Downloading Eclipse for Java in Windows

This section is for **Windows** users. Mac people can skip it.

Note

In order to reduce the likelihood of compatibility problems, we will taking both Oracle's and Eclipse's advice and will be downloading the **32 bit**, *not* the 64-bit, version of all software. This is not a performance sacrifice, since Java runs "above" the hardware. Many start-up errors are the result of students installing 64-bit versions of one of the packages below.

For very new hardware and software, see my cautionary permission to use 64-bit *everything*at the bottom of this page. Simply make sure you are consistent. If you choose to go with 64-bit, follow all the instructions on this page, but use 64-bit browers, 64-bit JRE and 64-bit Eclipse, rather than my 32-bit suggestions, everywhere. Again - I think 32-bit is safer for the near future.

CMP.1.1 Installing a Java Runtime Environment

Before you can begin Eclipse, you will need a preliminary component called a **Java Runtime Environment (JRE)**. You may already have one installed on your computer, but if you don't or if your **JRE** is not up-to-date or you have the 64-bit **JRE**, you will get the most up-to-date **32-bit JRE** in this step.

- **Let Oracle do an automatic, on-line install of the 32-bit JRE here**:

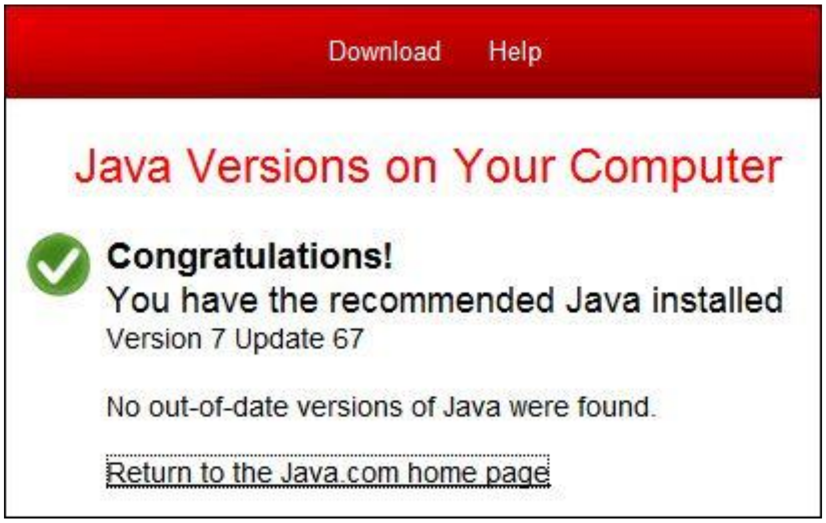
  http://www.java.com/en/index.jsp

  If you follow the recommended instructions, you will have nothing much to decide, Just click the big agree-and-install buttons as they are presented to you.   You will get a **jxpiinstall.exe**installer downloaded to you system, you will execute it, and eventually see the "Java successfully installed" message.

- **Confirm that Java was correctly installed. Go here:**

  http://www.java.com/en/download/testjava.jsp

  and let it run the ***testjava*** applet. It should tell you that you have the newest version, as shown below:



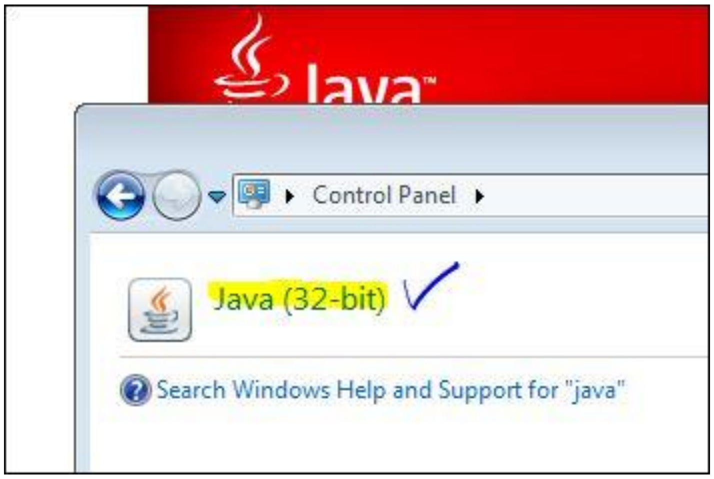
Java Versions on Your Computer

Congratulations!
You have the recommended Java installed
Version 7 Update 67

No out-of-date versions of Java were found.

Return to the Java.com home page

- **Make sure you are running 32-bit Java (***not* 64-bit Java). Oracle gave you what it thought was the right Java for your browser, and this was probably the 32-bit version. You can confirm that by navigating to your ***control panel*** and typing "java" in the search window (right top). You should see:



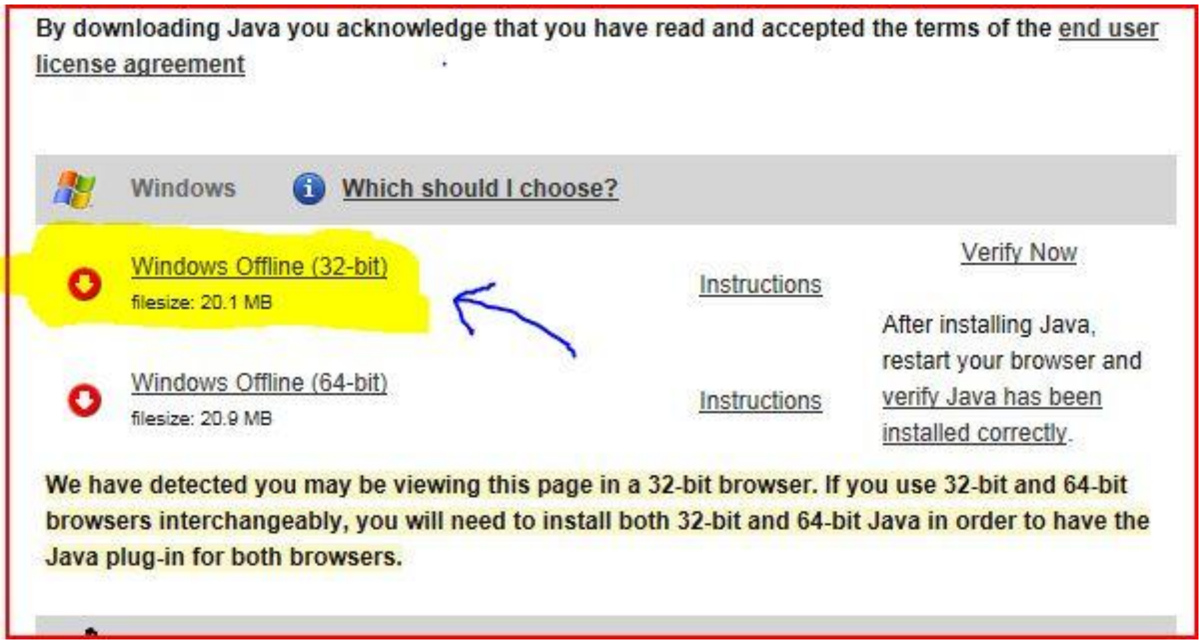
If it only says **Java**, without the "(32-bit)" qualification, it is probably the 64-bit version. (Another way to check is to go to ***control panel*** → ***programs and features*** and see which

Java is listed. In this case, if it says nothing, it's probably the 32 bit version, but if it says "(64-bit)" then you have the 64-bit version.)

- **If Oracle gave you the 64-bit version, you can do a manual install of the 32-bit JRE**. It's up to you whether to keep the 64-bit JRE in addition to the 32-bit; if you are an advanced user and like to use a 64-bit browser (rare), you can have both versions installed at once. You can get the 32-bit version here:

  http://www.java.com/en/download/manual.jsp.

  You will be able to force the choice to 32-bit, even if it wants to give you the 64-bit version:



By downloading Java you acknowledge that you have read and accepted the terms of the end user license agreement

After installing Java, restart your browser and verify Java has been installed correctly.

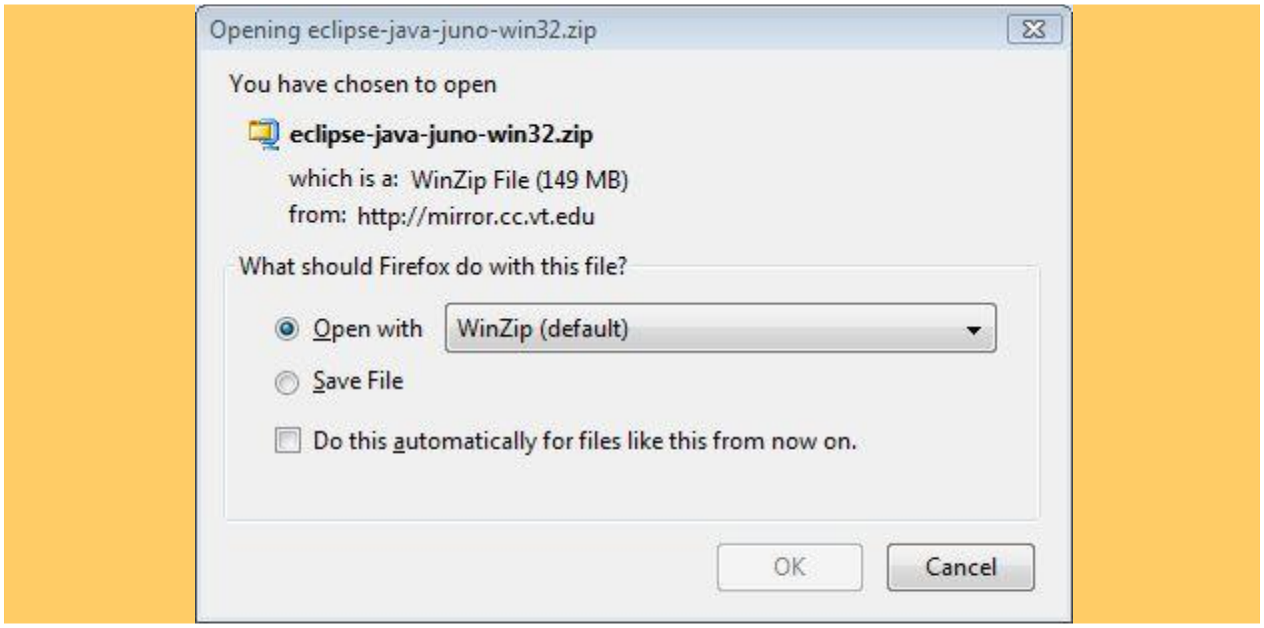
We have detected you may be viewing this page in a 32-bit browser. If you use 32-bit and 64-bit browsers interchangeably, you will need to install both 32-bit and 64-bit Java in order to have the Java plug-in for both browsers.

CMP.1.2 Getting Eclipse

Now you are ready to actually download ***Eclipse***.

Unzip Utility Required

You will need some kind of unzipping utility. This is included in most Windows versions or you can get something like ***WinZip*** (free). When you download these files, if you are given the option, select **Open** (not save to disk) in order to save a step, e.g.


Opening eclipse-java-juno-win32.zip

You have chosen to open

eclipse-java-juno-win32.zip

which is a: WinZip File (149 MB)

from: http://mirror.cc.vt.edu

What should Firefox do with this file?

[✓ Open with] WinZip (default)

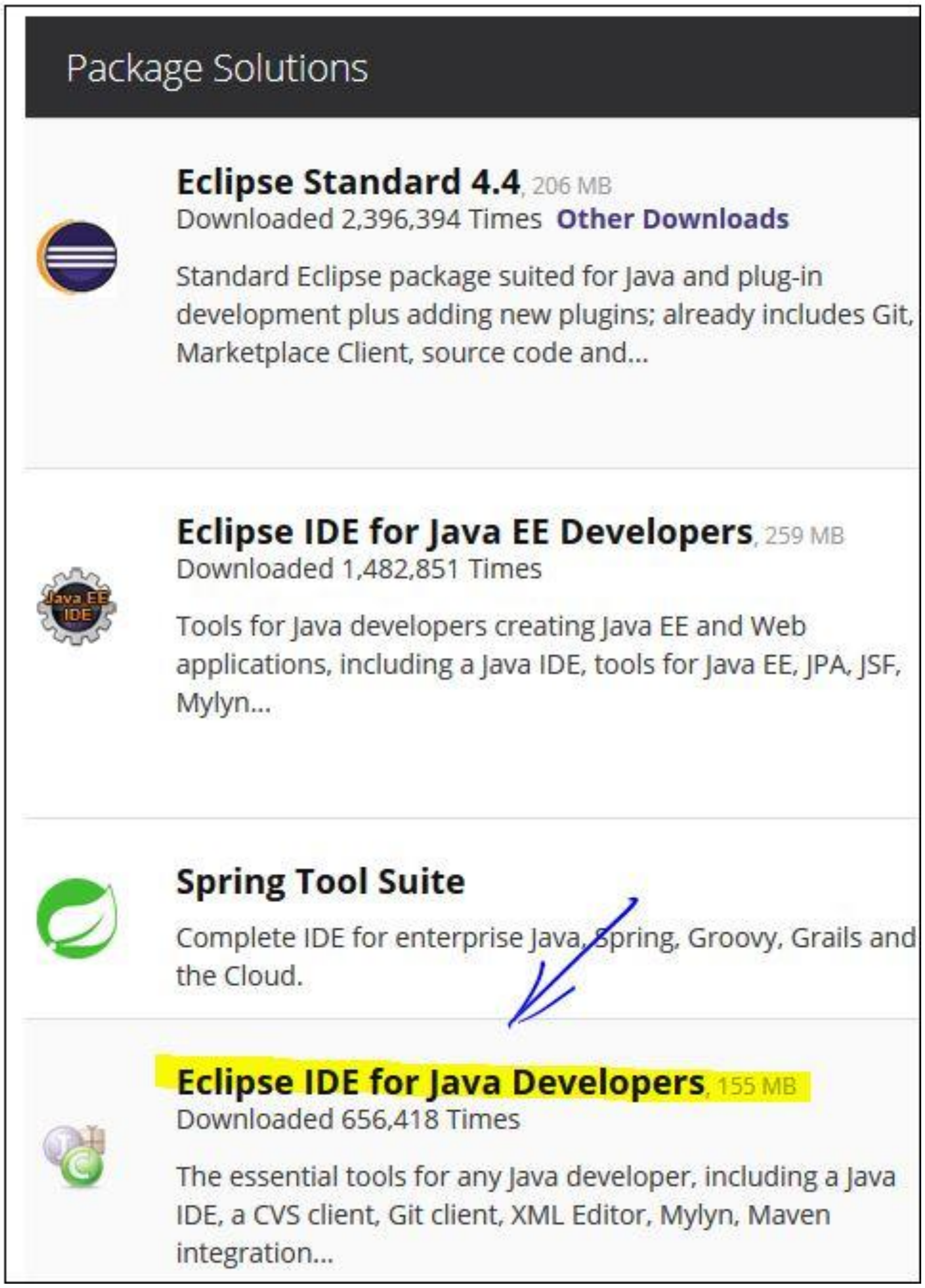
Save File

Do this automatically for files like this from now on.

