



# Java with Eclipse: Setup & Getting Started

Originals of Slides and Source Code for Examples:

<http://courses.coreservlets.com/Course-Materials/java.html>



**Customized Java EE Training:** <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, HTML5, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



For live Java-related training,  
see <http://courses.coreservlets.com/>  
or email [hall@coreservlets.com](mailto:hall@coreservlets.com).



**Taught by the author of *Core Servlets and JSP*, *More Servlets and JSP*, and this tutorial. Available at public venues, or customized versions can be held on-site at your organization.**

- Courses developed and taught by Marty Hall
  - JSF 2, PrimeFaces, servlets/JSP, Ajax, jQuery, Android development, Java 7 or 8 programming, custom mix of topics
  - Courses available in any state or country. Maryland/DC area companies can also choose afternoon/evening courses.
- Courses developed and taught by [coreservlets.com](http://coreservlets.com) experts (edited by Marty)
  - Spring, Hibernate/JPA, GWT, Hadoop, HTML5, RESTful Web Services

Contact [hall@coreservlets.com](mailto:hall@coreservlets.com) for details



# Topics in This Section

- Installing Java
- Installing and configuring Eclipse
- Importing sample projects
- Executing simple desktop programs
- Executing applets
- Eclipse shortcuts

© 2013 Marty Hall



## Installing Java



**Customized Java EE Training: <http://courses.coreservlets.com/>**

Java, JSF 2, PrimeFaces, HTML5, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

# Java Versions

- **Java SE (Standard Edition)**

- “Core” Java
  - Java version used in this class
- Latest is Java 7; Java 7 should be used for almost all new projects. **Java 8 coming in early 2014; see separate tutorial.**
- Naming conventions are confusing
  - Java 7 == Java 1.7 == JDK 1.7
  - Java 6 == Java 1.6 == JDK 1.6
  - Java 5 == Java 1.5 == JDK 1.5
  - Java 2, version 1.4 = Java 1.4 == JDK 1.4

- **Java EE (Enterprise Edition)**

- Same core language, but adds in many libraries for Web apps and other enterprise tasks
  - Many real-life deployments start with Java SE and then get server that is bundled with Java EE libraries

# Features of Recent Java Versions

- **Java 5**

- Major update. Generics, varargs, printf, @Override, new “for” loop.

- **Java 6**

- Minor update. Updates to collections, Swing, etc.

- **Java 7**

- Minor update. Diamond operator, Strings in switch statements, try-with-resources, updates to Swing (especially new look and feel).

- **Java 8**

- Major update. Lambdas for functional programming. Streams for bulk operations. Final version early 2014.
  - See tutorial at <http://www.coreservlets.com/java-8-tutorial/>

# Which Java SE Version Should You Use?

- **Server-Side Applications**

- Use the latest Java version that your app server supports
  - JDK 1.4 – 1.7, depending how old your server is
- If you can choose, use JDK 1.7 (but 1.6 still used)

- **Desktop Applications**

- For best power and speed, use Java 7 (aka 1.7)
  - If you install, use JDK 1.7 or 1.8
- Use older only if project is old (but consider upgrading)

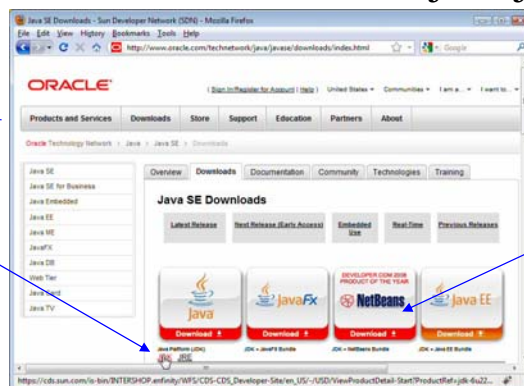
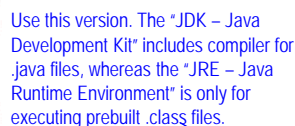
- **Applets**

