School of Computer Science Engineering and Technology

Course- BTech	Type- Core
Course Code- CSET205	Course Name-Software Engineering
Year-2022	Semester- Odd
Date-02/09/2022	Batch- 2021-2025

Lab Assignment No. 4 - Eclipse and Git Branch + Merge

CO Mapping

Exp. No.	Name	CO1	CO2	CO3
4	Working with the Eclipse IDE and GitHub	√	√	

So far, we understood that professional software developers work with IDE and a sub-version system like GitHub to push their code in the central repository. It is always challenging to push your changes in the main/**master** branch as it may become unstable due to bugs in the recent commit/changes. Hence, to avoid this hassle, industries use the concept of branch and merge.

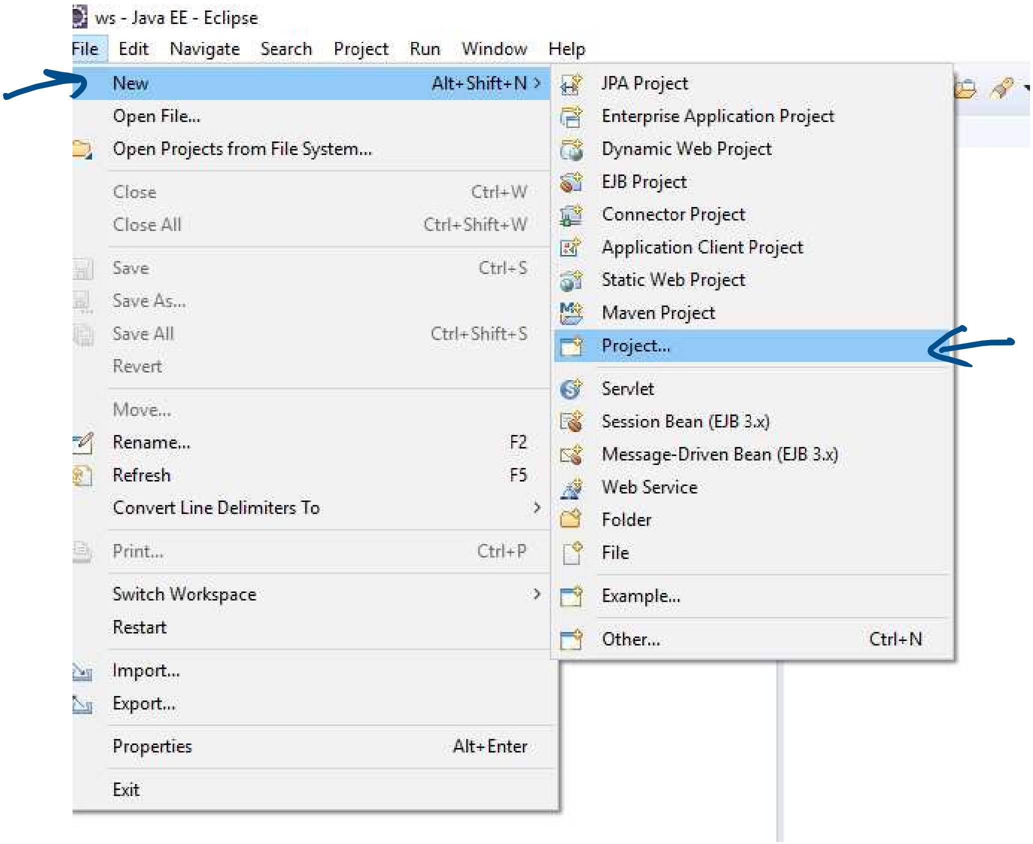
> **Branch** – take out a copy from the main branch to incorporate your changes.
>
> **Merge** – once the changes are complete and tested, the branch is merged back to the parent.

In this lab, you will be exposed to how IDE, when integrated with a sub-version system like Git, enables you to use branch and merge techniques. You have to develop two sorting techniques (one is in the master branch and one is in the develop branch). After testing the first technique, push it to the repository as the master branch (i.e., the stable version). Next, create a new branch and add one more sorting algorithm. Test (to check if it is working as intended) and merge it back to the master branch.

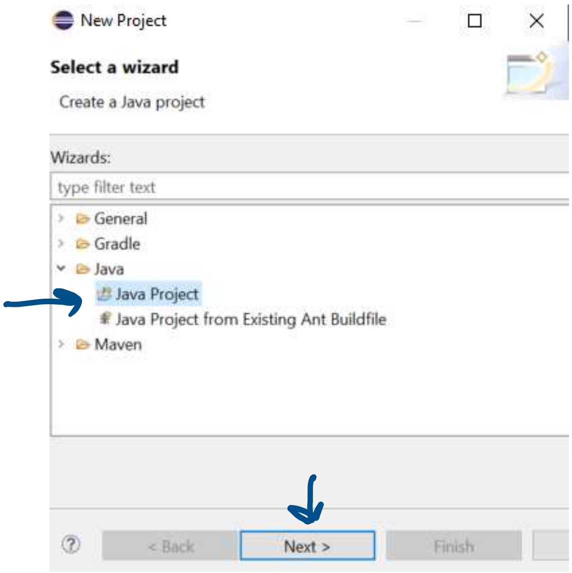
Follow the steps to achieve the above objective:

1. Open Eclipse IDE.

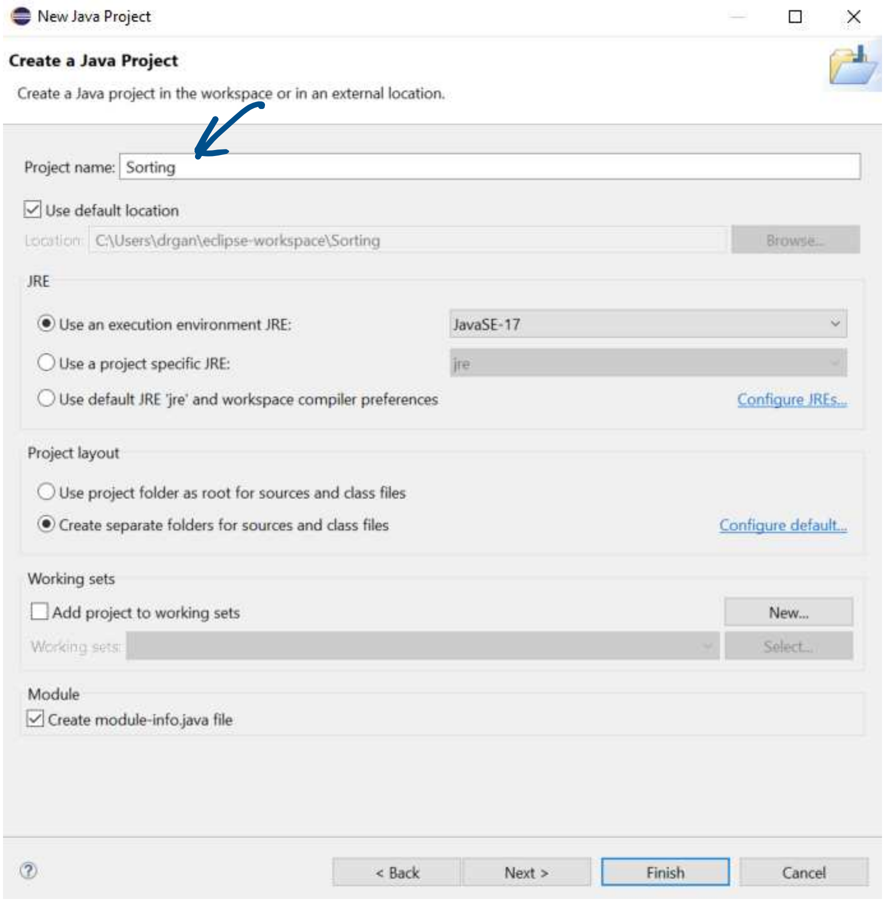
2. Click on file → **New project** option.