- Go to the ***Eclipse*** web page:

http://www.eclipse.org/

- Click on ***Download***, then on ***Eclipse IDE for Java Developers*** (one of the first five or so choices in the list).



Package Solutions

Eclipse Standard 4.4, 206 MB
Downloaded 2,396,394 Times **Other Downloads**

Standard Eclipse package suited for Java and plug-in development plus adding new plugins; already includes Git, Marketplace Client, source code and...

Eclipse IDE for Java EE Developers, 259 MB
Downloaded 1,482,851 Times

Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn...

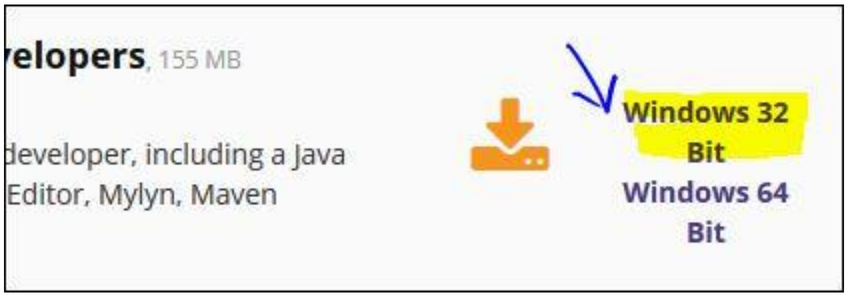
Spring Tool Suite

Complete IDE for enterprise Java, Spring, Groovy, Grails and the Cloud.

Eclipse IDE for Java Developers, 155 MB
Downloaded 656,418 Times

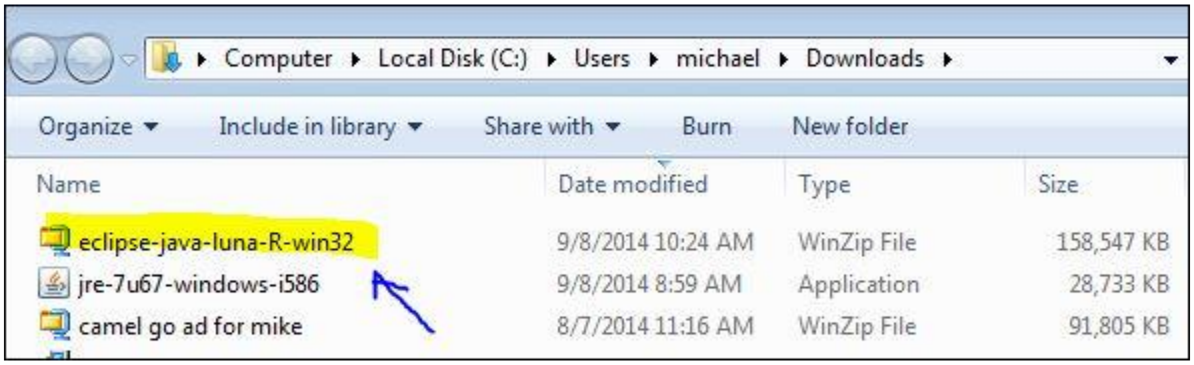
The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration...

Make sure you get the version that matches your Java. If you followed my earlier recommendation, you downloaded ***32-bit Java*** so you should match that with the ***32-bit Eclipse***, but if you got a 64-bit Java, you'll want the 64-bit Eclipse. I chose the 32-bit here:



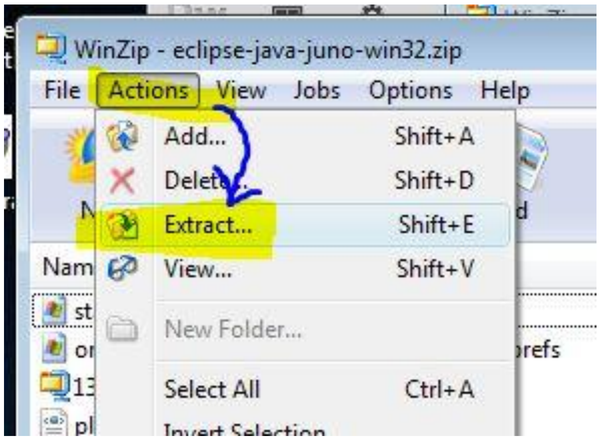
After selecting your download site it will take a few minutes to download.

- If you are given the option, select ***open,*** but if it ***saves to disk*** automatically, you will have to manually unzip it, which is just an extra step. In that case, open the containing folder and click to unzip:

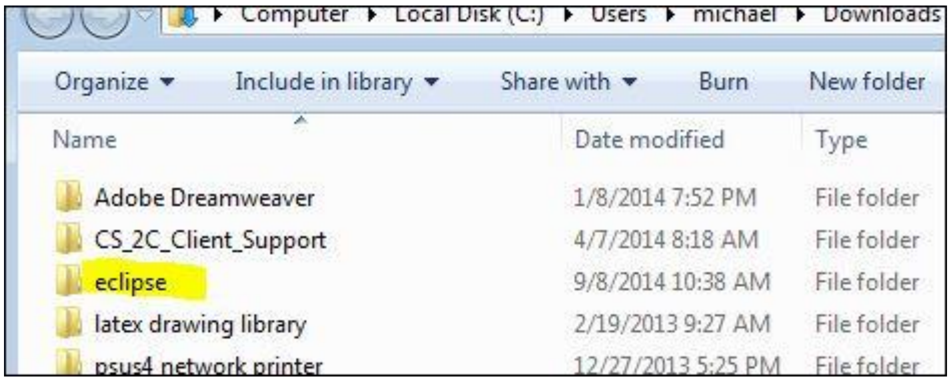


Name	Date modified	Type	Size
eclipse-java-luna-R-win32	9/8/2014 10:24 AM	WinZip File	158,547 KB
jre-7u67-windows-i586	9/8/2014 8:59 AM	Application	28,733 KB
camel go ad for mike	8/7/2014 11:16 AM	WinZip File	91,805 KB

Extract (**Action → Extract** and check ***all files and folders***) to any folder you wish on your system:



→



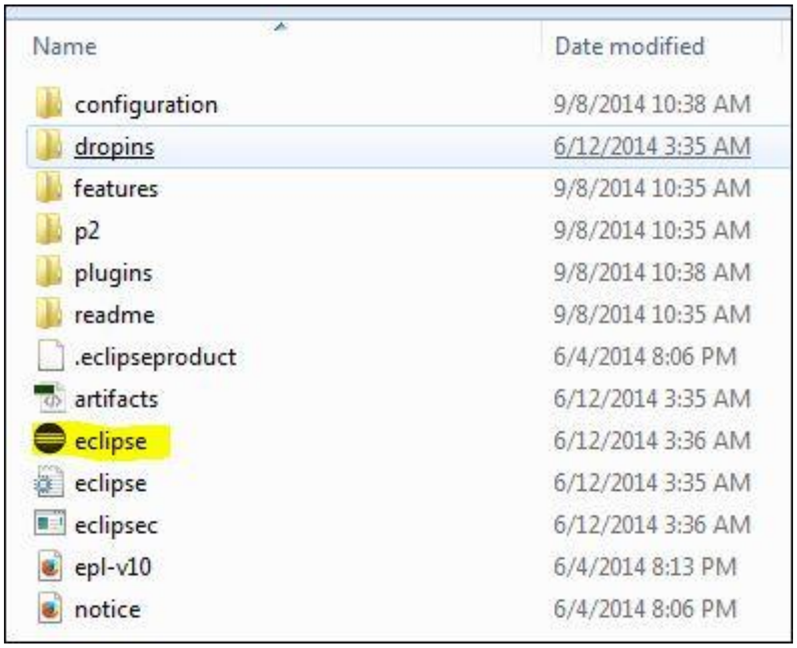
Computer ▸ Local Disk (C:) ▸ Users ▸ michael ▸ Downloads

Organize ▾ Include in library ▾ Share with ▾ Burn New folder

Name	Date modified	Type
Adobe Dreamweaver	1/8/2014 7:52 PM	File folder
CS_2C_Client_Support	4/7/2014 8:18 AM	File folder
eclipse	9/8/2014 10:38 AM	File folder
latex drawing library	2/19/2013 9:27 AM	File folder
psus4 network printer	12/27/2013 5:25 PM	File folder

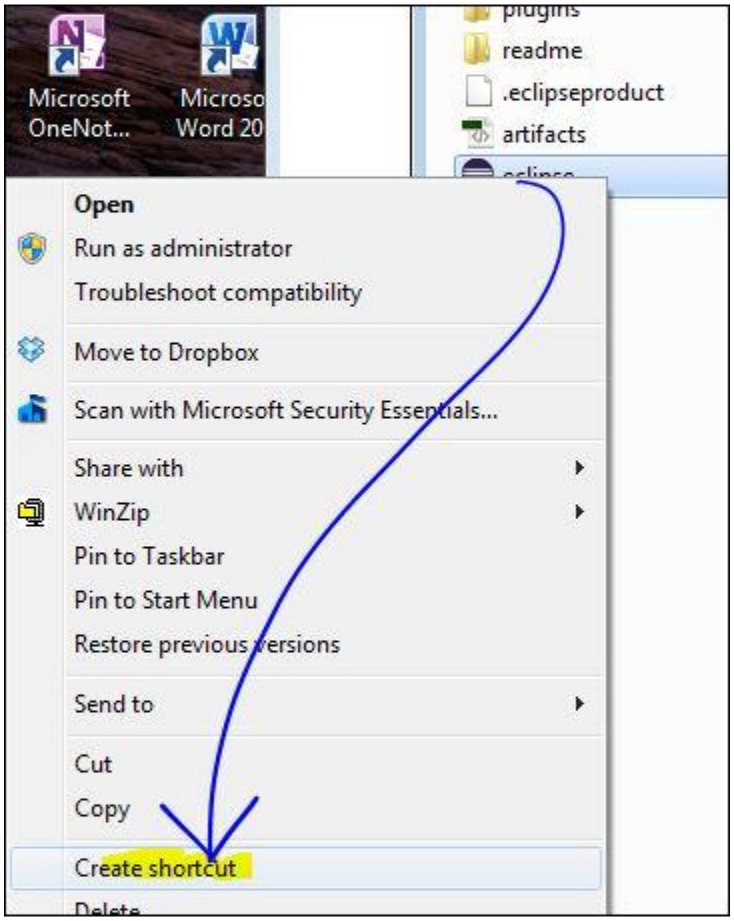
CMP.1.3 Making a Shortcut (or Menu Item)

- Open the ***Eclipse folder*** and you will see the ***Eclipse application*** contained therein.

