

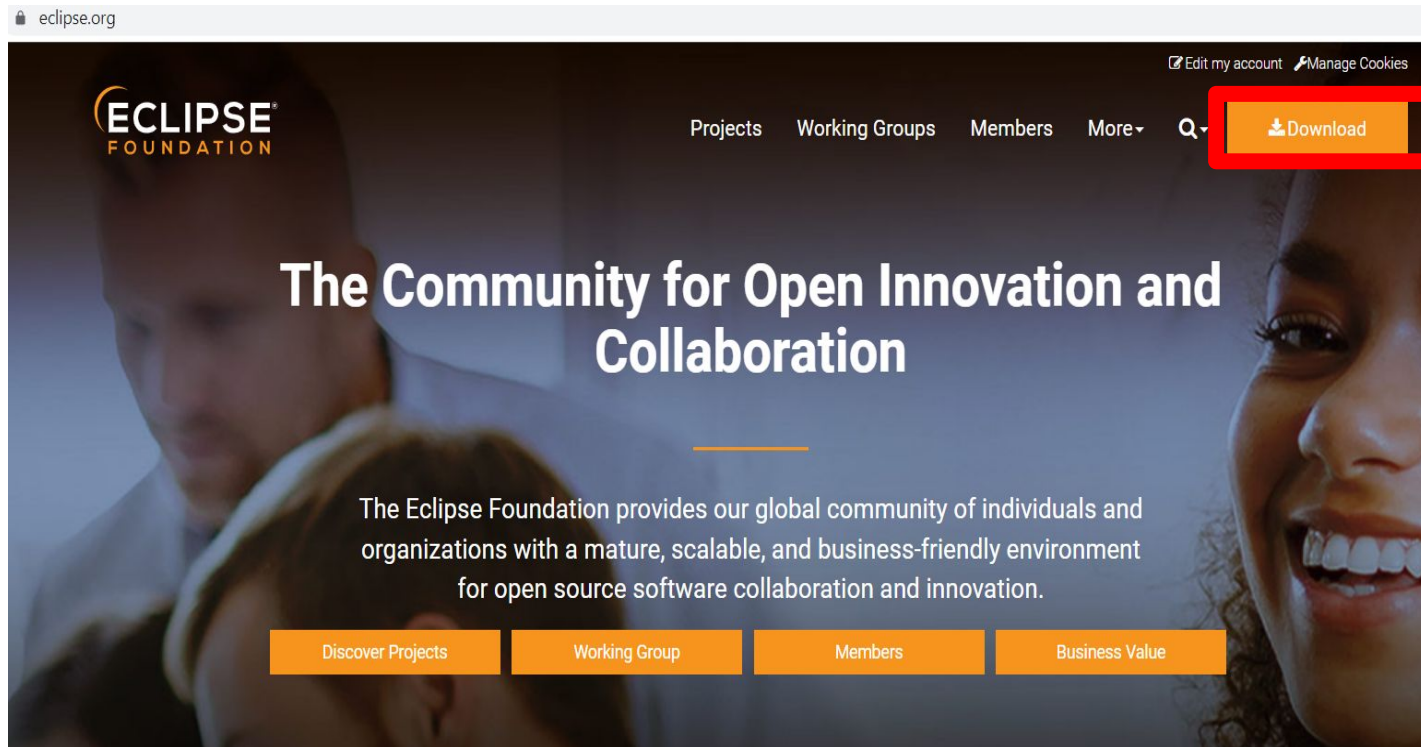
Tutorial 1

How to Download and Install Eclipse to Run Java

Step 1) Installing Eclipse

Open your browser and type
<https://www.eclipse.org/>

Step 2) Click on **Download** button.



Step 3) Click on **Download 64 bit** button

The Eclipse Installer 2022-09 R now includes a JRE for macOS, Windows and Linux.



Get **Eclipse IDE 2022-09**

Install your favorite desktop IDE packages.

Download x86_64

[Download Packages](#) | [Need Help?](#)

Step 4) Click on **Download** button

All downloads are provided under the terms and conditions of the [Eclipse Foundation Software User Agreement](#) unless otherwise specified.



Download from: Canada - Rafal Rzekowski (<https>)

File: [eclipse-inst-jre-win64.exe](#) SHA-512

[>> Select Another Mirror](#)