3. Select **Java Project**.



4. Give your project name as **Sorting** and click **Finish** button.



New Java Project

Create a Java Project

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

Project name: Sorting

☑ Use default location

Location: C:\Users\drgan\eclipse-workspace\Sorting

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

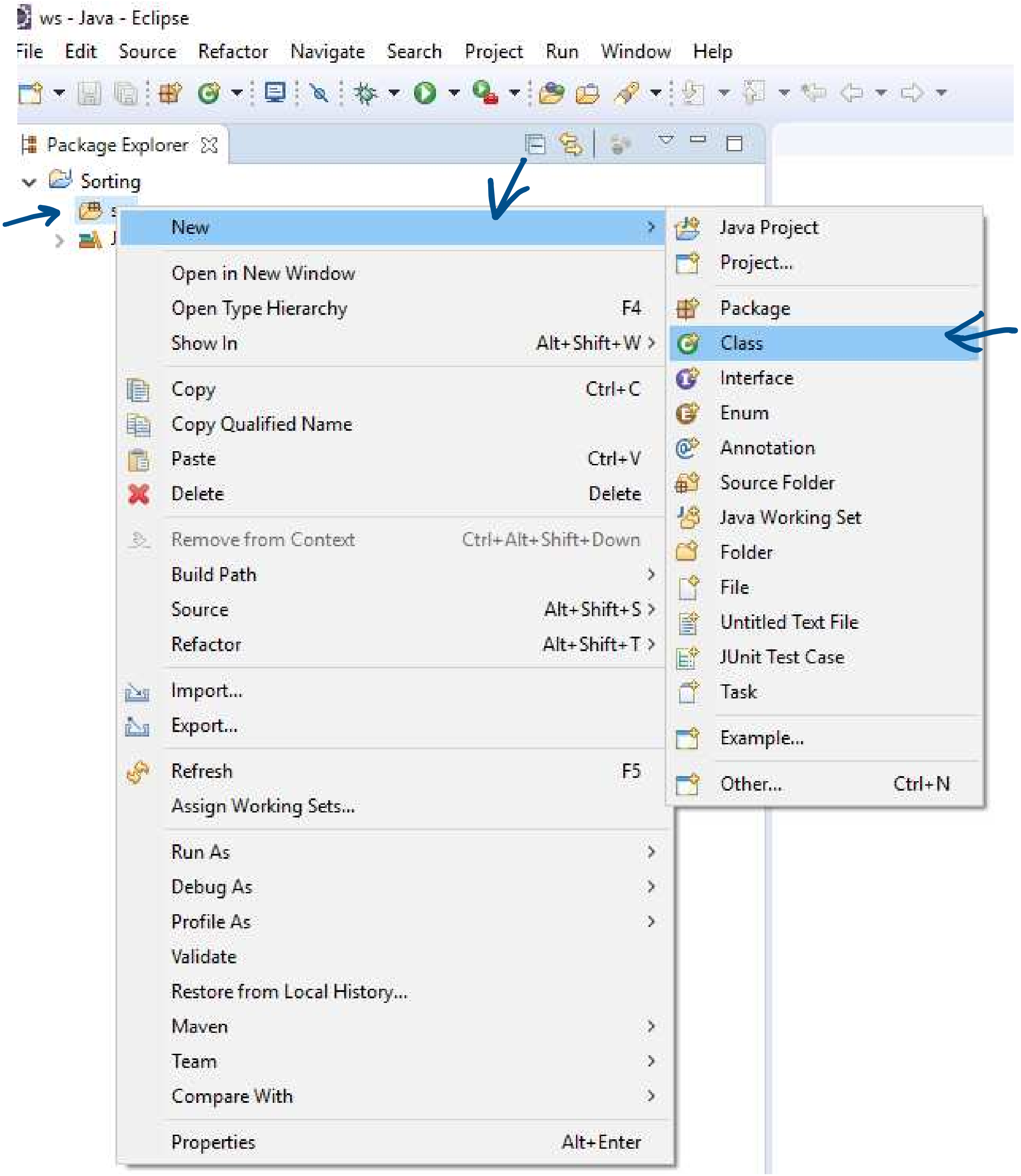
Working sets

□ Add project to working sets

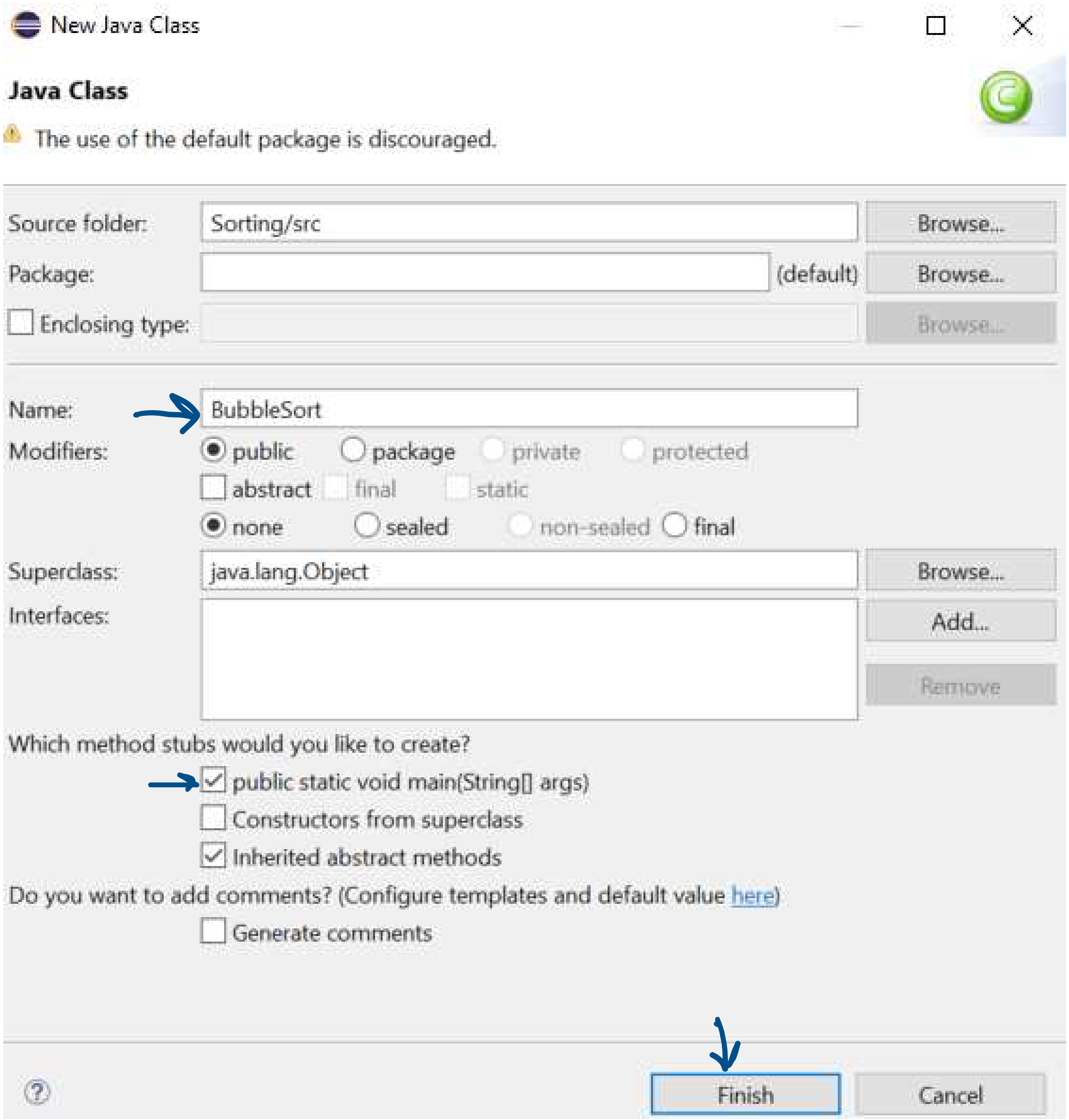
Module

☑ Create module-info.java file

5. Now, right click **src** under the project created by you → select **class** (i.e. Java class)



6. Give your Java class name as **BubbleSort** and make sure to check **void main box** → click **finish** button.



New Java Class

Java Class

The use of the default package is discouraged.

Source folder: Sorting/src

Package: (default)

□ Enclosing type:

Name: BubbleSort

Modifiers: ● public ○ package ○ private ○ protected

□ abstract □ final □ static

● none ○ sealed ○ non-sealed ○ final

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☑ public static void main(String[] args)

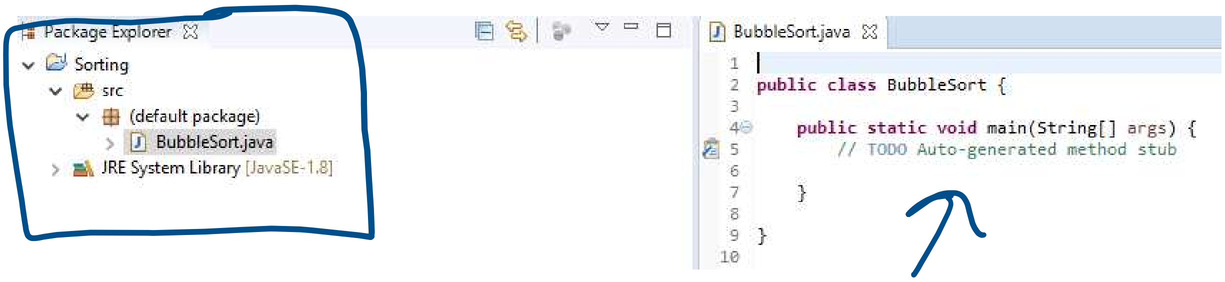
□ Constructors from superclass

☑ Inherited abstract methods

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

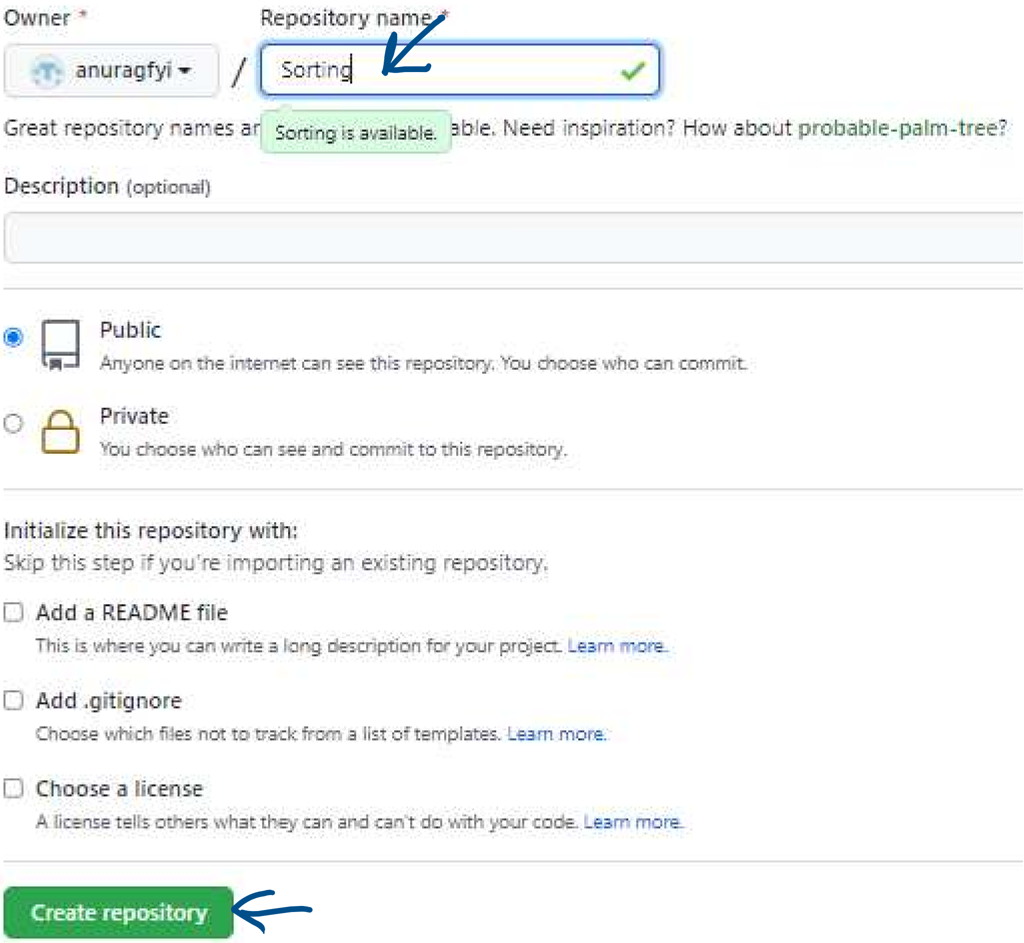
□ Generate comments

This is how it will look like.



```

public class BubbleSort {

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

	}

}

```

7. Go to your GitHub account and **create a new repository** with **same name as of project** (i.e., Sorting).



Owner * / Repository name *

anuragfyi / Sorting ✓

Great repository names ar Sorting is available. able. Need inspiration? How about probable-palm-tree?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

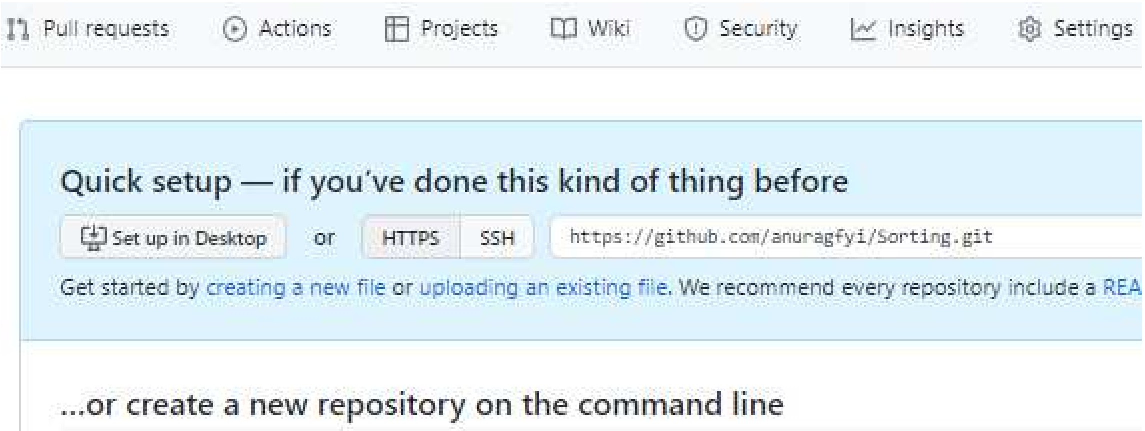
Initialize this repository with:
Skip this step if you're importing an existing repository.

□ Add a README file
This is where you can write a long description for your project. Learn more.

□ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

□ Choose a license
A license tells others what they can and can't do with your code. Learn more.

Screen shot after creating new repository. You can see the link of your repo as well. **Copy the link in a text file.**



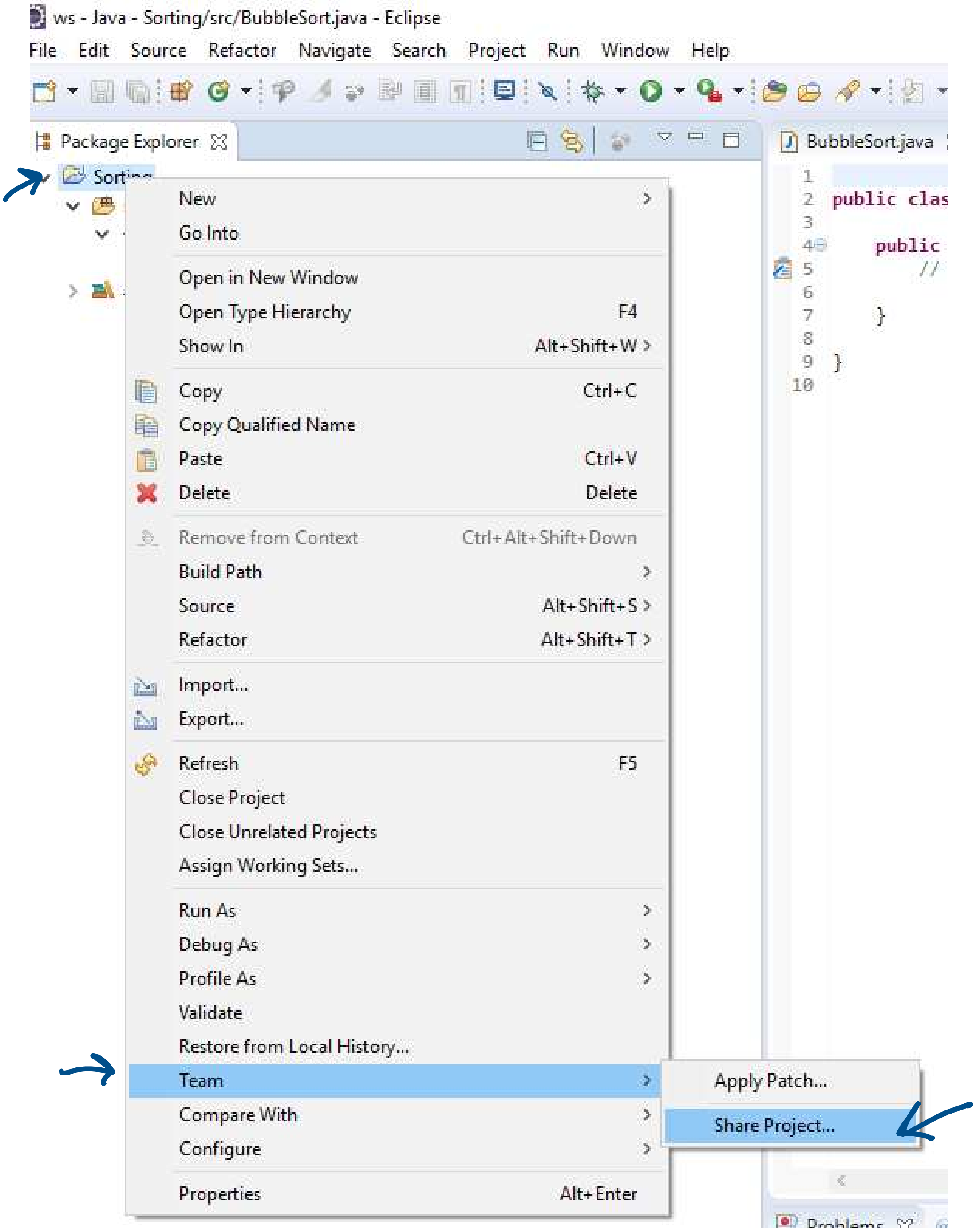
Quick setup — if you've done this kind of thing before

https://github.com/anuragfyi/Sorting.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a REA

