

COMP285 ECLIPSE & EGIT INSTRUCTIONS

In this class, you will need:

- Eclipse, an *Integrated Development Environment* (IDE) to program in Java
- EGit, a *Distributed Version Control System* (DVCS) to share work

1. INSTALLING ECLIPSE (ONE TIME)

In a web browser,

- (1) **Download Eclipse Classic 3.7**
 - (a) Go to <http://www.eclipse.org/downloads/>
 - (b) Select the 32-bit version of Eclipse Classic 3.7

In the file manager,

- (1) **Extract Eclipse**
 - (a) Locate the downloaded zip file
 - (b) Right click on it, select “Extract All”
 - (c) Follow the prompts
 - (d) Locate eclipse.exe in the extracted folder
- (2) **Start Eclipse**
 - (a) Double-click on eclipse.exe

2. INSTALLING EGIT (ONE TIME)

- (1) **Enter your Eclipse workspace**
 - (a) Click OK when asked for a Workspace at startup
 - (b) Click on the “Workbench” arrow (if visible)
 - (c) You are now in the Workbench
- (2) **Install EGit**
 - (a) In the top-level menu, click “Help” → “Install New Software...”
 - (b) A new dialog box called “Install” will appear
 - (c) Locate the drop down next to “Work with” (in the “Install” dialog box)
 - (d) Select “Indigo – <http://download.eclipse.org/releases/indigo>”
 - (e) A list of software categories will appear
 - (f) Click the triangle next to “Collaboration” to expand it
 - (g) Select “Eclipse EGit” (under “Collaboration”)
 - (h) Click “Next” and follow all prompts
 - (i) Restart Eclipse when prompted to do so

3. REMOVE A GIT REPOSITORY (LOCAL)

A *repository* stores project history. *Removing* only affects the local repository.

- (1) **Switch to the Git Repository Exploring perspective**
 - (a) In the top-level menu, click “Window” → “Open Perspective” → “Other...”
 - (b) Select “Git Repository Exploring”

- (c) Select “Window” → “Show View” → “Git Repositories”
- (d) You will now see the “Git Repositories” view
- (2) **Remove a git repository**
 - (a) Select the git repository to remove
 - (b) Right click on it
 - (c) Click “Delete Repository...”
 - (d) Check “Also delete repository content in working directory ...”
 - (e) Click OK

4. CLONE A PERSONAL GIT REPOSITORY

A *repository* stores project history. *Cloning* copies a *remote* repository locally to your machine.

- (1) **Enter your Eclipse workspace**
- (2) **Switch to the Git Repository Exploring perspective**
- (3) **Clone the git repository**
 - (a) In the “Git Repositories” view, select the button that says “Clone a Git Repository” (the middle of 3 GIT buttons)
 - (b) “Clone Git Repository” dialog box will appear
 - (c) Copy this to the URI field (modify wentworthLoginHere as appropriate):
`ssh://joeylawrance.com/var/git/COMP285/wentworthLoginHere.git`
 - (d) Your username is your WIT login WITHOUT @wit.edu
 - (e) Your password is your secret key in lower case
 - (f) Click “Next”
 - (g) Click “Next” again
 - (h) Click “Finish”
 - (i) If this didn’t work, email the professor at lawrancej@wit.edu

5. CONFIGURE YOUR USER NAME (ONE TIME)

Ensure that the repository stores proper details.

- (1) Select “Window” → “Show View” → “Properties”
- (2) In the “Git Repositories” view, select a repository.
- (3) In the “Properties” view, locate the drop down menu.
- (4) Select “Global configuration”
- (5) Locate the “Edit” button (it has a pencil on it)
- (6) Click on “Edit”
- (7) The “Git Configuration Editor” dialog box will appear
- (8) Change the value of user.name to your first and last name
- (9) Change the value of user.email to your Wentworth email address
- (10) Click OK

6. CREATE A NEW PROJECT (FOR EACH ASSIGNMENT)

Create one new project per assignment, while naming projects properly.