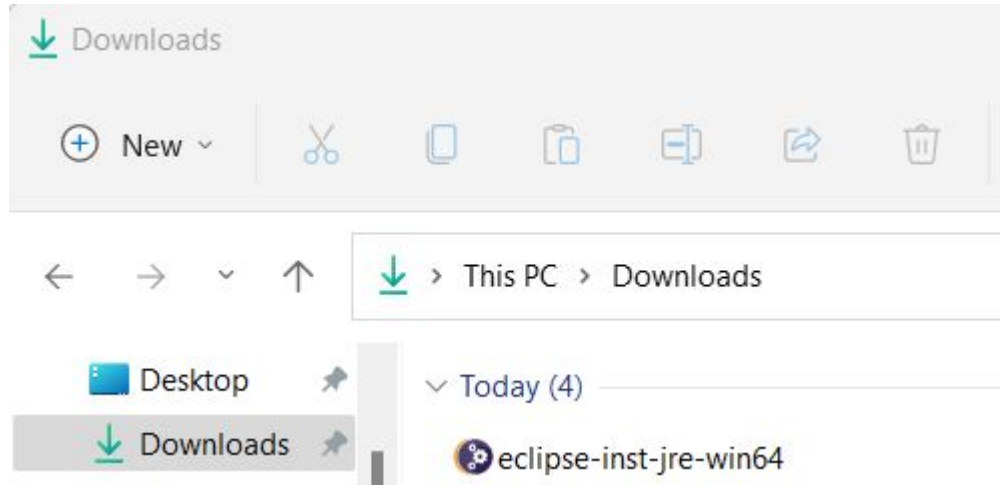


eclipse-inst-jre-wi....exe

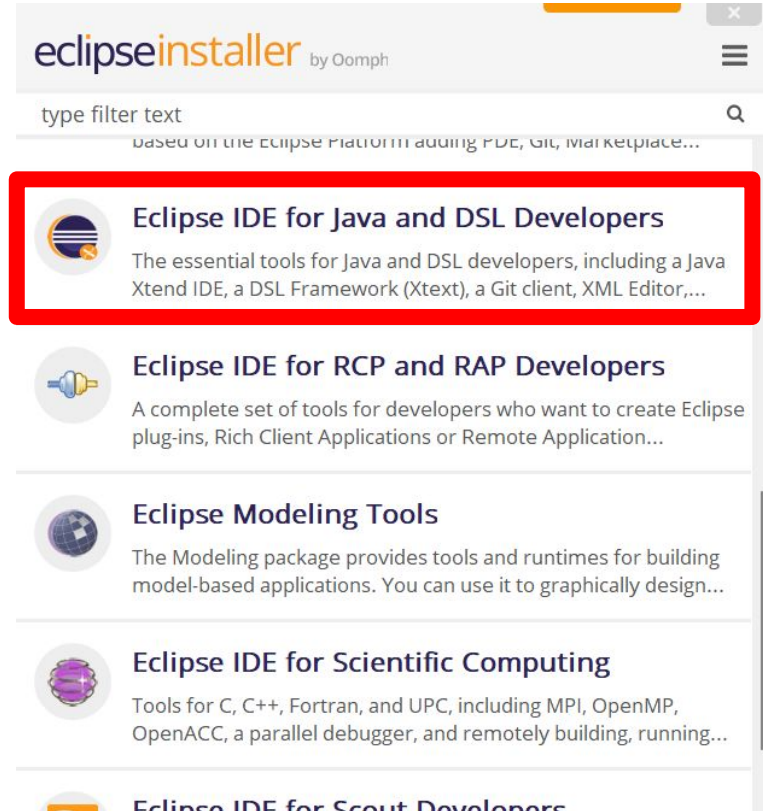


Sep 4) Install Eclipse.

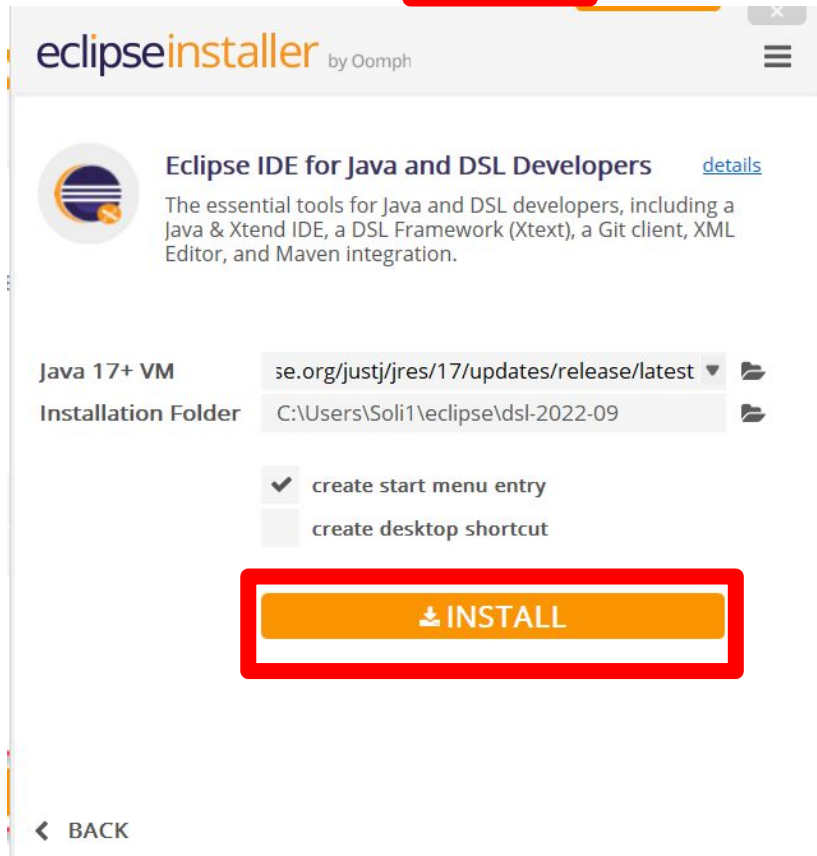
1. Click on downloads folder in Windows
2. Click on eclipse-inst-win64.exe file



Step 6) Click on Eclipse IDE for Java and DSL Developers



Step 7) Click on **INSTALL** button



Eclipse IDE for Java and DSL Developers

eclipseinstaller by Oomph



Eclipse IDE for Java and DSL Developers [details](#)

The essential tools for Java and DSL developers, including a Java & Xtend IDE, a DSL Framework (Xtext), a Git client, XML Editor, and Maven integration.