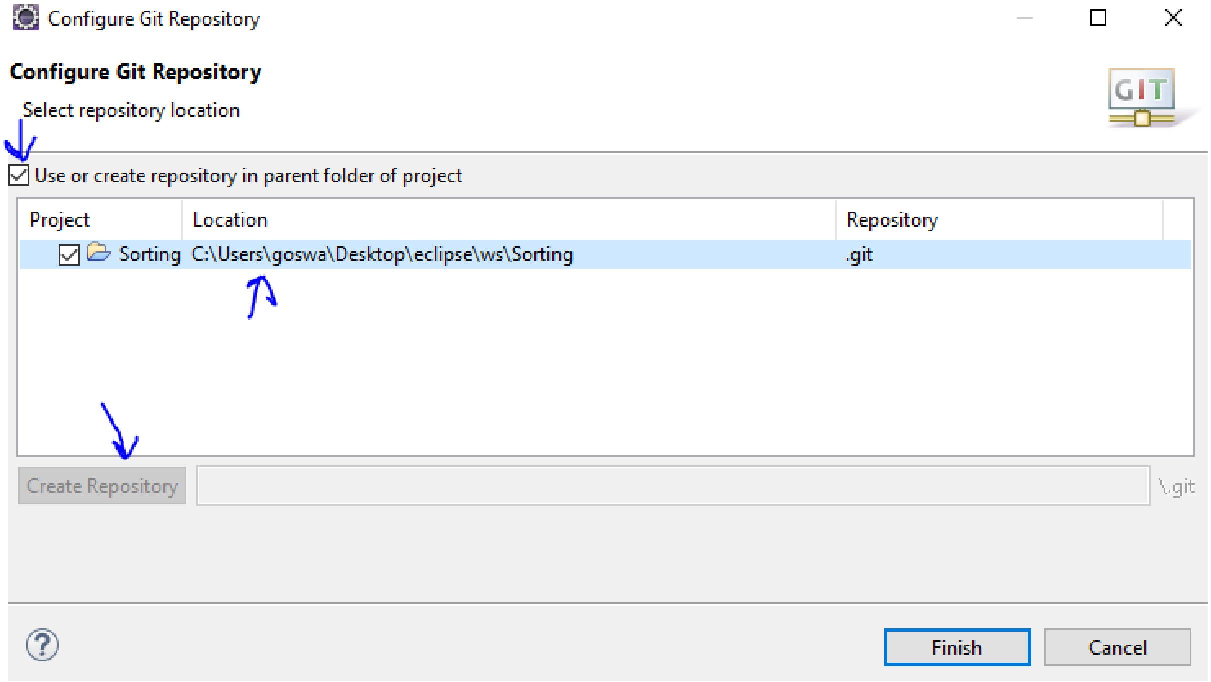
...or create a new repository on the command line

8. Right click on your project in eclipse and go to **Team** option → click on **share project**.



9. It will ask for configuring Git Repository. Make sure to **check User or Create** check box, select the path, and click on **create repository button**.

   Note: Eclipse may ask you for your GitHub credentials here.

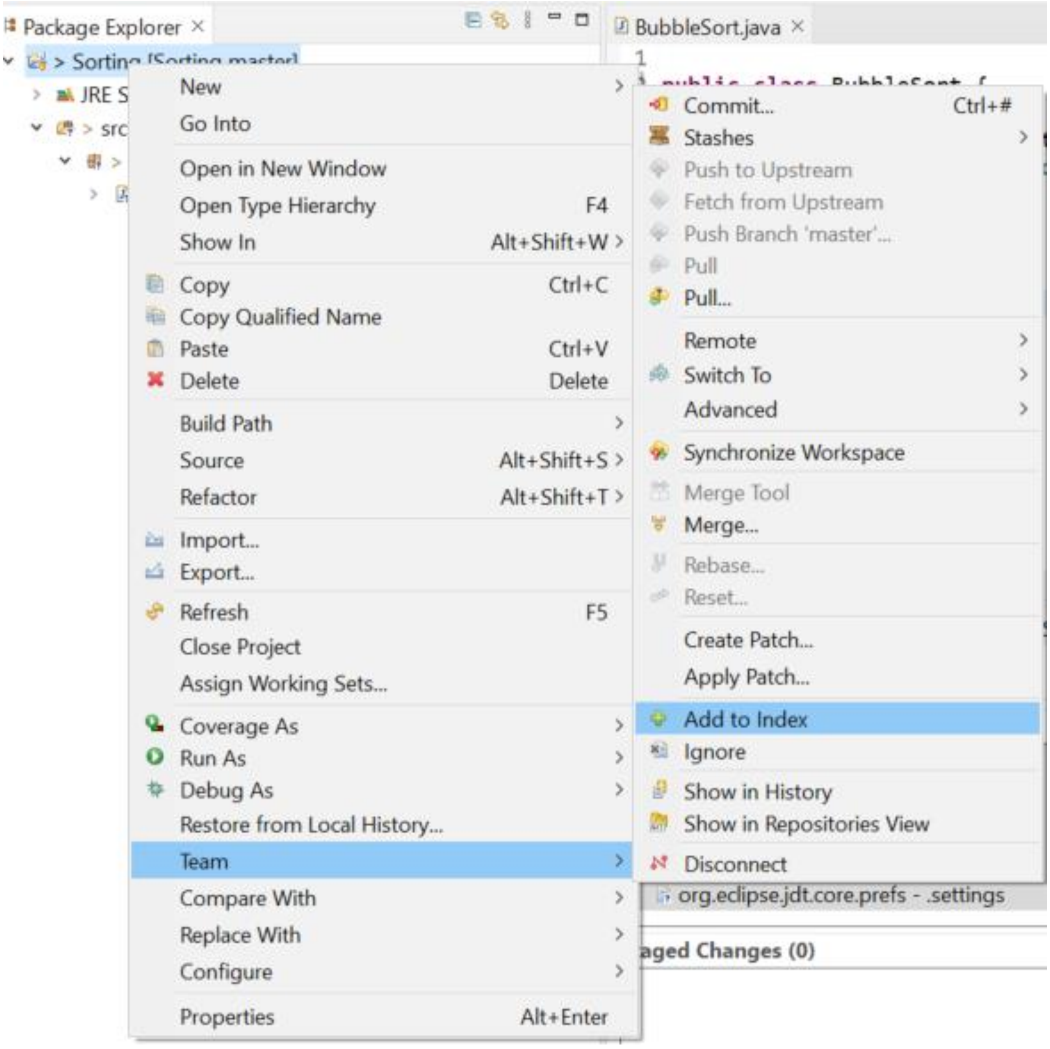


☑ Use or create repository in parent folder of project

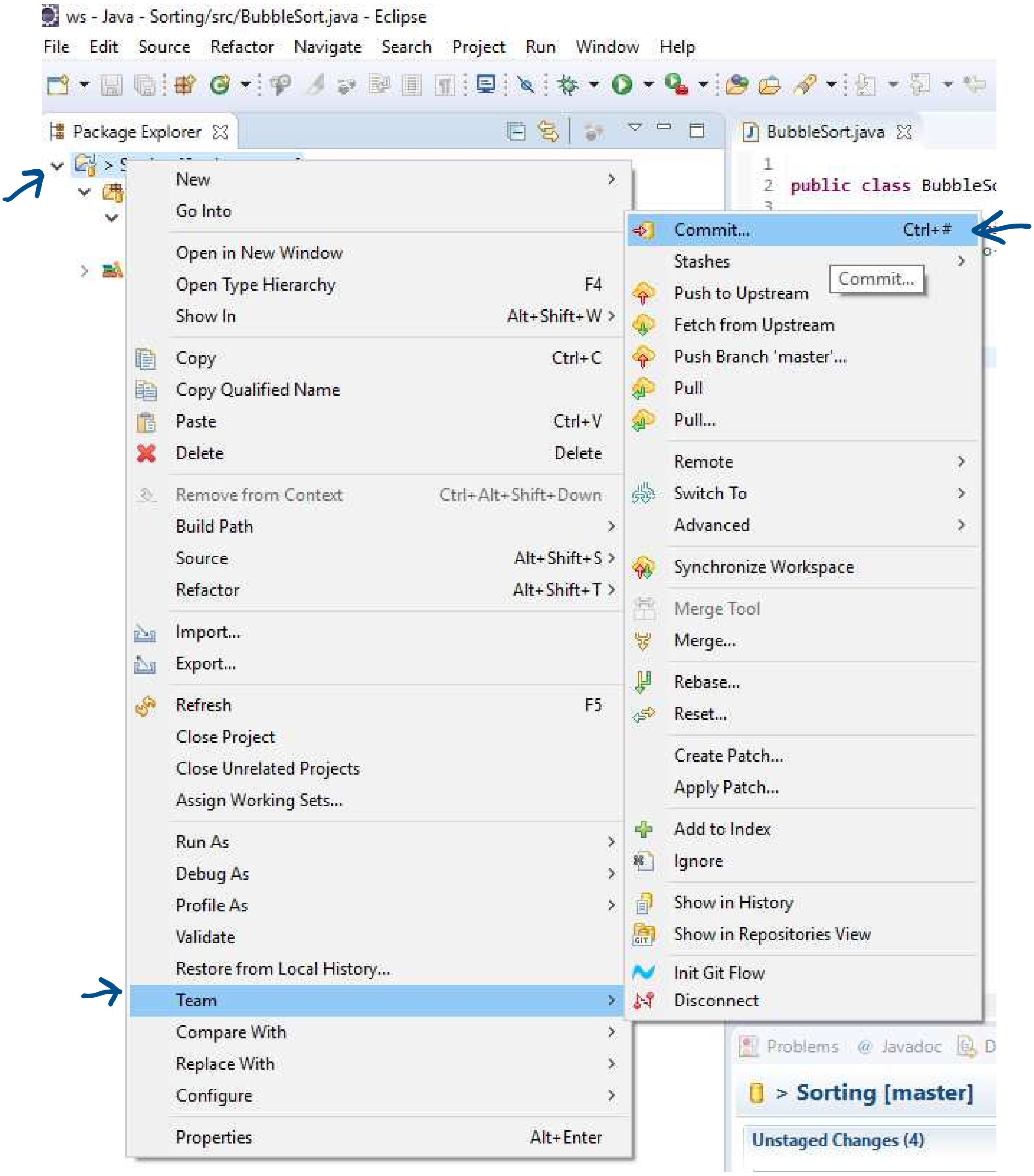
Project	Location	Repository
☑ Sorting	C:\Users\goswa\Desktop\eclipse\ws\Sorting	.git

10. Create your bubble sort program and execute/run it.

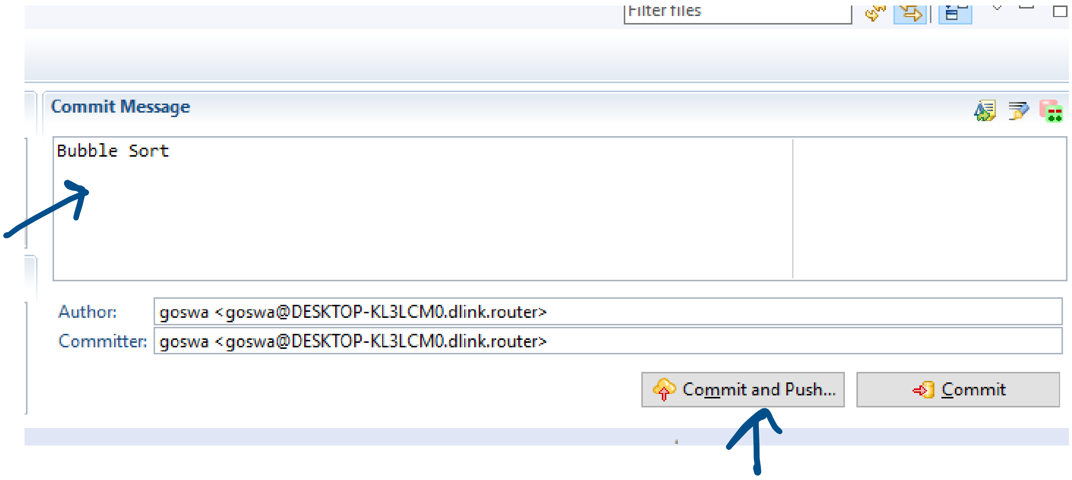
11. Right click on your project → go to **team** option → click on **Add to Index.** This will mark the files to be committed to the repository.