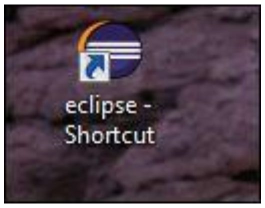


Name	Date modified
configuration	9/8/2014 10:38 AM
dropins	6/12/2014 3:35 AM
features	9/8/2014 10:35 AM
p2	9/8/2014 10:35 AM
plugins	9/8/2014 10:38 AM
readme	9/8/2014 10:35 AM
.eclipseproduct	6/4/2014 8:06 PM
artifacts	6/12/2014 3:35 AM
eclipse	6/12/2014 3:36 AM
eclipse	6/12/2014 3:35 AM
eclipsec	6/12/2014 3:36 AM
epl-v10	6/4/2014 8:13 PM
notice	6/4/2014 8:06 PM

- Without moving that application from its folder, make a ***shortcut***, and move the ***shortcut*** (*not the original*) to your desktop for easier access:



→

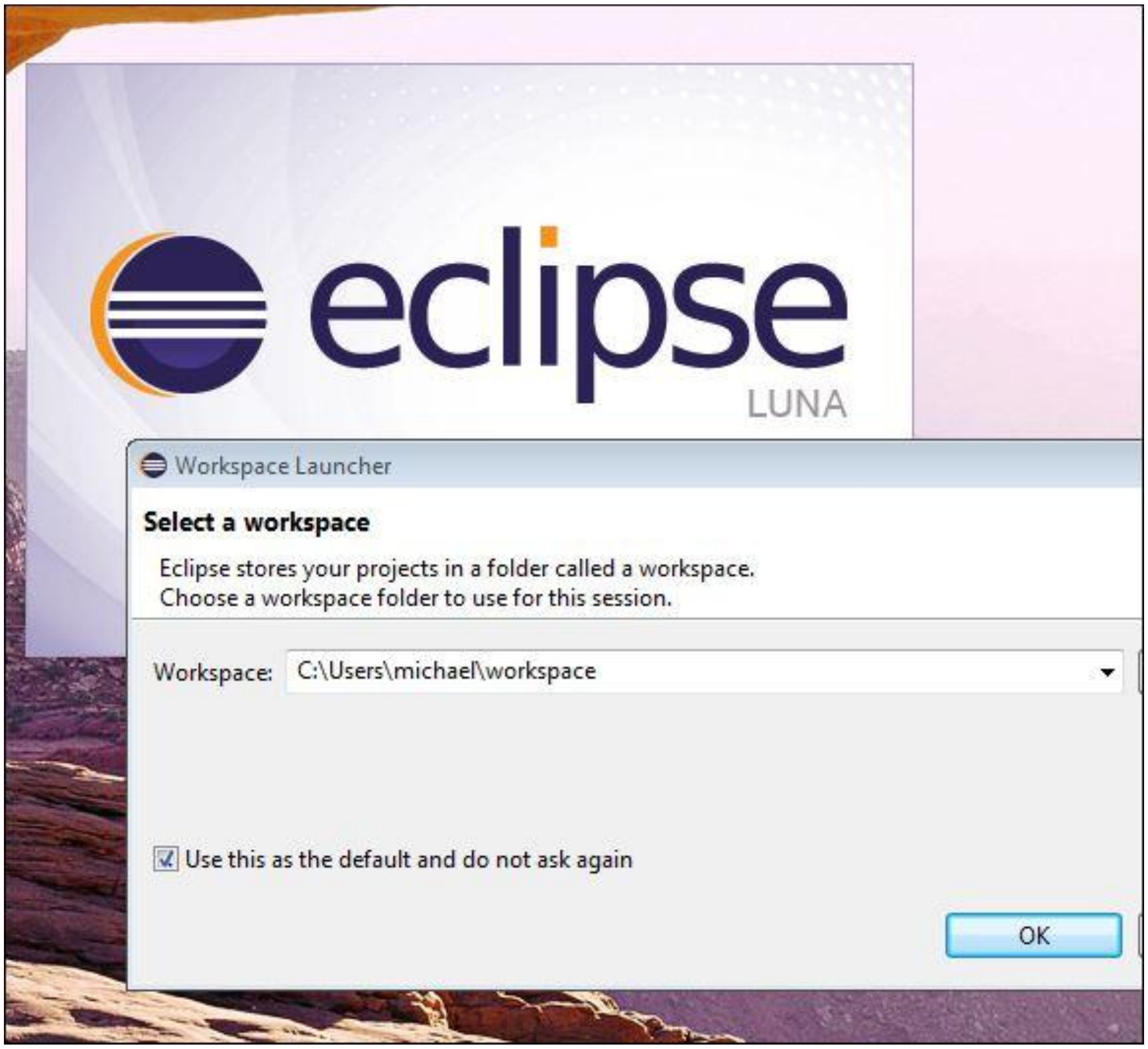


Extreme Caution

Do not try to move the Eclipse application to the desktop. Move only the shortcut you just made. If you accidentally move the application, move it back and move the shortcut to the desktop.

CMP.1.4 Launch (and Exit) Eclipse

- Confirm that your installation is working by double-clicking on your desktop short-cut. You should get a preliminary window (check the box):



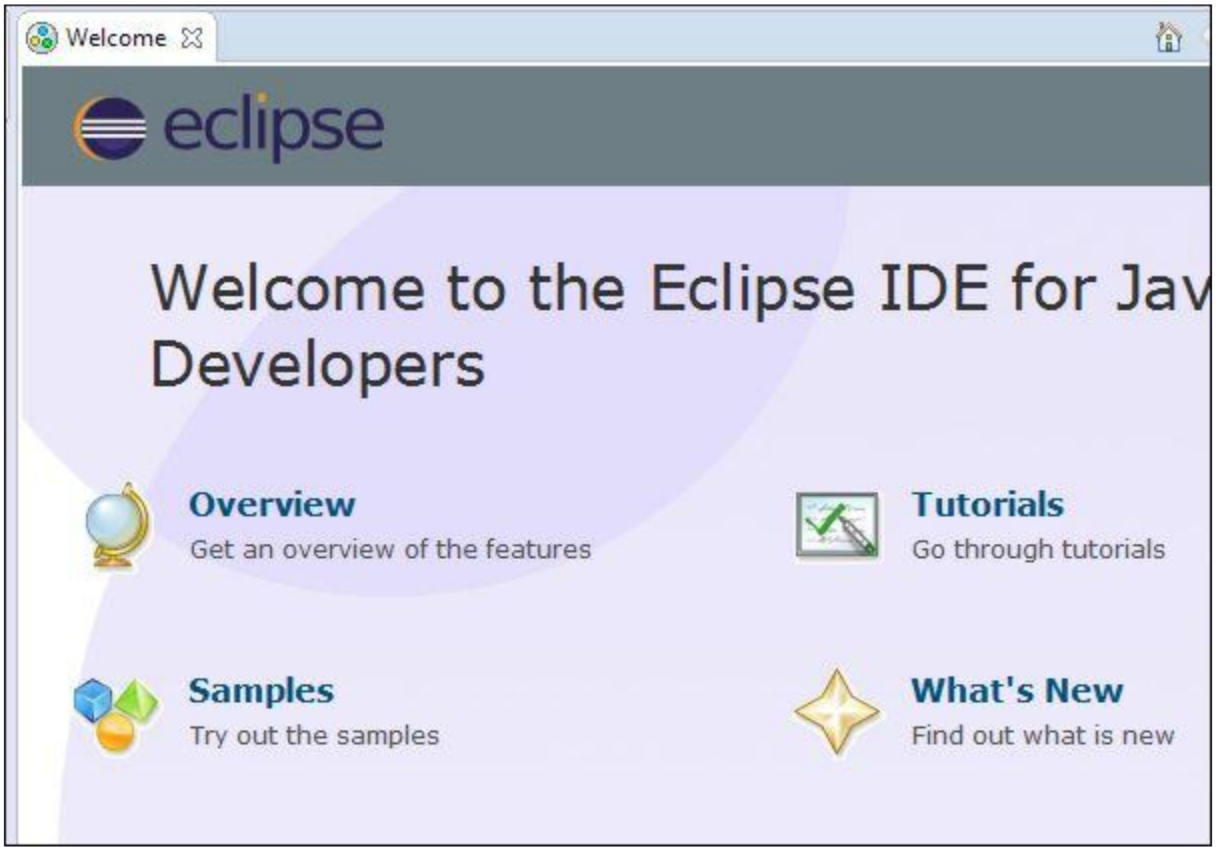
Select a workspace

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

Workspace: C:\Users\michael\workspace

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

followed by the Eclipse welcome screen if it is your first time using any version of Eclipse:



If you have used Eclipse in the past, you will probably see a project perspective rather than this welcome screen.

- Now, ***close Eclipse before you get into trouble***. In a few pages, I'll show you what to do next.

Congratulations. You have completed the installation of ***Eclipse for Java development***. Next, we configure it so it is set up for the remainder of the course.

IMPORTANT

Do not try to create or run a program yet. We have to configure the compiler in the next**section**, and then work through another **module** detailing how to compile and run. You must complete *all* the **module** reading in the first week and only start a new program when I direct you to do so, or you will have questions that are answered in this week's sections.

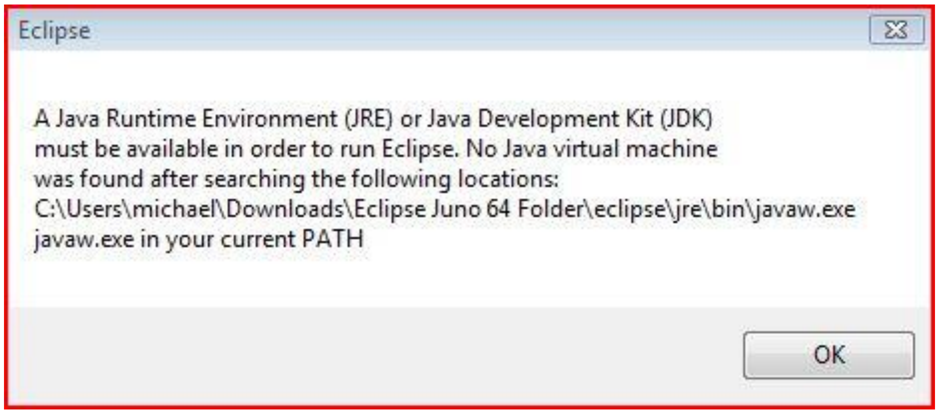
CMP.1.5 Some 32 vs. 64 Bit Notes

For Advanced Users or Folks with the Newest "Everything"

If you *do* have the 64-bit JRE (because you like to run the optional 64-bit browsers), and *want to try it*, go ahead and download the 64-bit Eclipse. If it works, great. If not, my only advice is to drop back to 32-bit version of both.

Detecting a 32-64 Bit Conflict

You will know if you have a 32-bit Java and a 64-bit Eclipse, or vice versa, if you get this error message:



A Java Runtime Environment (JRE) or Java Development Kit (JDK) must be available in order to run Eclipse. No Java virtual machine was found after searching the following locations:
C:\Users\michael\Downloads\Eclipse Juno 64 Folder\eclipse\jre\bin\javaw.exe
javaw.exe in your current PATH

If you get that message ... that's right: go back to the beginning and get 32-bit version of everything.

Section 2 - Downloading Eclipse for Java on the Mac

This section is for **Mac** users. Windows people can skip it.

CMP.2.1 Confirming the Java Runtime Environment

Before you can begin Eclipse, you will need a preliminary component called a **Java Runtime Environment (JRE)**. Good news for most Mac users: this is a non-step. Apple has its own **JRE** (not from Oracle) included as part of OS X. By keeping your system X software up-to-date, you may have the most recent JRE available on a Mac.

Not-so-good news for Mac users: the Mac JRE is often a generation behind the Oracle/MS Windows version. However, we will never need something that is only in the newest JRE, so that won't matter to us.

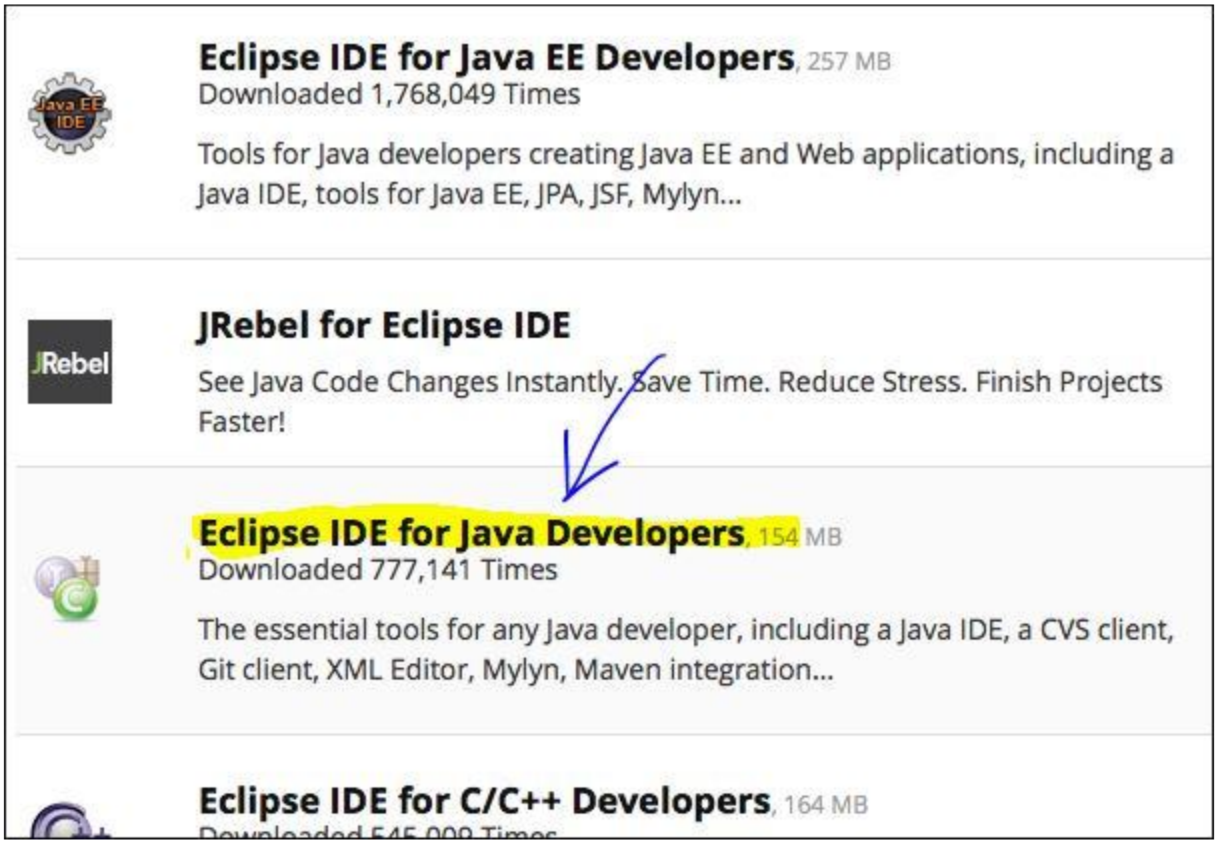
CMP.2.2 Getting Eclipse (Mac)

Now you are ready to actually download ***Eclipse***.