Java 17+ VM

se.org/justj/jres/17/updates/release/latest

Installation Folder

C:\Users\Soli1\eclipse\dsl-2022-09



create start menu entry



create desktop shortcut



INSTALLING

✕ Cancel Installation

< BACK



Eclipse IDE for Java and DSL Developers [details](#)

The essential tools for Java and DSL developers, including a Java & Xtend IDE, a DSL Framework (Xtext), a Git client, XML Editor, and Maven integration.

Java 17+ VM

se.org/justj/jres/17/updates/release/latest

Installation Folder

C:\Users\Soli1\eclipse\dsl-2022-09



create start menu entry



create desktop shortcut

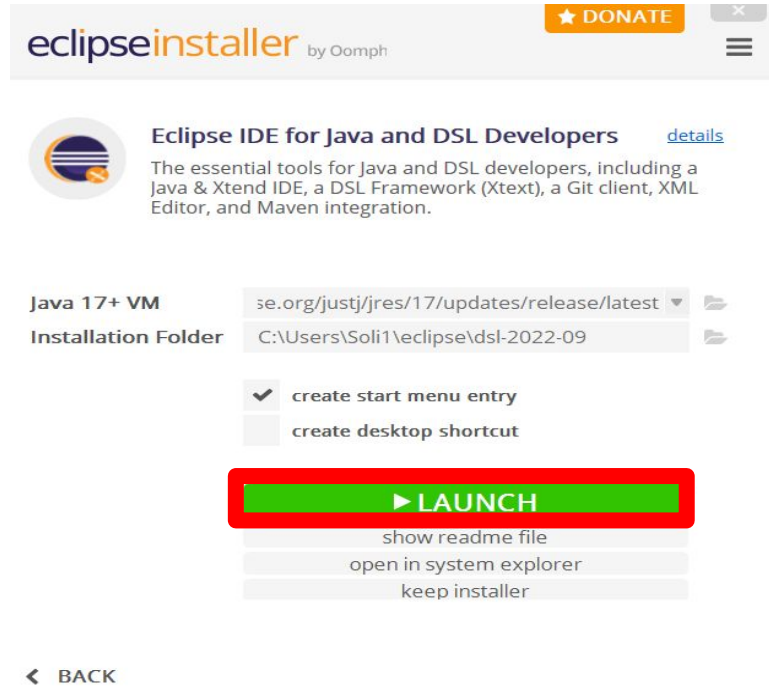


INSTALLING

✕ Cancel Installation


< BACK

Step 8) Click on **LAUNCH** button



eclipseinstaller by Oomph

★ DONATE

 **Eclipse IDE for Java and DSL Developers** [details](#)

The essential tools for Java and DSL developers, including a Java & Xtend IDE, a DSL Framework (Xtext), a Git client, XML Editor, and Maven integration.

Java 17+ VM

Installation Folder

☒ create start menu entry

☐ create desktop shortcut

▶ LAUNCH

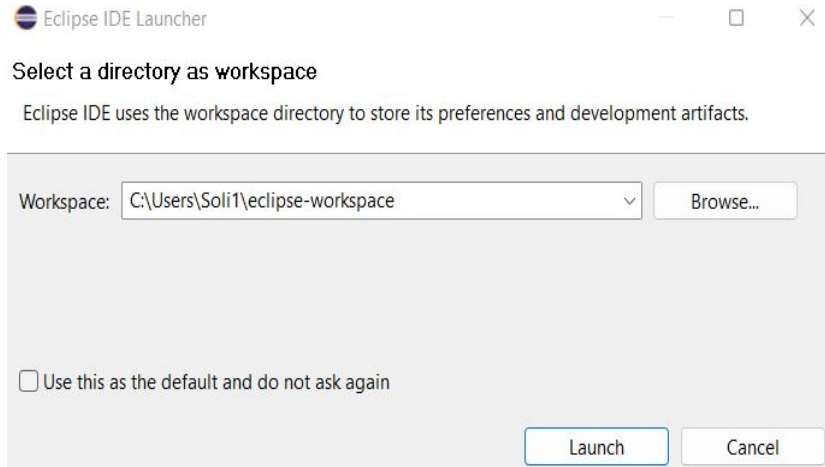
[show readme file](#)

[open in system explorer](#)

[keep installer](#)