12. Right click on your project → go to **team** option → select **commit**.



13. Type a commit message (think about the concept of writing good commit messages) → click on **commit and push** button.



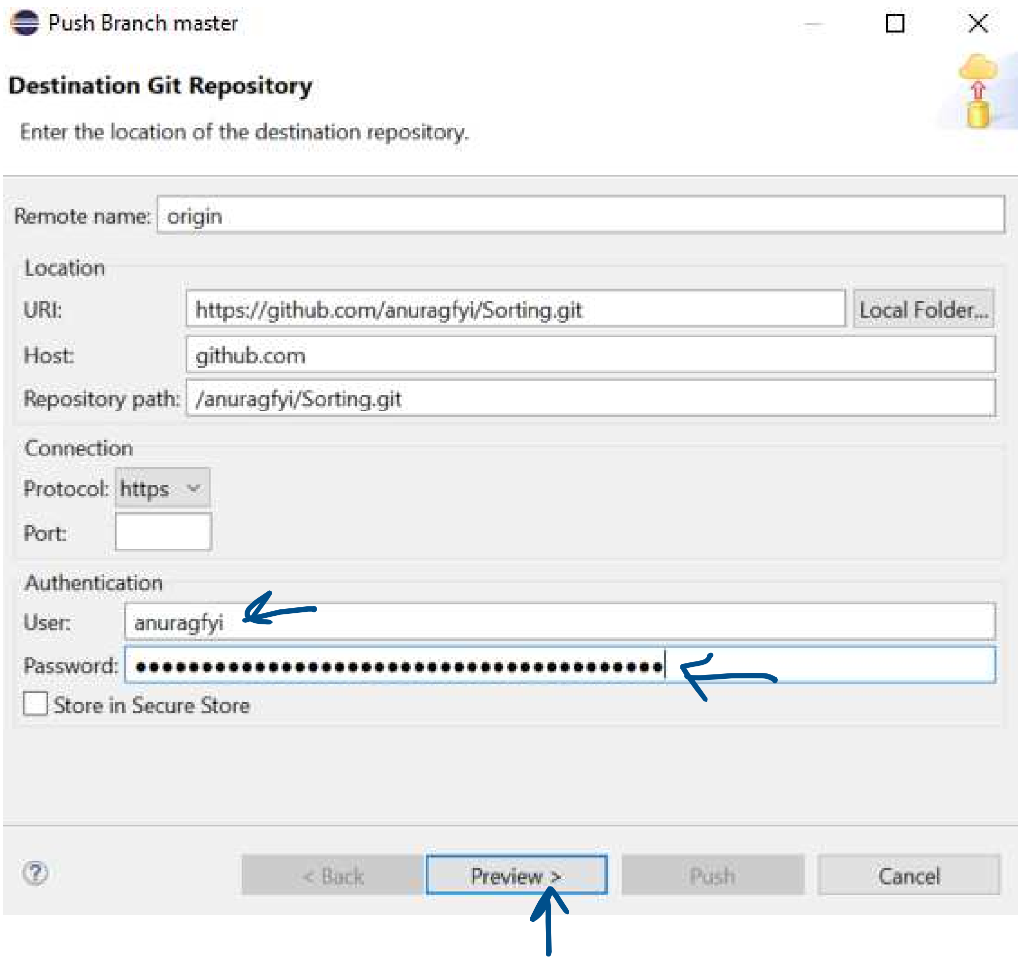
Commit Message

Bubble Sort

Author: goswa <goswa@DESKTOP-KL3LCM0.dlink.router>

Committer: goswa <goswa@DESKTOP-KL3LCM0.dlink.router>

14. Eclipse will ask for your GitHub credentials → enter details [**use token for password – check previous lab for steps]** → Click **Preview button**.



Push Branch master

Destination Git Repository

Enter the location of the destination repository.

Remote name: origin

Location

URI: https://github.com/anuragfyi/Sorting.git

Host: github.com

Repository path: /anuragfyi/Sorting.git

Connection

Protocol: https

Port:

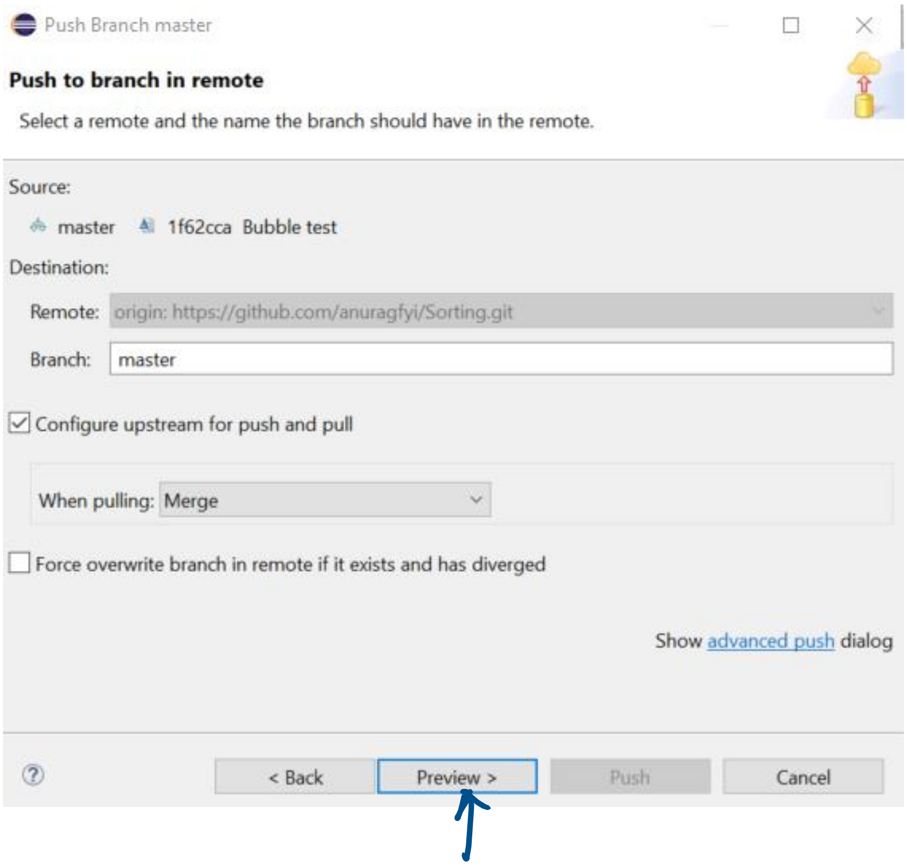
Authentication

User: anuragfyi

Password: ●●

□ Store in Secure Store

15. Click **Preview button** → click **Push button**



Push Branch master

Push to branch in remote

Select a remote and the name the branch should have in the remote.

Source:

master 1f62cca Bubble test

Destination:

Remote: origin: https://github.com/anuragfyi/Sorting.git

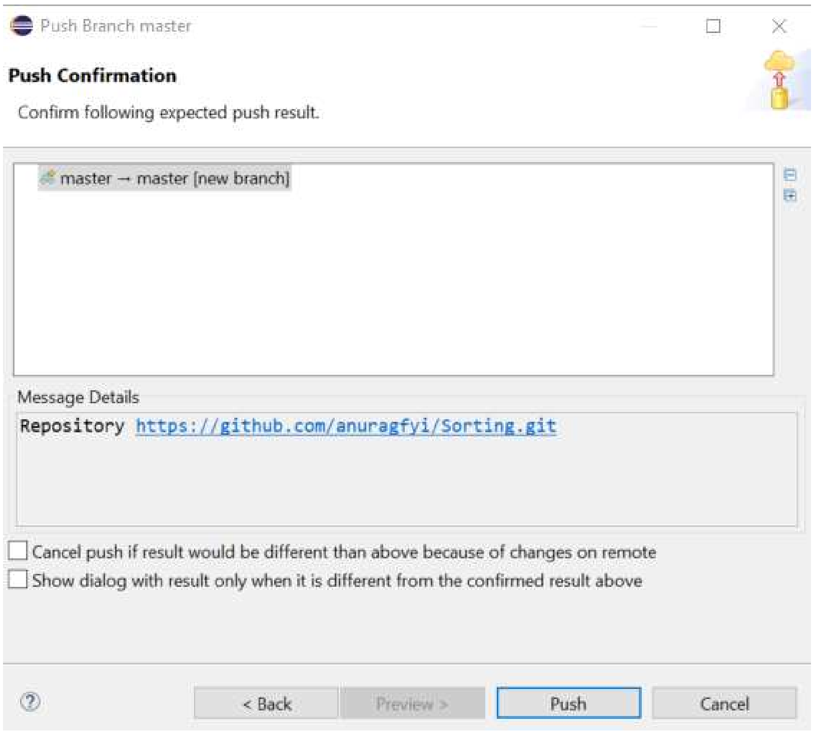
Branch: master

☑ Configure upstream for push and pull

When pulling: Merge

□ Force overwrite branch in remote if it exists and has diverged

Show advanced push dialog



Push Branch master

Push Confirmation

Confirm following expected push result.

master → master [new branch]

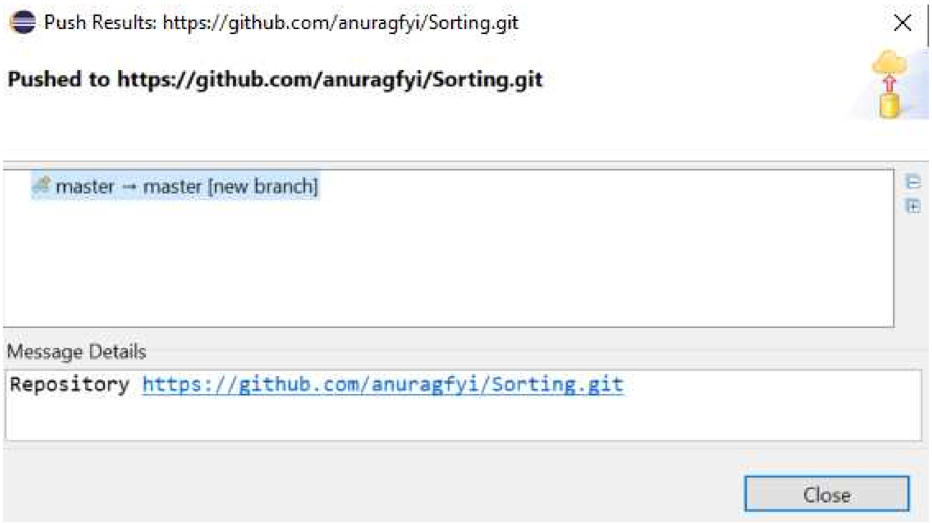
Message Details

Repository https://github.com/anuragfyi/Sorting.git

□ Cancel push if result would be different than above because of changes on remote

□ Show dialog with result only when it is different from the confirmed result above

16. Your project will be pushed in the Git repository.



Push Results: https://github.com/anuragfyi/Sorting.git

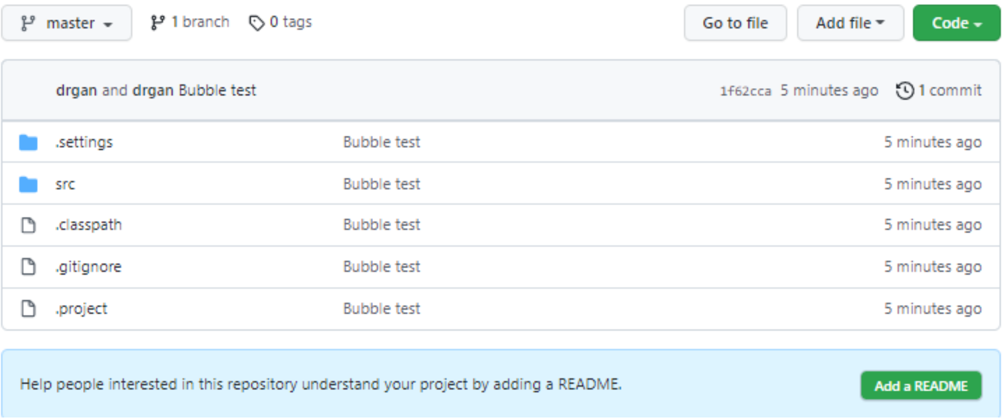
Pushed to https://github.com/anuragfyi/Sorting.git

master → master [new branch]