- Go to the ***Eclipse*** web page:

  http://www.eclipse.org/

- Click on ***Download***, then on ***Eclipse IDE for Java Developers*** (one of the first five or so choices in the list).



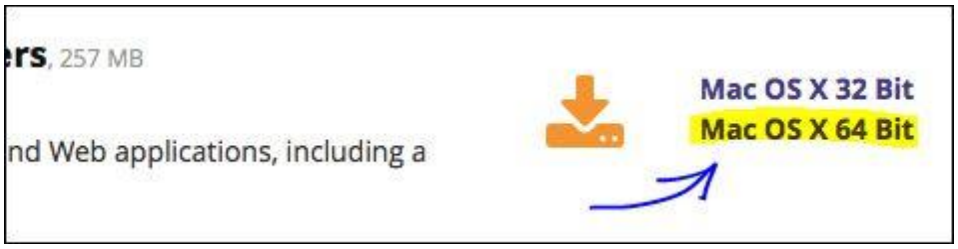
Eclipse IDE for Java EE Developers, 257 MB
Downloaded 1,768,049 Times
Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn...

JRebel for Eclipse IDE
See Java Code Changes Instantly. Save Time. Reduce Stress. Finish Projects Faster!

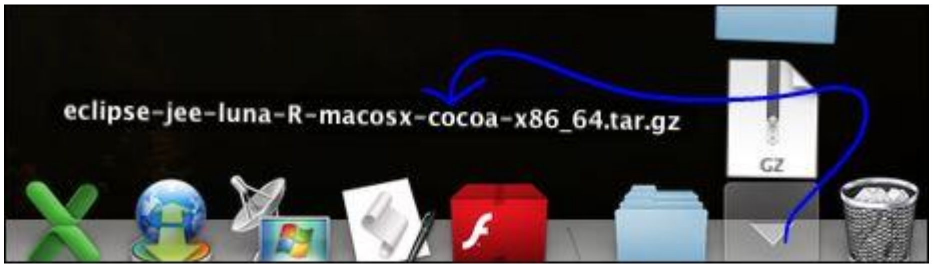
Eclipse IDE for Java Developers, 154 MB
Downloaded 777,141 Times
The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration...

Eclipse IDE for C/C++ Developers, 164 MB

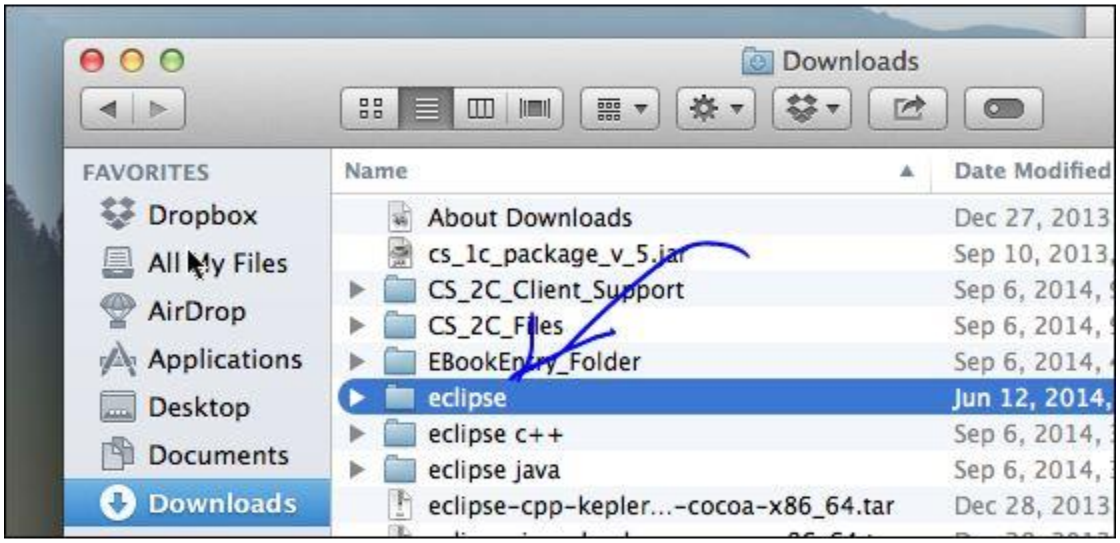
Make sure you get the 64-bit version to optimize for Mac OS X:



- After a few minutes to an hour, you'll see the Eclipse .tar file in your downloads stack. Click it to expand the Eclipse Directory:



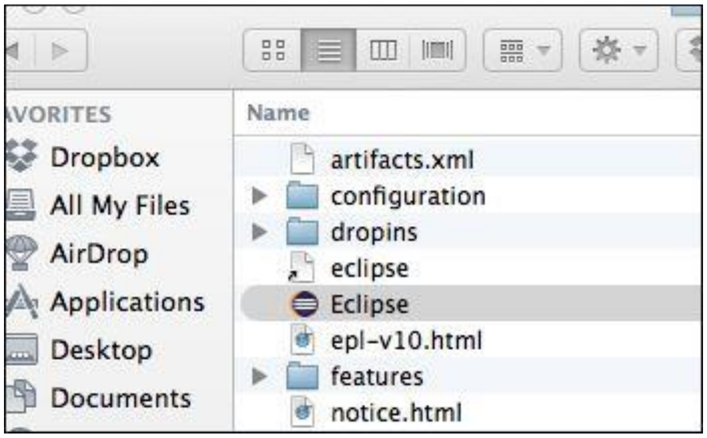
→



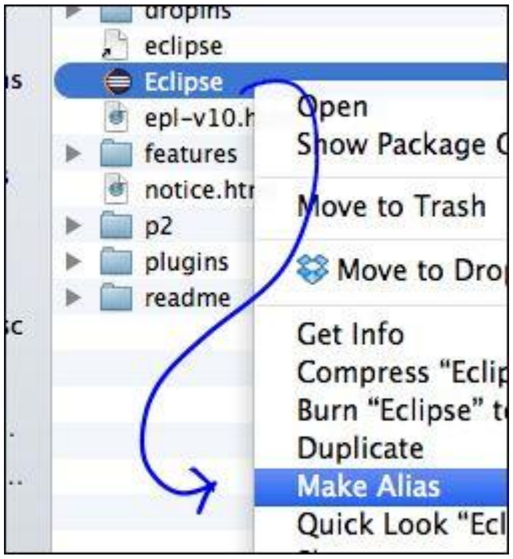
Name	Date Modified
About Downloads	Dec 27, 2013
cs_1c_package_v_5.jar	Sep 10, 2013
CS_2C_Client_Support	Sep 6, 2014,
CS_2C_Files	Sep 6, 2014,
EBookEntry_Folder	Sep 6, 2014,
eclipse	Jun 12, 2014,
eclipse c++	Sep 6, 2014,
eclipse java	Sep 6, 2014,
eclipse-cpp-kepler...-cocoa-x86_64.tar	Dec 28, 2013

CMP.2.3 Making an Alias on Your Desktop

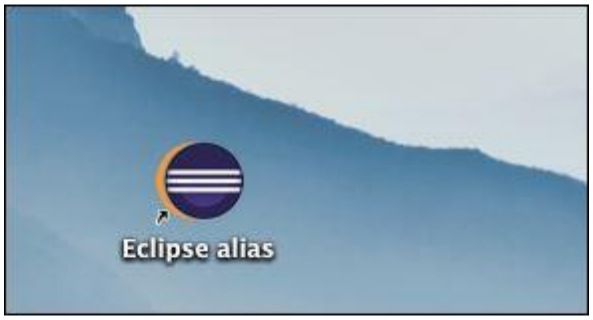
- Open the ***Eclipse folder*** and you will see the ***Eclipse application*** contained therein.



- Without moving that application from its folder, make an ***alias***, and move the ***alias*** (*not the original*) to your desktop for easier access:



→



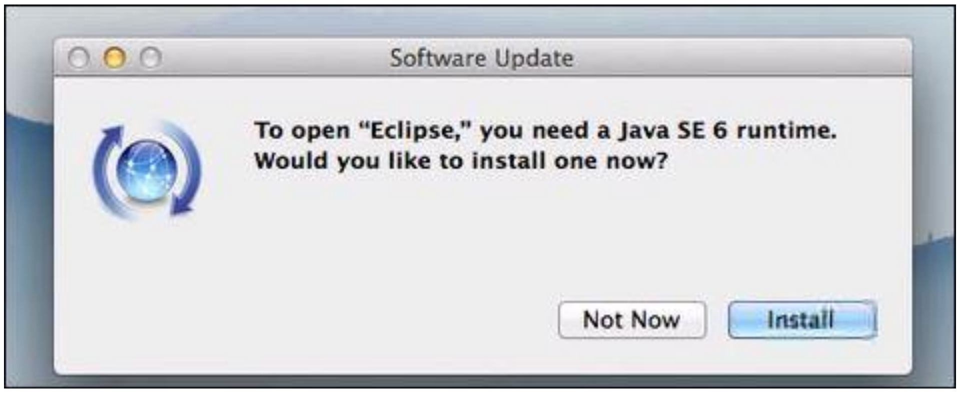
Extreme Caution

Do not try to move the Eclipse application to the desktop. Move only the alias you just made. If you accidentally move the application, move it back and move the alias to the desktop.

CMP.1.4 Launch (and Exit) Eclipse

Confirm that your installation is working by double-clicking on your desktop short-cut.

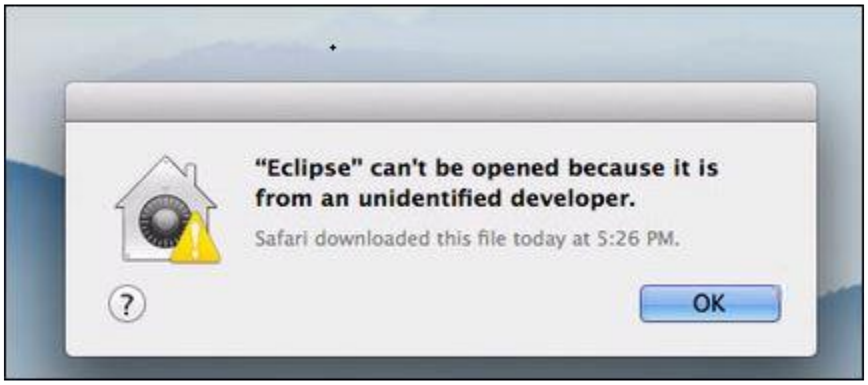
The first minor bump you may have to negotiate is that you get an alert that looks like this:



To open "Eclipse," you need a Java SE 6 runtime. Would you like to install one now?

Just click **Install** and wait. You will never have to do this again for **Eclipse**.

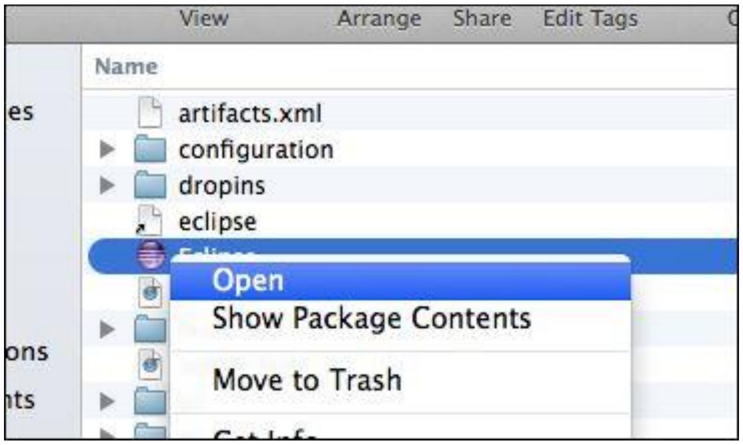
When **Java SE 6** is installed you can try to open **Eclipse** a second time. I say *try*, because you may hit another bump:



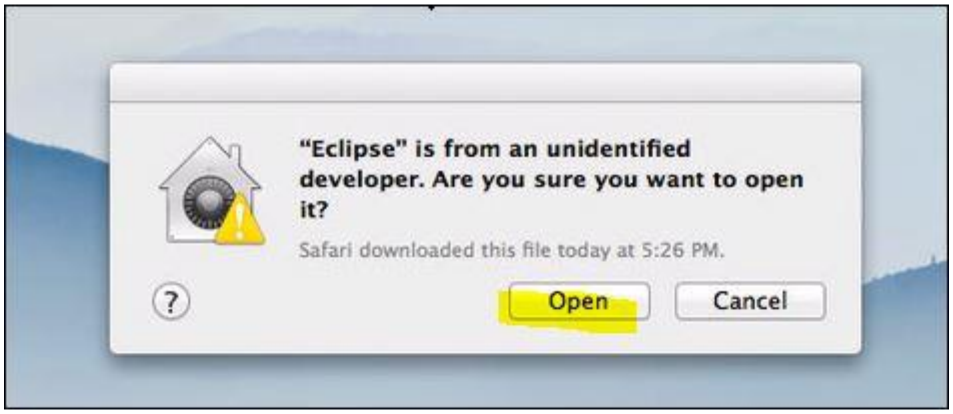
"Eclipse" can't be opened because it is from an unidentified developer.

Safari downloaded this file today at 5:26 PM.