◀ BACK

Step 9) Click on **Launch** button



Step 10) Click on Create a new project link



Review IDE configuration settings

Review the IDE's most fiercely contested preferences



Overview

Get an overview of the features



Visit Xtext Documentation

Features a 15 minute tutorial on getting started with Xtext



Tutorials

Go through tutorials



Create a new Xtext project

Create a new Xtext Eclipse project



Samples

Try out the samples



Checkout projects from Git

Checkout Eclipse projects hosted in a Git repository



What's New

Find out what is new



Import existing projects

Import existing Eclipse projects from the filesystem or archive



Launch the Eclipse Marketplace

Enhance your IDE with additional plugins and install your Marketplace favorites

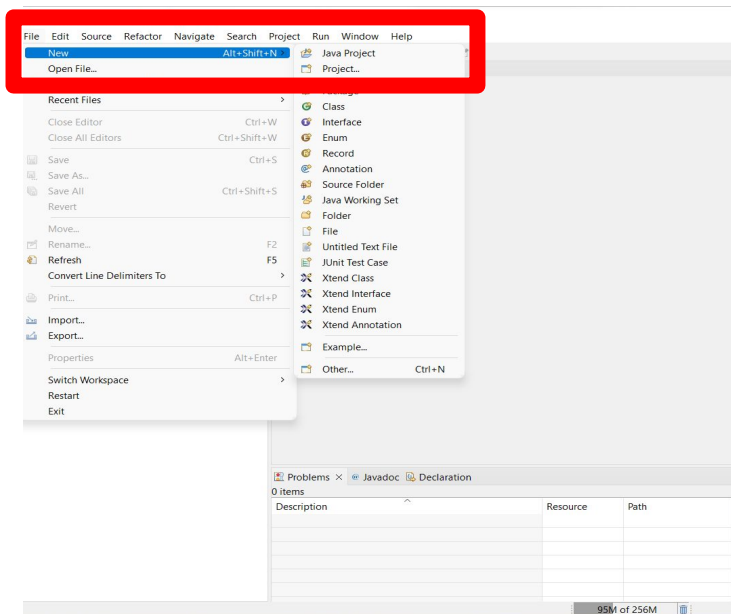


Open an existing file

Open a file from the filesystem

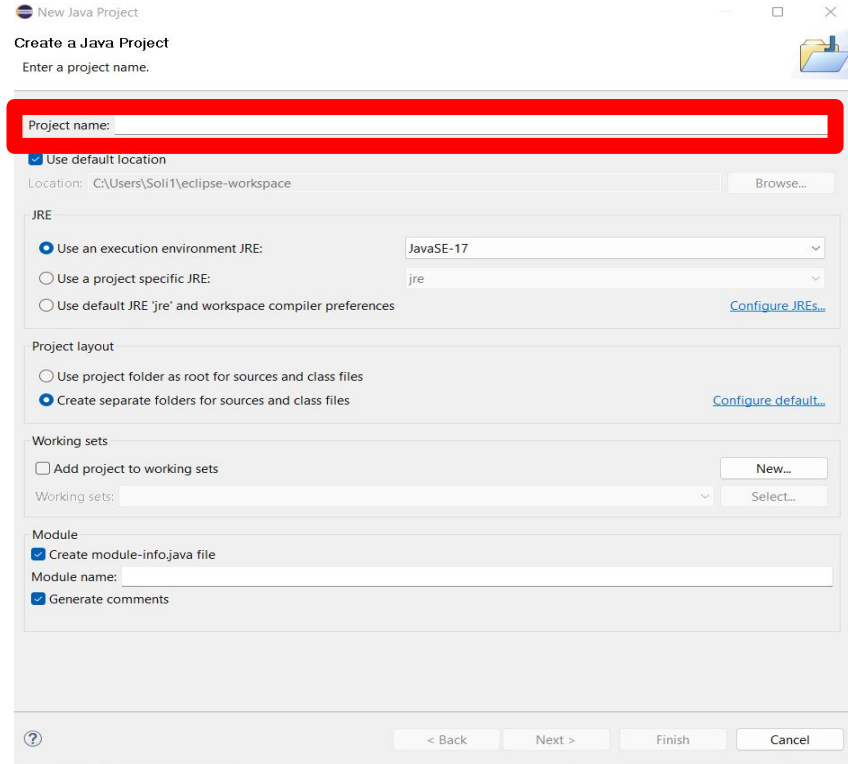
Step 11) Create a new Java Project

File->new->Java Project



Step 12) Create a new Java Project

1. Write project name
2. Click on Finish button



New Java Project

Create a Java Project

Enter a project name.

Project name:

☒ Use default location

Location: C:\Users\Solmaz\workspace [Browse...](#)

JRE

☒ Use an execution environment JRE: JavaSE-17

☐ Use a project specific JRE: jre

☐ Use default JRE 'jre' and workspace compiler preferences [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets [New...](#)

Working sets: [Select...](#)

Module

☒ Create module-info.java file

Module name:

☒ Generate comments

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

Step 12) Create a new Java Project

1. Write **project name** (I write **CS380Test**)
2. Click on **Finish** button

New Java Project

Create a Java Project

Discouraged module name. By convention, module names usually start with a lowercase letter

Project name: CS380Test

Location: C:\Users\Solmaz\workspace\CS380Test

JRE

☒ Use an execution environment JRE: JavaSE-17

☐ Use a project specific JRE: jre

☐ Use default JRE 'jre' and workspace compiler preferences

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files

Working sets

☐ Add project to working sets

Working sets:

Module

☒ Create module-info.java file

Module name:

☒ Generate comments

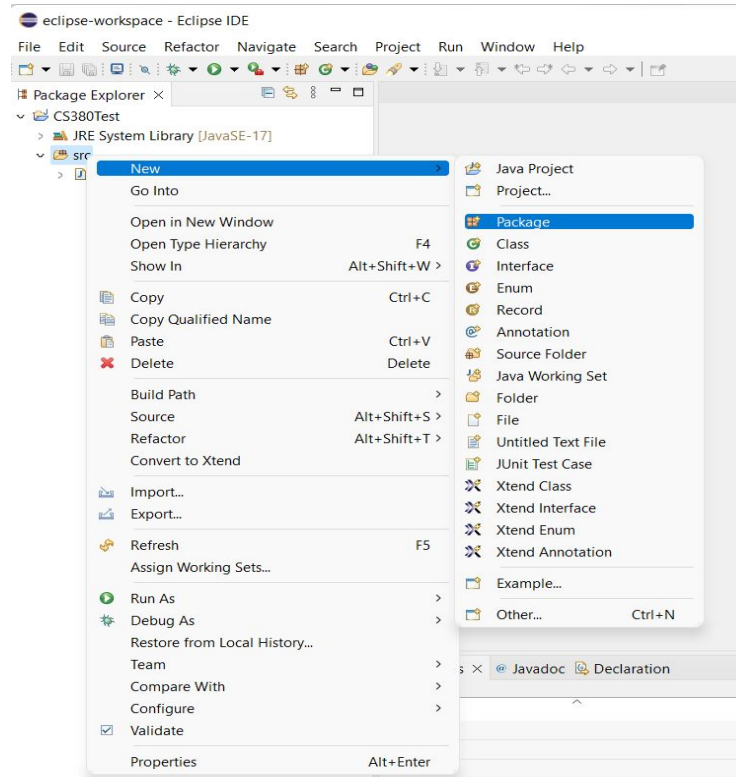
module name will be "CS380Test" (if no module is specified, then project name will be used as module name)

