GETTING STARTED WITH ECLIPSE AND GITHUB

CONTENTS

GETTING STARTED WITH ECLIPSE AND GIT

INTRODUCTION

Eclipse Software Development Kit (Eclipse SDK) is a vast extendable set of tools for software development. Here we are interested in Eclipse's Integrated Development Environment (IDE) component for writing Java software. Eclipse is an open source project of Eclipse Foundation; you can find information about Eclipse Project at http://www.eclipse.org/eclipse. Eclipse is available free of charge under the Eclipse Public License.

Eclipse was developed by software professionals for software professionals; it may seem overwhelming to a novice. This document describes the very basics of Eclipse, enough to get started with Java in an educational setting.

Eclipse runs on multiple platforms including Windows, Linux, and Mac OS. There may be minor differences between Eclipse versions for different platforms and operating systems, but the core features work the same way. Here we will use examples and screen shots from Windows.

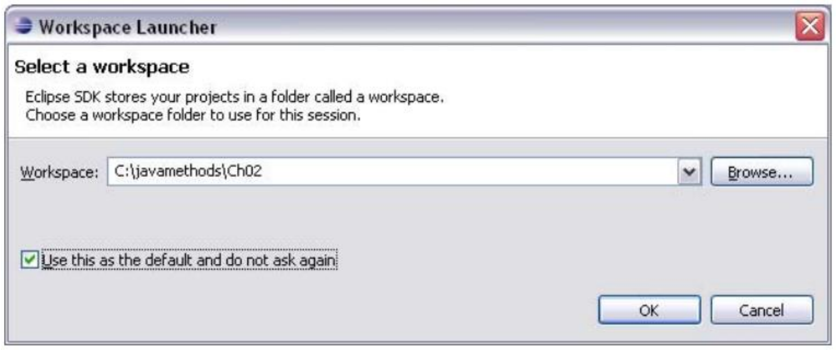
This document describes Version 4.5, Eclipse Mars for Java Developers.

GETTING STARTED WITH ECLIPSE

Go to http://www.eclipse.org and click on "Download Eclipse." In the displayed list of installations click on "Eclipse IDE for Java Developers." On the right, choose the desired operating system and architecture (for example, "Windows 64-bit" if you have a 64-bit system running Windows).

Under Windows, you will download a zip file, for example, eclipse-java-mars-R-win32.zip (for 32-bit systems) or eclipse-java-mars-R-win32-x86_64.zip (for 64-bit systems). Extract the eclipse folder from the zip file into your file system into C:\Program Files and rename it Eclipse (with a capital "E") for consistency with the names of other application folders. (It may take a few minutes to extract Eclipse files from the zip file.) You will find eclipse.exe in the eclipse folder. This is the Eclipse executable. Create a shortcut to it on the desktop (by dragging eclipse.exe to the desktop while holding down Ctrl+Shift).

Double click on the shortcut or on eclipse.exe. A dialog box will pop up asking you to choose a workspace:


Workspace Launcher

Select a workspace

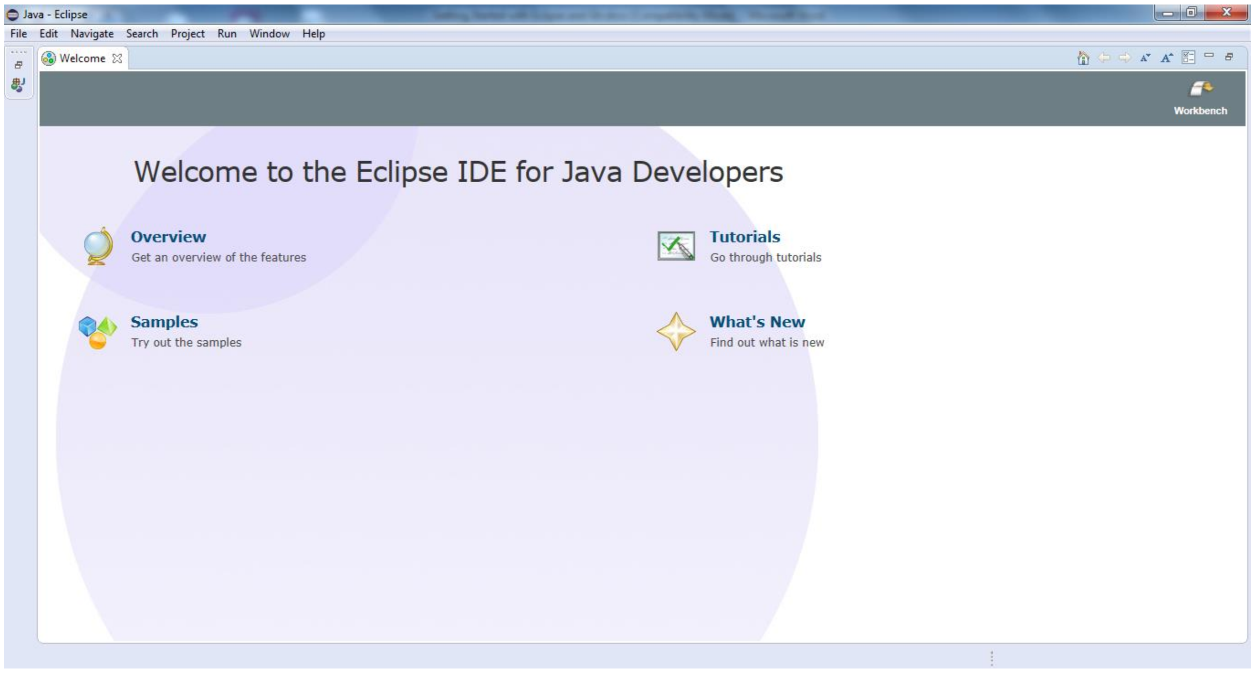
Eclipse SDK stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\javamethods\Ch02

[✓] Use this as the default and do not ask again

In Eclipse a workspace is basically a folder that contains project files and configuration data. Enter a folder of your choice, for example H:\ICS4U\Programs. Check the "Use this as the default" box — you can change it later. Click **OK**.