To fix this one (once and for all) find the original **Eclipse** application on your desktop (or in the **Eclipse** folder) and**right-click** → **Open**:

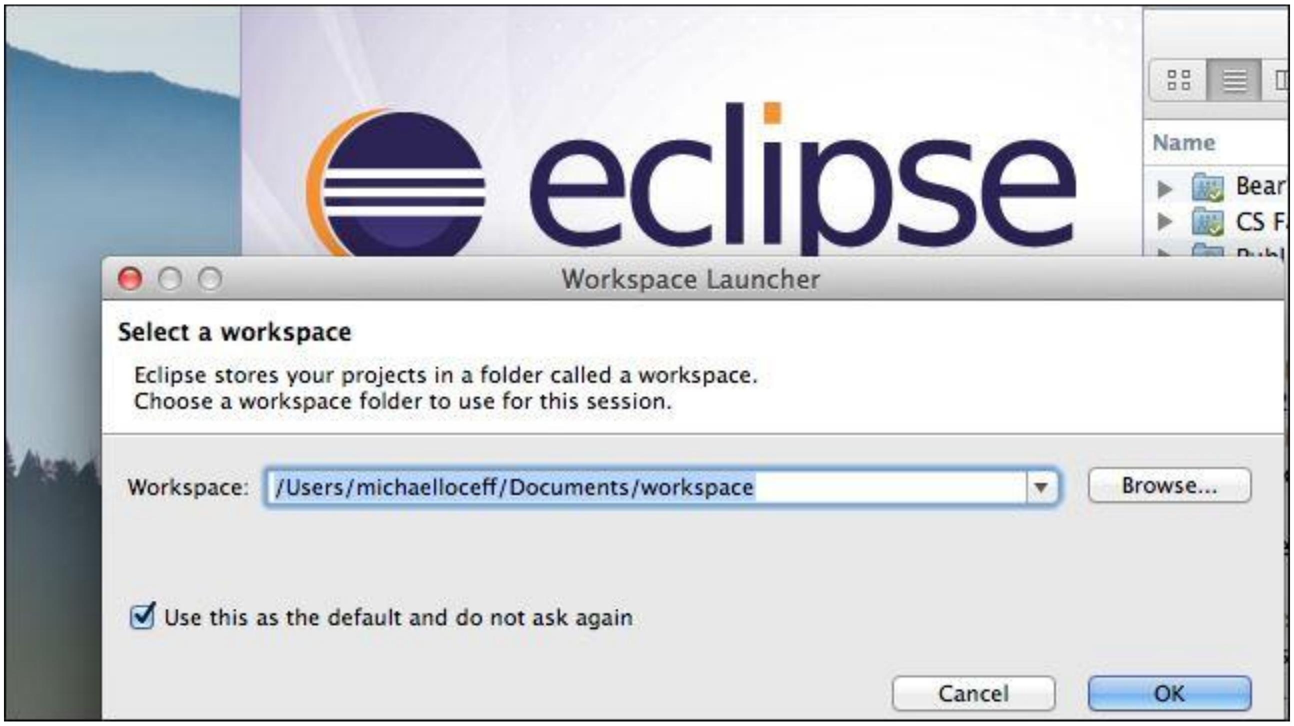


You will get a dialog that allows you to override the warning, and your Mac will remember that for all time:


"Eclipse" is from an unidentified developer. Are you sure you want to open it?

Safari downloaded this file today at 5:26 PM.

Once launched, **Eclipse** will show you a ***Select Workspace*** window (***check the box*** and click **OK**, so it never shows you that window again) …



Workspace Launcher

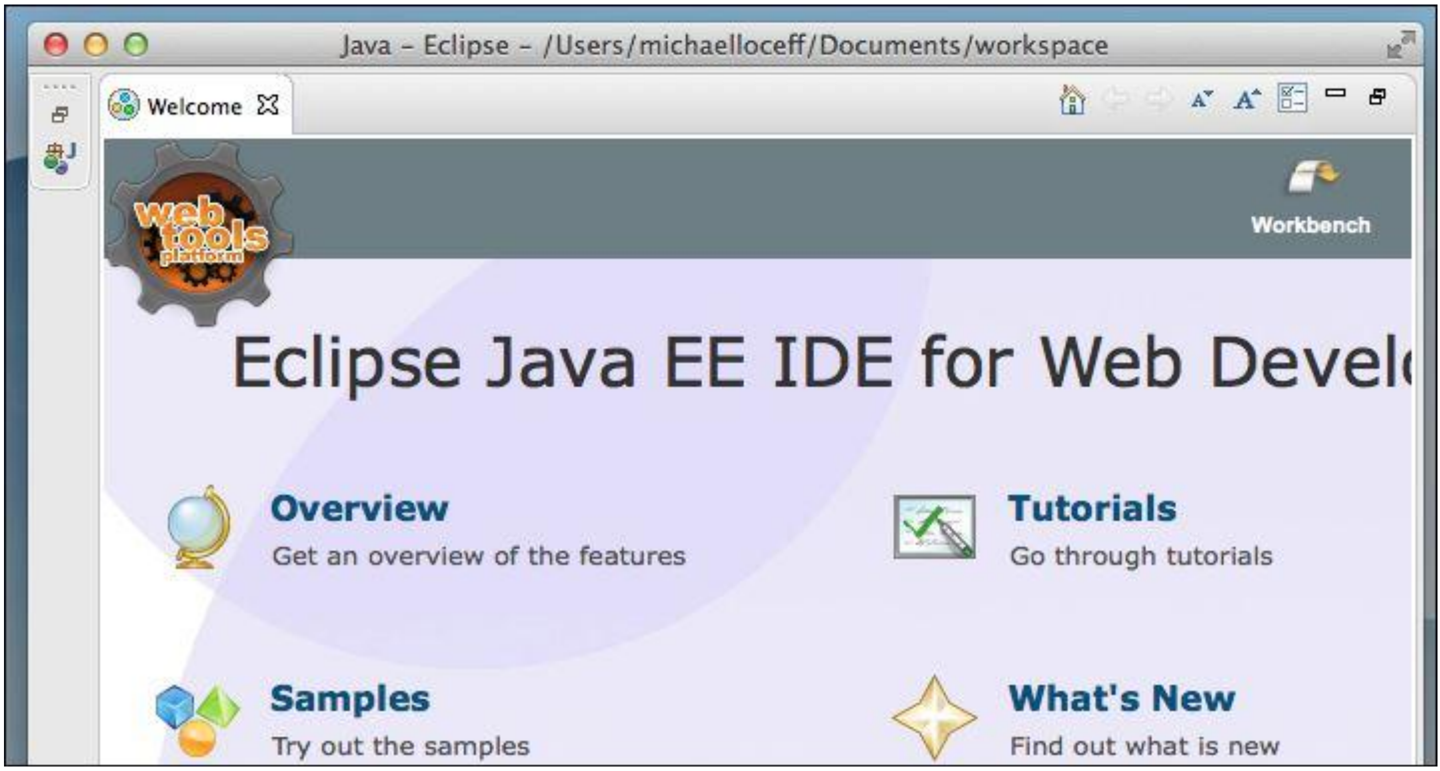
Select a workspace

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

Workspace: /Users/michaelloceff/Documents/workspace

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

… followed by the **Eclipse** welcome screen if it is your first time using any version of **Eclipse**:



Java - Eclipse - /Users/michaelloceff/Documents/workspace

Eclipse Java EE IDE for Web Devel

Overview Get an overview of the features

Tutorials Go through tutorials

Samples Try out the samples

What's New Find out what is new

If you have used **Eclipse** in the past, you will probably see a project perspective rather than this welcome screen.

Now, ***close Eclipse before you get into trouble***. In a few pages, I'll show you what to do next.

Congratulations. You have completed the installation of ***Eclipse for Java development***. Next, we configure it so it is set up for the remainder of the course.

IMPORTANT

Do not try to create or run a program yet. We have to configure the compiler in the next**section**, and then work through another **module** detailing how to compile and run. You must complete *all* the **module** reading in the first week and only start a new program when I direct you to do so, or you will have questions that are answered in this week's sections.

Section 3 - Configuring Eclipse for the Correct Java Version

CMP.3.1 The Latest Java Version

You have installed the latest version of Java on your computer, and you want to make sure you are *using* that version. (As it turns out, you can ask Eclipse to work with an earlier version of Java than you installed. You might do that, for example, if you intended to distribute your program to an audience that had on an older Java platform. However, we will gladly use the latest and greatest, so let's see how to make sure that happens.)

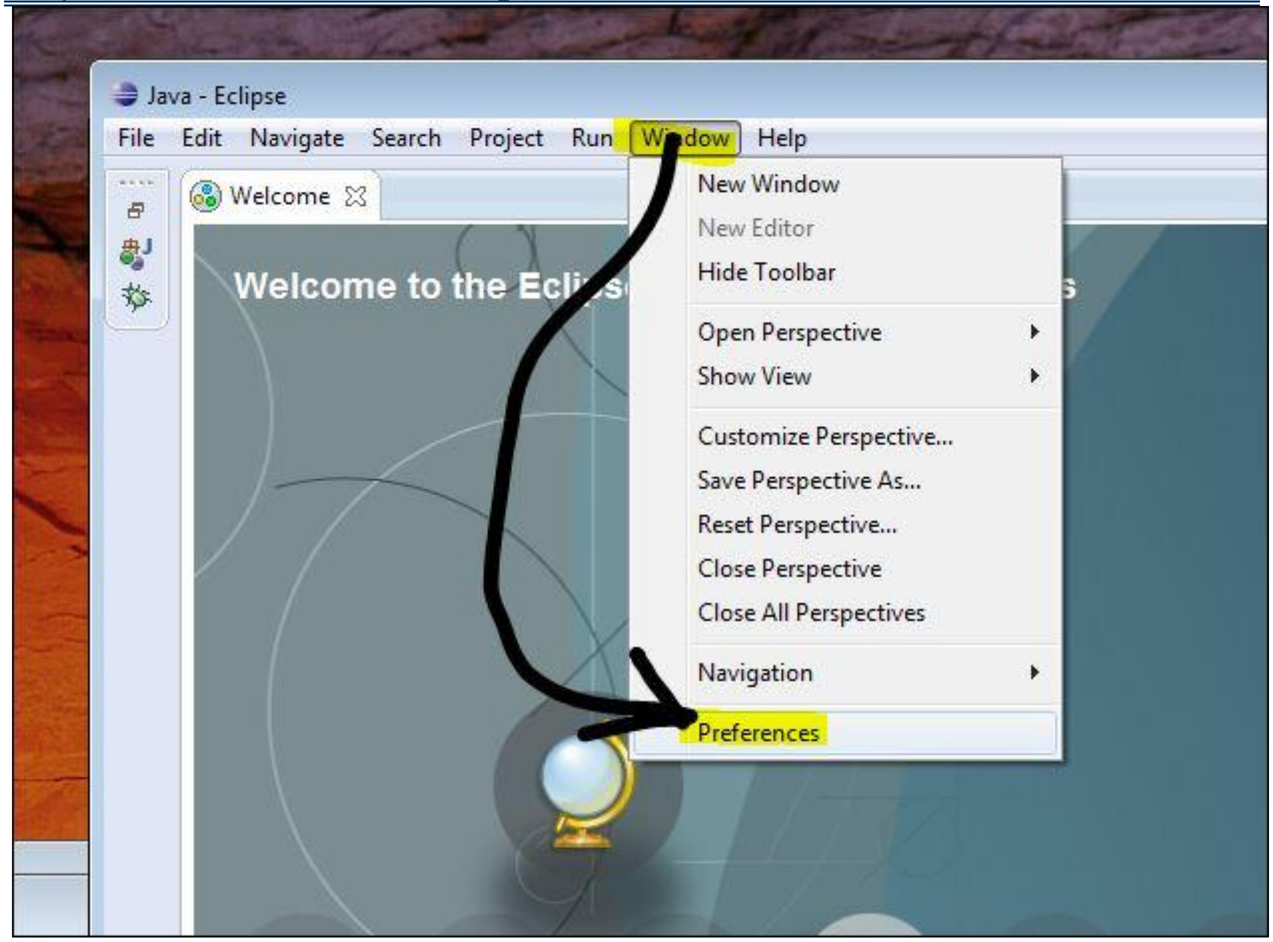
Start Eclipse

Find your **Eclipse** icon and launch it. You will see the **Eclipse** menu along the top (pics below). We will be tweaking the***preferences***.