Message Details

```
Repository https://github.com/anuragfyi/Sorting.git
```

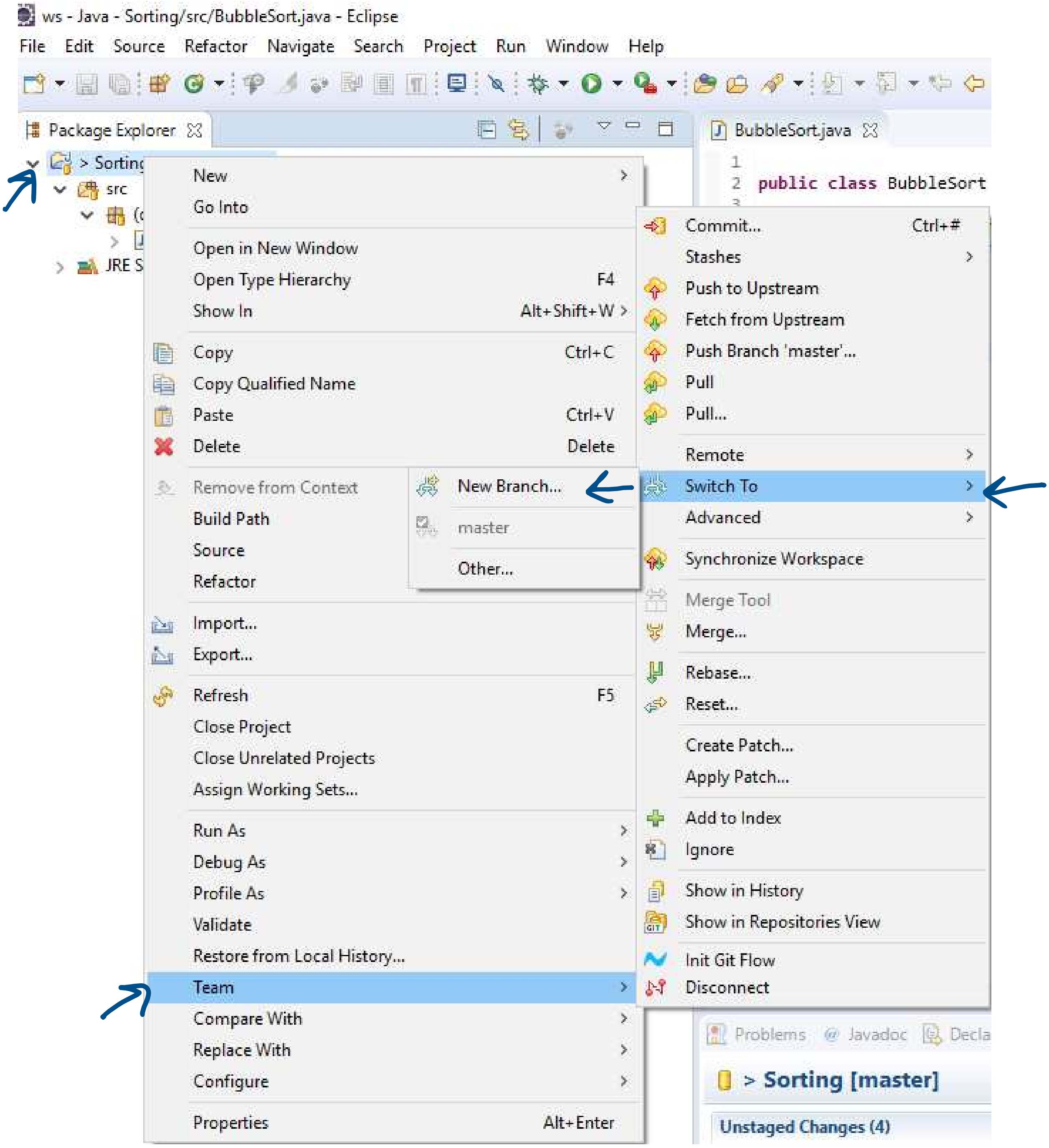
17. Refresh your Git repository (i.e., sorting) and look for BubbleSort.java file in your repository under **src** folder.



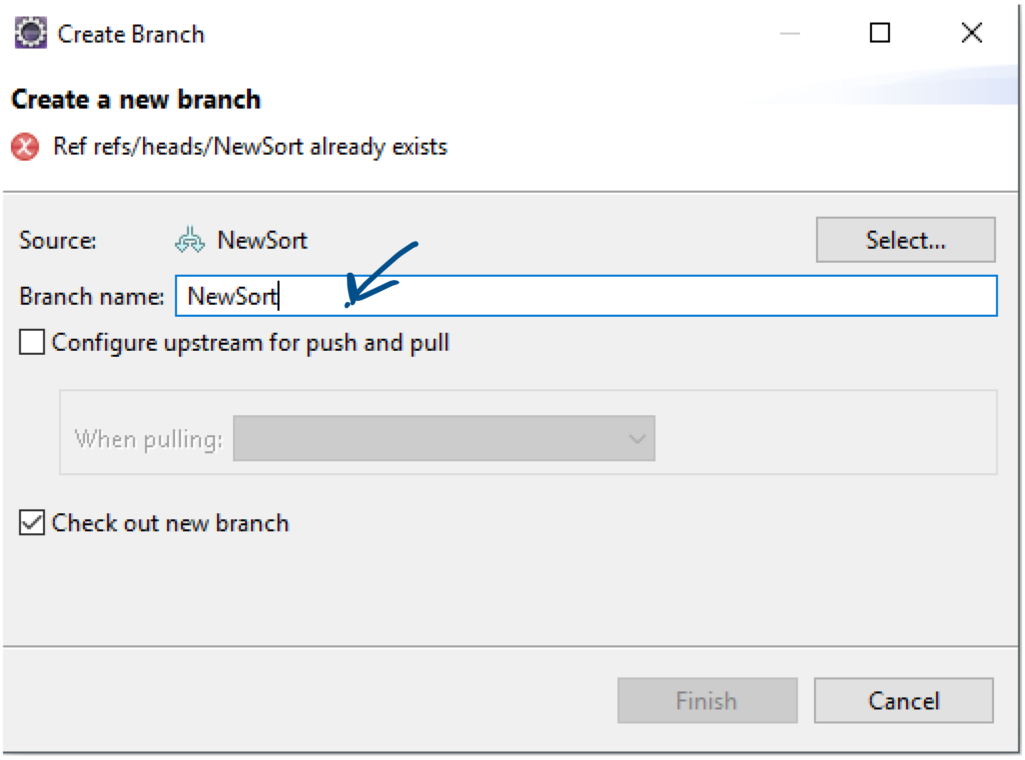
drgan and drgan Bubble test		1f62cca 5 minutes ago 1 commit
.settings	Bubble test	5 minutes ago
src	Bubble test	5 minutes ago
.classpath	Bubble test	5 minutes ago
.gitignore	Bubble test	5 minutes ago
.project	Bubble test	5 minutes ago

Help people interested in this repository understand your project by adding a README.

18. To create a new branch in Eclipse IDE, right click on project → go to **team** option → **switch to** → click on **new branch**.