Eclipse then comes up with a Welcome screen:



Welcome to the Eclipse IDE for Java Developers

Overview
Get an overview of the features

Tutorials
Go through tutorials

Samples
Try out the samples

What's New
Find out what is new

Go over the overview and/or tutorials or click on Workbench (arrow in the top right corner) to start working.

CONFIGURING ECLIPSE

Here are a few suggestions for setting preferences.

- **In Eclipse, the Preferences command is located under the Window menu.**

Click on it.

Under **General→Editors→Text Editors** check “Show line numbers”.

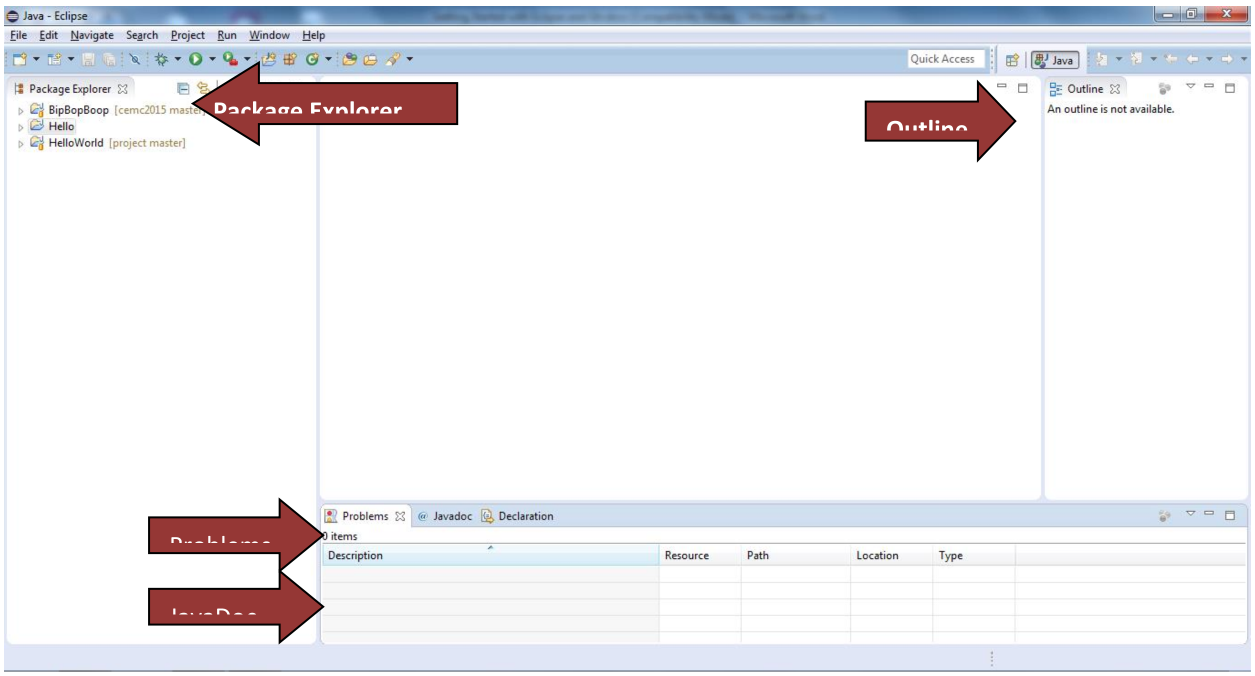
Under **General→Startup and Shutdown** check “Refresh workspace on startup”. Also increase the number of recent workspaces under **General→Startup and Shutdown→Workspaces**.

Under **General→Workspace** check “Save automatically before build.”

When you are finished setting the preferences, click **OK**. If Eclipse asks you whether it is OK to reload the workspace, click **Yes**.

SETTING UP YOUR PERSPECTIVE

If your screen does not look like the one below, go to **Window→Perspective→Open Perspective** and select Java. If Java is not shown click on “other” then scroll through the list until you see Java. If you are missing any of the panes click on **Window→Show View…** and select what you are missing. Any additional panes on your screen can be closed.



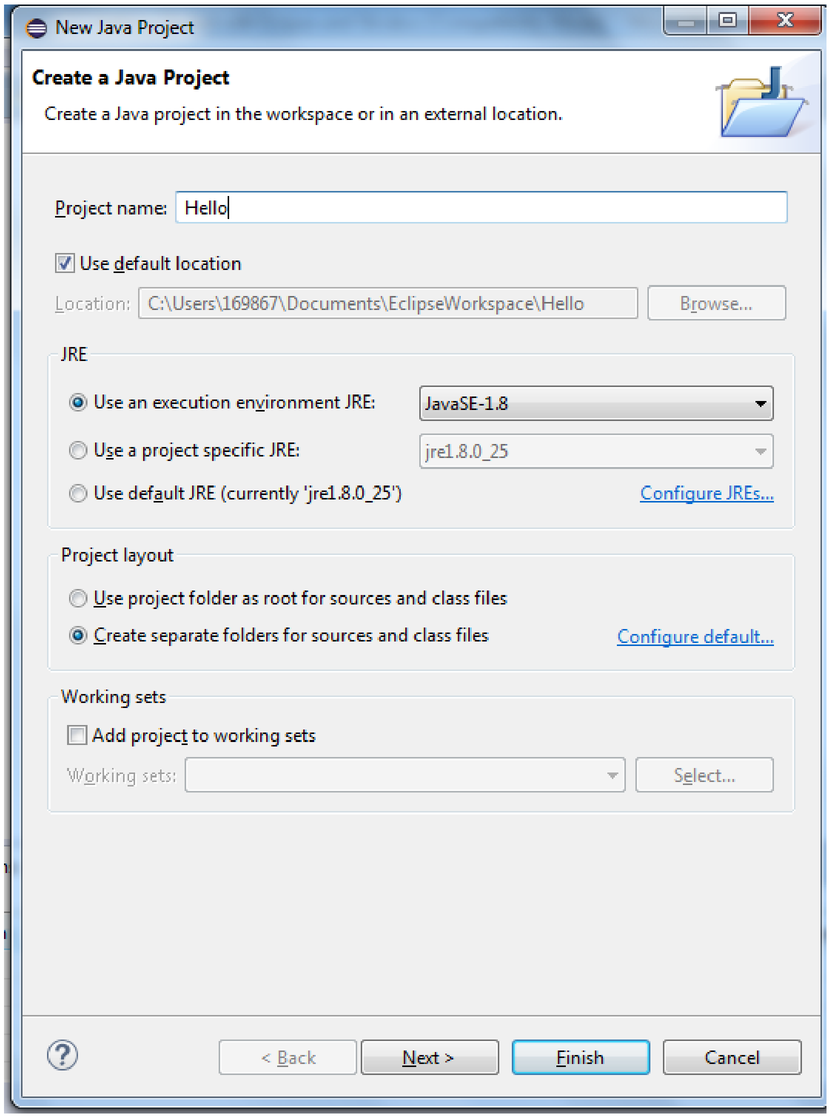
An outline is not available.

