

# SVN with GoogleCode

Supplementary for  
COMP3111/H tutorial

# What is SVN?

- Apache Subversion (often abbreviated SVN) is one of the software versioning and revision control systems
  - Other systems are available: CVS, Git
- Developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation.
- Subversion was created by CollabNet Inc. in 2000.

# Setup SVN with GoogleCode

- URL: <https://code.google.com>
- Click “Create a new project”, you should sign in using your Google account

The screenshot shows the Google Code homepage at <https://code.google.com>. The page has a header with the Google logo and a search bar. Below the header, there's a section for "Project Hosting" with a description: "Project Hosting on Google Code provides a free collaborative development environment for open source projects." A "Search Projects" input field and a "Search Projects" button are present. To the left, there's a "Google Developers" sidebar with links to "Looking for Google APIs and Tools?", "developers.google.com", and a "Create a new project" button, which is circled in red. At the bottom, there are links for "Terms of Service", "Privacy Policy", and "Site Directory". Language options at the very bottom include English, Español, 日本語, 한국어, Português, Pycckий, 中文(简体), and 中文(繁體).

# Setup a new project

Project name  Example: my-project-name

Project summary

Description

Version control system  Git  Mercurial  Subversion Subversion means “SVN”

Source code license

Project label(s)    [add another row](#)

**Create project**

# A sample SVN project is created

 my-sample-svn-project

This is a sample SVN project (for COMP3111)

[Project Home](#) [Wiki](#) [Issues](#) [Source](#) [Administer](#)

[Summary](#) [People](#)

Tip: Project owners, see our [Getting Started guide](#)

**Project Information**

Starred by 0 users  
[Project feeds](#)

Code license  
[GNU GPL v3](#)

Labels  
Academic, Demo

Members  
[hk peterpeter](#)

Your role  
[Owner](#)

This is a sample SVN (Subversion) project (for COMP3111). It helps students learn how to use GoogleCode.

# Project settings in GoogleCode

- In this example:
  - URL (marked in red)
  - Username (marked in black)
  - Password (click inside the link marked in green)

 my-sample-svn-project  
This is a sample SVN project (for COMP3111)

Project Home Wiki Issues **Source** Administer

Checkout Browse Changes Request code review

How-to: Explore this project's source code by clicking the "Browse" and "Changes" links above. X

Command-line access

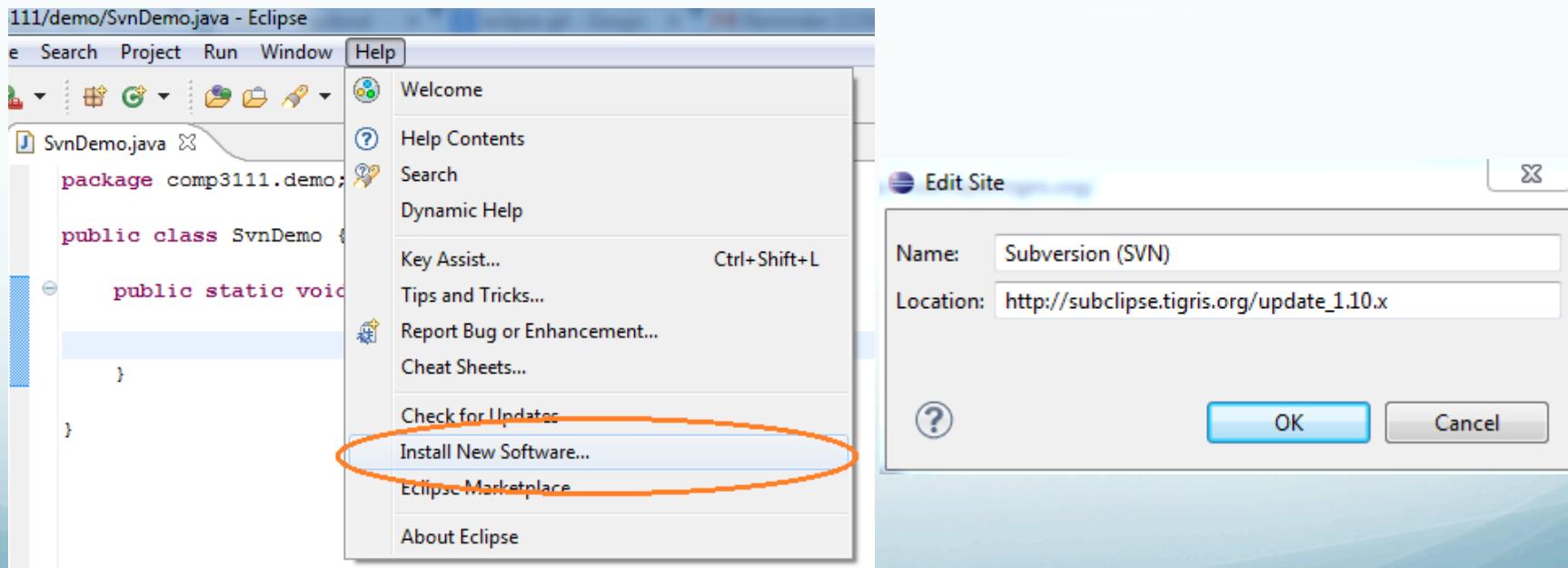
If you plan to make changes, use this command to check out the code as yourself using HTTPS:

```
# Project members authenticate over HTTPS to allow committing changes.  
svn checkout https://my-sample-svn-project.googlecode.com/svn/trunk/ my-sample-svn-project --username hk peterpeter@gmail.com
```

When prompted, enter your generated [googlecode.com password](#).

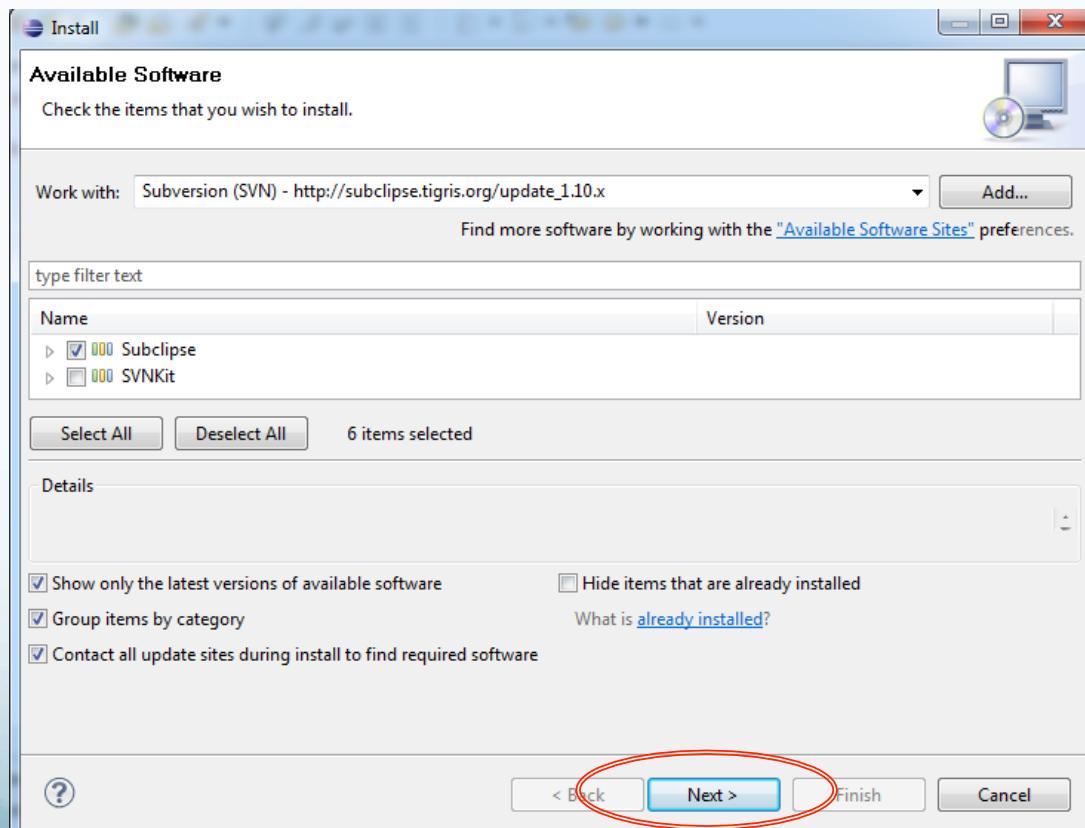
# Install SVN plugin in Eclipse (if it is not installed)