19. Enter name of your new branch and click on finish button.



Create a new branch

Ref refs/heads/NewSort already exists

Source: NewSort

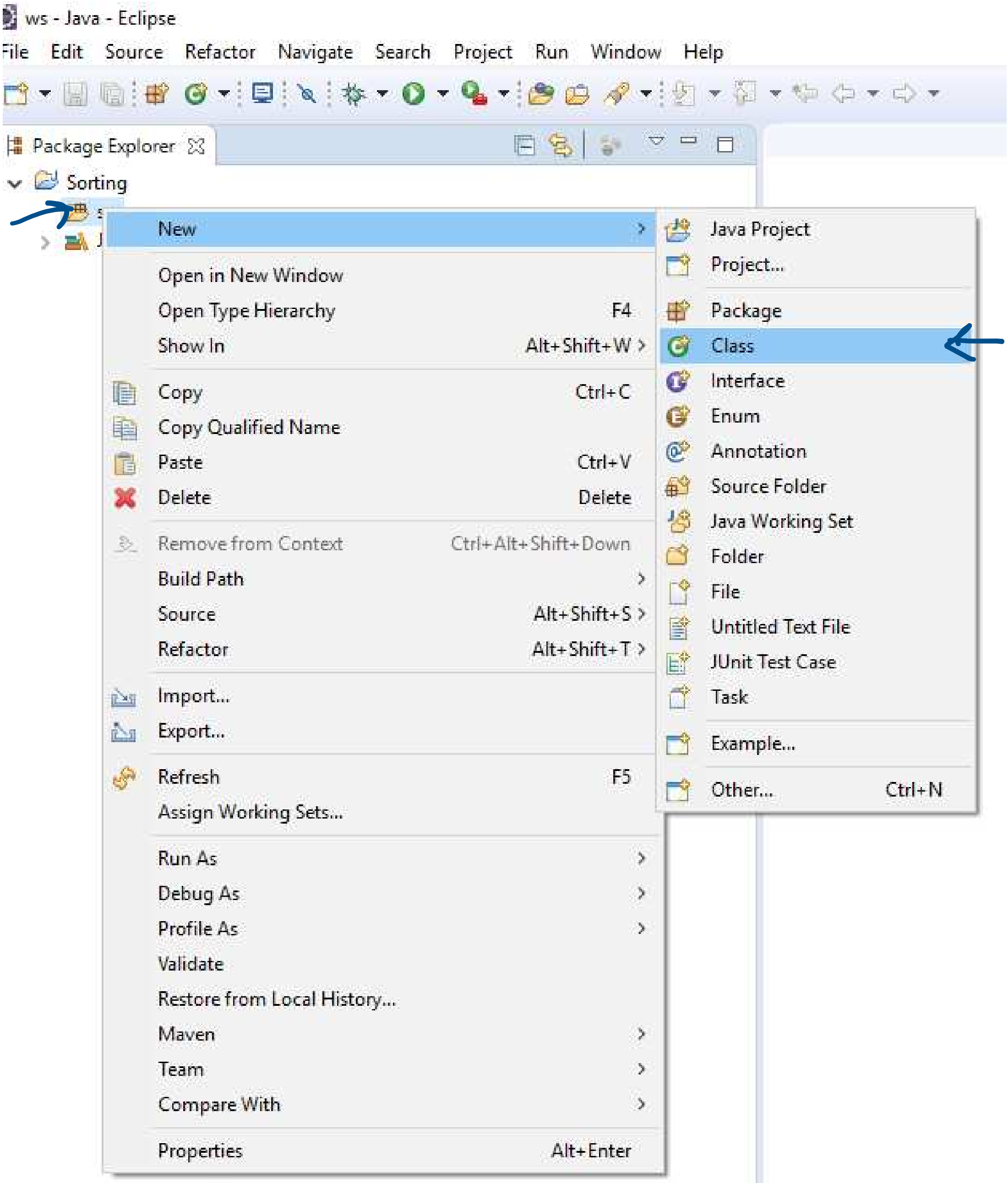
Branch name: NewSort

☐ Configure upstream for push and pull

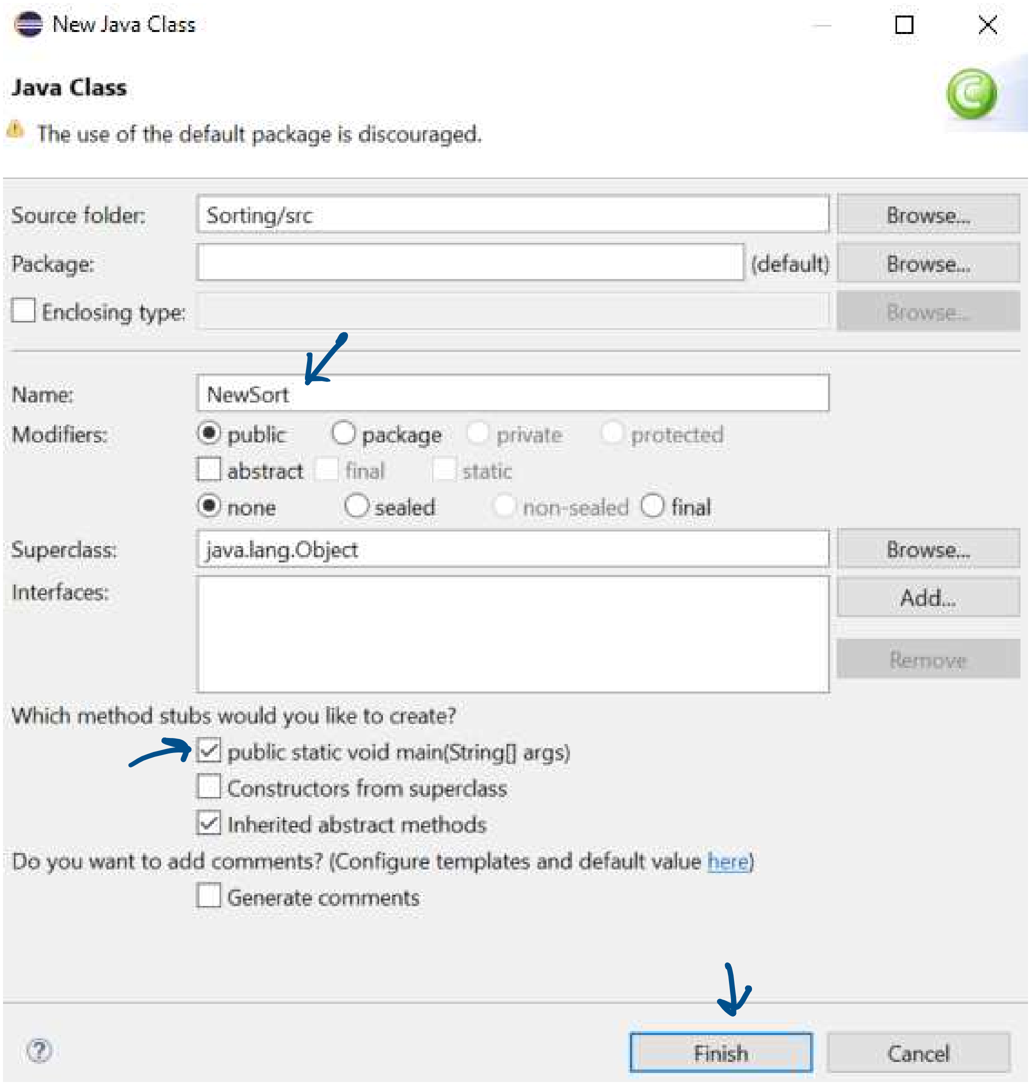
When pulling:

☑ Check out new branch

20. Now, right click **src** under your project → select **class** (i.e. Java class).



21. Type the name of the class that you prefer (I used NewSort) and make sure to check **void main** box → click **finish** button.



New Java Class

Java Class

The use of the default package is discouraged.

Source folder: Sorting/src

Package: (default)

□ Enclosing type:

Name: NewSort

Modifiers: ● public ○ package ○ private ○ protected

□ abstract □ final □ static

● none ○ sealed ○ non-sealed ○ final

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☑ public static void main(String[] args)

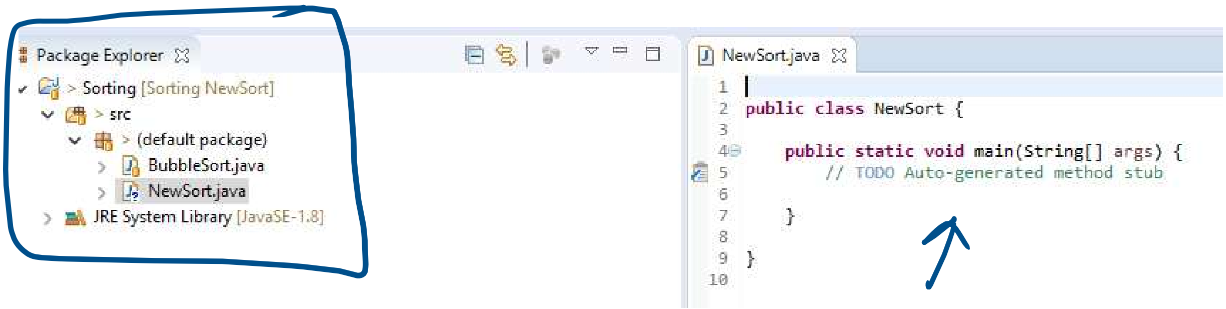
□ Constructors from superclass

☑ Inherited abstract methods

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

□ Generate comments

This is how it looks like.



Package Explorer

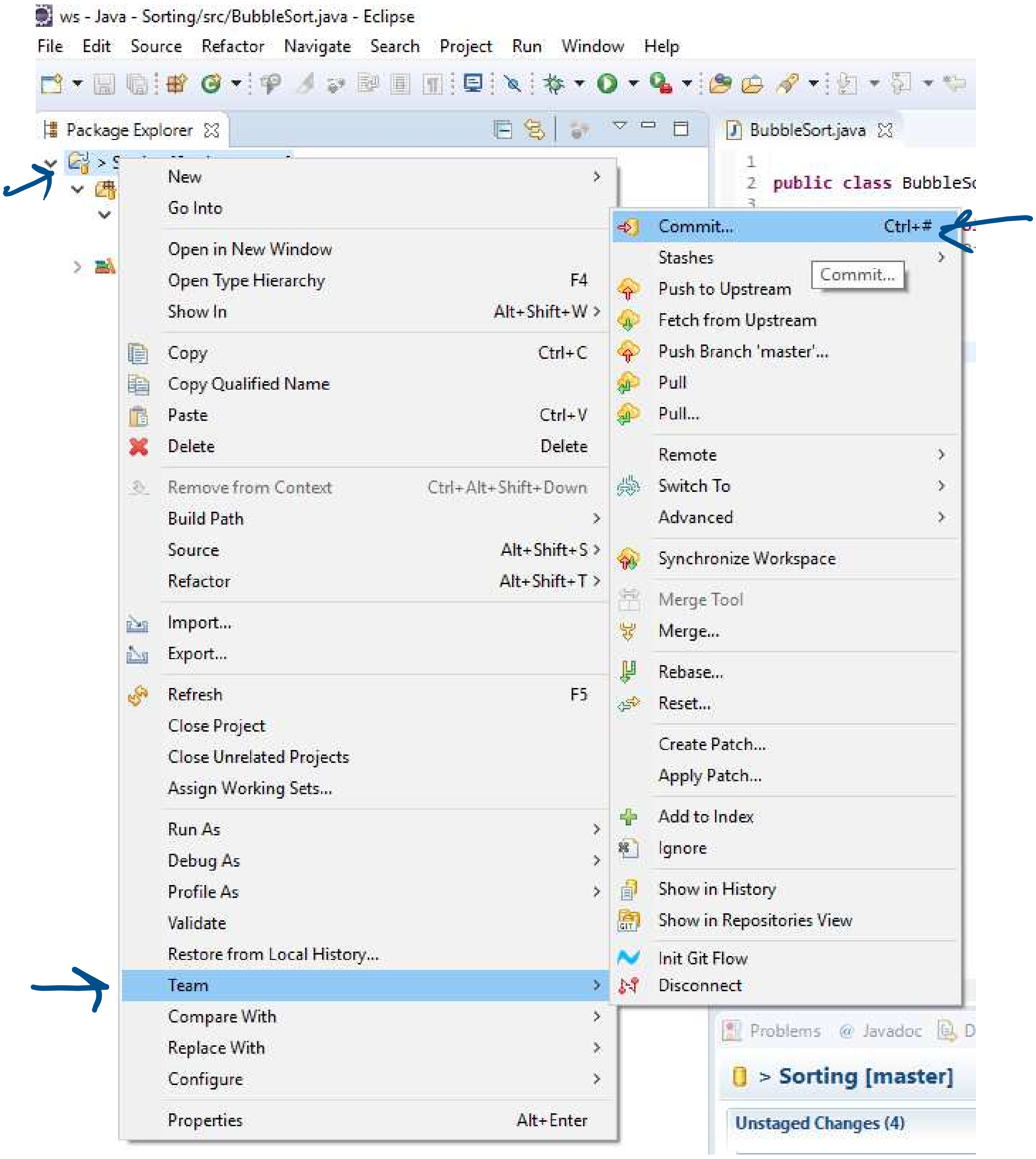
- Sorting [Sorting NewSort]
  - src
    - (default package)
      - BubbleSort.java
      - NewSort.java
  - JRE System Library [JavaSE-1.8]

NewSort.java

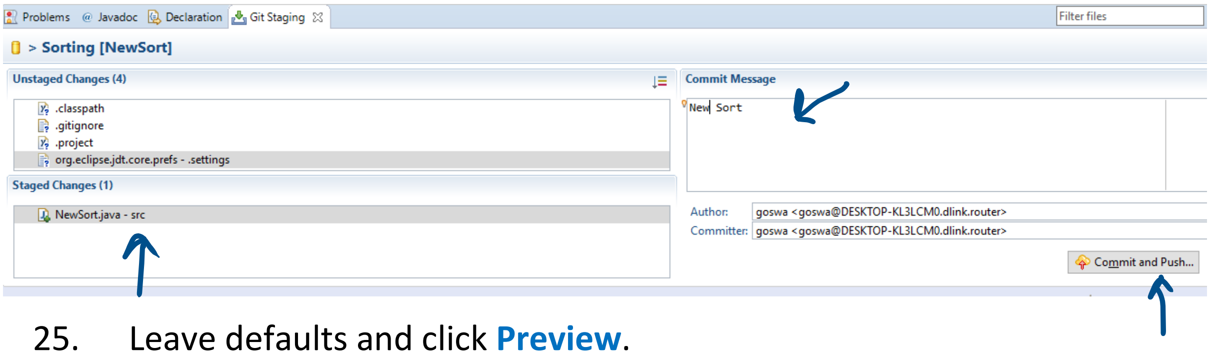
```java

public class NewSort {

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

    }

}

```

22. Create your new sorting program → **save** → **run**.

23. Right click on your project → go to **team** option → select **commit**.



24. Type commit message → drag new file to **Staged changes** box → click **commit and push**.



> Sorting [NewSort]

Unstaged Changes (4)

- .classpath
- .gitignore
- .project
- org.eclipse.jdt.core.prefs - .settings

Staged Changes (1)

- NewSort.java - src

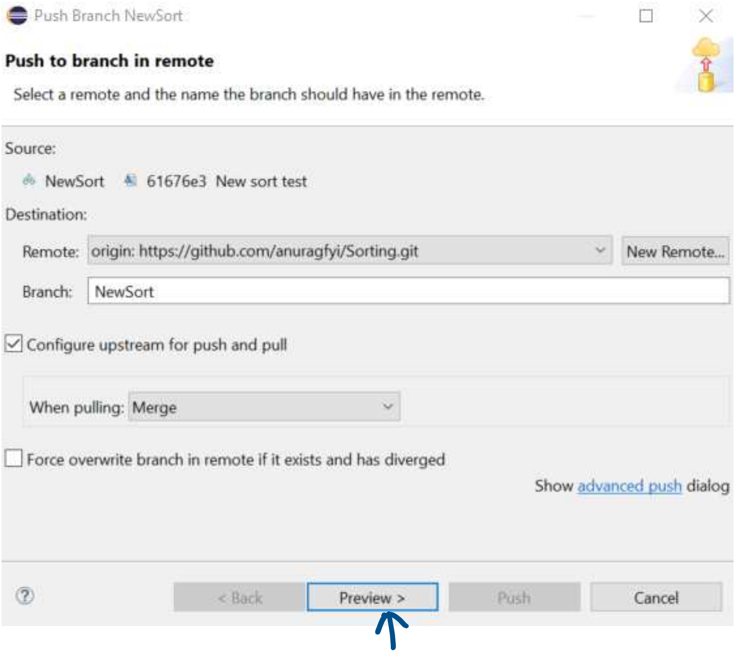
Commit Message

```
New Sort
```

Author: goswa <goswa@DESKTOP-KL3LCM0.dlink.router>

Committer: goswa <goswa@DESKTOP-KL3LCM0.dlink.router>

25. Leave defaults and click **Preview**.



Push to branch in remote

Select a remote and the name the branch should have in the remote.