0 items

- ❖ One great feature of this version is that if you mess up the layout of your screen you can always go to **Window→Perspective** and select "Reset Perspective" and it will put all the blocks back in their original place.

RUNNING "HELLO WORLD"

On the File menu choose **New→Java Project** (or click on the pull-down arrow next to the New button on the toolbar and choose Java Project). A dialog box pops up. Enter the project name, for example, "Hello"; leave the "Use default location" box checked:


New Java Project

Create a Java Project

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

Project name: Hello

[✓] Use default location

Location: C:\Users\169867\Documents\EclipseWorkspace\Hello

JRE

(●) Use an execution environment JRE: JavaSE-1.8

( ) Use a project specific JRE: jre1.8.0_25

( ) Use default JRE (currently 'jre1.8.0_25')

Project layout

( ) Use project folder as root for sources and class files

(●) Create separate folders for sources and class files

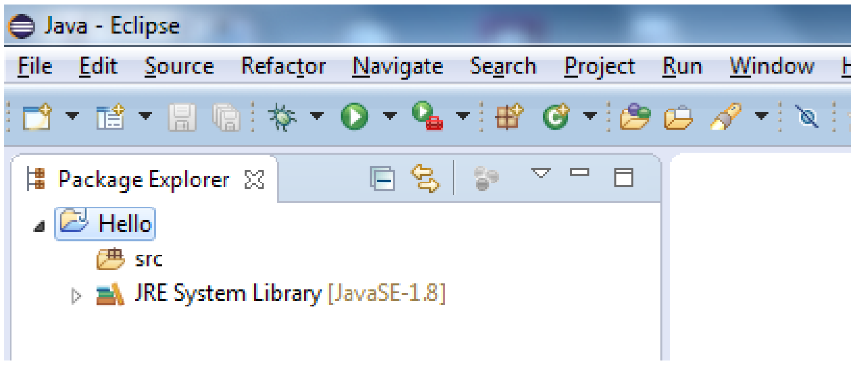
Working sets

[ ] Add project to working sets

Working sets:

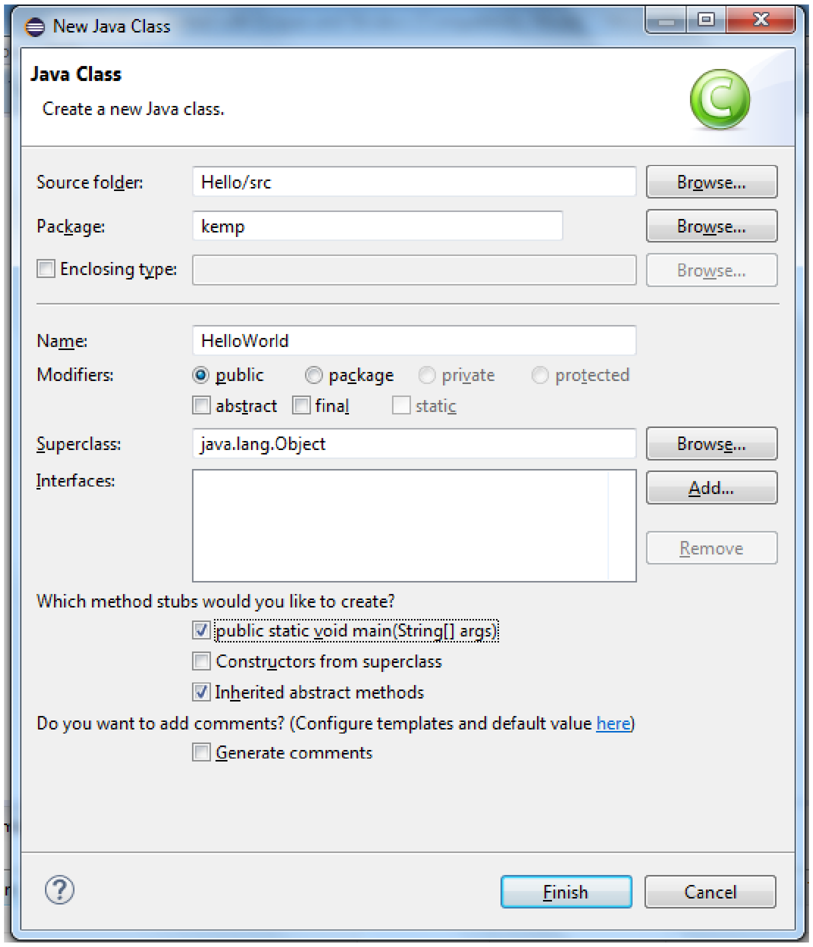
Click **Finish**.

If you expand the Hello folder in Package Explorer, you will see the src subfolder:


Java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window
Package Explorer
Hello
src
JRE System Library [JavaSE-1.8]

That's where Java source files go.