- Separate Java Plug-In is required in all recent browsers
  - For intranet apps, use Java 7
  - On internet, most Java applet sites (e.g., *Yahoo! Games*) use JDK 1.5 or 1.6, but *all* require end-user installation

# Installing Java SE (Standard Edition)

- **Install Java 7**

<http://www.oracle.com/technetwork/java/javase/downloads/>



This tutorial uses Eclipse, but if you prefer the NetBeans environment, it is very easy to adapt the instructions to that development environment. So, if you prefer NetBeans or your organization has standardized on it, use this download instead of (not in addition to) the one on the left.

- **Bookmark the Java API (“JavaDocs”)**

- <http://docs.oracle.com/javase/7/docs/api/>
  - This is the most important Java reference for developers. Eclipse integrates this API, but a separate link is still good





# Installing Eclipse



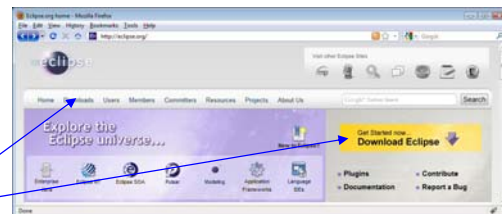
**Customized Java EE Training: <http://courses.coreservlets.com/>**

Java, JSF 2, PrimeFaces, HTML5, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

## Installing Eclipse

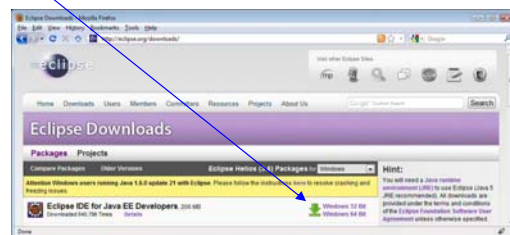
### • Overview

- Eclipse is a free open source IDE. Support for Java, Android, HTML, CSS, JavaScript, C++, PHP, JSF, servlets, and more.
  - <http://eclipse.org/downloads/>
  - Choose “Eclipse IDE for Java EE Developers”
    - I use latest (Kepler).



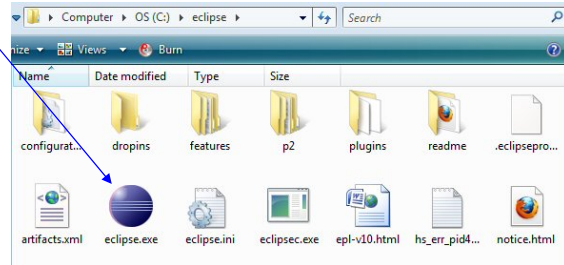
### • Features

- Checks your syntax as you type
- Automatically compiles every time you save file
- Many tools: refactoring, debugging, server integration, templates for common tasks, etc.
  - Low learning curve: beginners can use Eclipse without knowing these tools



