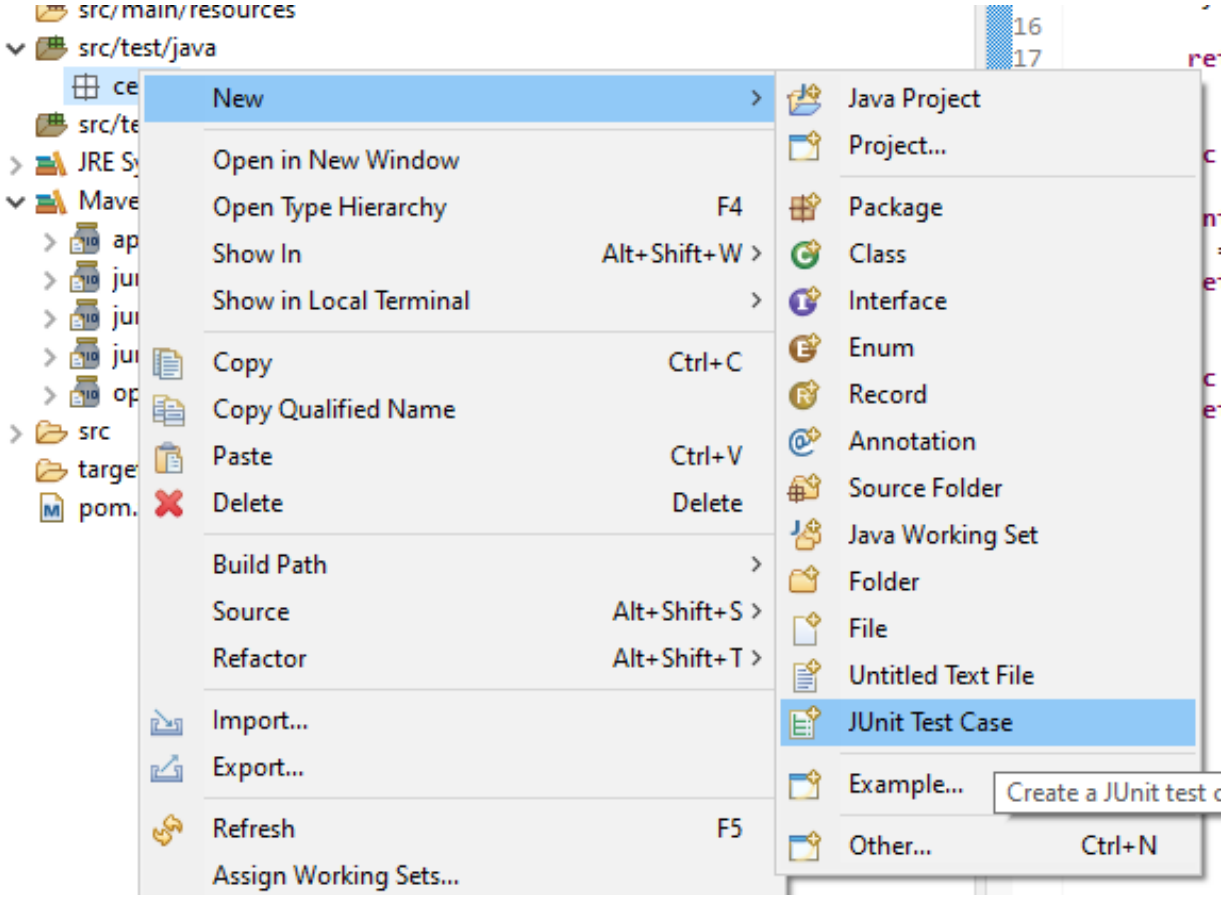
0.187.12.29 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test Now lets create tests inf src/test/java