If you are starting from scratch, click on the New Java Class button on the toolbar (small green C). In the dialog box that pops up, enter the package name (for our purposes your name), the name for your class (for example, HelloWorld), and the features you want automatically generated for your class (for example public static void main).



New Java Class

Java Class

Create a new Java class.

Source folder: Hello/src

Package: kemp

Enclosing type: [ ]

Name: HelloWorld

Modifiers: [✓ public] package private protected; [ ] abstract [ ] final [ ] static

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

Click **Finish**.

Enter the code for your class in the editor:

```
/**
 * Displays a "Hello World!" message on the screen
 */

public class HelloWorld
{
     public static void main(String[] args)
     {
          System.out.println("Hello World!");
     }
}
```

If the indentation is not correct select all the code (Ctrl+A) then press Ctrl+I.

In the toolbar click the green play button or click Ctrl+F11. This will build or rebuild your project, if necessary, and run it. When you run your program for the first time, Eclipse might ask you: Run as Java applet or application? Choose “application.”

- **It is easy and convenient in Eclipse to have several applets and/or applications in the same project and choose which one of them to run.**

If you press Ctrl+F11 when a Java class is selected or open it in the editor, Eclipse will run the selected class. If there are several classes that have a main method in them and nothing is selected, Eclipse will ask you which one to run.

COMMAND-LINE ARGUMENTS AND USER INPUT

In the same project “Hello,” create two more classes,

Greetings —

```
/**
 * This program expects two command-line arguments
 * -- a person's first name and last name.
 */

public class Greetings
{
 public static void main(String[] args)
 {
      String firstName = args[0];
      String lastName = args[1];
      System.out.println("Hello " + firstName + " " + lastName);
      System.out.println("Congratulations on your second
      program!");
 }
}
```

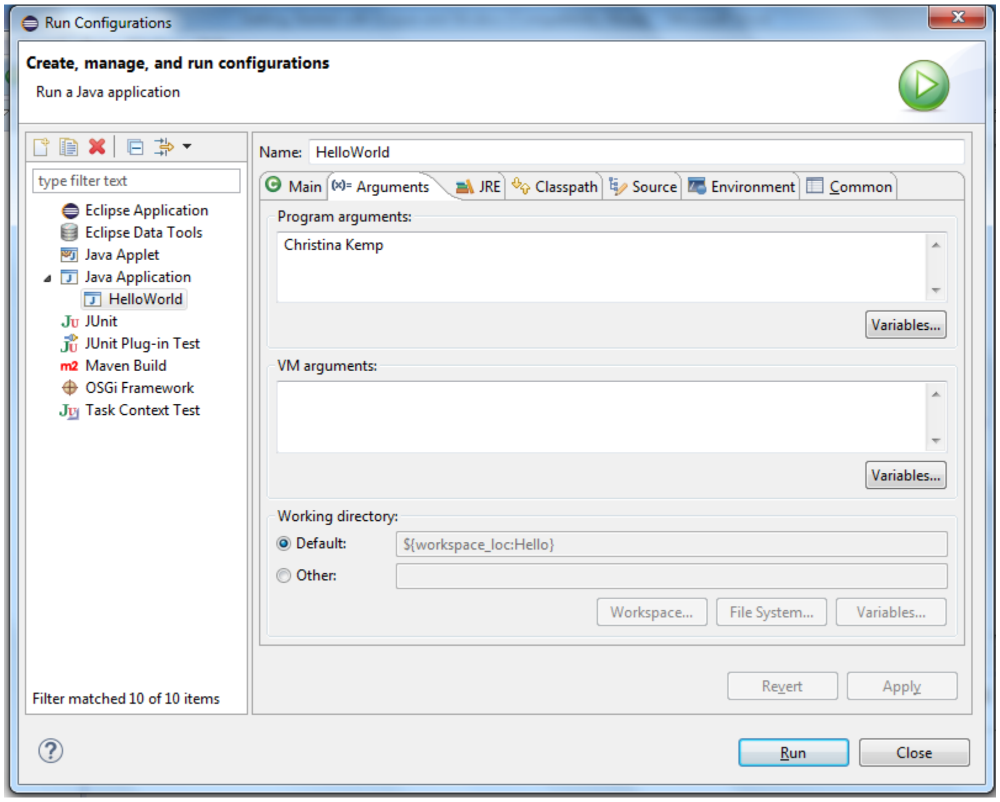
— and Greetings2:

```
/* This program prompts the user to enter his or her first name
and last name and displays a greeting message. Author: Maria
Litvin
*/

import java.util.Scanner;
public class Greetings2
{
 public static void main(String[] args)
 {
     Scanner scan = new Scanner(System.in);
     System.out.print("Enter your first name: ");
     String firstName = scan.nextLine();
     System.out.print("Enter your last name: ");
     String lastName = scan.nextLine();
     System.out.println("Hello " + firstName + " " + lastName);
     System.out.println("Welcome to Java!");
 }
}
```

Greetings expects command-line arguments and Greetings2 accepts input from the user.

If your application expects run-time parameters from the command line, you need to define a run-time configuration. From the Run menu choose Run Configurations and click on the "(x)=Arguments" tab. Enter the program arguments in the top text area. For example:


Run Configurations

Create, manage, and run configurations

Run a Java application

type filter text

- Eclipse Application
- Eclipse Data Tools
- Java Applet
- Java Application
  - HelloWorld
- JUnit
- JUnit Plug-in Test
- Maven Build
- OSGi Framework
- Task Context Test

Filter matched 10 of 10 items

Name: HelloWorld

Main | Arguments | JRE | Classpath | Source | Environment | Common

Program arguments:

```
Christina Kemp
```

VM arguments:

Working directory:

(●) Default: ${workspace_loc:Hello}

( ) Other:

Then click **Run**. You only have to do it once, as long as you keep the same program arguments.

You can double click on any Java file to open it in the editor window.