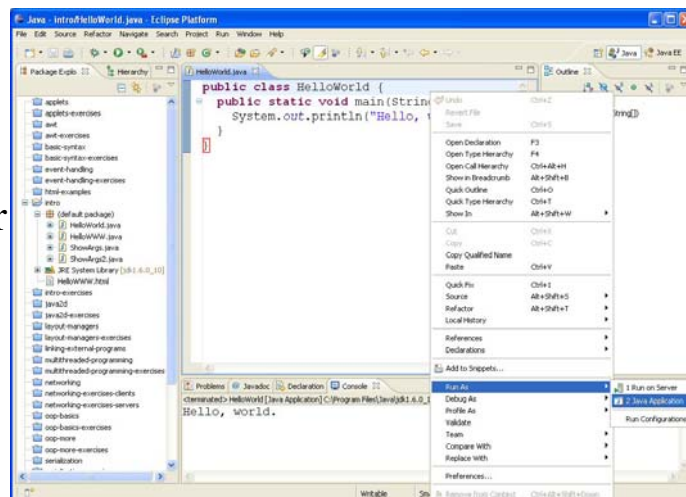
# Running Eclipse

- **Unzip the downloaded file (no installer!)**
  - Call the folder you unzip into “installDir”
- **Double click eclipse.exe**
  - From *installDir/bin*
- **Click on “Workbench” icon**
  - Next time you bring up Eclipse, it will come up in workbench automatically
- **Shortcut**
  - Many developers put Eclipse link on their desktop
    - R-click eclipse.exe, Copy, then go to desktop, R-click, and Paste Shortcut (not just Paste!)



# Eclipse: Running Programs

- **Executing program from existing project**
  - Open existing project
  - Double click Java file to bring it up in editor
  - R-click anywhere in code
  - Select Run As → Java Application
  - Output goes in Console at bottom
    - Note: Class must have a “main” method



# Eclipse: Making Projects

- **Creating new project**

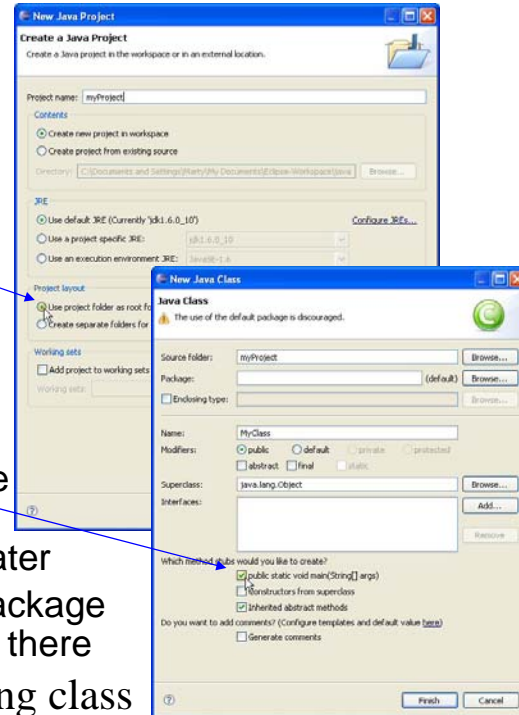
- File → New → Project → Java → Java Project

- Pick any name
- To simplify applets later, choose Sources/Classes in same folder

- **Creating new class**

- R-click → New → Class

- You can have Eclipse make “main” when class created, or use shortcut to insert it later
- Eventually you will make package (subdir) first, then put class there
- Can also copy/rename existing class



© 2013 Marty Hall



## Building Java Manually



Customized Java EE Training: <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, HTML5, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

# Creating and Running Program

- **Create the File**

- Write and save a file (say **Test.java**) that defines `public class Test`
  - Other than “real” Java IDEs (e.g., Eclipse, NetBeans, IntelliJ IDEA), there are a number of text editors (e.g., TextPad, UltraEdit, Brief) with good Java syntax support.
- File and class names are case sensitive

- **Compile the program**

- Compile Test.java
  - > **javac Test.java**
    - This step creates a file called Test.class

- **Run the program**

- > **java Test**
  - This step assumes your class has “main” method

© 2013 Marty Hall



## Simple Examples



**Customized Java EE Training:** <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, HTML5, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



# Installing Sample Projects

- **Code from all tutorials is available online**
  - <http://courses.coreservlets.com/>
    - Click on Java tutorial on top left of page
- **Import project into Eclipse**
  - Click on appropriate tutorial section
  - Download ZIP file
    - The one for this section is called “intro”
  - Start Eclipse and go to Workbench
  - File → Import → General → Existing Projects into Workspace → Select archive file (not “Select root directory”).
    - Then browse to ZIP file you downloaded, OK, Finish