Source:

NewSort 61676e3 New sort test

Destination:

Remote: origin: https://github.com/anuragfyi/Sorting.git

Branch: NewSort

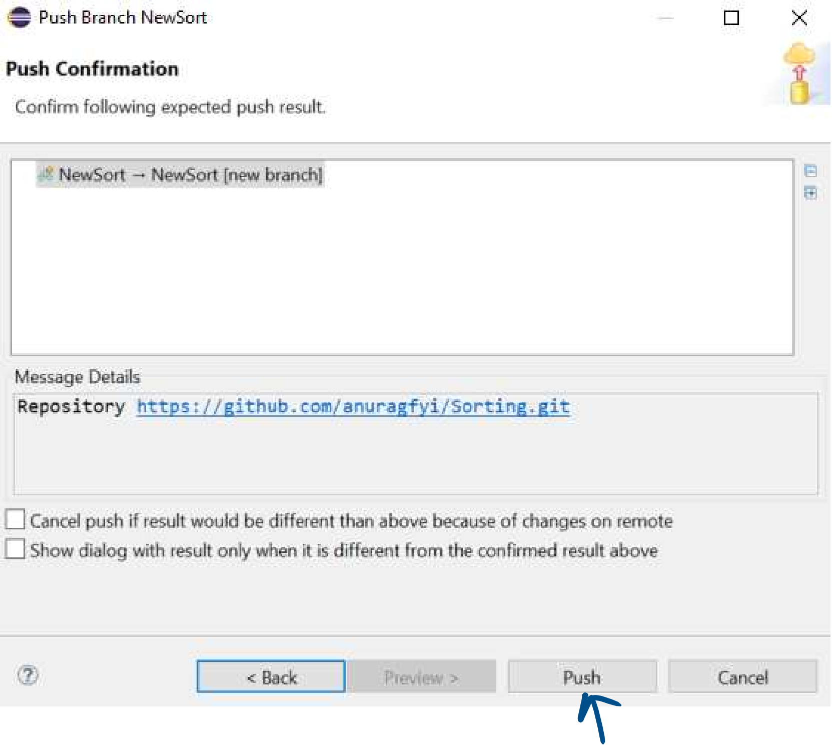
☑ Configure upstream for push and pull

When pulling: Merge

☐ Force overwrite branch in remote if it exists and has diverged

Show advanced push dialog

26. Click **Push**. It will push your file to repository in the new branch.



Push Branch NewSort

Push Confirmation

Confirm following expected push result.

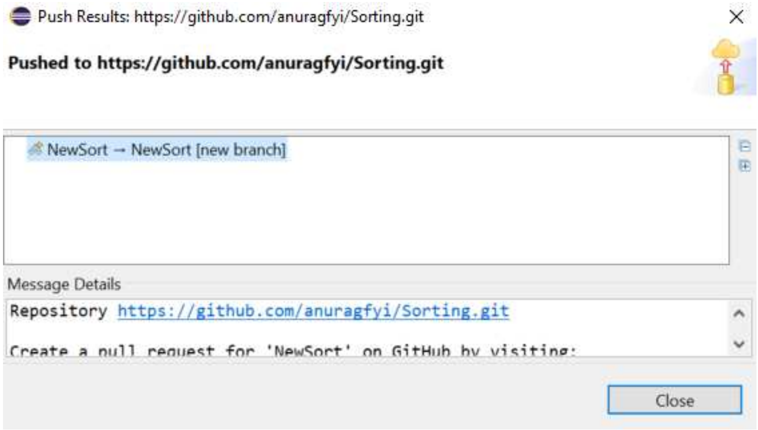
NewSort → NewSort [new branch]

Message Details

```
Repository https://github.com/anuragfyi/Sorting.git
```

□ Cancel push if result would be different than above because of changes on remote

□ Show dialog with result only when it is different from the confirmed result above



Push Results: https://github.com/anuragfyi/Sorting.git

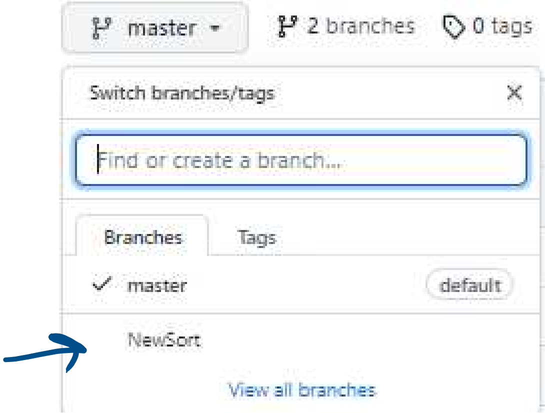
Pushed to https://github.com/anuragfyi/Sorting.git

NewSort → NewSort [new branch]

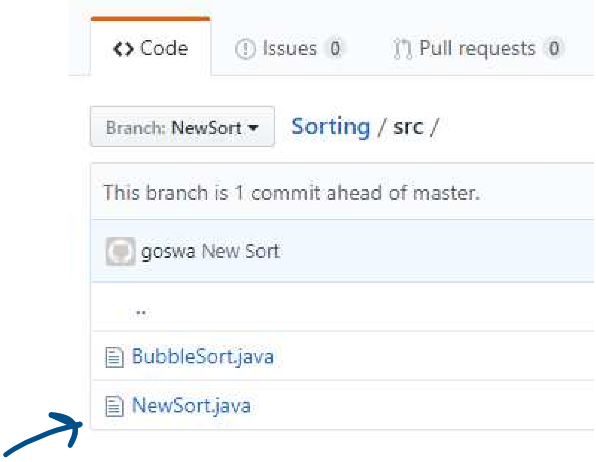
Message Details

```
Repository https://github.com/anuragfyi/Sorting.git

Create a pull request for 'NewSort' on GitHub by visiting:
```

27. **Refresh** your repository and **select the new branch** (in my case it was NewSort). ***Take a snapshot of the screen where it shows two branches.***



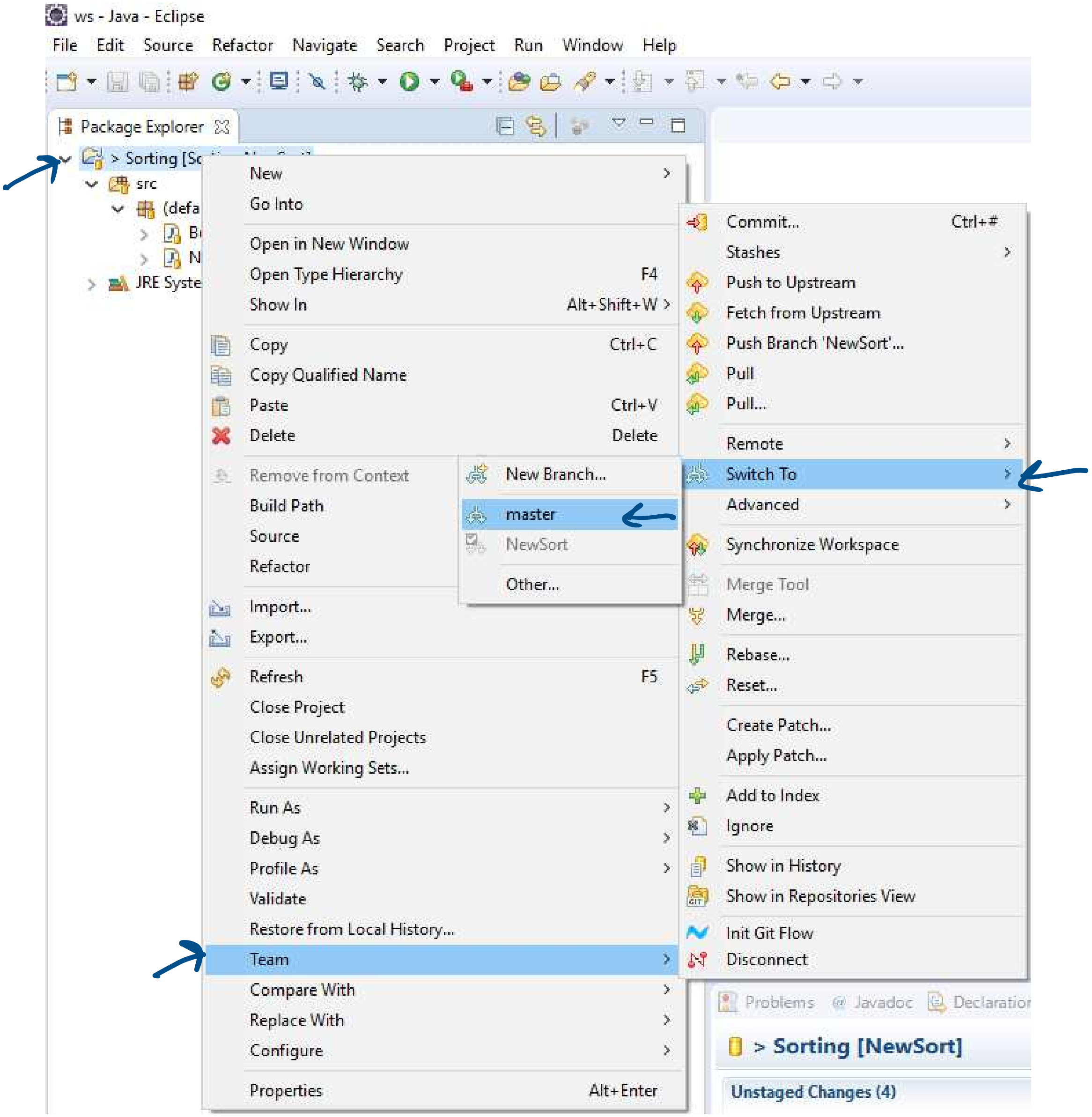
28. You will see a new java file under **src** folder (i.e., NewSort.java) is added to NewSort branch.



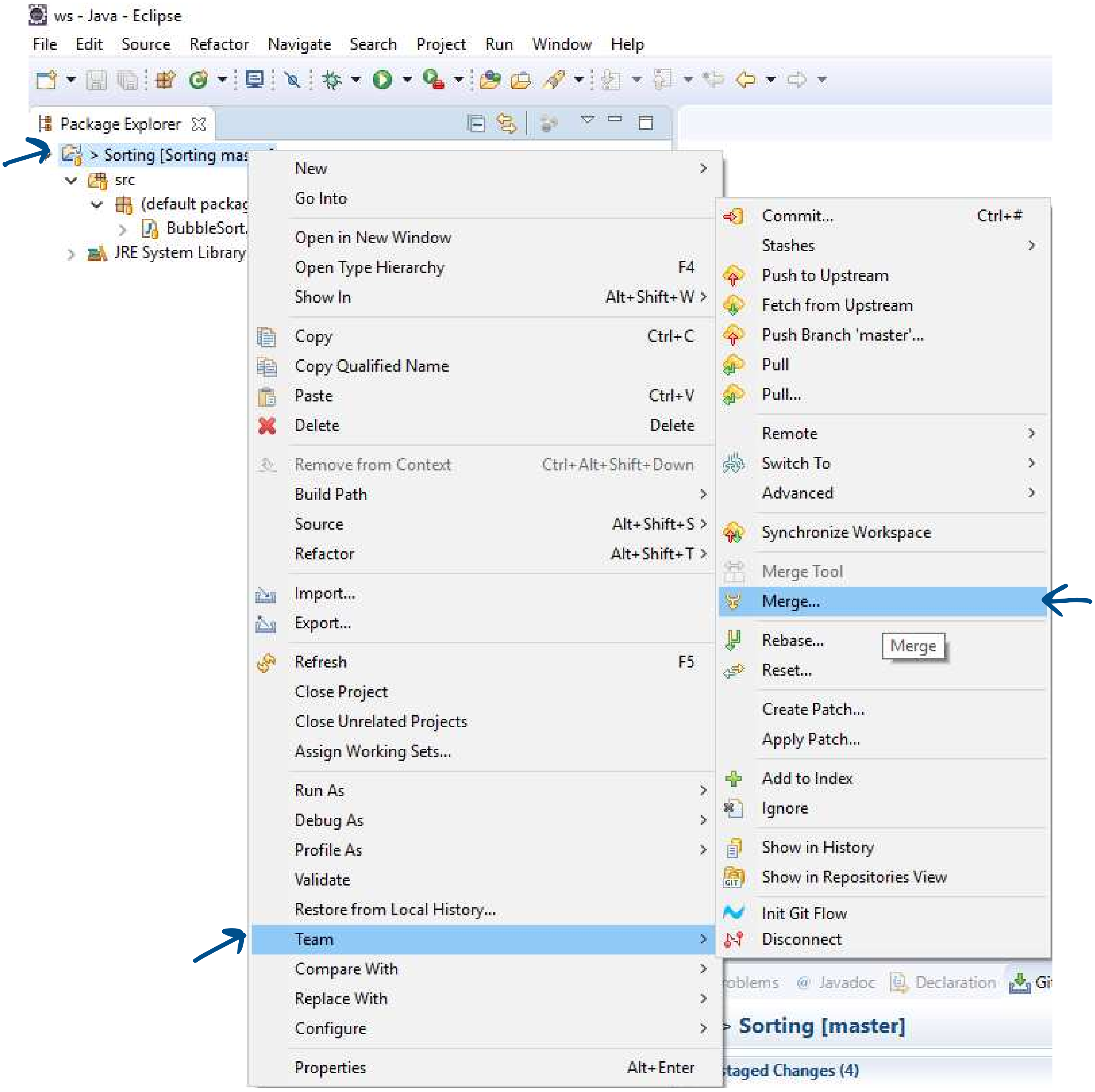
This branch is 1 commit ahead of master.