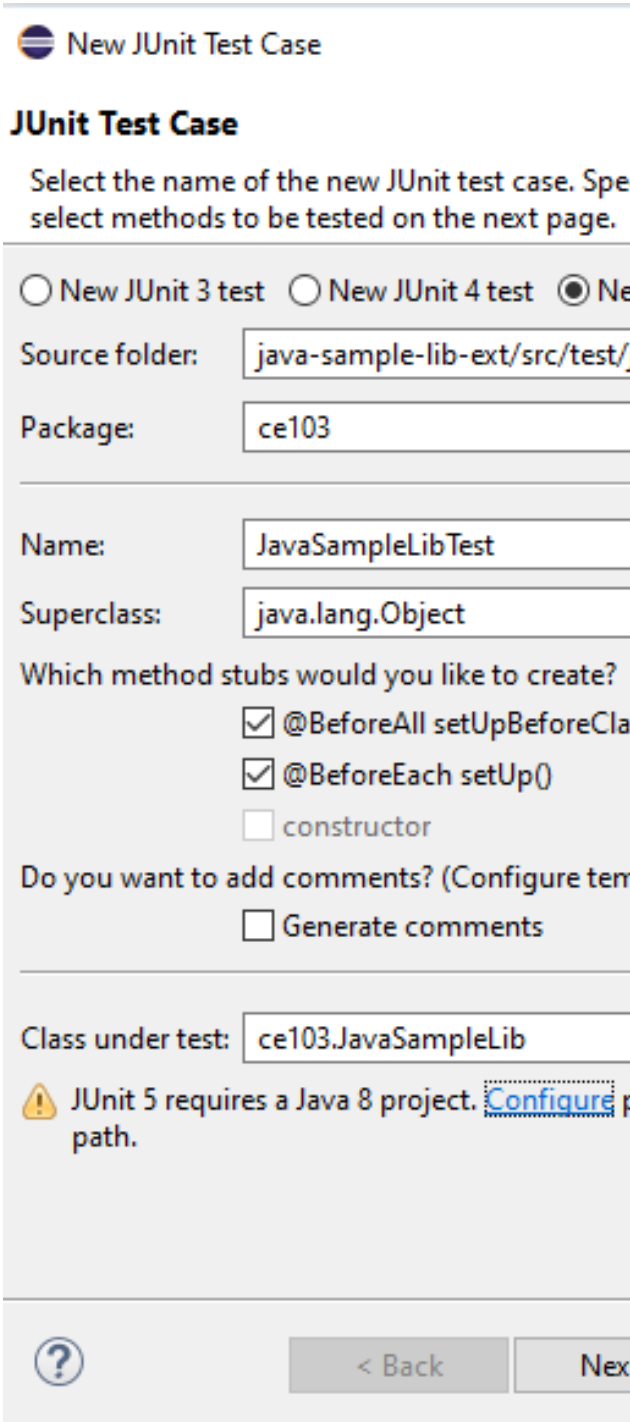




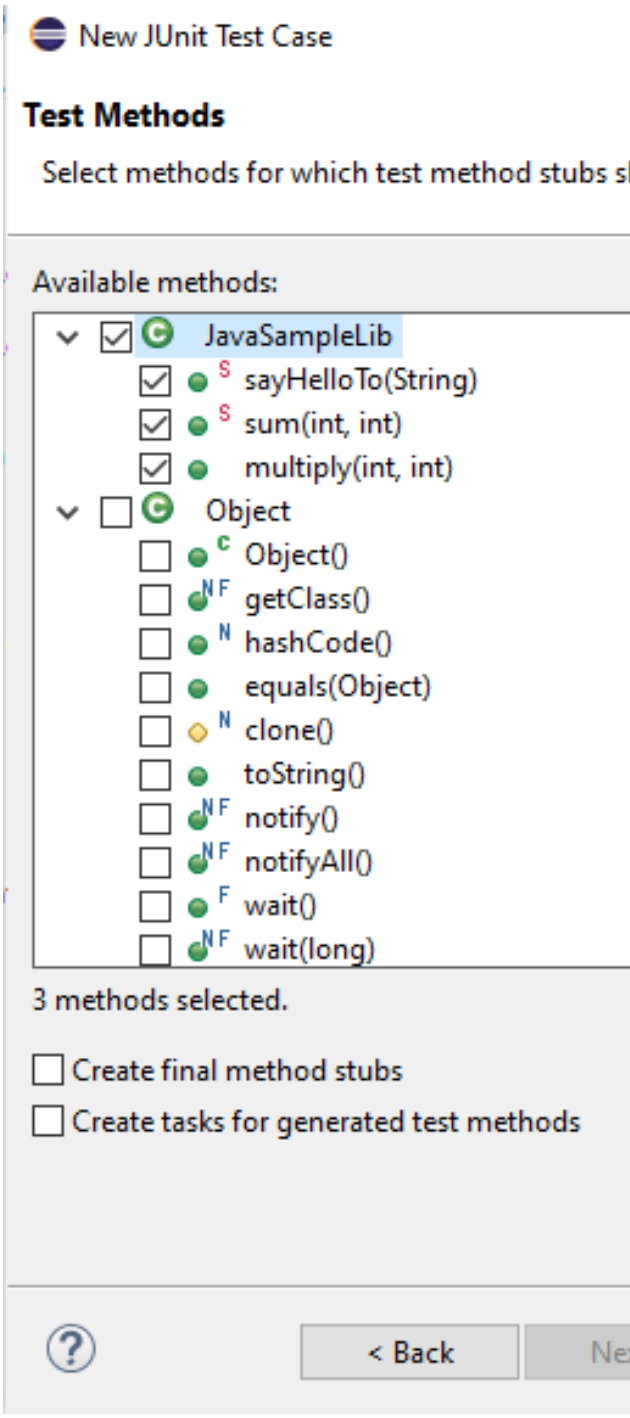
0.187.12.30 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test