- Instructions: <http://subclipse.tigris.org/>
- The latest version is located at  
[http://subclipse.tigris.org/update\\_1.10.x](http://subclipse.tigris.org/update_1.10.x)



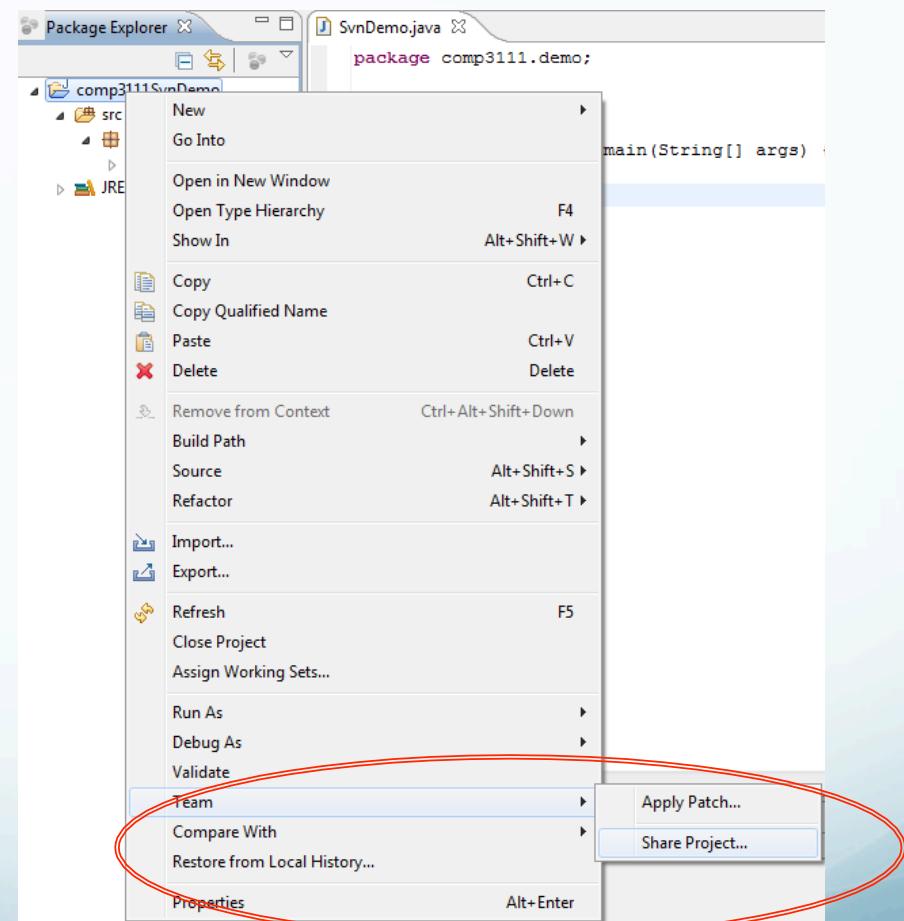
# Install SVN plugin in Eclipse

- After that, follow the instructions to install the “Subclipse” Eclipse plugin and restart Eclipse



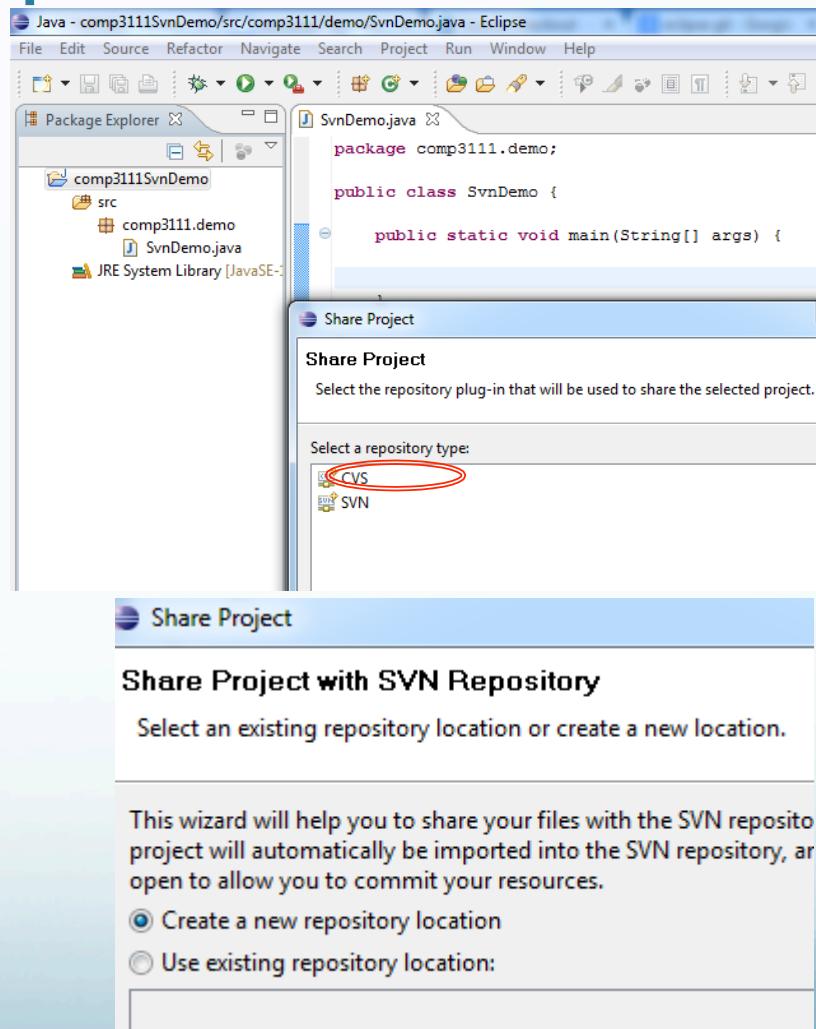
# [By project owner] Share a project using SVN/Eclipse

- Will be done by the project owner
- Create a simple Java project
- Right-click the project folder
  - Choose Team > Share Project...