< Back Next > Finish Cancel

Step 13) Create Java Package

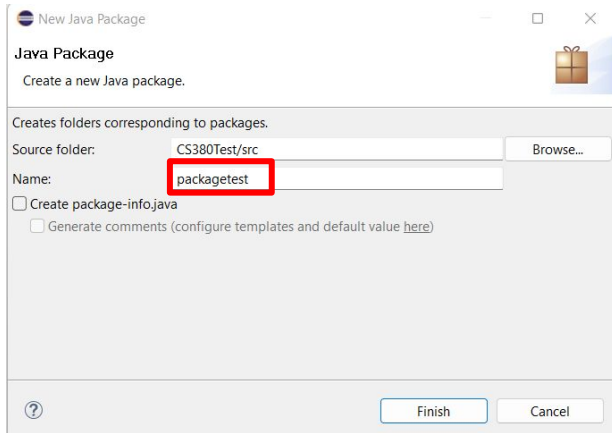
Right click on src

1. Goto **src**
2. Click on **New**
3. Click on **Package**

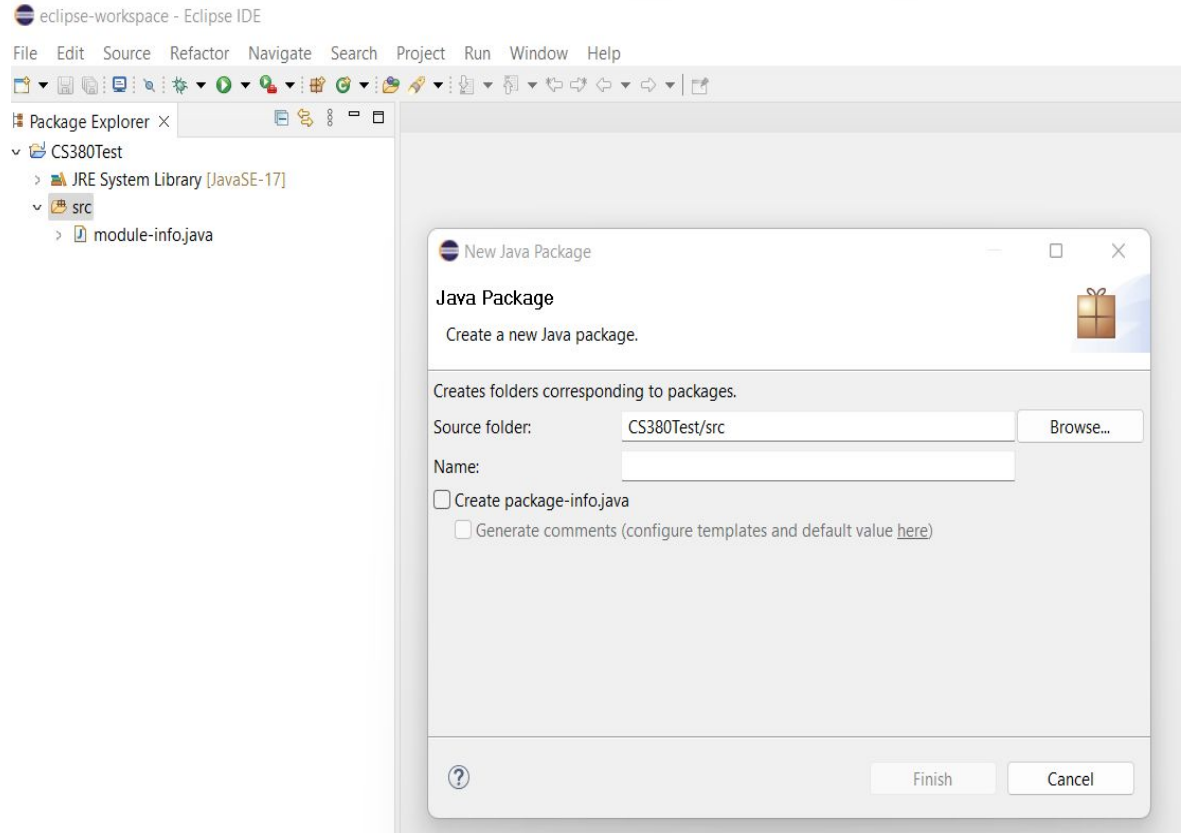


Step 14) Writing package name

1. Write name of the package
2. Click on Finish button

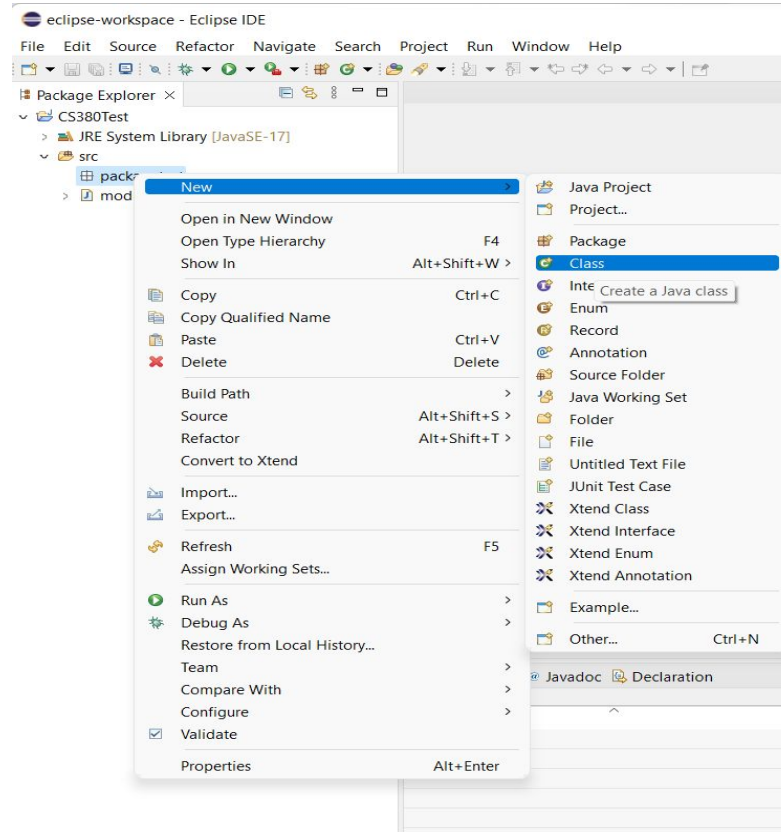


I write
packagetest



Step 15) Creating Java Class

1. Click on package you have created.
2. Click on **New**
3. Click on **Class**



Step 16) Defining Java Class

1. Write class name: **CS380_lab0**
2. Click on public static void main (String[] args) checkbox
3. Click on **Finish** button.

New Java Class

Java Class

Create a new Java class.

Source folder: CS380Test/src Browse...

Package: packetest Browse...

☐ Enclosing type: Browse...

Name: 1 CS380_Lab0

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

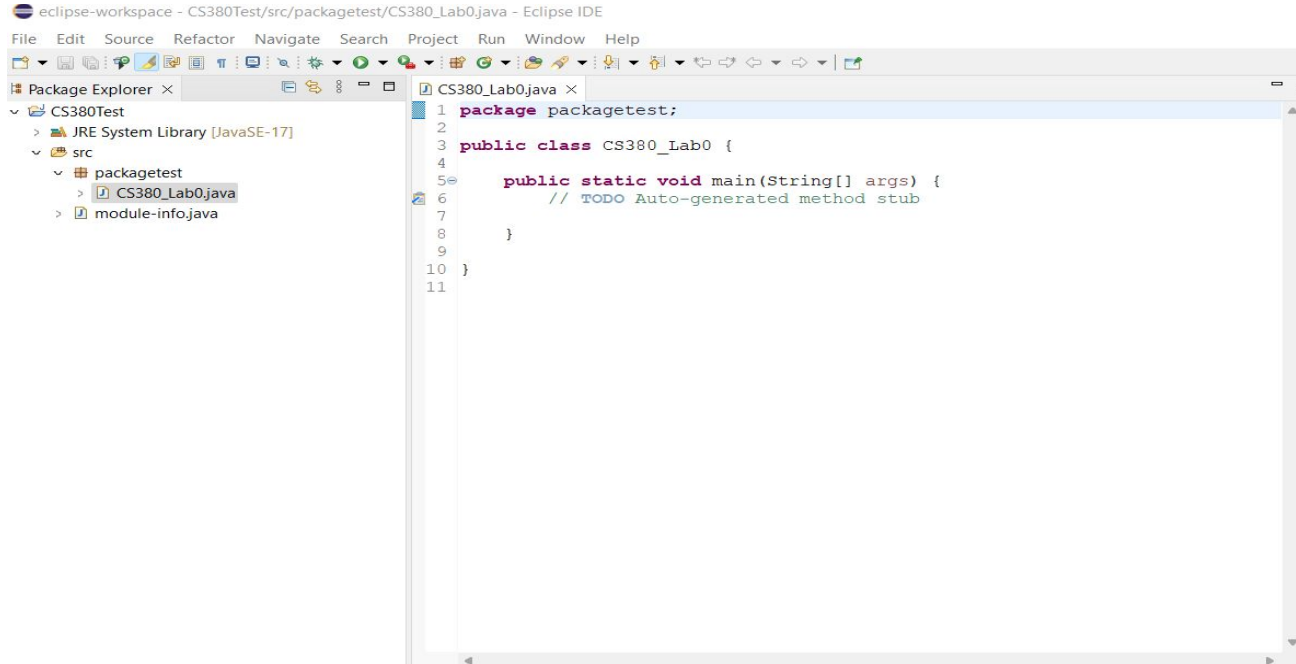
Which method stubs would you like to create? 2