goswa New Sort

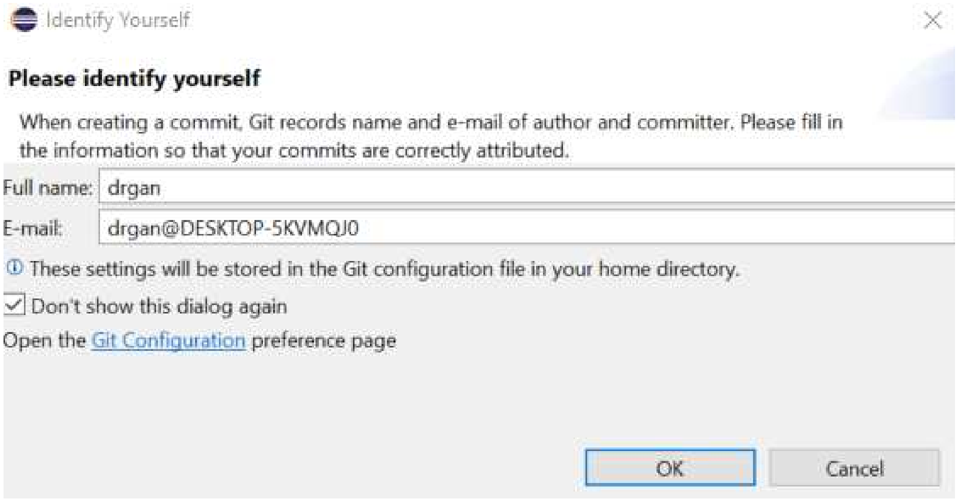
29. Go back to Eclipse IDE. Your new sorting program is working fine, and you want it to merge in your master branch. To do so, switch to master branch. Right click on project → **team** → **switch to** → **master**.



30. Right click on project → **team** → **merge**.



31. If Eclipse asks for identification, then click OK button.



Please identify yourself

When creating a commit, Git records name and e-mail of author and committer. Please fill in the information so that your commits are correctly attributed.

Full name: drgan

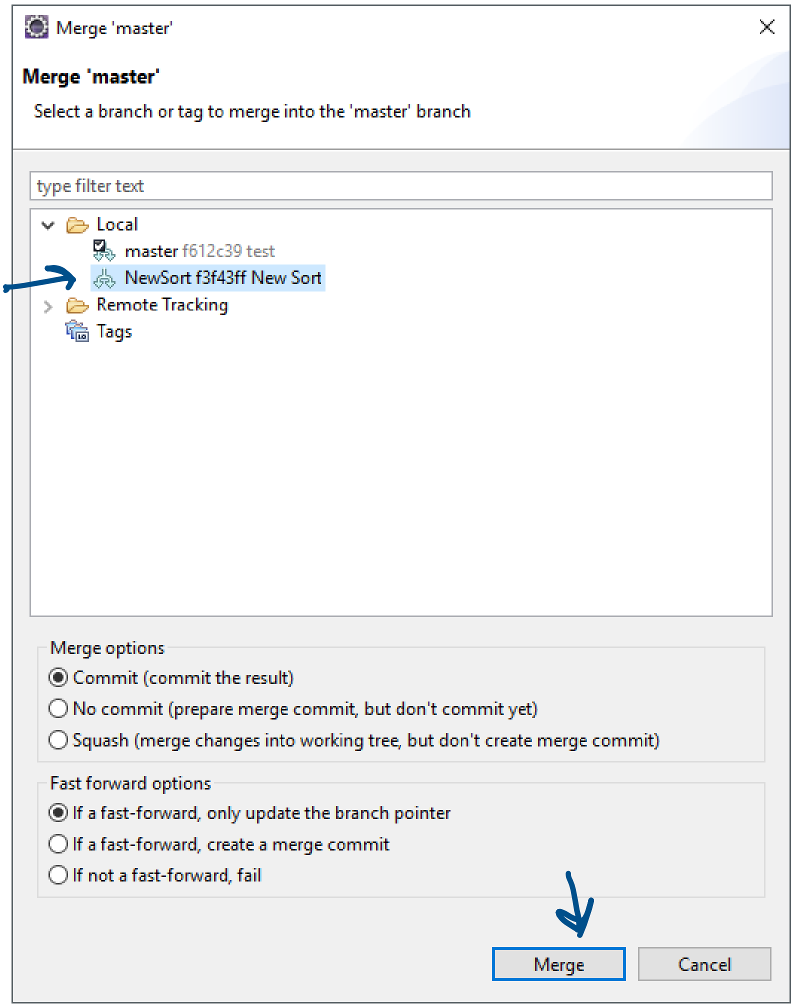
E-mail: drgan@DESKTOP-5KVMQJ0

These settings will be stored in the Git configuration file in your home directory.

☑ Don't show this dialog again

Open the Git Configuration preference page

32. Select **NewSort** branch and click **merge** button.



Merge 'master'

Select a branch or tag to merge into the 'master' branch

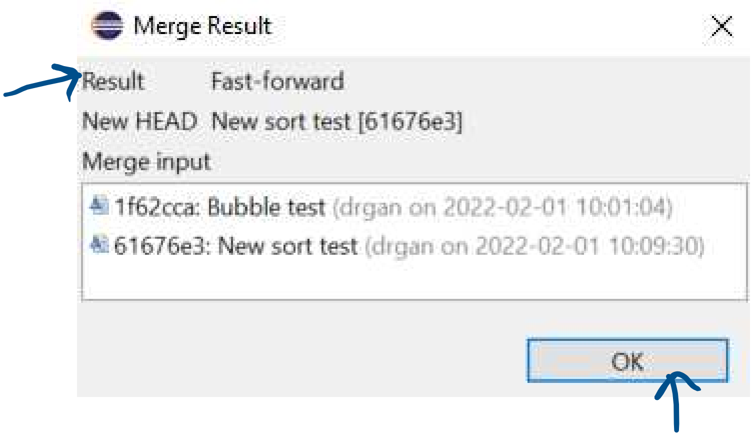
Merge options

● Commit (commit the result)
○ No commit (prepare merge commit, but don't commit yet)
○ Squash (merge changes into working tree, but don't create merge commit)

Fast forward options

● If a fast-forward, only update the branch pointer
○ If a fast-forward, create a merge commit
○ If not a fast-forward, fail

33. Click **OK** button after the merge result is displayed.



Merge Result

Result Fast-forward

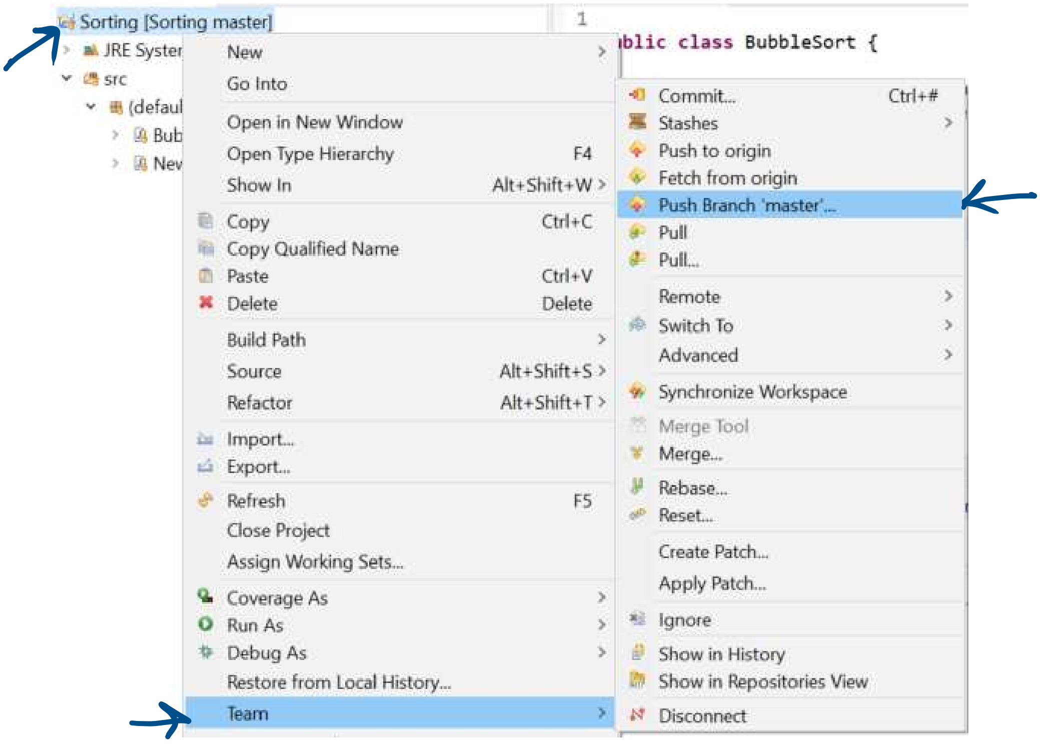
New HEAD New sort test [61676e3]

Merge input

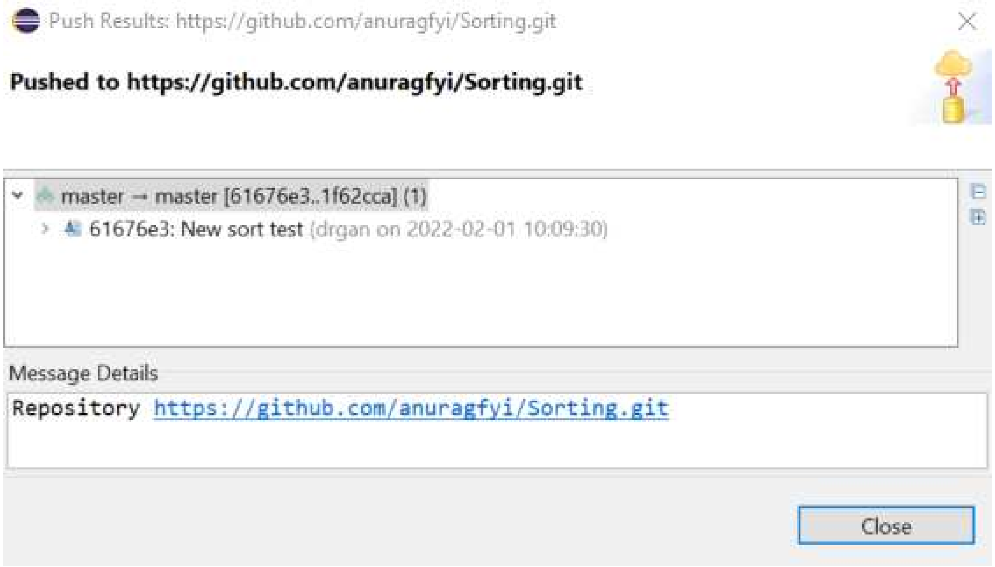
1f62cca: Bubble test (drgan on 2022-02-01 10:01:04)

61676e3: New sort test (drgan on 2022-02-01 10:09:30)

34. Right click on project → **team** → **Push Branch 'master'**.



35. Follow all the previous steps to Push and check form the confirmation at the end.



Push Results: https://github.com/anuragfyi/Sorting.git

Pushed to https://github.com/anuragfyi/Sorting.git

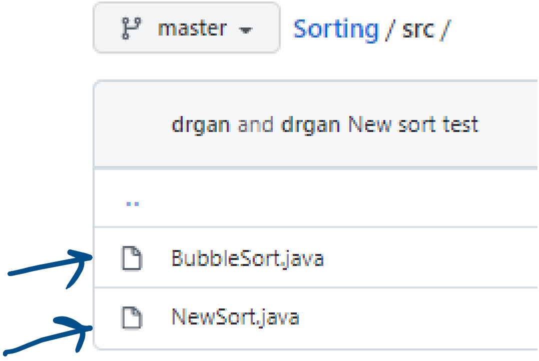
master → master [61676e3..1f62cca] (1)

61676e3: New sort test (drgan on 2022-02-01 10:09:30)

Message Details

Repository https://github.com/anuragfyi/Sorting.git

36. Refresh your repository and switch to **master branch**. You will see both the programs in master branch. ***Take a snapshot of your screen that shows master branch with both JAVA files.***



drgan and drgan New sort test

..

BubbleSort.java

NewSort.java

Submit screenshot (e.g., image in step #28 and #36) of **new branch** (mine is NewSort) **as well as master** that you created along with **both the sorting programs** (i.e., java files) as well as **link to your Git repository** in a text file as your lab assignment 4.

Total five files: 2 screenshots, 2 java files, and link in 1 text file