0.187.12.31 Eclipse IDE (JUnit4 , JUnit5) + Java Unit Test create a JUnit Case

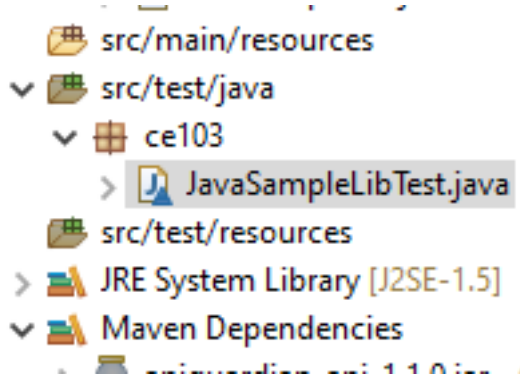

src/main/resources
src/test/java
New
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Assign Working Sets...
Java Project
Project...
Package
Class
Interface
Enum
Record
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
JUnit Test Case
Example...
Other... Ctrl+N
Create a JUnit test



0.187.12.32 Eclipse IDE (JUnit4, JUnit5) + Java Unit Test



0.187.12.33 Eclipse IDE (JUnit4, JUnit5) + Java Unit Test



0.187.12.34 Eclipse IDE (JUnit4, JUnit5) + Java Unit Test

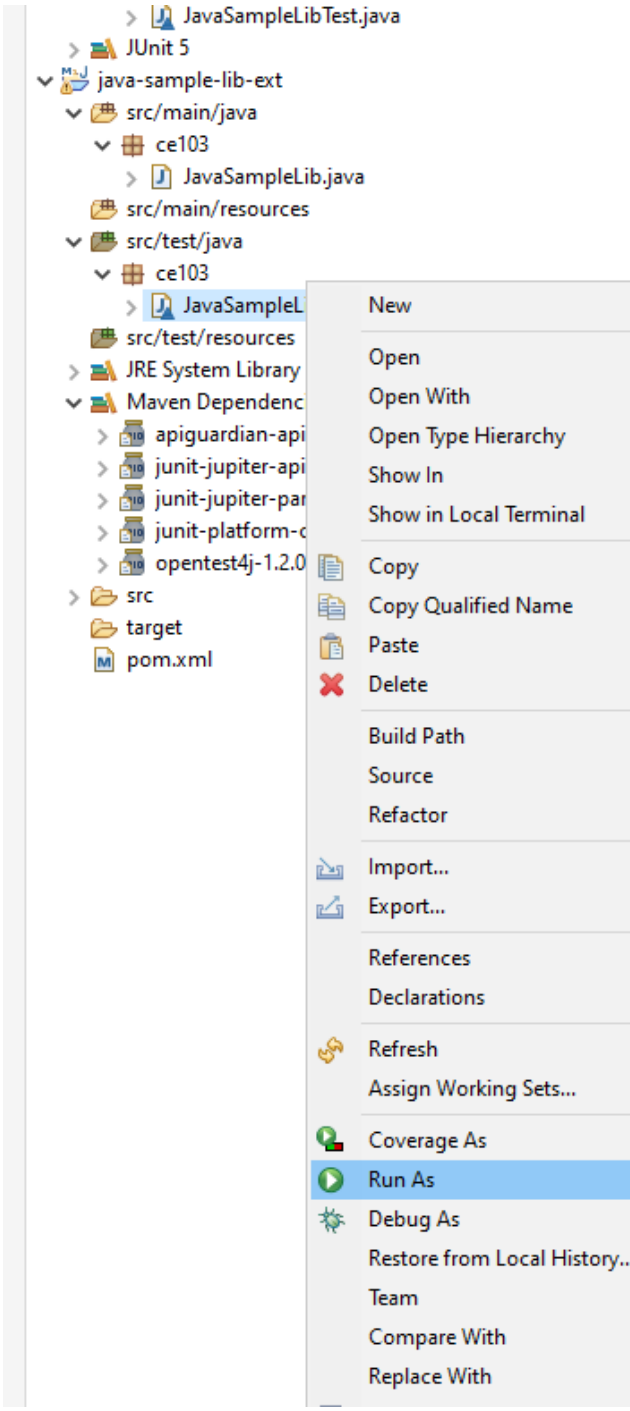
0.187.12.35 Eclipse IDE (JUnit4, JUnit5) + Java Unit Test you will simple template

```
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");
```