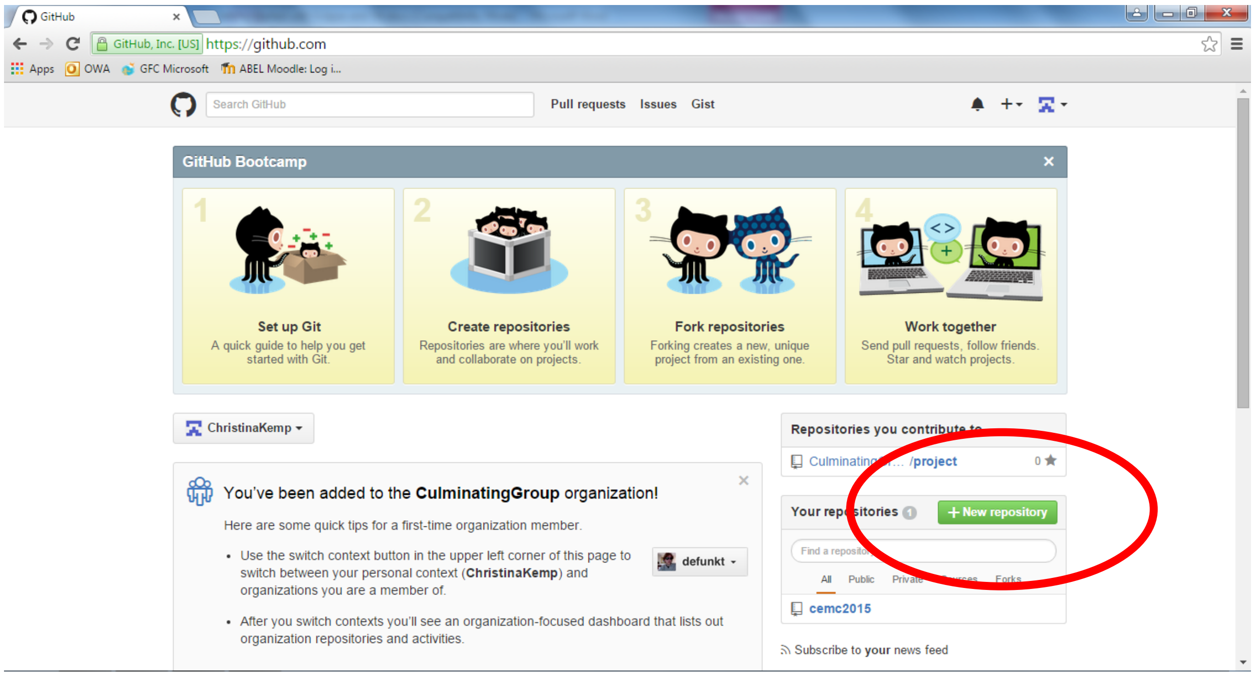
The third application in our “Hello” project, Greetings2, prompts the user for input.

- **Unfortunately, Eclipse doesn’t position the cursor correctly on the input line in the console window. If the cursor is in an editor window and you start typing, you will mess up the source code. Make sure to click on the console window before responding to the prompt.**

- **If your program reads data files (.txt, .wav, .gif, etc.), place them into the project folder (such as Hello, one step above src and bin).**

CREATING A GIT REPOSITORY IN GITHUB

Open your browser and login to GitHub (www.github.com). Create a new Git repository:


GitHub Bootcamp

1 Set up Git
A quick guide to help you get started with Git.

2 Create repositories
Repositories are where you'll work and collaborate on projects.

3 Fork repositories
Forking creates a new, unique project from an existing one.

4 Work together
Send pull requests, follow friends. Star and watch projects.

ChristinaKemp

You've been added to the **CulminatingGroup** organization!

Here are some quick tips for a first-time organization member.

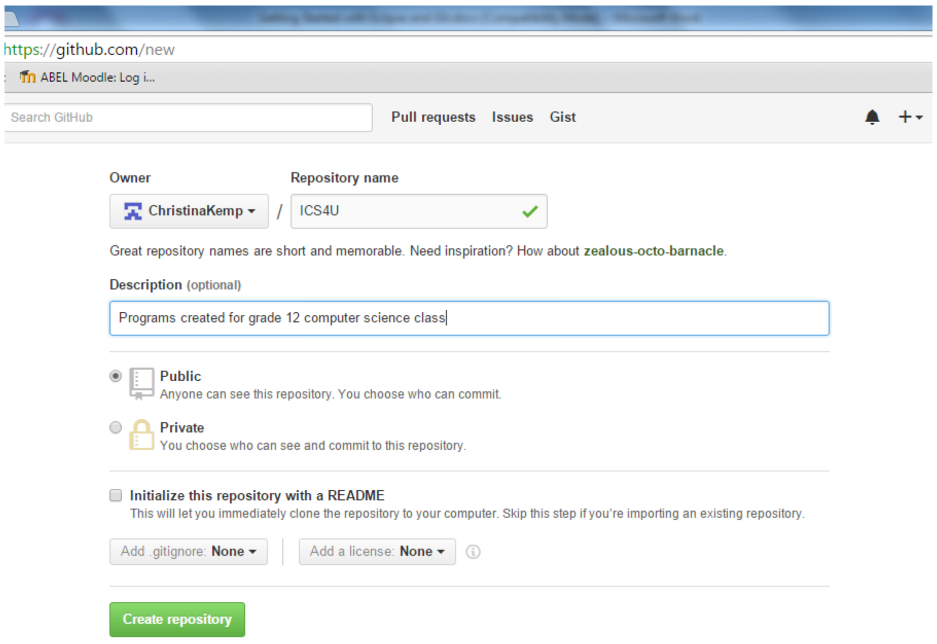
- Use the switch context button in the upper left corner of this page to switch between your personal context (**ChristinaKemp**) and organizations you are a member of.
- After you switch contexts you'll see an organization-focused dashboard that lists out organization repositories and activities.