- (1) **Switch to the Java Perspective, if necessary**
 - (a) In the top-level menu, click “Window” → “Open Perspective” → “Other...”
 - (b) Select “Java (default)”
 - (c) Click “OK”

- (2) **Create a new Java project**
 - (a) In the menu, select “File” → “New” → “Java Project”
 - (b) Enter a project name as follows:
Assignment00wentworthLoginHere, Assignment01wentworthLoginHere, ...
 - (c) Click “Finish”
- (3) **Connect your project to the git repository**
 - (a) In the Package Explorer (on the left) select the new project
 - (b) Right click on the project name. Select “Team” → “Share Project”
 - (c) Choose “Git” as the repository type
 - (d) Click “Next”
 - (e) Select your personal repository
 - (f) Leave “Path within repository” blank
 - (g) Click “Finish”

7. ADDING A CLASS TO A PROJECT

In Eclipse,

- (1) **Switch to the Java Perspective, if necessary**
- (2) In the menu, select “File” → “New” → “Class”
- (3) Enter the name of your new class
- (4) Click “Finish”

8. SHARING YOUR CODE & CHANGES

Commits record history, *pushes* update *remote* repositories, *pulls* update the *local* repository.

- (1) **Switch to the Git Repository Exploring perspective**
- (2) **Commit (record changes locally)**
 - (a) Right click on your repository. Select “Commit...”
 - (b) If you see: “No changed items were selected. Do you wish to amend the last commit?”
Click “No” if you already pushed your commit.
 - (c) If you get a dialog that says “Please Identify Yourself”
For User name, enter your first and last name
For User e-mail, enter your Wentworth email address
Click OK
 - (d) Enter a commit message (describe what you did)
 - (e) Ensure the Author and Committer are: Your Name <yourlogin@wit.edu>
 - (f) Check the “Select All” box (looks like a checkbox) under “Files”
 - (g) Click “Commit” (this records changes locally without sharing with others)
- (3) **Pull (only for final project) and Push (share changes with everyone else)**
 - (a) Right click on your repository.
 - (b) Select “Pull” (grabs the latest revisions, only necessary for team projects)
 - (c) Right click on your repository, select “Push to Upstream” (shares commits)

9. REVIEWING YOUR CODE

IMPORTANT: Solutions must have the problem statement in comments
Without a problem statement, semantics are unsatisfactory (fail)
 Use this template at the top of your classes:

```

/**
 * Problem statement and specification goes here.
 * Without this, semantics are automatically unsatisfactory.
 * @author wentworthLoginHere
 */

```

- (1) **Switch to the Git Repository Exploring perspective**
- (2) **Update README to describe what your reviewer should review**
 - (a) Select and expand your repository
 - (b) Select and expand “Working directory”
 - (c) Double-click on “README”
 - (d) Edit and save (Ctrl-S) the file to instruct reviewers
- (3) **Commit, and push**
- (4) **Review your code according to the rubric. BE Tough but Fair**
 - (a) Read, compile, and run the code to review it
 - (b) Use the rubric at <http://joeylawrance.com/COMP285/rubrics.pdf>
 - (c) Complete the code review at <http://bit.ly/COMP285CodeReview>

10. REVIEWING OTHERS’ CODE

In Eclipse,

- (1) **Get the reviewing assignments**
 - (a) Go to: <http://bit.ly/COMP285ReviewingAssignments>
 - (b) Locate your user name in the leftmost column
 - (c) You will review the code of those listed to the right
- (2) **Switch to the Git Repository Exploring perspective**
- (3) **(First time) Clone the relevant repository**
 - (a) Follow instructions in section 4.
Clone their `wentworthLogin.git` using your username and password
- (4) **(Subsequent times) Pull from the relevant repository**
 - (a) Right click on their git repository
 - (b) Select “Pull”
- (5) **Import projects from your reviewing assignment**
 - (a) Right-click on the relevant user’s personal repository
 - (b) Select “Import projects...”
 - (c) Click “Next”
 - (d) Click “Finish”
- (6) **Review their README file. You may need to review previous assignments**
- (7) **Switch back to the Java perspective**
- (8) **Review their code according to the rubric. BE Tough but Fair**

11. REVIEWING HISTORY

In Eclipse,

- (1) **Switch to the Git Repository Exploring perspective**
- (2) In the top-level menu, click “Window” → “Show View” → “History”

12. HOW DO I...?

If you want to do anything else with git,

- (1) Go to http://wiki.eclipse.org/EGit/User_Guide
- (2) Email lawrancej@wit.edu if you have questions