



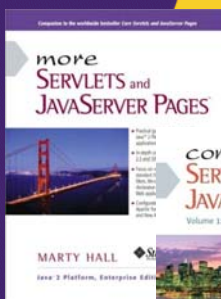
# Java with Eclipse: Setup & Getting Started

Originals of slides and source code for examples: <http://courses.coreservlets.com/Course-Materials/java.html>  
Also see Java 8 tutorial: <http://www.coreservlets.com/java-8-tutorial/> and many other Java EE tutorials: <http://www.coreservlets.com/>  
Customized Java training courses (onsite or at public venues): <http://courses.coreservlets.com/java-training.html>



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

Java 7, Java 8, JSF 2, PrimeFaces, Android, JSP, Ajax, jQuery, Spring MVC, RESTful Web Services, GWT, Hadoop  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



**For customized Java-related training at  
your organization, email [hall@coreservlets.com](mailto:hall@coreservlets.com)**  
**Marty is also available for consulting and development support**



**Taught by lead author of *Core Servlets & JSP*, co-author of *Core JSF* (4<sup>th</sup> Ed), & 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 2.2, PrimeFaces, servlets/JSP, Ajax, JavaScript, jQuery, Android, Java 7 or 8 programming, GWT, 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 experts (edited by Marty)
  - Spring MVC, Core Spring, Hibernate/JPA, 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 from Eclipse
- Executing applets from Eclipse
- Executing Java programs manually
- Using Beanshell
- Eclipse shortcuts

© 2015 Marty Hall



## Installing Java



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

Java 7, Java 8, JSF 2, PrimeFaces, Android, JSP, Ajax, jQuery, Spring MVC, RESTful Web Services, GWT, Hadoop  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

# Java SE vs. Java EE

- **Java SE (Standard Edition)**
  - “Core” Java
  - Java version used in this video course
- **Java EE (Enterprise Edition)**
  - Same core language, but adds in many libraries for Web apps and other enterprise tasks
    - For tutorials on building Web apps in Java, please see JSF and PrimeFaces tutorials at [coreservlets.com](http://coreservlets.com)
  - Many or most real-life deployments start with Java SE and then get a server that is bundled with the needed Java EE libraries
    - So, most developers download Java SE, not Java EE

# Java SE Versions

- **Latest Java SE version**
  - Latest is Java 8; Java 8 should be used for almost all new projects.
    - Java 8 was final in March 2014. This tutorial covers general Java programming integrated with Java 8, but for just the Java-8-specific topics, see <http://www.coreservlets.com/java-8-tutorial/>.
- **Java SE naming conventions**
  - Naming conventions are confusing
    - Java 8 == JDK 1.8
    - Java 7 == JDK 1.7
    - Java 6 == JDK 1.6
    - Java 5 == JDK 1.5
    - Java 2, version 1.4 == JDK 1.4

# 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 March 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 1.5 – 1.8, depending on how old your server is
  - If you can choose, use JDK 1.8 (but 1.7 still common)
- **Desktop apps**
  - For best power and speed, use Java 8 (aka JDK 1.8)
  - Use older only if project is old (but consider upgrading)
- **Android phone apps**
  - As of 2015, Google supports only Java 6 (aka JDK 1.6)
- **Browser apps (Applets or Java WebStart)**
  - Separate Java Plug-In is required in all recent browsers
    - For intranet apps, use Java 8. Rarely used for internet apps.

# Installing Java SE (Standard Edition)

- **Install Java**

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

Use this version. The "JDK – Java Development Kit" includes compiler for .java files, whereas the "JRE – Java Runtime Environment" is only for executing prebuilt .class files.



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 below.