Repositories you contribute to
Culminating… /project 0

Your repositories 1 + New repository
cemc2015

Subscribe to your news feed

Name your new repository ICS4U and give it a meaningful description.


https://github.com/new

Owner: ChristinaKemp / Repository name: ICS4U

Great repository names are short and memorable. Need inspiration? How about **zealous-octo-barnacle**.

Description (optional): Programs created for grade 12 computer science class

[✓ Public] Anyone can see this repository. You choose who can commit.

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

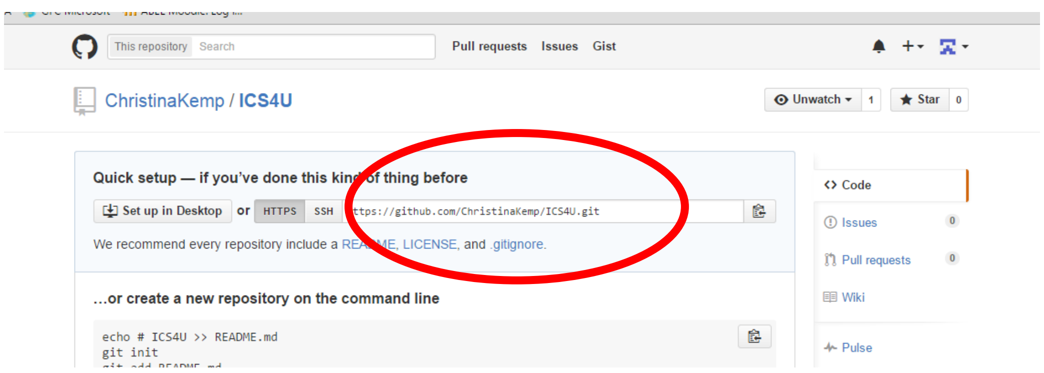
☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

Create repository

Click **Create Repository**.

Once it is created you should see a page that says Quick Setup at the top and a link right below it. Copy the https link given for your repository.


This repository | Search

Pull requests Issues Gist

ChristinaKemp / ICS4U

Unwatch 1 | Star 0

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

Set up in Desktop or HTTPS SSH ttps://github.com/ChristinaKemp/ICS4U.git

We recommend every repository include a README, LICENSE, and .gitignore.

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

```
echo # ICS4U >> README.md
git init
```

Code

Issues 0

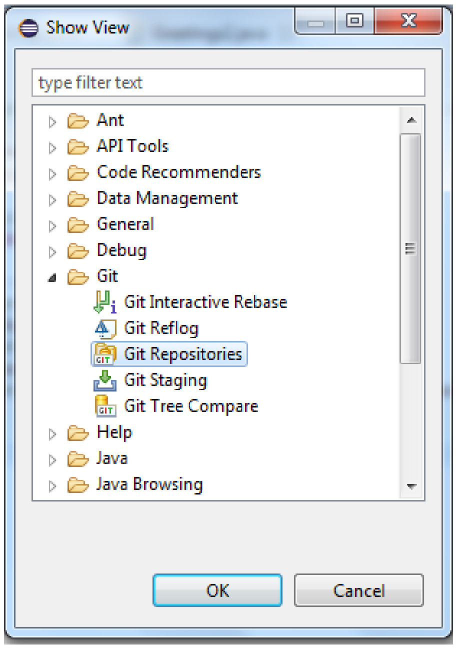
Pull requests 0

Wiki

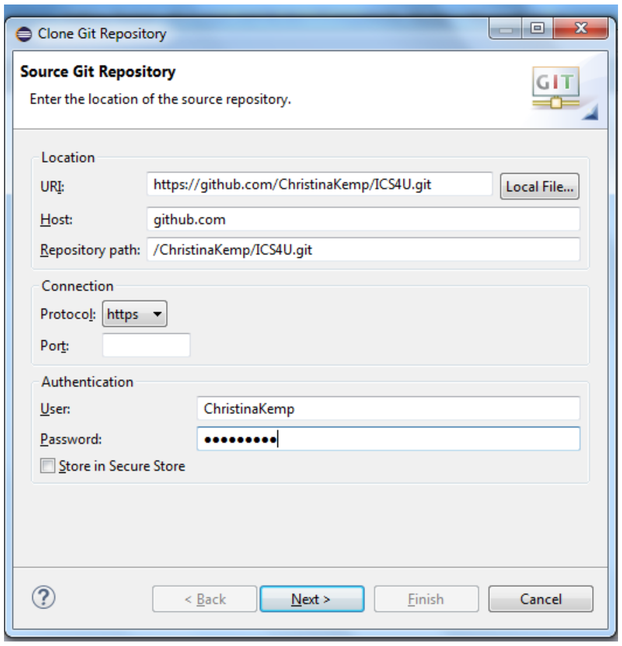
Pulse

ADDING YOUR GIT REPOSITORY TO ECLIPSE

In Eclipse, go to **Window→Show View→Other…** Then select **Git→Git Repositories**



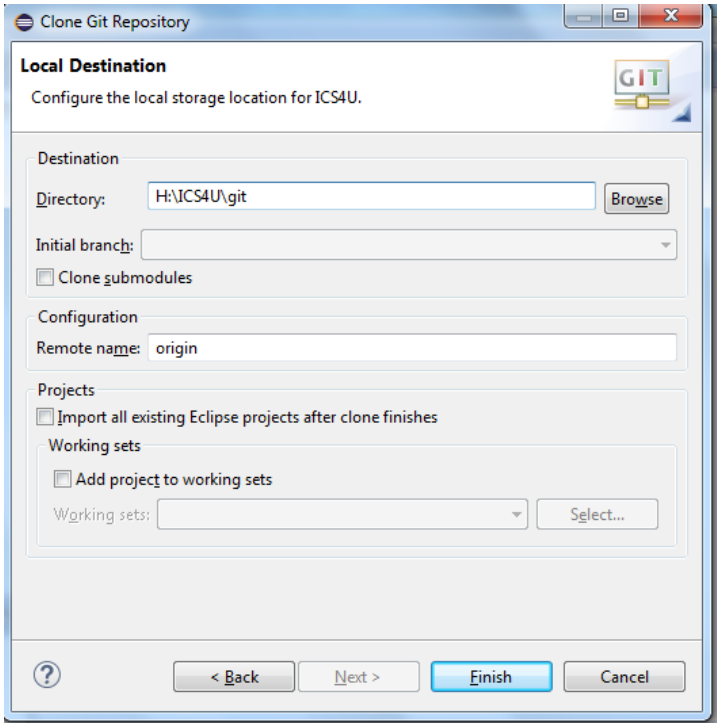
In the Git Repositories box, choose "Clone a Git repository". Paste the link you copied from your GitHub repository into the URI text field and add your username and password for your GitHub account under "Authentication":