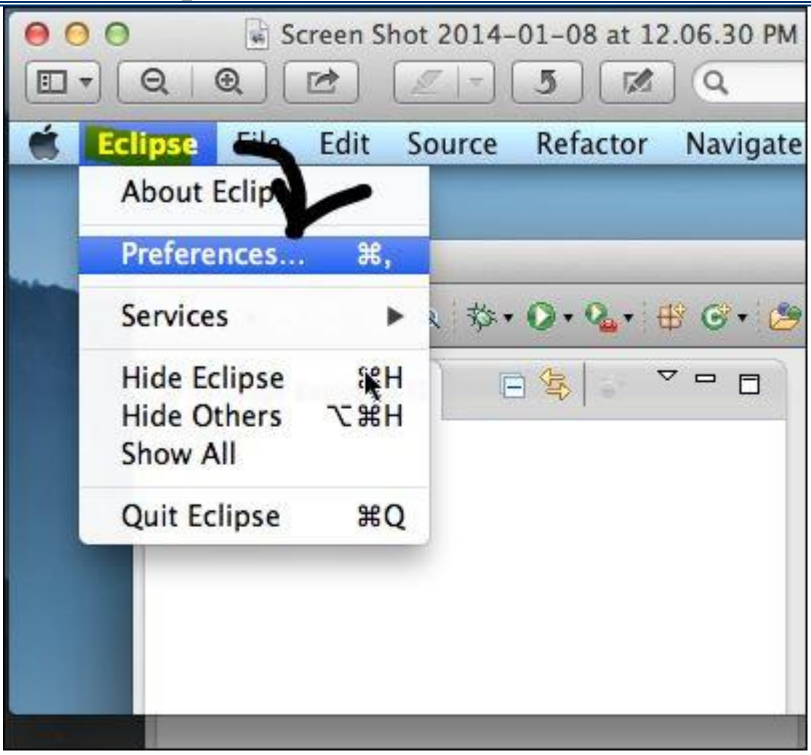
Preferences in Mac and Windows

When I tell you to go to the **Preferences** menu, you do so on a **Mac** by going to **Eclipse** → **Preferences**, while on a **Windows OS**, you get there via **Window** → **Preferences.**

Preferences in Windows Eclipse:

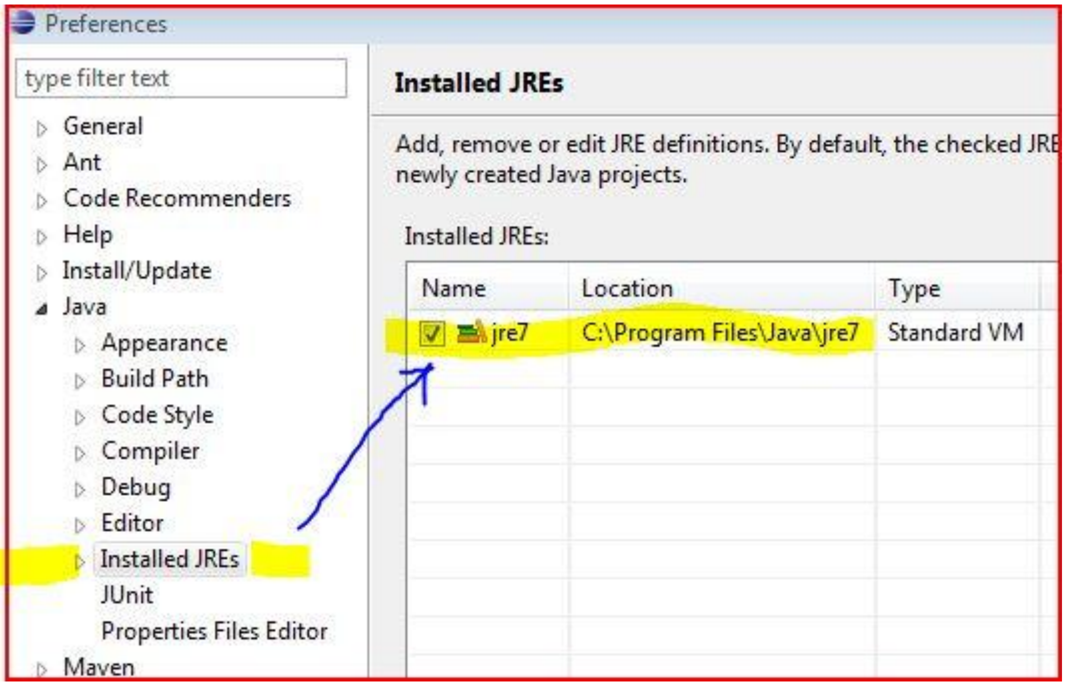


Preferences in Mac Eclipse:



CMP.3.2 Confirm that Eclipse Sees the Latest Version

Even though you have installed the latest version of Java on your system, Eclipse might be trying to use an older version that you forgot to uninstall, or intentionally left there. We go to **Preferences** → **Java** → **Installed JREs**:



Installed JREs

Add, remove or edit JRE definitions. By default, the checked JRE newly created Java projects.

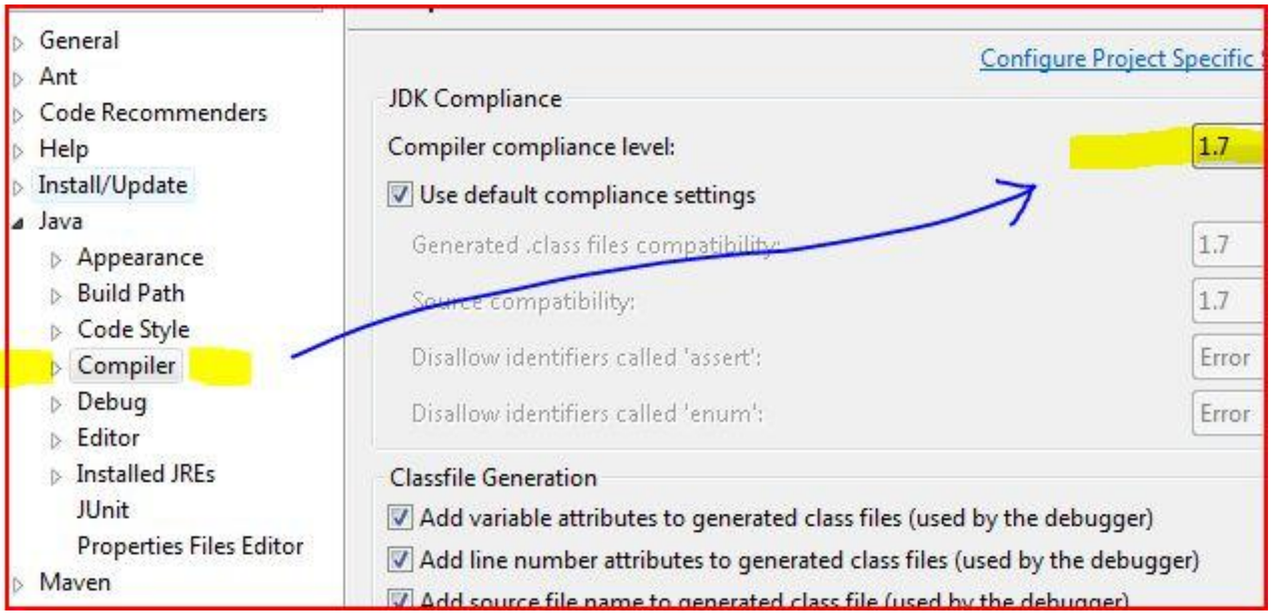
Installed JREs:

Name	Location	Type
✓ jre7	C:\Program Files\Java\jre7	Standard VM

You should see, at a minimum, your recent version of Java, and possibly some older versions, sitting on your system Check the box next to the latest version, or, if it already checked, just back out (**cancel**). *It may be correct, already.*

CMP.3.3 Tell Eclipse to Use the Latest Version

Even though you told Eclipse to use the latest JRE, you still have to tell it which version of Java you want to write for. This will determine what kind of code, errors and warnings Java is going to give you as you compile. You set this from**Preferences → Java → Compiler:** *Compiler Compliance Level*:



JDK Compliance

Compiler compliance level: 1.7

[✓] Use default compliance settings

Generated .class files compatibility: 1.7

Source compatibility: 1.7

Disallow identifiers called 'assert': Error

Disallow identifiers called 'enum': Error

Classfile Generation

[✓] Add variable attributes to generated class files (used by the debugger)

[✓] Add line number attributes to generated class files (used by the debugger)

[✓] Add source file name to generated class file (used by the debugger)

Set this to the same version that you saw in the last screen. E.g., if Java 6 (=1.6) was on your previous screen, above, set this window to 6 (=1.6), as well. On the other hand, if your installed JRE is 7, set the compliance level here to 7 (= 1.7). *It will probably be correct, without your having to change anything.*

Section 4 - Configuring Eclipse for Proper Style

This section helps you configure your newly installed Integrated Development Environment (IDE) so that it will automatically assist you with indentation and style.

CMP.4.1 Why I Take Off Points For Bad Style and Tabs

Programs are works-in-progress, not finished pieces of art, even if they seem to work perfectly. Therefore, someone will always have to go back in and fix or adjust something. That's why a working program has no value if another programmer cannot easily read, understand and modify it. You want to become the kind of programmer that colleagues and employers will value. If your code is clean and adheres to good style, others will enjoy reviewing it, and not worry that it might contain costly hidden errors.

Tabs

Tabs expand differently on different systems. You cannot control how your source will look when you send it to someone else with tabs in it. Therefore, I require that you remove all tabs in your programs and replace them with 3 or 4 spaces each. This module will explain how to set up Eclipse so that you always get spaces automatically inserted in place of tabs whenever you hit the tab key.

Style

There are three or four standard and accepted styles in programming. If you use one of them, you will not lose points. It is in your best interest, however, to use *my* style in this course. That way, you are sure that I will have an easy time reading through your code. This page will show you how to set up Eclipse so that it automatically generates code in my preferred style.

Do This Now

If you do this correctly the first time, you will never have to repeat it - all of your future projects will obey these choices. So, don't start programming until you have completed this section and made the changes to your compiler. Open up Eclipse and follow these step-by-step instructions.

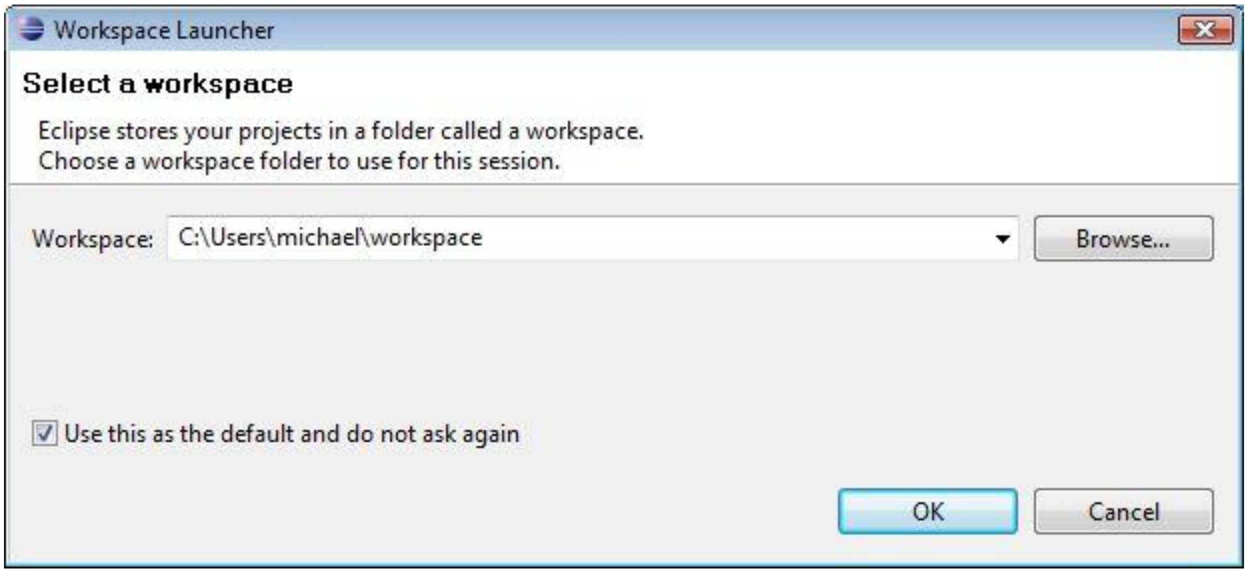
You Should Check, Manually, Anyway

Even after setting up Eclipse as described below, things can happen. You should review your code line-by-line to be sure it meets my style requirements each week before you submit. You can adjust things manually, if you see a problem. Also, there's a trick I will show you (a little later) to help fix bad indentation if it sneaks in, despite these measures.

CMP.4.2 Start Eclipse

You should not have tried to do anything in Eclipse yet, but now I authorize you to start it and click through the welcome screens to the ***Workbench***.

If this is your first time working with ***Eclipse***, accept the default workspace (and check the box that does this automatically in the future):


Workspace Launcher

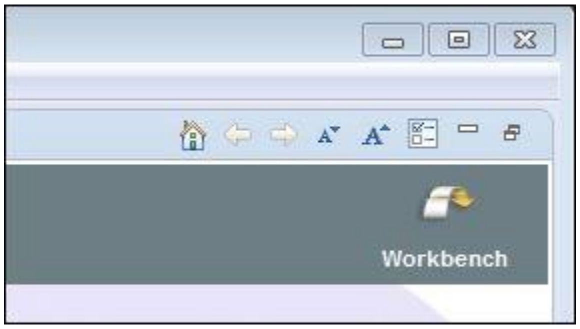
Select a workspace

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

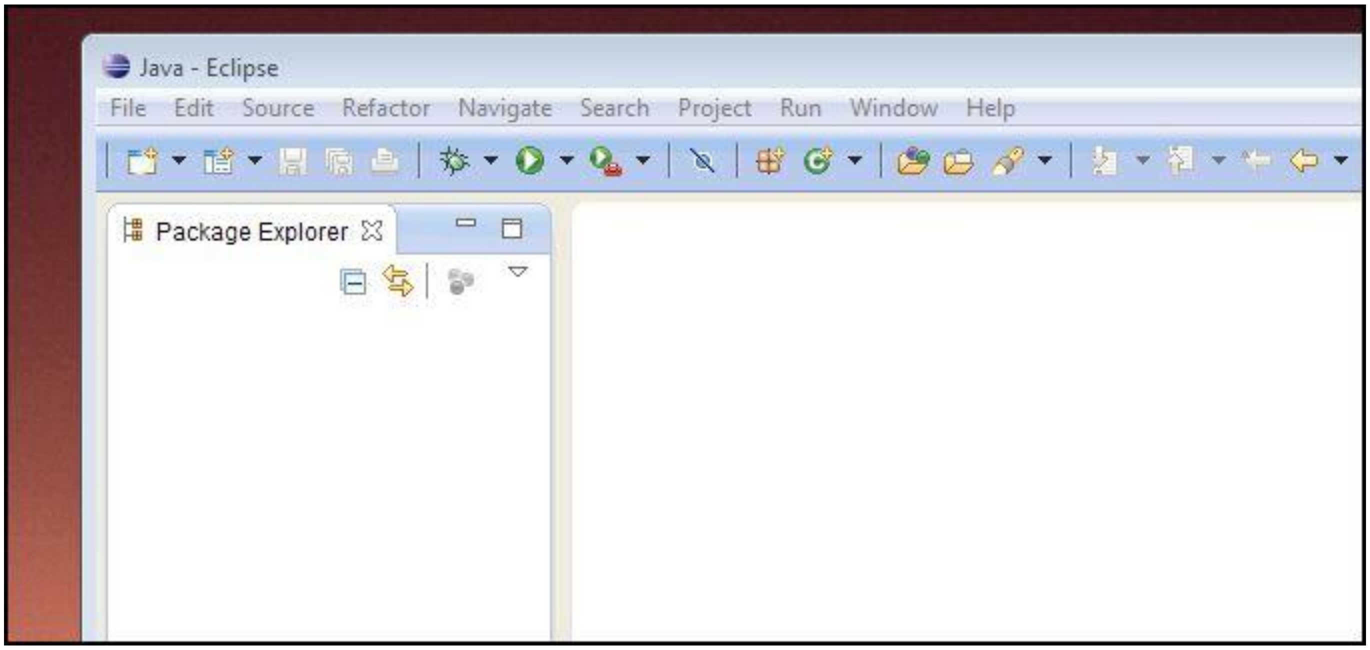
Workspace: C:\Users\michael\workspace

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

Also, the very first time you run Eclipse, you may have to manually ask to go to the ***Workbench***. (This step will not recur in future launches.) Click on the ***Workbench*** icon:



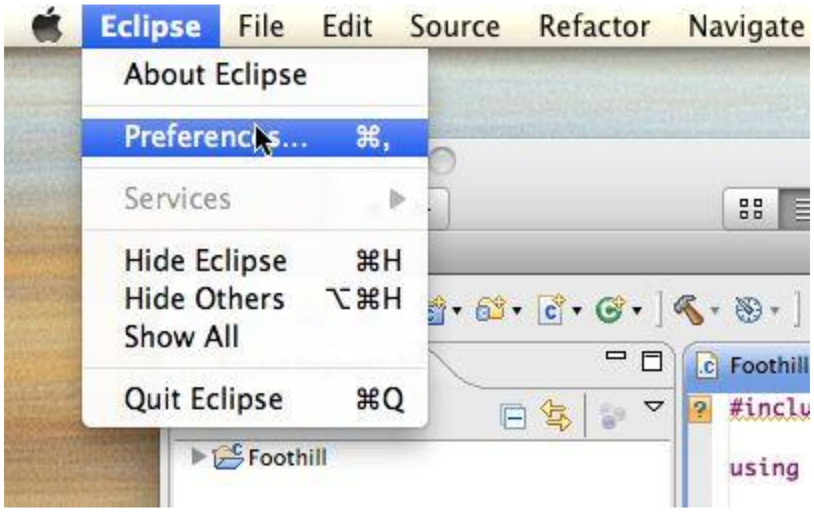
You will see a work screen that looks something like this:


Java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer

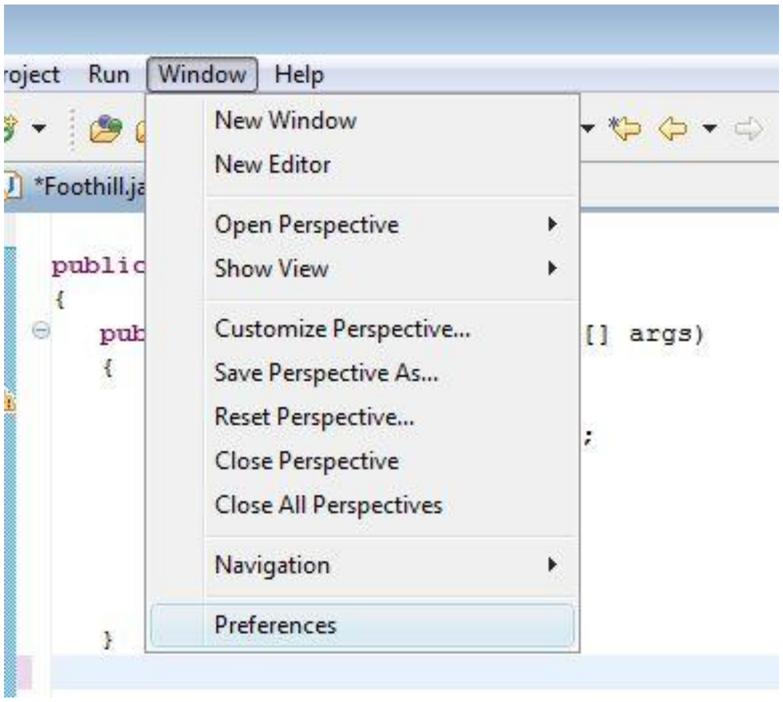
Next, we take three general actions to set-up Eclipse for use throughout the course. We do this once, and we'll never have to do it again.

CMP.4.3 Setting Tab Preferences for All Languages

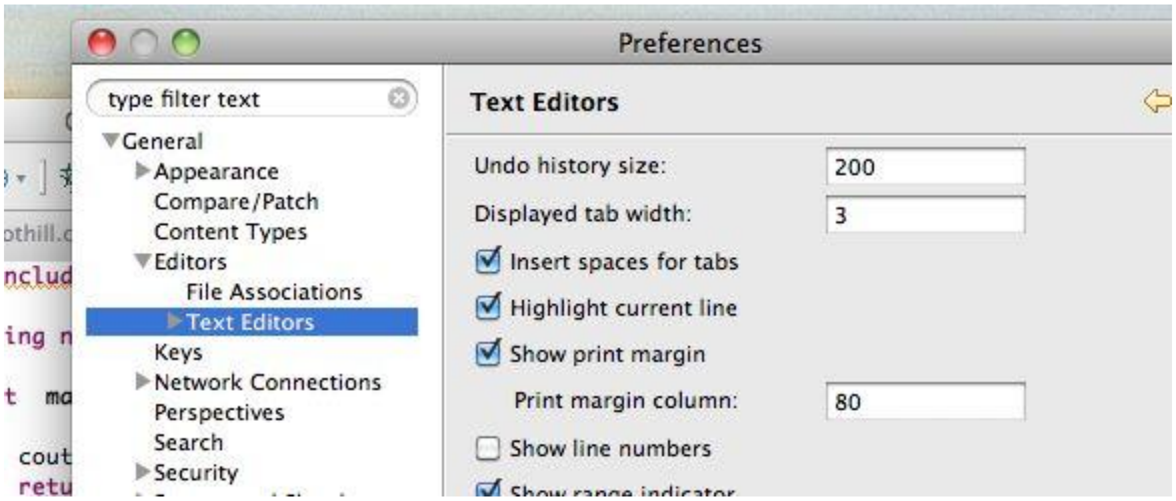
Select **Eclipse → Preferences (on a Mac)**



or **Window → Preferences (on Windows)**



Then, on the left explorer tree, expand **General → Editors** and select **Text Editors** (same for both **PC** and **Mac**; **Mac**shown):



Text Editors

Undo history size: 200
Displayed tab width: 3
☑ Insert spaces for tabs
☑ Highlight current line
☑ Show print margin
Print margin column: 80
☐ Show line numbers
☑ Show range indicator

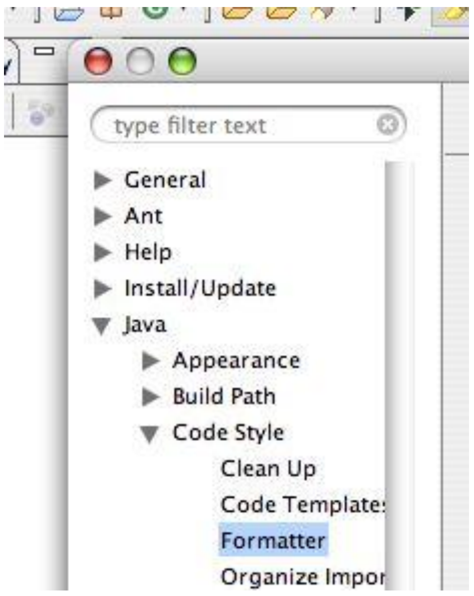
As shown above, check **Insert spaces for tabs** and put **3** in the **Displayed tab width** text box.

While you are in that window, help yourself to avoid losing points for long lines by checking **Show print margin** and leaving (or typing) **80** in the **Print margin column** text box. This will put a vertical line in column **80** of your source files, so you will know where your "too long line" maximum is. Click **OK** to save.

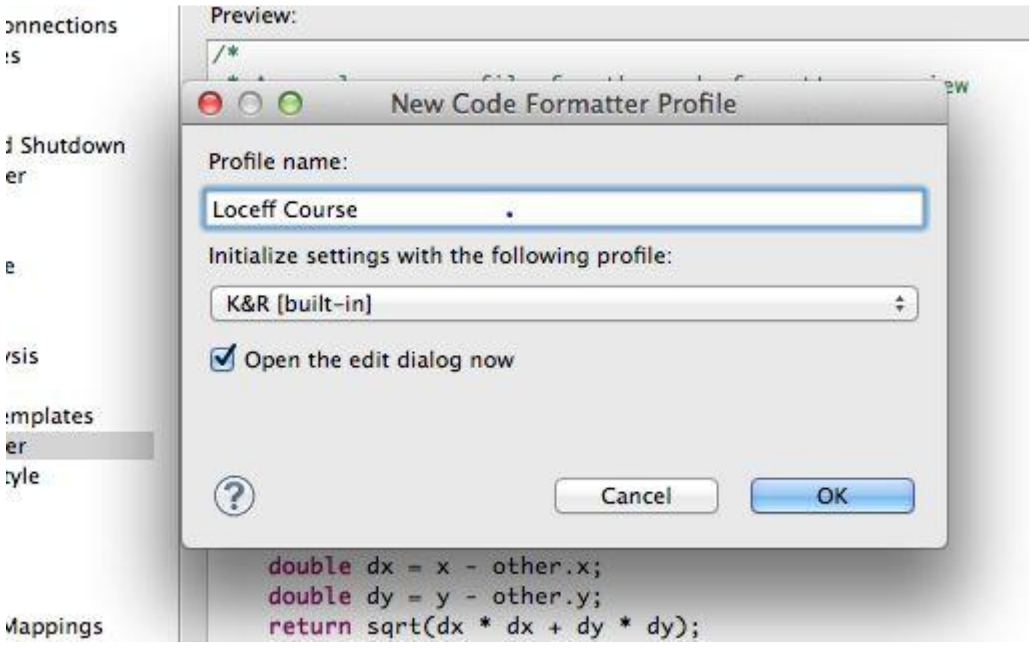
CMP.4.4 Setting Tab Preferences as Custom Code Style for Java (or C++)

There is a second measure that will add extra protection against unwanted tabs.

Start with the same **Preferences** menu, but instead of **General**, expand **Java (or C++)** → **Code Style** and select**Formatter**:



The rest is the same for both **Java** and **C++** (although the pics are from a **C++** operation). Toward the right side something like, **Eclipse [built-in]**, **K&R (built-in)** for **C++**, or **Java Conventions [built-in]** for **Java** will be selected. Under that, click **New...** and pick a name for your new formatting profile (I chose "**Loceff Course**"):


New Code Formatter Profile

Profile name:

Loceff Course

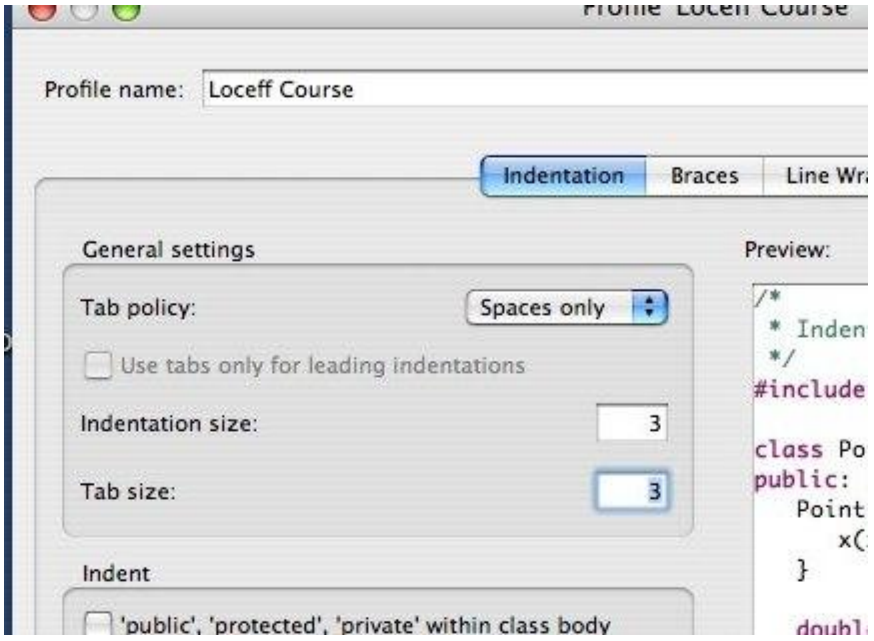
Initialize settings with the following profile:

K&R [built-in]

✓ Open the edit dialog now

```
    double dx = x - other.x;
    double dy = y - other.y;
    return sqrt(dx * dx + dy * dy);
```