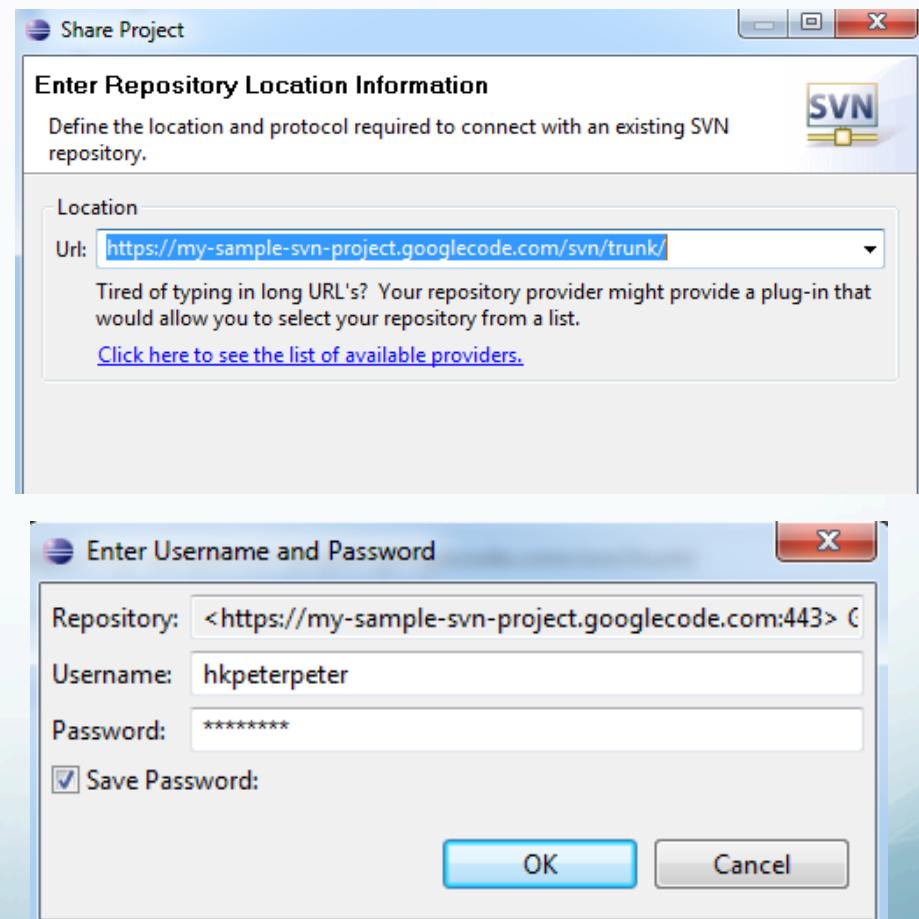
# Share a project using SVN/ Eclipse

- Make sure the Subversion Eclipse plugin (Subclipse) is correctly installed
- Select “SVN” as a repository type
- Choose “Create a new repository location”



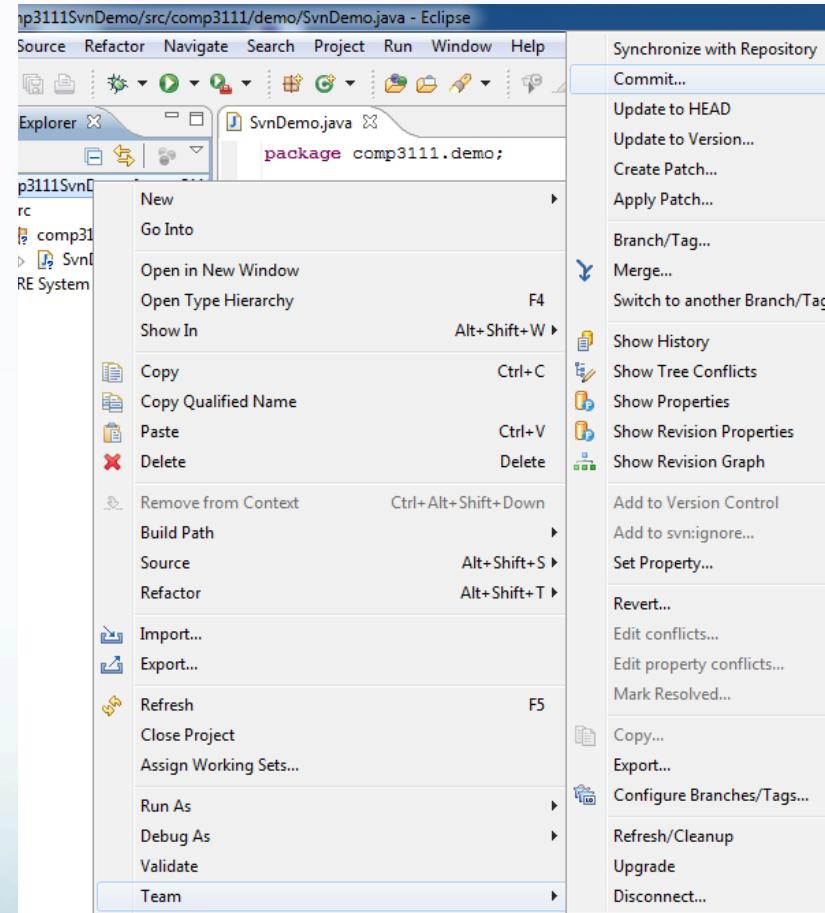
# Share a project using SVN/ Eclipse

- Enter the URL, username and password from the GoogleCode settings



# Share a project using SVN/ Eclipse

- Once you have connected your project to the SVN repository
  - Right-click the project folder, select Team > Commit...
  - Type in the commit log and then commit



# Check the GoogleCode project page

- The initial project is uploaded to the repository

My-sample-svn-project	
is a sample SVN project (for COMP3111)	
<a href="#">Wiki</a> <a href="#">Issues</a> <a href="#">Source</a> <a href="#">Administer</a>	
Browse	<a href="#">Changes</a> <a href="#">Request code review</a>
svn/	
Filename	SvnDemo.java
11SvnDemo	
ings	
omp3111	
demo	