# Basic Hello World Application

- **File HelloWorld.java:**

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, world.");  
    }  
}
```
- **Notes**
  - “Application” is lingo for a stand-alone Java program
  - An application is a Java class that contains “main”
    - Most Java classes do not contain “main”, but only those that do can be *directly* executed



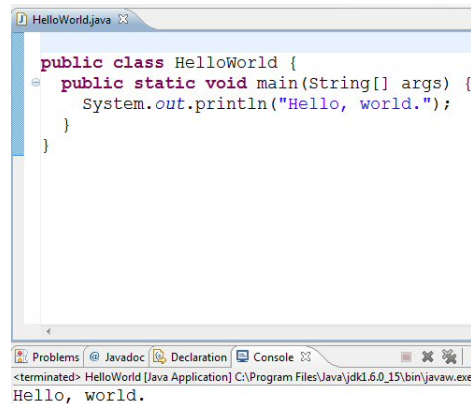
# Executing HelloWorld

- **In Eclipse (recommended)**

- Compiling
  - Done automatically whenever you save a file
- Executing
  - R-click inside window, then Run As → Java Application
- Output (see “Console” tab at bottom)
  - Hello, World

- **Manually**

- Compiling
  - `javac HelloWorld.java`
- Executing
  - `java HelloWorld`
- Output
  - Hello, World



# Basic Hello WWW Applet

- **File HelloWorld.java:**

```
import java.applet.Applet;  
import java.awt.*;  
  
public class HelloWorld extends Applet {  
    public void init() {  
        setBackground(Color.BLACK);  
        setForeground(Color.WHITE);  
        setFont(new Font("SansSerif", Font.BOLD, 30));  
    }  
  
    public void paint(Graphics g) {  
        g.drawString("Hello, World Wide Web.", 5, 35);  
    }  
}
```

# Basic Hello WWW Applet (Continued)

- **File HelloWorld.html:**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
  <title>HelloWWW: Simple Applet Test.</title>
</head>
<body>
<h1>HelloWWW: Simple Applet Test.</h1>
<p>
  <applet code="HelloWWW.class" width="460" height="50">
    <b>Error! Must have Java enabled in your browser.</b>
  </applet>
</p>
</body></html>
```

# Basic Hello WWW Applet (Continued)

- **Compiling**

- Eclipse: save the file
- Manual: `javac HelloWorld.java`

- **Running:**

Load **HelloWWW.html** in a Java-enabled browser  
With Eclipse, just drag the file onto browser





# Some Eclipse Shortcuts



Customized Java EE Training: <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, HTML5, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

## A Few Eclipse Tricks

- **Making a new project**
  - File → New → Project → Java → Java Project
    - Choose Source/Classes in Same Folder
- **Making a new class**
  - R-click project, New → Class
- **Autocompletion**
  - Type part of a class or method name, Control-Space
- **Inserting main method**
  - Type the word “main”, then Control-Space
- **Inserting System.out.println**
  - Type the word “sysout”, then Control-Space
- **Renaming a variable or method**
  - Select method or variable, R-click, Refactor → Rename
    - Will also change all places that refer to it





# Wrap-Up



**Customized Java EE Training:** <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, HTML5, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

## Summary

- **Downloading Java**
  - <http://www.oracle.com/technetwork/java/javase/downloads/>
- **Bookmarking the Java API**
  - <http://docs.oracle.com/javase/7/docs/api/>
- **Downloading Eclipse**
  - <http://eclipse.org/downloads/>
- **Downloading sample projects**
  - <http://courses.coreservlets.com/>
    - Click on Java Programming tutorial on top left
  - Import with File → Import → Existing Projects ...
- **Executing a class that has “main”**
  - R-click in code, Run As → Java Application



# Questions?

JSF 2, PrimeFaces, Java 7 or 8, HTML5, Ajax, jQuery, Hadoop, RESTful Web Services, Android, Spring, Hibernate, Servlets, JSP, GWT, and other Java EE training.



**Customized Java EE Training: <http://courses.coreservlets.com/>**

Java, JSF 2, PrimeFaces, HTML5, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.