☒ 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

3 Finish Cancel

CS380_lab0.java file will be created as shown below



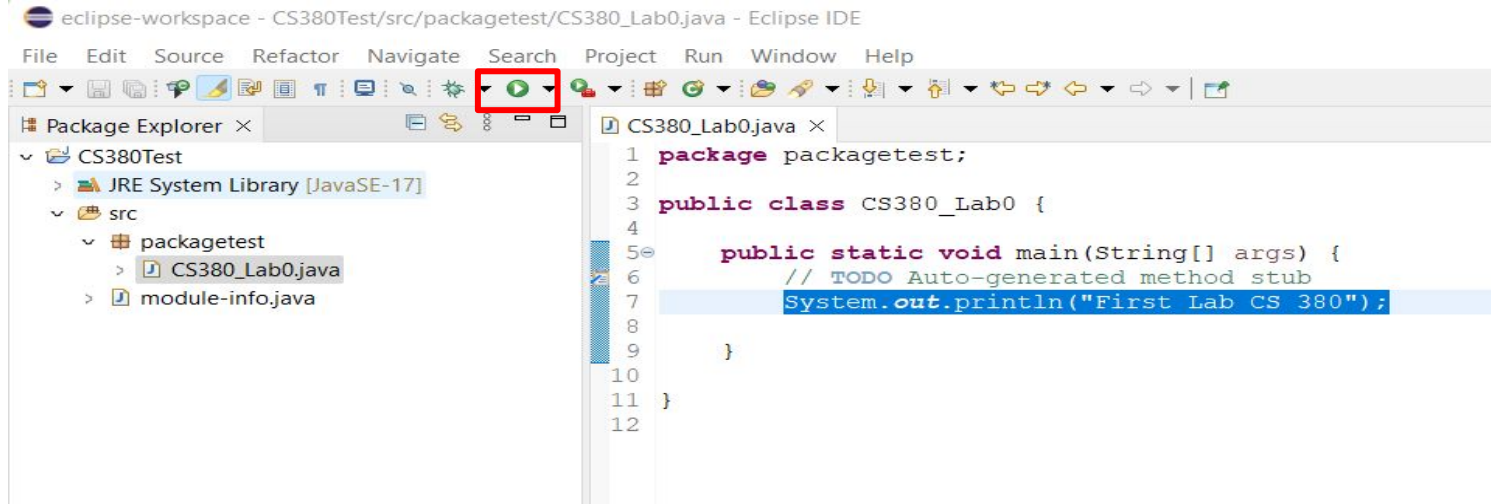
The screenshot shows the Eclipse IDE interface. The title bar reads "eclipse-workspace - CS380Test/src/packageTest/CS380_Lab0.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The Package Explorer on the left shows a project named "CS380Test" with a "src" folder containing a "packageTest" package. Inside "packageTest", the file "CS380_Lab0.java" is selected. The main editor window displays the following Java code:

```
1 package packageTest;
2
3 public class CS380_Lab0 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7     }
8 }
9
10
11
```

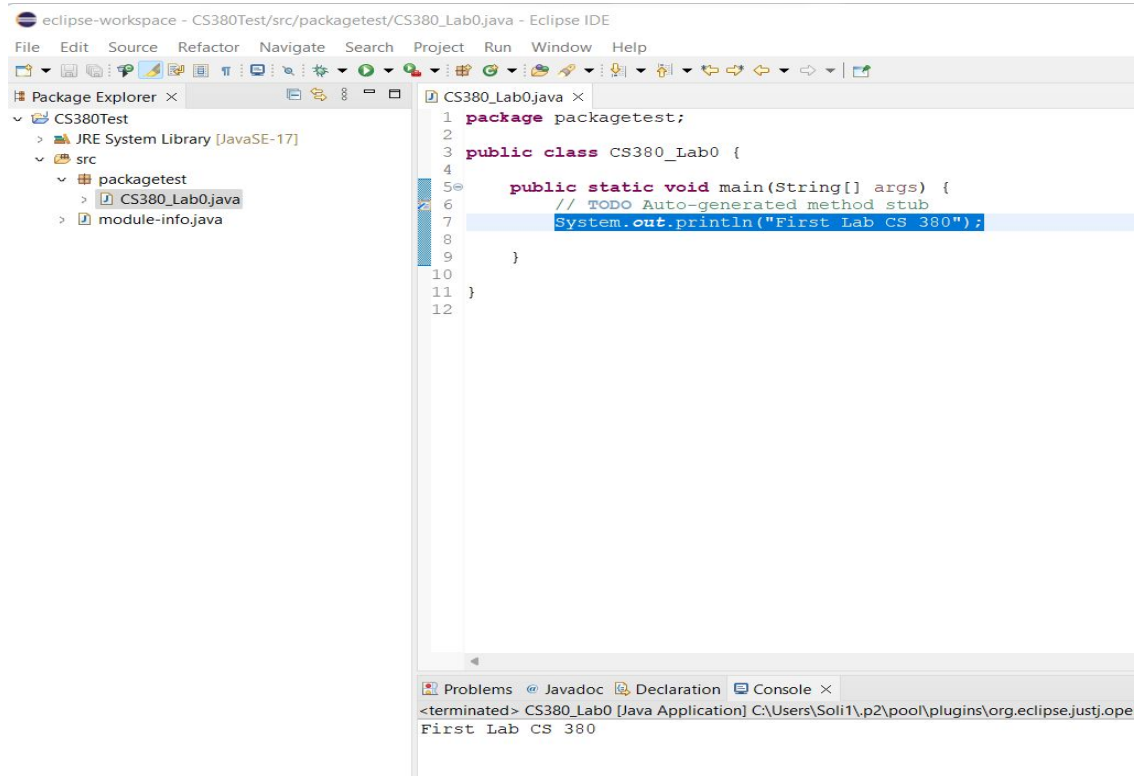
Add this line code:

```
System.out.println("First Lab CS 380");
```