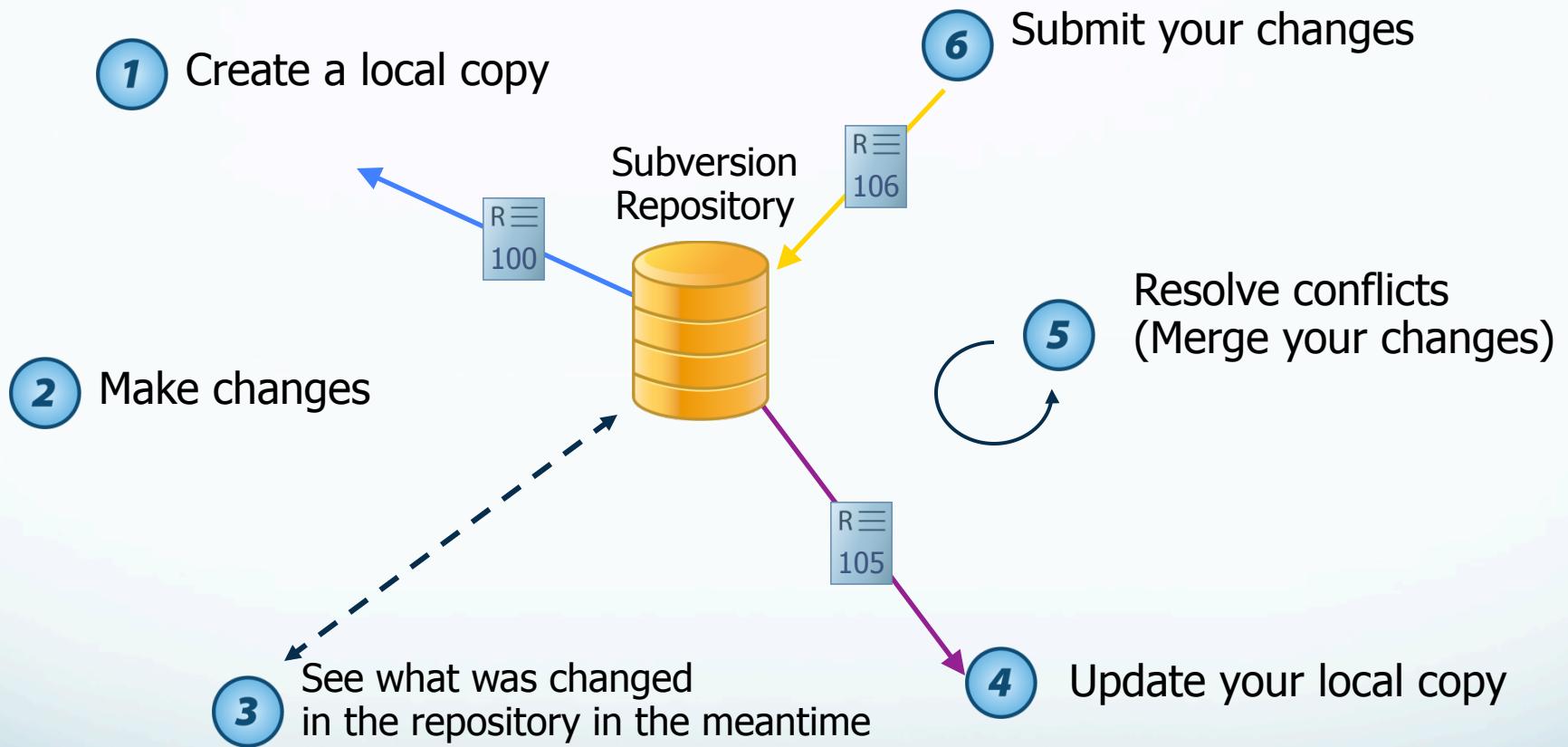
# Share with other users

- For the project owner, click Administer > Sharing

The screenshot shows the 'Sharing' tab of a Google Project's Administer section. It includes three main sections for managing project access:

- Project owners:** A text input field containing "hkpeterpeter@gmail.com". To its right is an instruction box: "Instructions: Specify each project participant by his or her Google Account email address. Each person must have already created a Google Account with that email address." Below this is a note: "Separate addresses with commas and/or newlines."
- Project committers:** An empty text input field. To its right is a note: "Project owners may make any change to this project. Project committers may work in the project, but may not reconfigure it."
- Project contributors:** An empty text input field. To its right is a note: "Project contributors start with the same permissions as non-members, but their role in the project is visible. Additional permissions can be granted to committers and contributors on the project's [people](#) sub-tab."

# [By contributors] Life cycle of SVN

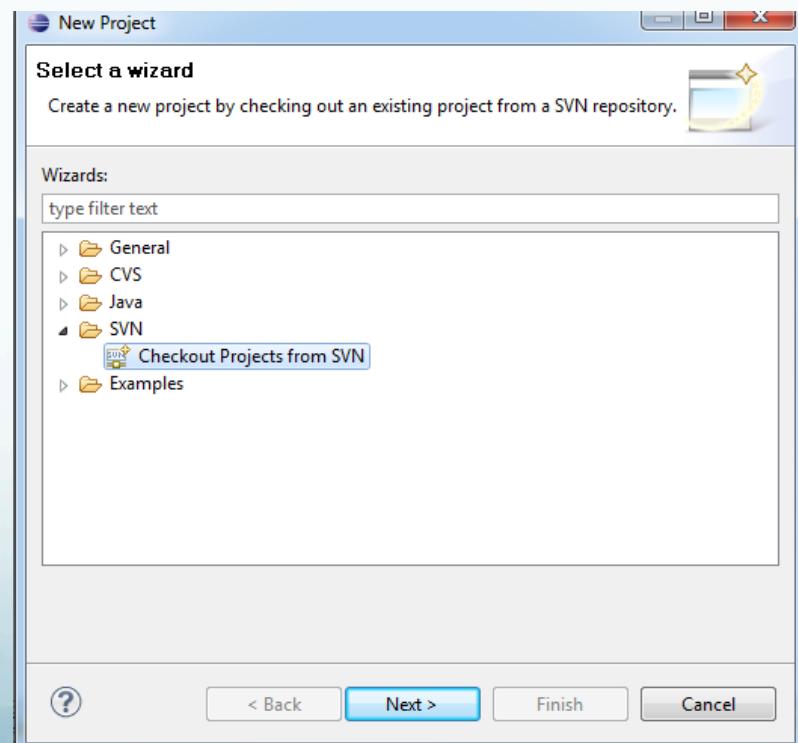
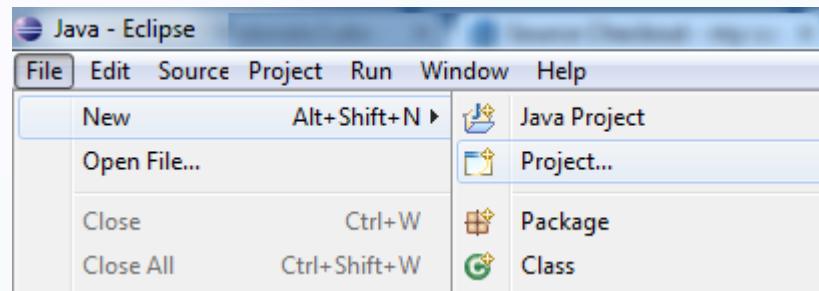


# SVN life cycle

- Once a project is created, other users can start the project development
  - [Checkout] a local copy
  - Make changes to your local copy
  - [Compare] with the latest changes in the repository
  - [Update] your local copy
  - [Resolve conflicts] if needed
  - [Commit] your changes