- **Bookmark the Java API ("JavaDocs")**

- <http://docs.oracle.com/javase/8/docs/api/>
- <http://docs.oracle.com/javase/7/docs/api/> (if you need old version)
  - This is the most important Java reference for developers
  - Eclipse integrates this API, but a separate link is still good

© 2015 [Marty Hall](#)



## Installing Eclipse



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

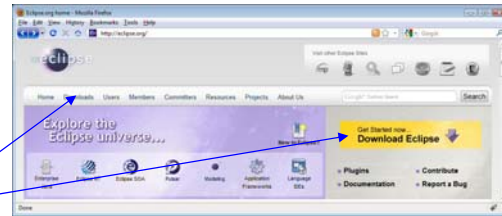
Java 7, Java 8, JSF 2, PrimeFaces, Android, JSP, Ajax, jQuery, Spring MVC, RESTful Web Services, GWT, Hadoop  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



# Installing Eclipse

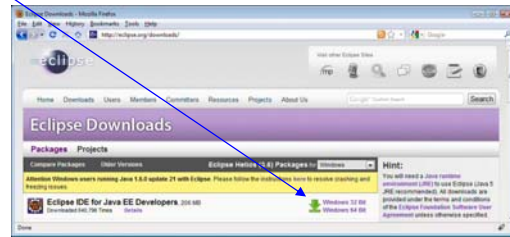
- **Overview**

- 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”



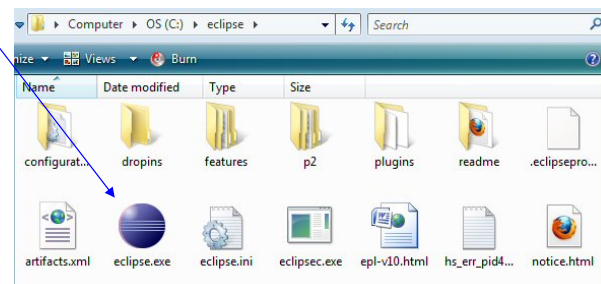
- **Features**

- 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

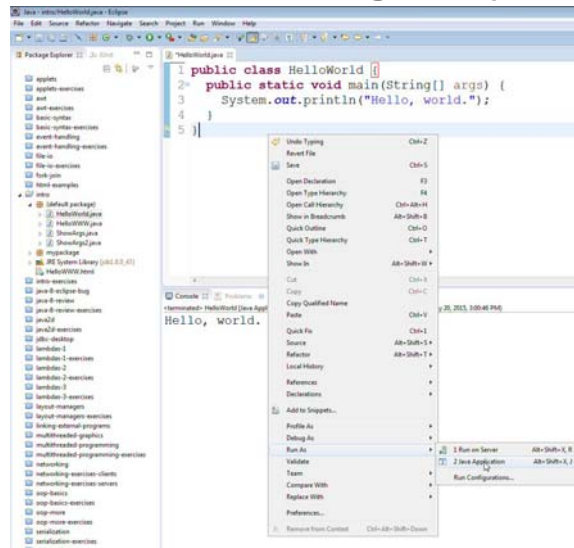
- **Unzip the downloaded file (no installer!)**
  - Call the folder you unzip into “installDir”
- **Double click eclipse.exe (Mac/Linux similar)**
  - From *installDir/bin*
    - Pic is for Windows, but Mac and Linux is similar
- **Click on “Workbench” icon**
  - Next time you bring up Eclipse, it will 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

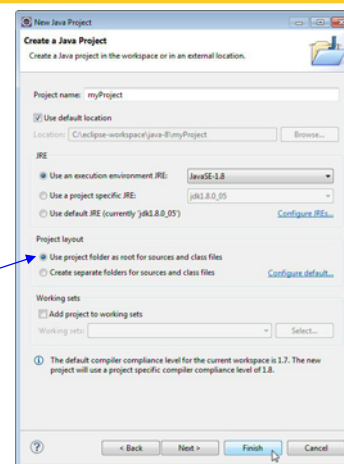


- Note: Class must have a “main” method – this is explained in the upcoming basic syntax section

# Eclipse: Making Projects

- **Main steps**

- File → New → Project → Java → Java Project
  - Pick any name
- If you will ever use applets
  - Choose sources/classes in project folder



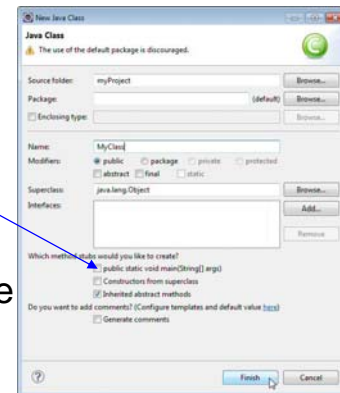
# Eclipse: Creating Classes

- **Main steps**

- R-click on project → New → Class
  - You can have Eclipse make “main” when class created, but easier to use shortcut to insert it later
  - Eventually you will make package (subdirectory) first, then put class there