Click **OK** and you will then be given a page with some choices along the top. First, select **Indentation** from the top choices and set the **Tab policy** to be **Spaces only**. Next, replace the **4** in the two text boxes with **3** as shown:


Profile name: Loceff Course

General settings

Tab policy: Spaces only

☐ Use tabs only for leading indentations

Indentation size: 3

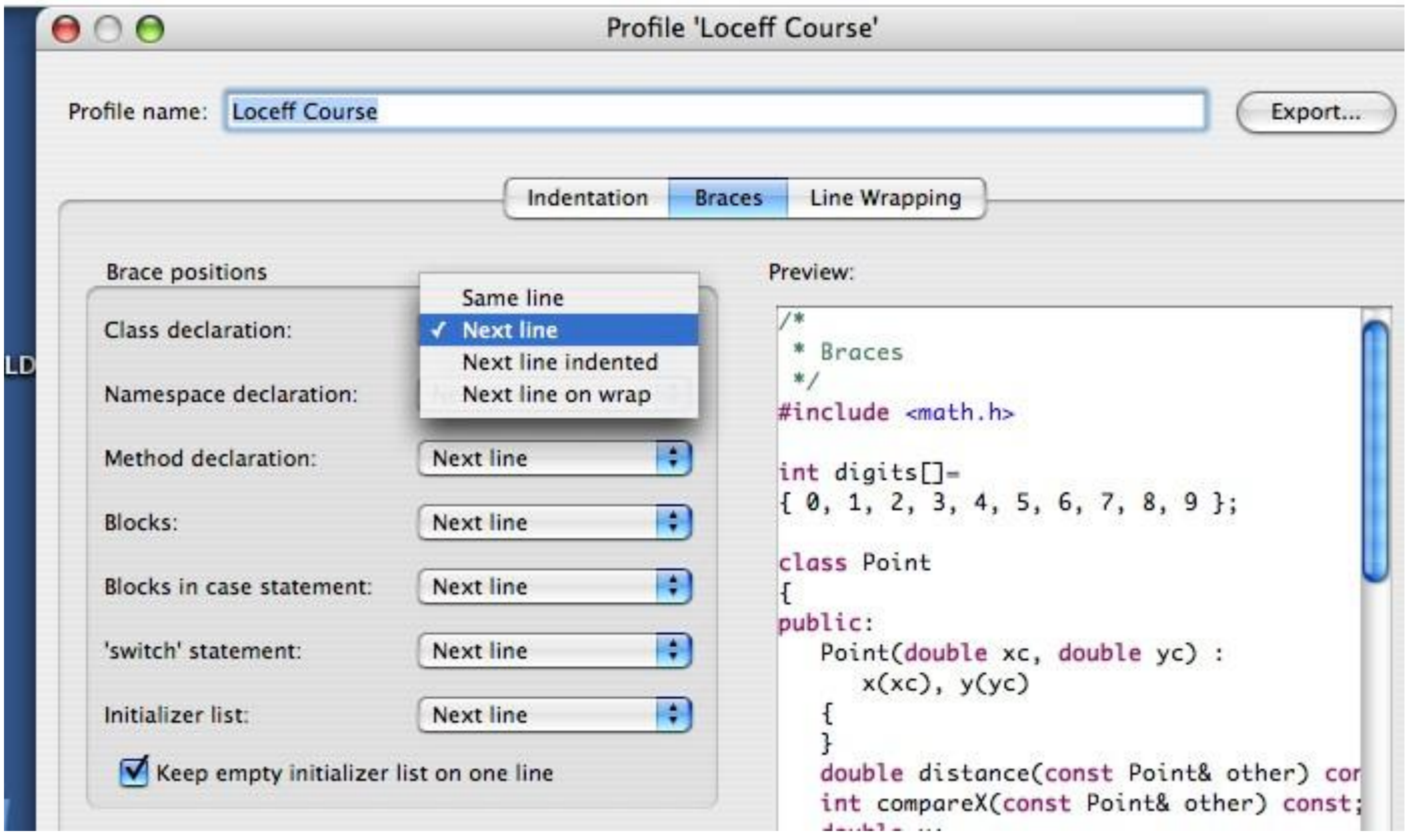
Tab size: 3

Indent

☐ 'public', 'protected', 'private' within class body

CMP.4.5 Setting Style Preferences in Custom Code Style

While you are in the code style menu, you should set up the brace-placement so it is identical to mine (unless you are already a programmer and have a coherent style that you prefer). To do this, turn your attention, again, to the top choices, and select the **Braces** tab. Once on that screen, set all of the pull-downs so that they are set to **Next line:**



Profile name: Loceff Course

Brace positions

Class declaration:	Next line
Namespace declaration:	
Method declaration:	Next line
Blocks:	Next line
Blocks in case statement:	Next line
'switch' statement:	Next line
Initializer list:	Next line

[✓] Keep empty initializer list on one line

```
/*
 * Braces
 */
#include <math.h>

int digits[]=
{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

class Point
{
public:
   Point(double xc, double yc) :
      x(xc), y(yc)
   {
   }
   double distance(const Point& other) co
   int compareX(const Point& other) const;
```

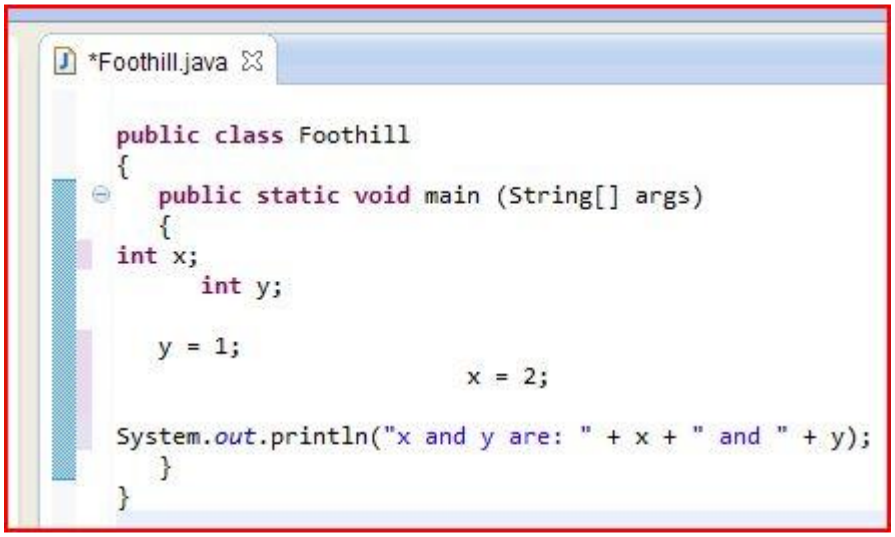
Select **OK** all the way back (there will be at least two **OKs**) to return to the main ***IDE*** window.

These settings help Eclipse help you, but they don't guarantee anything. That's why I remind you that you should do a manual check, each week, and fix anything that looks wrong.

If Bad Style Creeps In ...

Assuming you have made the above settings, it is easy to correct a bad-looking program if you managed to force some bad indentation on it (which is possible). You won't be able to test this now, because you don't have any code on which to tinker, but in a few days, you will. Still, watch what I do, and remember to come back here if it happens:

Here is an example of some code in which I forced bad style in several places (this is Java code, but it is very similar to C++ and the steps are the same):

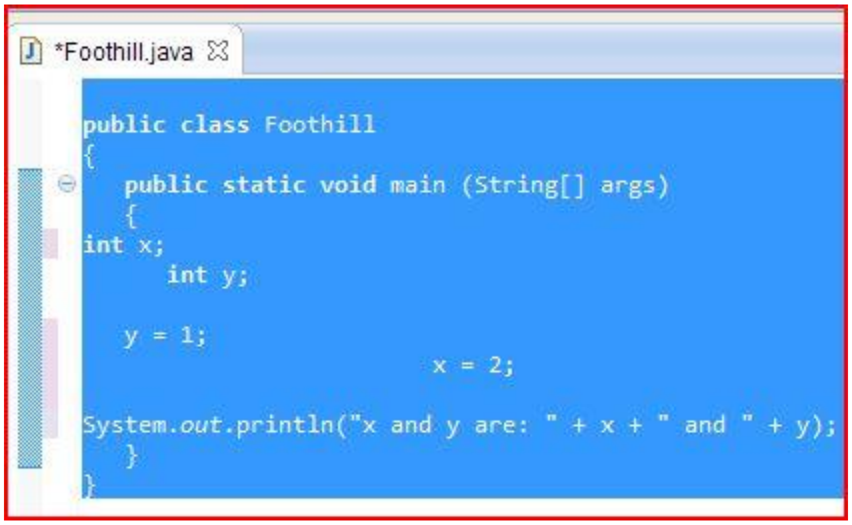


*Foothill.java

```
public class Foothill
{
   public static void main (String[] args)
   {
int x;
      int y;

   y = 1;
                      x = 2;

System.out.println("x and y are: " + x + " and " + y);
   }
}
```

To fix it, we do two things:

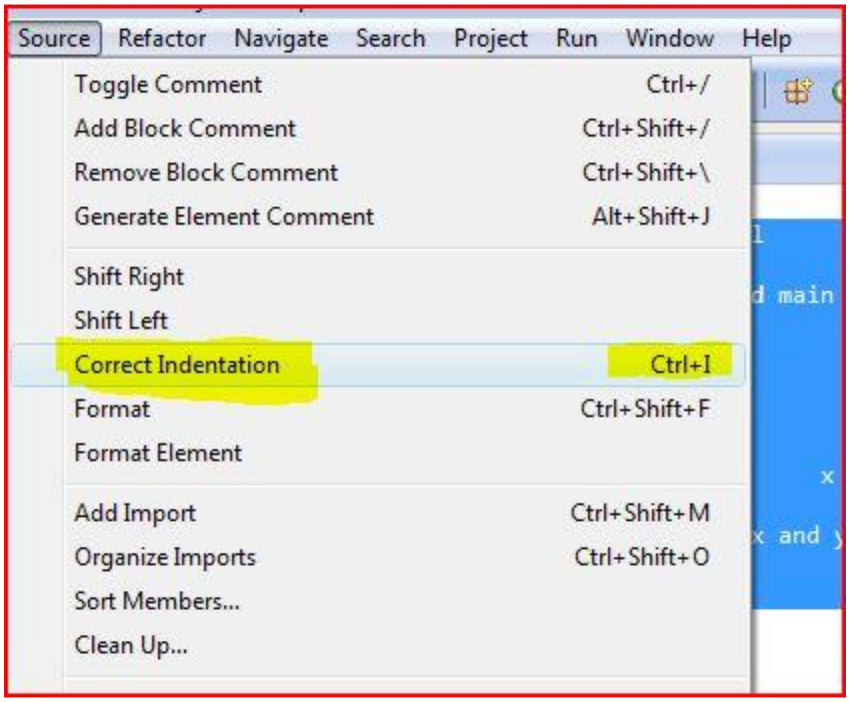
1. Select the entire program (or whatever part we want to correct).



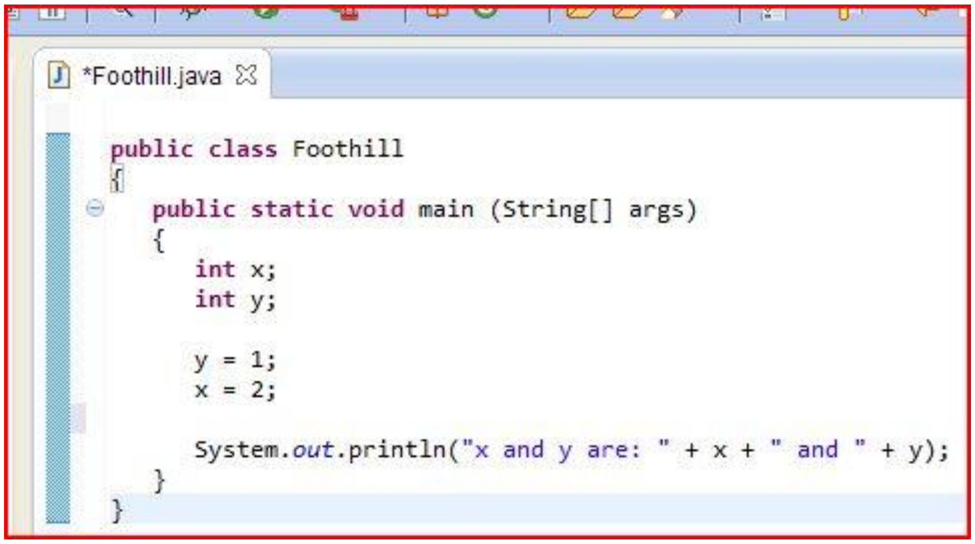
*Foothill.java

```
public class Foothill
{
   public static void main (String[] args)
   {
int x;
      int y;

   y = 1;
                      x = 2;

System.out.println("x and y are: " + x + " and " + y);
   }
}
```

2. Choose **Source → Correct Indentation**.



The indentation is now fixed:



```
public class Foothill
{
   public static void main (String[] args)
   {
      int x;
      int y;

      y = 1;
      x = 2;

      System.out.println("x and y are: " + x + " and " + y);
   }
}
```

Warning

There are other things besides indentation and tabs that need to be correct stylistically - you'll find out about them. So even this does not guarantee a penalty-free piece of code. Skipped lines in the wrong place, excessively long lines, and other things have to be fixed, and there is no auto-fix for those. So you still must scrutinize your program style and make manual corrections before submitting.

I'm on your side: I don't want you to lose points for bad style. That's why I wrote this module.