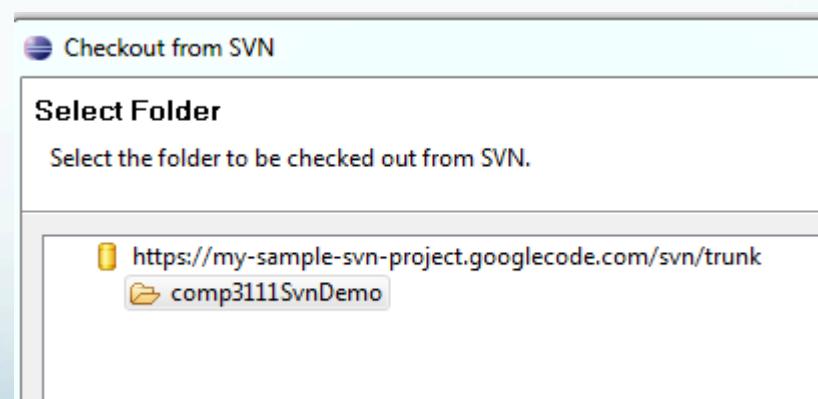
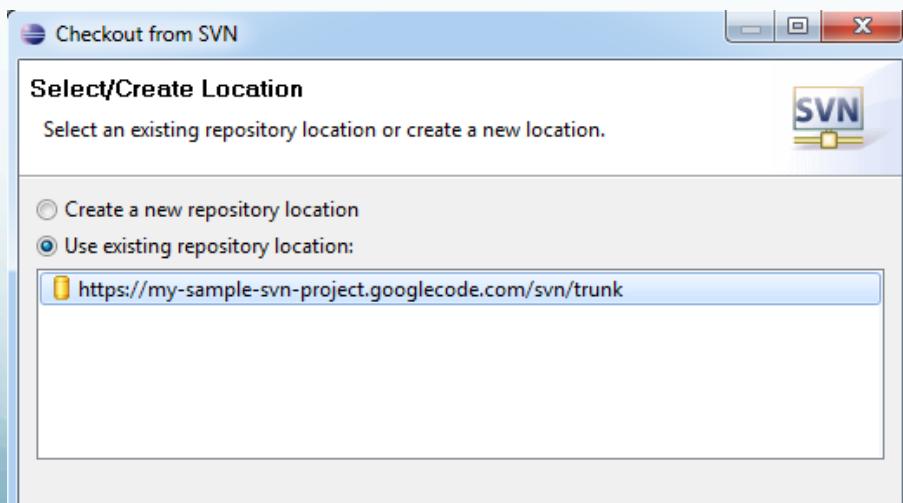
# Step1: Checkout a local copy

- Please note that a SVN Eclipse plugin must be installed first
- Select File > New > Project...
- Select “Checkout Projects from SVN”

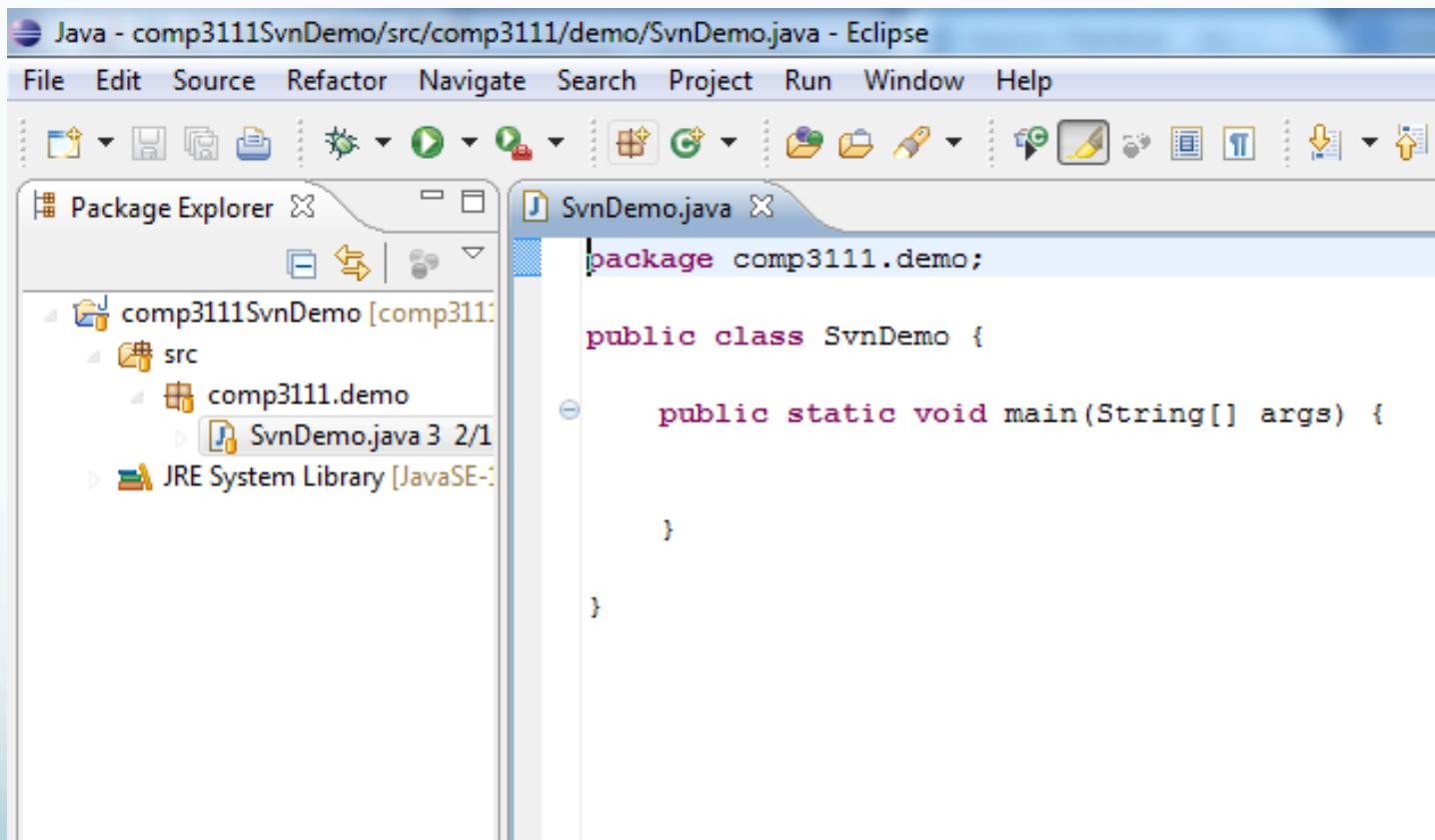


# Step1: Checkout a local copy

- If you have previously connected and saved repository locations, you can select from the list. Otherwise, you can create a new repository location
- Pick the project you want to checkout