- **Alternative**

- Can also copy/rename existing class



© 2015 [Marty Hall](#)



## Building Java Manually (Rare)



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

Java 7, Java 8, JSF 2, PrimeFaces, Android, JSP, Ajax, jQuery, Spring MVC, RESTful Web Services, GWT, Hadoop  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



# Creating and Running Program

- **Create the .java 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, vi, emacs) with good Java support.
  - File and class names are case sensitive
- **Compile the the .java file**
  - Compile Test.java
    - > **javac Test.java**
      - This step creates a file called Test.class
- **Run the .class file**
  - > **java Test**
    - This step assumes your class has “main” method

© 2015 Marty Hall



## Simple Examples



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

Java 7, Java 8, JSF 2, PrimeFaces, Android, JSP, Ajax, jQuery, Spring MVC, RESTful Web Services, GWT, Hadoop  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

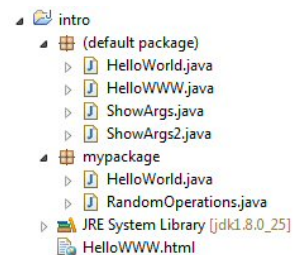
# Installing Sample Projects

- **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 contain “main” can be *directly* executed

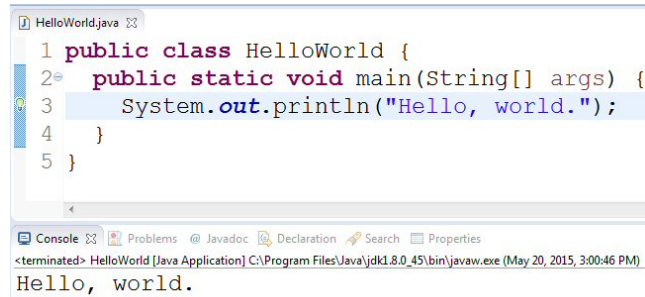
# Executing HelloWorld

- **In Eclipse (recommended)**

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

- **Manually (rare)**

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



The screenshot shows the Eclipse IDE with the HelloWorld.java file open. The code is as follows:

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hello, world.");  
4     }  
5 }
```

The console output at the bottom shows:

```
<terminated> HelloWorld [Java Application] C:\Program Files\Java\jdk1.8.0_45\bin\javaw.exe (May 20, 2015, 3:00:46 PM)  
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, drag HTML file onto browser, or (no browser setup needed), R-click in Java, Run As → Java Applet







# Using Beanshell



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

Java 7, Java 8, JSF 2, PrimeFaces, Android, JSP, Ajax, jQuery, Spring MVC, RESTful Web Services, GWT, Hadoop  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

## Overview

- **Way to run Java commands interactively**
  - Like the command prompt in Python, Basic, Lisp, etc.
  - Good for beginner practice

- **Examples**

```
bsh % System.out.println("hi");  
hi  
bsh % String message = "Hello";  
bsh % System.out.println(message);  
Hello  
bsh % public int getFavoriteNum() { return(7); }  
bsh % System.out.println("Favorite num is " + getFavoriteNum());  
Favorite num is 7
```

# Steps

- **Download beanshell**

- <http://www.beanshell.org/download.html>
- Beanshell is included in “intro” project of this tutorial