URI: https://github.com/ChristinaKemp/ICS4U.git
Host: github.com
Repository path: /ChristinaKemp/ICS4U.git
Protocol: https
Port:
User: ChristinaKemp
Password: •••••••••
☐ Store in Secure Store

Click **Next**, then **Next** again.

Set the directory for the git repository to be stored to be H:\ICS4U\git *(or whatever the home directory is for your school)*:


Clone Git Repository

Local Destination

Configure the local storage location for ICS4U.

Destination

Directory: H:\ICS4U\git

Initial branch:

☐ Clone submodules

Configuration

Remote name: origin

Projects

☐ Import all existing Eclipse projects after clone finishes

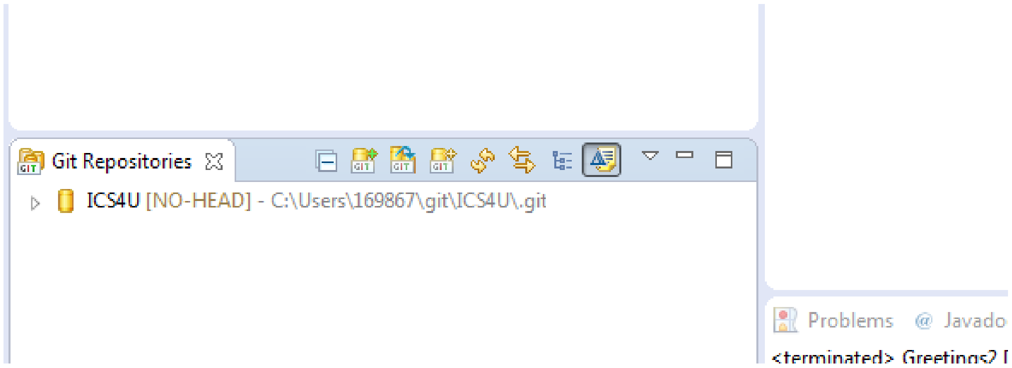
Working sets

☐ Add project to working sets

Working sets:

Click **Finish**.

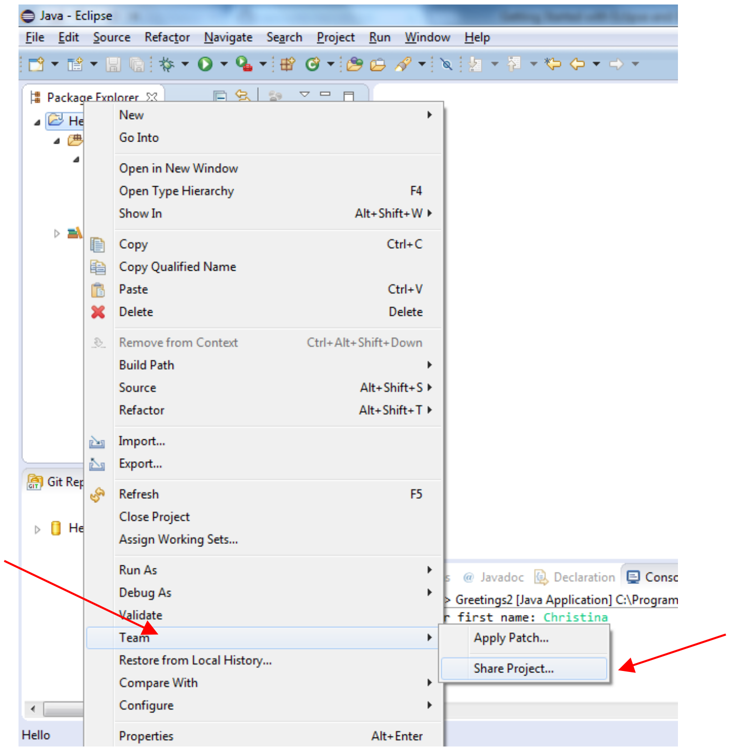
You should now see your repository from GitHub located in the Git Repositories box.


Git Repositories

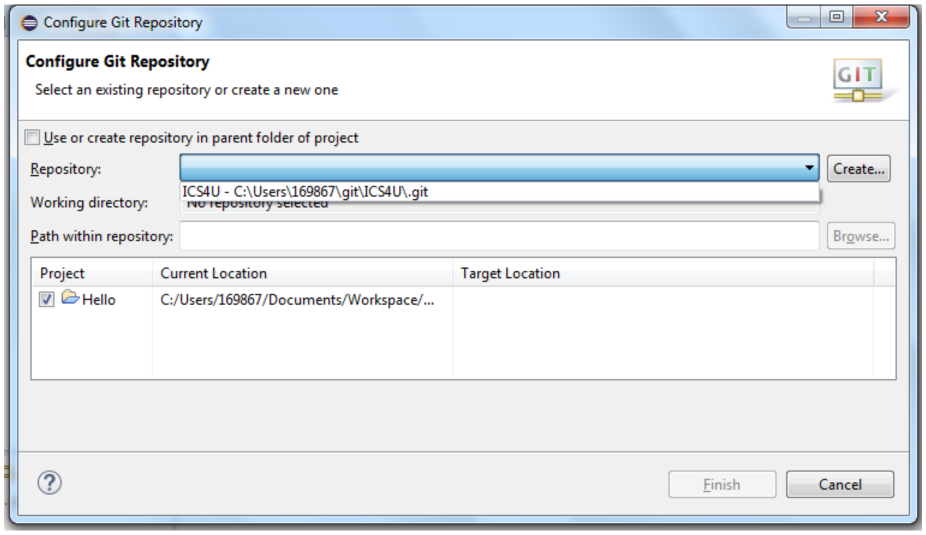
ICS4U [NO-HEAD] - C:\Users\169867\git\ICS4U\.git