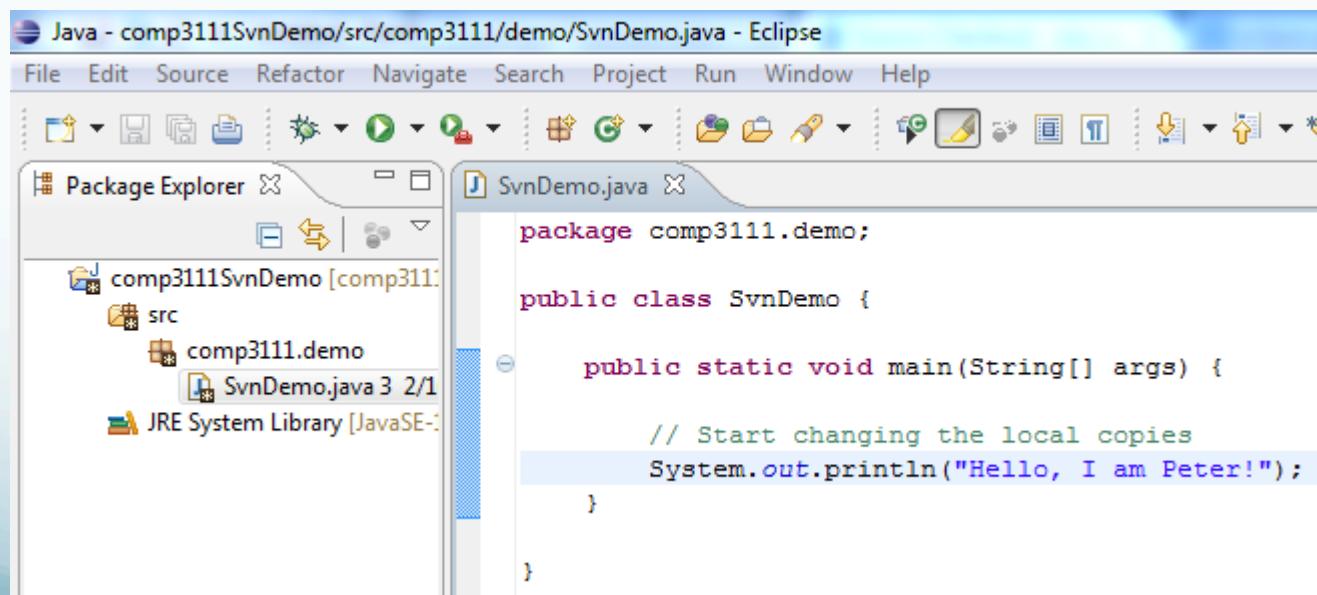


# [DONE] Step1: Checkout a local copy



# Step 2: Make changes to your local copy

- After you have checked out the project, you can start contributing the project by editing the local copy
  - Please note that all changes are now in the local machine

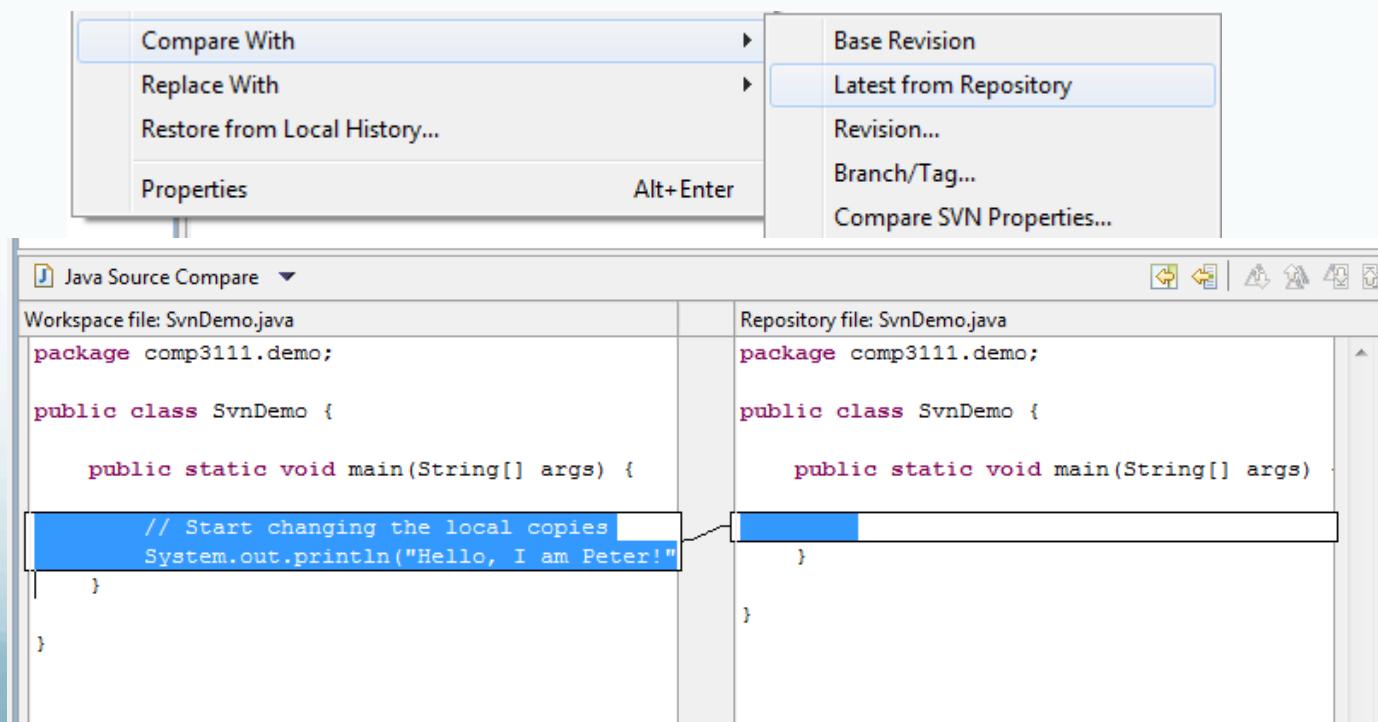


# Actions before committing the changes

- At the same time, someone may edit the same file
- He/she may commit the changes before your last checkout
- Before you commit your contributions, you should
  - Compare with the latest version
  - Update your local copy
  - Resolve conflicts if needed
  - Commit your changes

# Step 3: Compare with the latest version

- How can I compare with the latest version?
  - Right-click a file you would like to compare
  - Select Compare With > Latest from Repository



# Step 3: Compare with the latest version

- Other possible scenario
  - We may need to resolve conflicts

The screenshot shows a Java Source Compare interface. On the left, the 'Workspace file: SvnDemo.java' contains the following code:

```
package comp3111.demo;

public class SvnDemo {

    public static void main(String[] args) {
        // Start changing the local copies
        System.out.println("Hello, I am Peter!");
    }
}
```

On the right, the 'Repository file: SvnDemo.java' contains this code:

```
package comp3111.demo;

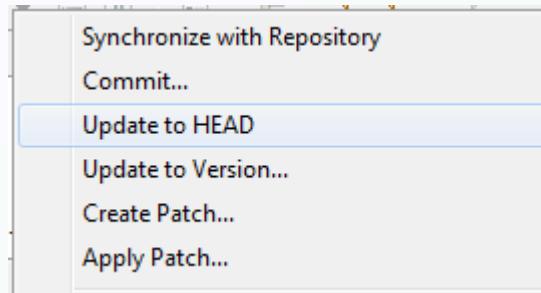
public class SvnDemo {

    public static void main(String[] args) {
        System.out.println("Hey, some other guys update this file!!!");
        System.out.println("It may cause some conflicts!");
    }
}
```

The code in the repository file has three new lines highlighted in blue, indicating they are additions. The original code in the workspace file is shown in grey.