Step 17) Click on **Run** button



Output will be displayed as shown below



The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows the project structure: CS380Test > JRE System Library [JavaSE-17] > src > packetest > CS380_Lab0.java. The main editor window shows the source code of CS380_Lab0.java, which includes a package declaration, a class declaration, and a main method with a single println statement. The console at the bottom shows the output of the program.

```
eclipse-workspace - CS380Test/src/packetest/CS380_Lab0.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer x
  CS380Test
    JRE System Library [JavaSE-17]
    src
      packetest
        CS380_Lab0.java
        module-info.java
CS380_Lab0.java x
1 package packetest;
2
3 public class CS380_Lab0 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         System.out.println("First Lab CS 380");
8
9     }
10
11 }
12
Problems @ Javadoc Declaration Console x
<terminated> CS380_Lab0 [Java Application] C:\Users\Soli1\p2\pool\plugins\org.eclipse.justj.ope
First Lab CS 380
```

Exercise 1:

Reverse a Number in Java

Glven Number

12345

Reverse Number

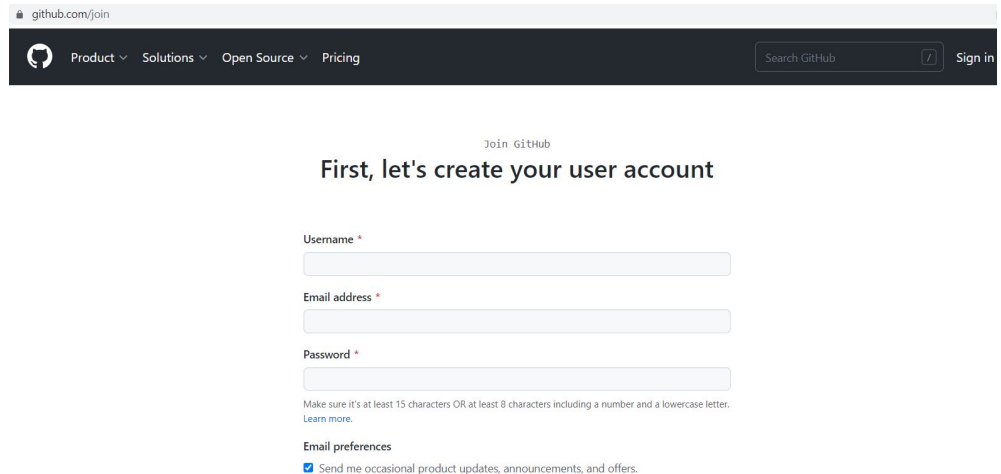
54321

To send your exercise create a github account

What Is the Process for Uploading Eclipse Projects to GitHub?

First things first, download git from <http://git-scm.com/>.
Then go to <http://github.com/> and create an account and repository.

<https://github.com/join>

A screenshot of the GitHub website's sign-up page. The browser address bar shows 'github.com/join'. The navigation bar includes links for Product, Solutions, Open Source, and Pricing, along with a search bar and a 'Sign in' button. The main heading is 'Join GitHub' followed by 'First, let's create your user account'. There are three input fields for 'Username', 'Email address', and 'Password', each with a red asterisk indicating it is required. Below the password field, there is a note: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.' with a 'Learn more' link. At the bottom, there is an 'Email preferences' section with a checked checkbox and the text 'Send me occasional product updates, announcements, and offers.'

What Is the Process for Uploading Eclipse Projects to GitHub?

On your machine, first you will need to navigate to the project folder using git bash.

When you get there you do:

```
git init
```

which initiates a new git repository in that directory.

When you've done that, you need to register that new repo with a remote (where you'll upload -- push -- your files to), which in this case will be github. This assumes you have already created a github repository. You'll get the correct URL from your repo in GitHub.

```
git remote add origin https://github.com/[username]/[reponame].git
```


What Is the Process for Uploading Eclipse Projects to GitHub?

You need to add you existing files to your local commit:

```
git add .    # this adds all the files
```

Then you need to make an initial commit, so you do:

```
git commit -a -m "Initial commit" # this stages your files locally for commit.  
                                   # they haven't actually been pushed yet
```

Now you've created a commit in your local repo, but not in the remote one. To put it on the remote, you do the second line you posted:

```
git push -u origin --all
```