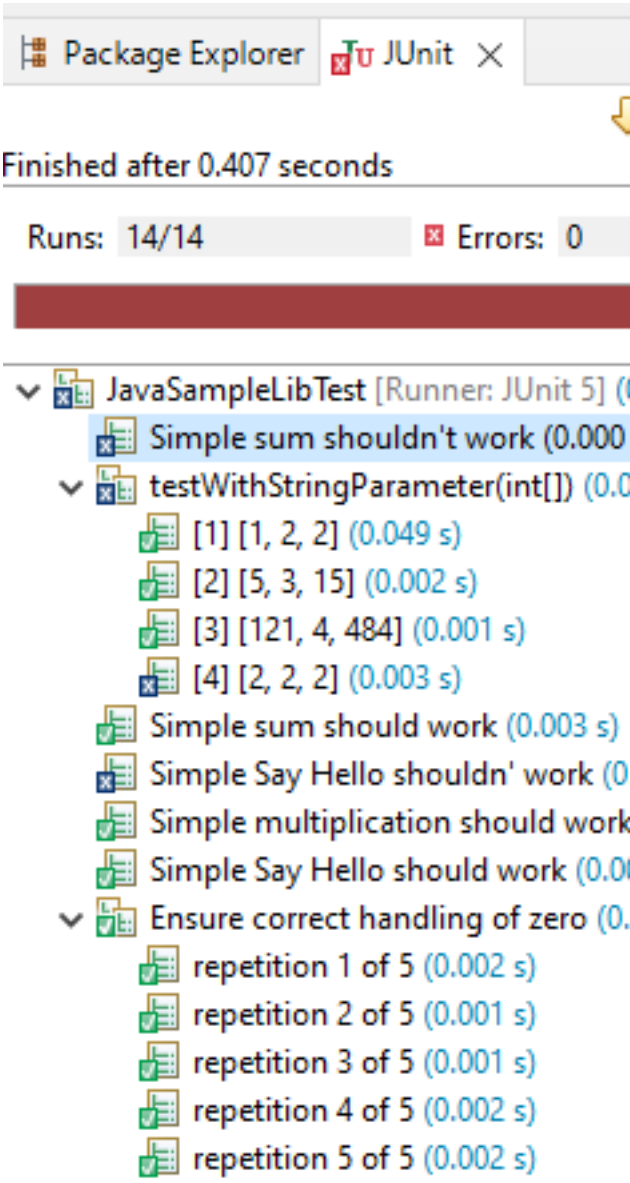
```
    }

}
```

0.187.12.36 Eclipse IDE (JUnit4, JUnit5) + Java Unit Test now lets copy tests from other projects

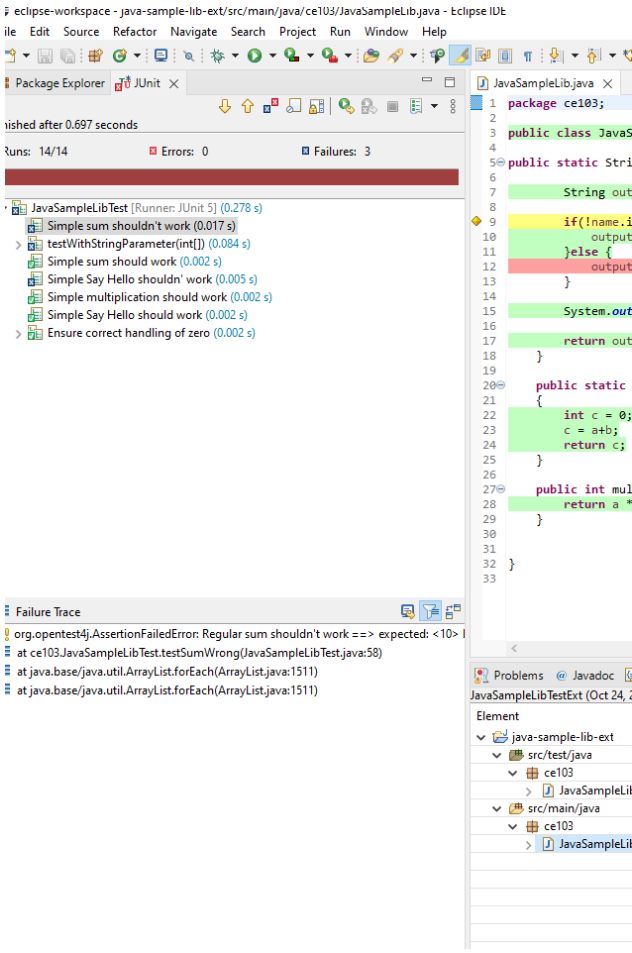
```
Convert source codes to java codes...
```