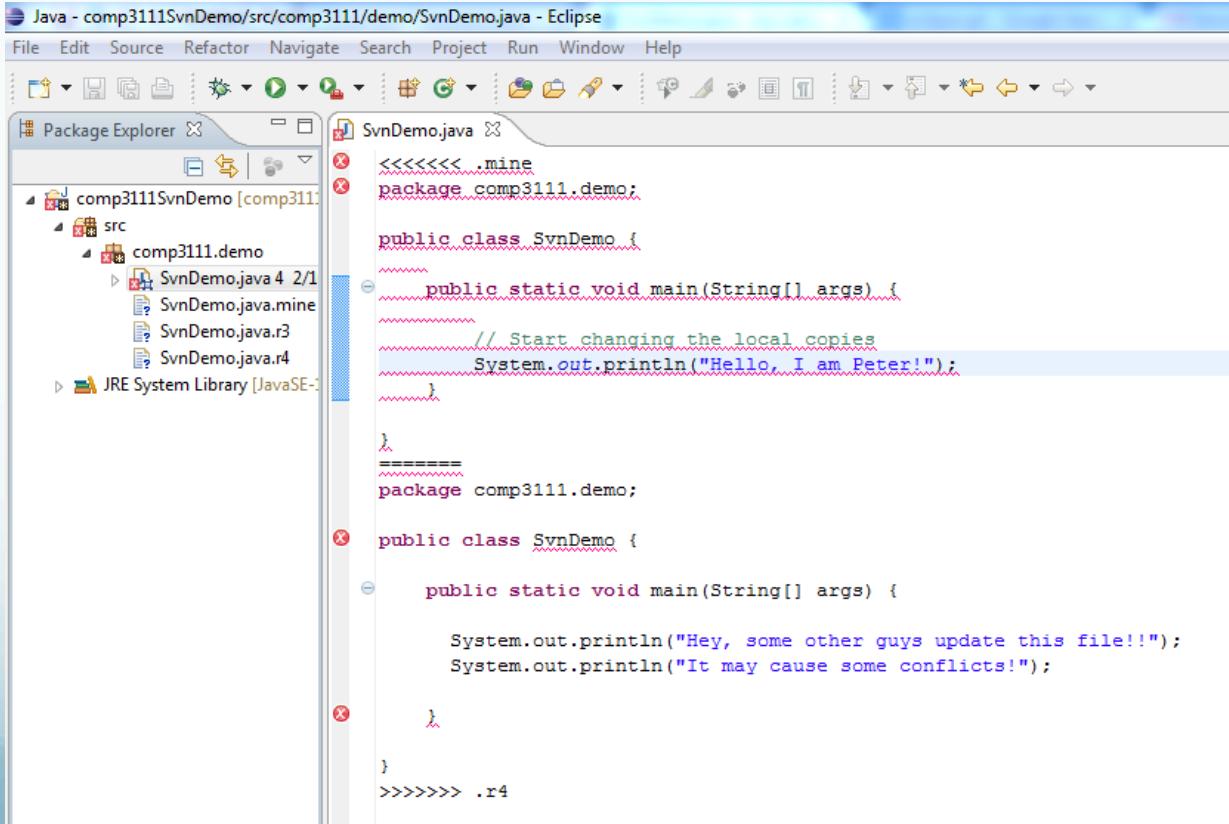
# Step 4: [Update] your local copy

- In any case, we should first update our local copy
- Update to HEAD (to the latest version)
  - Right-click the file you would like to update
  - Select Team > Update to HEAD



# NightMare: Conflict appears

- In the worst case, you will encounter conflicts after you updated your local copy



The screenshot shows the Eclipse IDE interface with the title bar "Java - comp3111SvnDemo/src/comp3111/demo/SvnDemo.java - Eclipse". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The toolbar has various icons for file operations. The left sidebar is the "Package Explorer" showing a project structure with "comp3111SvnDemo [comp3111]" containing "src" and "comp3111.demo" which contains "SvnDemo.java". The right side is the "SvnDemo.java" editor. The code is as follows:

```
<<<<< .mine
package comp3111.demo;

public class SvnDemo {

    public static void main(String[] args) {
        // Start changing the local copies
        System.out.println("Hello, I am Peter!");
    }
}
=====
package comp3111.demo;

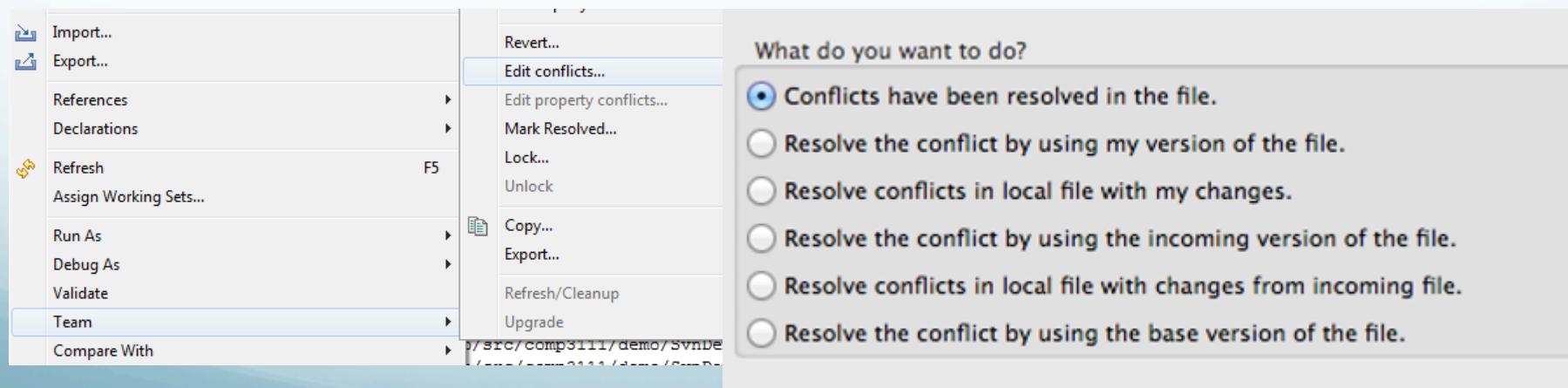
public class SvnDemo {

    public static void main(String[] args) {
        System.out.println("Hey, some other guys update this file!!!");
        System.out.println("It may cause some conflicts!");
    }
}
>>>>> .r4
```

The code is annotated with conflict markers: `<<<<< .mine` at the top, `=====` in the middle, and `>>>>> .r4` at the bottom. The package declaration and the first class definition are marked with red X's, indicating they are conflicted. The code content itself is mostly in blue, with some parts in pink.