ADDING A PROJECT TO YOUR GIT REPOSITORY

To add your Java project to your Git Repository Right Click on the project (Hello) in the Project Explorer, go to **Team→Share Project…**



In the "Configure Git Repository" window click the drop down list next to "Repository" then select the ICS4U repository you added. Ensure your Hello project is selected in the bottom of the window.



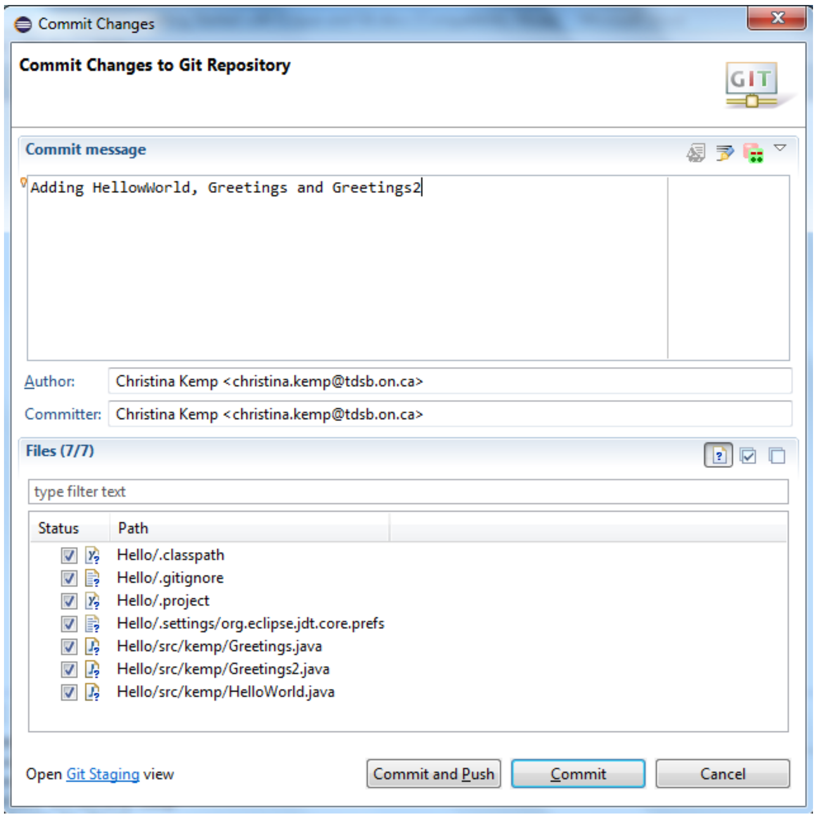
Project	Current Location	Target Location
☑ Hello	C:/Users/169867/Documents/Workspace/...	

Click **Finish**.

You should now see [ICS4U NO-HEAD] next to the project name in the Package Explorer.

To upload the project to github.com right click on the project (Hello), go to **Team→Commit…**

In the "Commit Changes" window add a description about what you are adding. Ensure the Author and Committer are both your name and email address. Select all files in the bottom window.



```
Adding HellowWorld, Greetings and Greetings2
```

Author: Christina Kemp <christina.kemp@tdsb.on.ca>

Committer: Christina Kemp <christina.kemp@tdsb.on.ca>

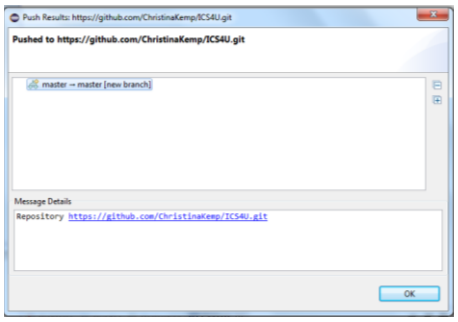
Status	Path
✓	Hello/.classpath
✓	Hello/.gitignore
✓	Hello/.project
✓	Hello/.settings/org.eclipse.jdt.core.prefs
✓	Hello/src/kemp/Greetings.java
✓	Hello/src/kemp/Greetings2.java
✓	Hello/src/kemp/HelloWorld.java

Click **Commit and Push**.

❖ Adding a description is important to help you (and others) know what changes were made each time the code was committed.

When the "Push Branch Master" dialog box opens, click **Next**. If the "Login" window opens enter your GitHub username and password then click Okay. Click **Finish**.

You should get a confirmation message once your files have been pushed:



```
Push Results: https://github.com/ChristinaKemp/ICS4U.git
Pushed to https://github.com/ChristinaKemp/ICS4U.git
master → master [new branch]
Message Details
Repository https://github.com/ChristinaKemp/ICS4U.git
```

Open your browser to your Git repository ICS4U. Click on <> Code on the right side of the page. You should now see your project, “Hello”. You can click through the project to get to your source files, and should see your three java files, HelloWorld, Greetings and Greetings2.

- **Every time you update your java project in Eclipse you simply have to “Commit and Push” and all your changes will appear in GitHub.**

- **If you are working on the same project on another computer you will need to set the second computer up the same way. Make sure you always “Pull” from GitHub before you start working on your project to ensure you have the most recent version on that particular computer, and “Commit and Push” when you are finished.**

- **GitHub can be used to collaborate on one project with multiple people using Organizations. For more information see Mark Hoel’s video here.**