0.187.12.37 Eclipse IDE (JUnit4, JUnit5) + Java Unit Test



0.187.12.38 Eclipse IDE (JUnit4, JUnit5) + Java Unit Test



0.187.12.39 Eclipse IDE (JUnit4, JUnit5) + Java Unit Test

0.187.12.40 Eclipse IDE (JUnit4, JUnit5) + Java Unit Test That's a part of java unit testing...

0.188 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

0.189 Test and Deployment Automation Management

There are several Continues-Integration services online as follow; - Travis-CI - Appveyor - Jenkins - CircleCI - GitLab - Pantheon - GitHub - Bitrise - Flosum - Buddy - Semaphore

---

0.190 Test and Deployment Automation Management

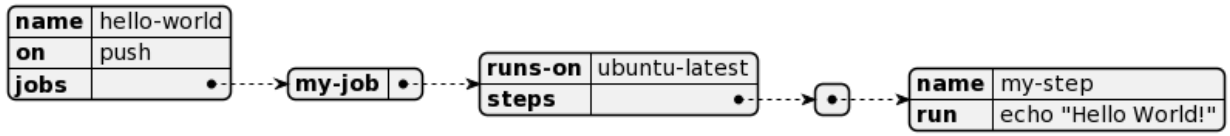
- Github provides Github Actions for Releases and Tests
- Jenkins has on promise solutions private development

0.191 Test and Deployment Automation Management

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

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

0.192 Test and Deployment Automation Management



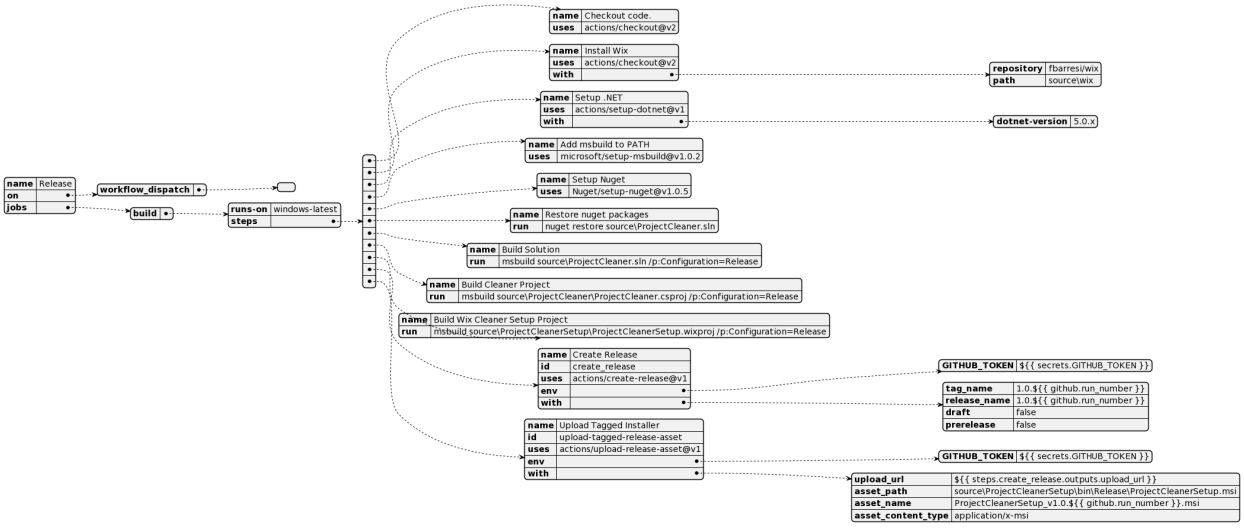
0.193 Test and Deployment Automation Management

- 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

---



1 References

GitHub - MicrosoftDocs/cpp-docs: C++ Documentation[30]

$$End - Of - Week - 4$$

[30] https://github.com/MicrosoftDocs/cpp-docs