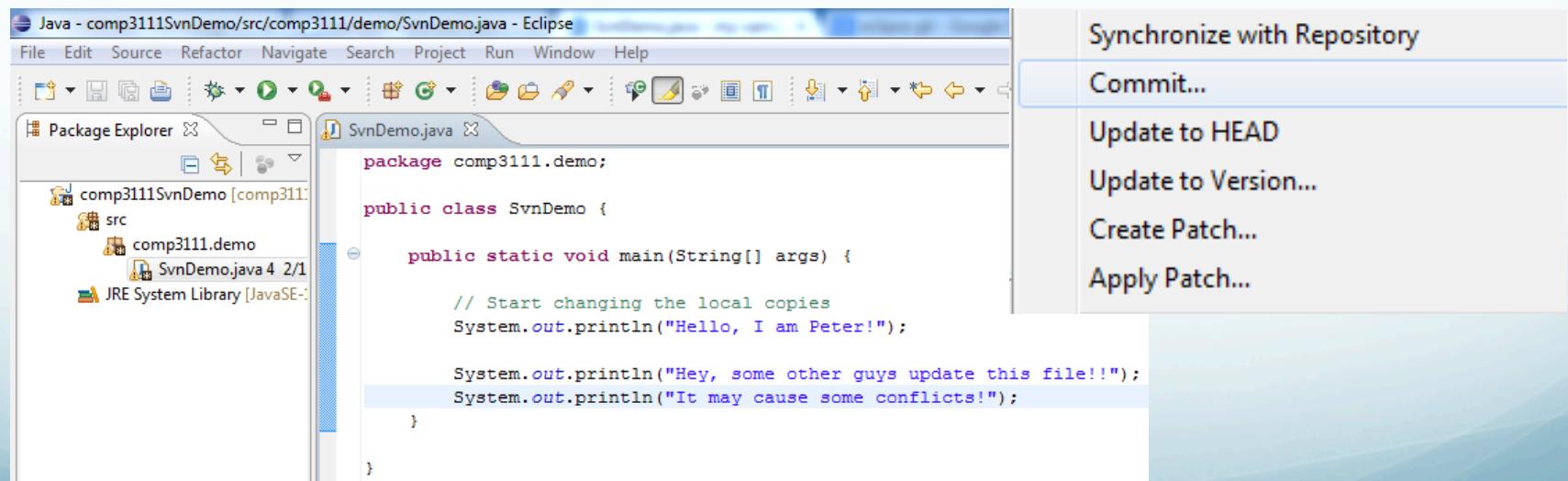
# Step 5: Resolve conflicts if needed

- Try your best to edit the conflicts (i.e. integrate the changes contributed by yourself with the latest repository)
- After that, you can “Mark Resolved...”
  - Otherwise, you may need to choose “revert...”



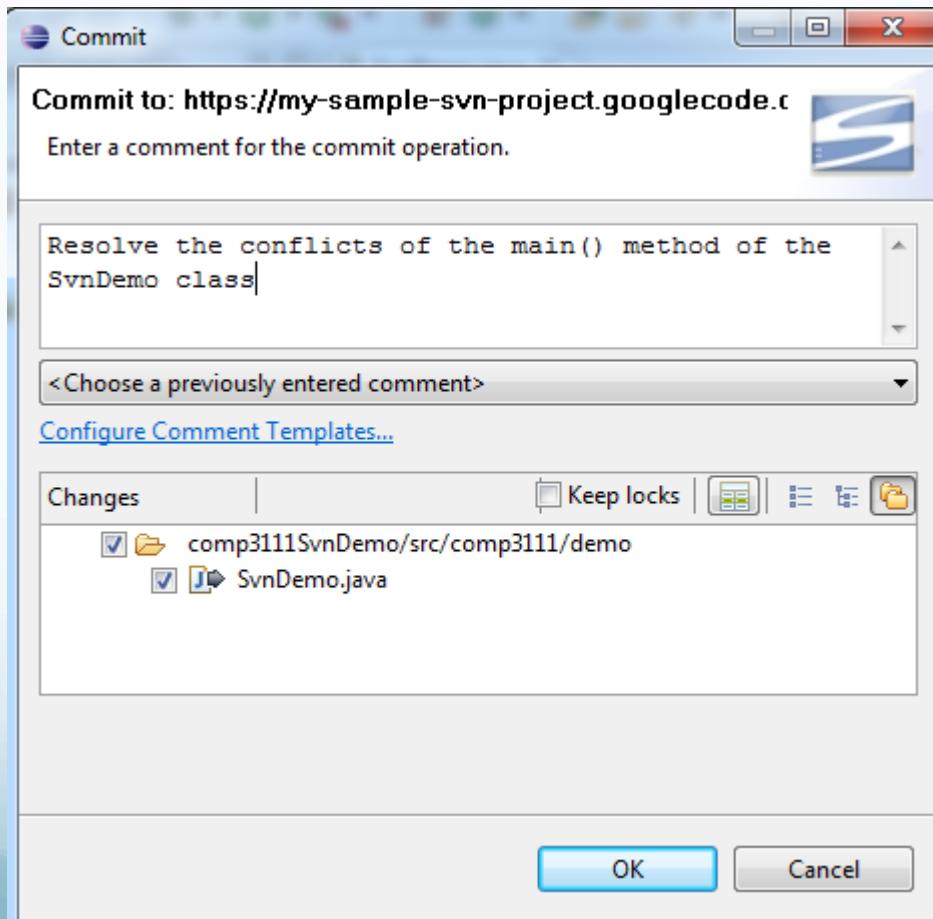
# Step 6: [Commit] your changes

- Once the conflict is resolved, we would like to update the repository
- Right-click the file, select Team > Commit...



# Step 6: [Commit] your changes

- Type in a commit log (a good practise)



# Result in the GoogleCode

- Useful information on the right-hand side

my-sample-svn-project  
This is a sample SVN project (for COMP3111)

Project Home Wiki Issues Source Administer

Checkout Browse Changes Request code review

Source path: [svn/ trunk/ comp3111SvnDemo/ src/ comp3111/ demo/ SvnDemo.java](#)

```
1 package comp3111.demo;
2
3 public class SvnDemo {
4
5     public static void main(String[] args) {
6
7         // Start changing the local copies
8         System.out.println("Hello, I am Peter!");
9
10        System.out.println("Hey, some other guys update this file!");
11        System.out.println("It may cause some conflicts!");
12    }
13}
14}
```

[Edit file](#) [r4 r5](#) [Hide details](#)

**Change log**

[r5 by hkpeterpeter on Today \(3 minutes ago\) Diff](#)  
Resolve the conflicts of the main() method of the SvnDemo class

Go to: [...o/src/comp3111/demo/SvnDemo.java ▾](#)

Double click a line to add a comment

**Older revisions**

[+ r4 by hkpeterpeter@gmail.com on Today \(18 minutes ago\) Diff](#)  
[+ r3 by hkpeterpeter on Today \(48 minutes ago\) Diff](#)  
[All revisions of this file](#)

**File info**

Size: 323 bytes, 14 lines  
[View raw file](#)