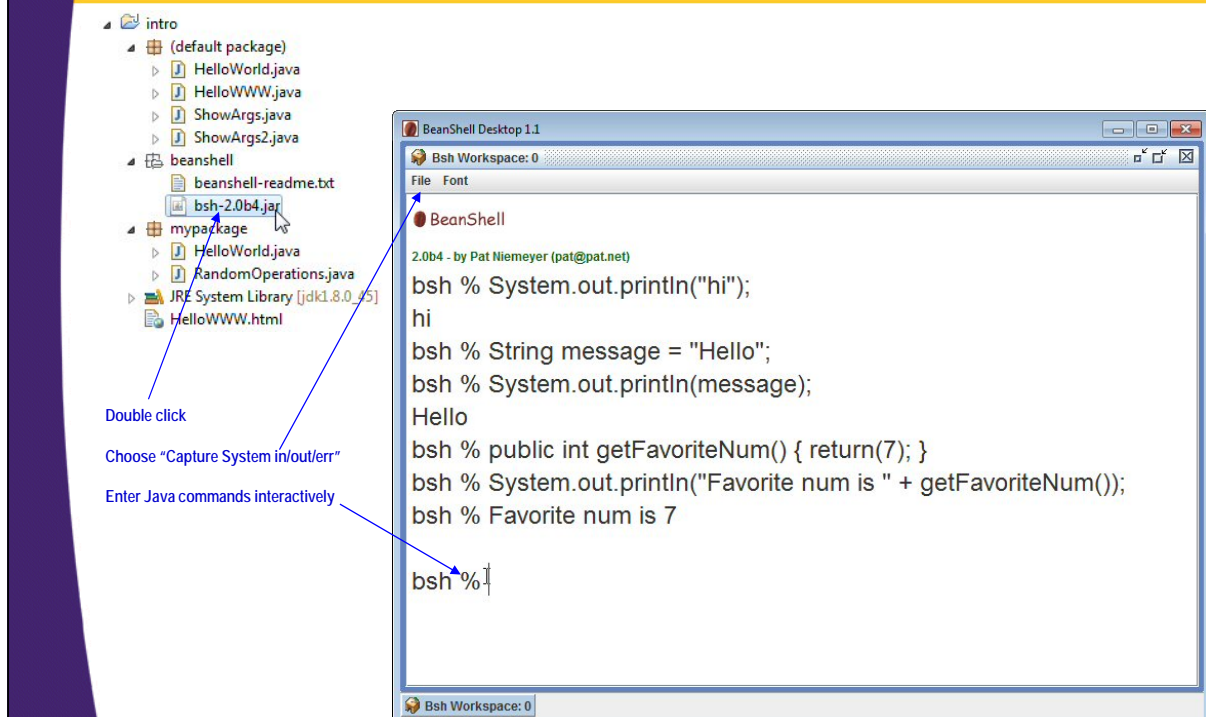
- **Run beanshell**

- Double-click the .jar file
  - If it won't launch on double click, see <http://stackoverflow.com/questions/8511063/how-to-run-jar-file-by-double-click-on-windows-7-64>
- Go to File menu and select “Capture System in/out/err”
- Enter Java commands and see the results
  - Define variables, define methods, call methods, etc.

- **More info**

- [http://www.beanshell.org/manual/bshmanual.html#Quick\\_Start](http://www.beanshell.org/manual/bshmanual.html#Quick_Start)

# Example





# Some Eclipse Shortcuts



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

Java 7, Java 8, JSF 2, PrimeFaces, Android, JSP, Ajax, jQuery, Spring MVC, RESTful Web Services, GWT, Hadoop  
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
- **Making new package**
  - R-click project, New → Package
- **Making a new class**
  - R-click package, 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 class, variable, or method**
  - Select class, variable, or method, R-click, Refactor → Rename
    - Will also change all places that refer to it



# Wrap-Up



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

Java 7, Java 8, JSF 2, PrimeFaces, Android, JSP, Ajax, jQuery, Spring MVC, RESTful Web Services, GWT, Hadoop  
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/8/docs/api/> (or .../7/...)
- **Downloading Eclipse**
  - <http://eclipse.org/downloads/>
- **Downloading sample projects**
  - <http://www.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?

More info:

<http://courses.coreservlets.com/Course-Materials/java.html> – General Java programming tutorial

<http://www.coreservlets.com/java-8-tutorial/> – Java 8 tutorial

<http://courses.coreservlets.com/java-training.html> – Customized Java training courses, at public venues or onsite at *your* organization

<http://coreservlets.com/> – JSF 2, PrimeFaces, Java 7 or 8, Ajax, jQuery, Hadoop, RESTful Web Services, Android, HTML5, Spring, Hibernate, Servlets, JSP, GWT, and other Java EE training

Many additional free tutorials at [coreservlets.com](http://coreservlets.com) (JSF, Android, Ajax, Hadoop, and lots more)



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

Java 7, Java 8, JSF 2, PrimeFaces, Android, JSP, Ajax, jQuery, Spring MVC, RESTful Web Services, GWT, Hadoop

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