

Eclipse Basics

- The JDK and JRE
- Eclipse and IDEs
- Perspectives and Views
- Projects and Workspaces
- Your First Project

Introduction to Java



See Also

Install JDK:

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

Install Eclipse:

<https://eclipse.org/downloads/>

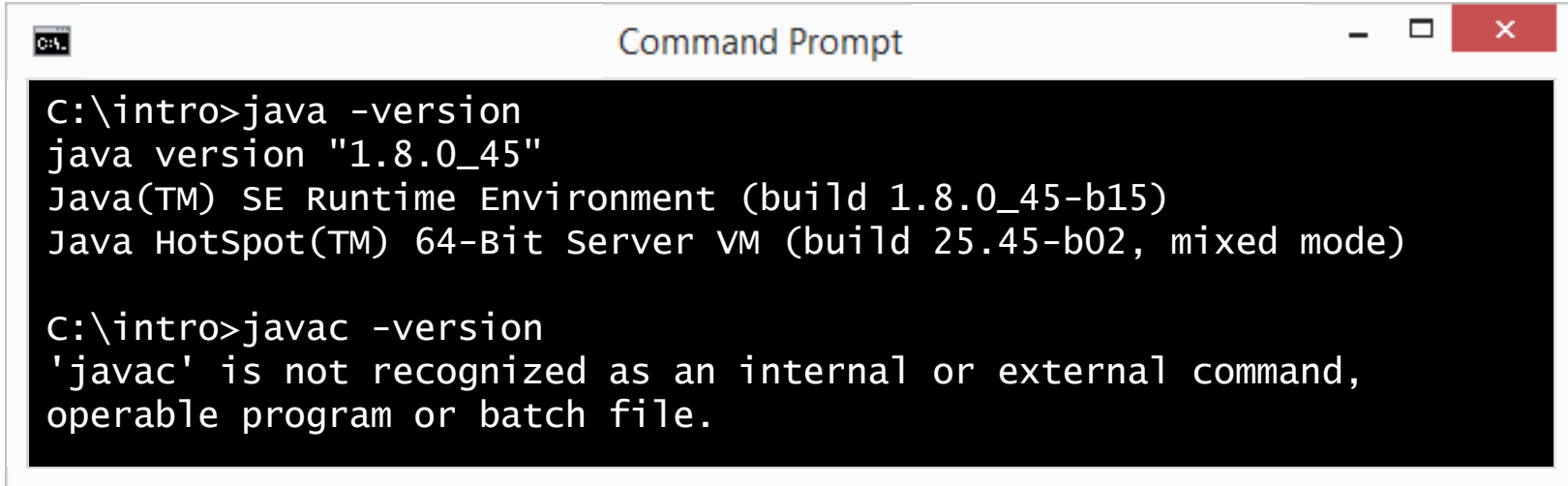
Online Eclipse documentation:

<http://help.eclipse.org/mars/index.jsp>



Runtime and Development

- JRE Java Runtime Environment
 - Runs a compiled program (.class or .jar)
- JDK Java Development Kit
 - Developer tools (like the compiler)

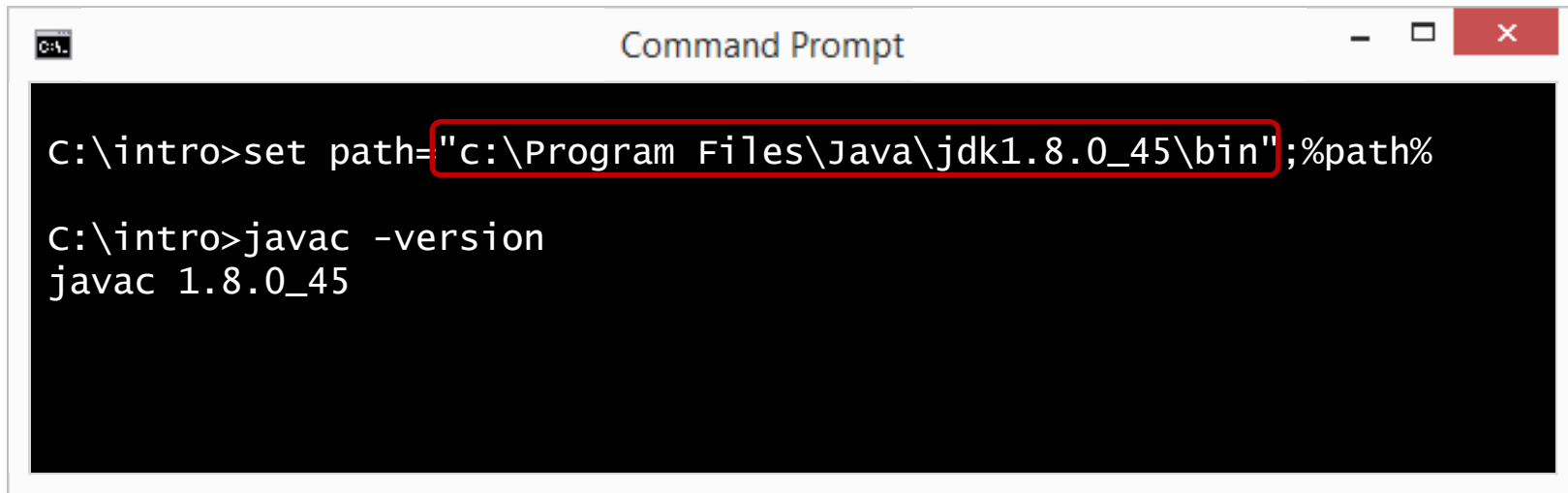


```
C:\intro>java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b15)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)

C:\intro>javac -version
'javac' is not recognized as an internal or external command,
operable program or batch file.
```

The Windows Path

- Set it temporarily at the command prompt
 - Only for this window
 - Create a batch file



```
C:\intro>set path="c:\Program Files\Java\jdk1.8.0_45\bin";%path%

C:\intro>javac -version
javac 1.8.0_45
```

The Windows Path

- Set it permanently through windows
 - Remains in effect for all programs

Control Panel > System and Security > System

Environment Variables

User variables for Topher

Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

New... Edit... Delete

System variables

Variable	Value
NUMBER_OF_P...	8
OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;...

New... Edit... Delete

OK Cancel

Edit System Variable

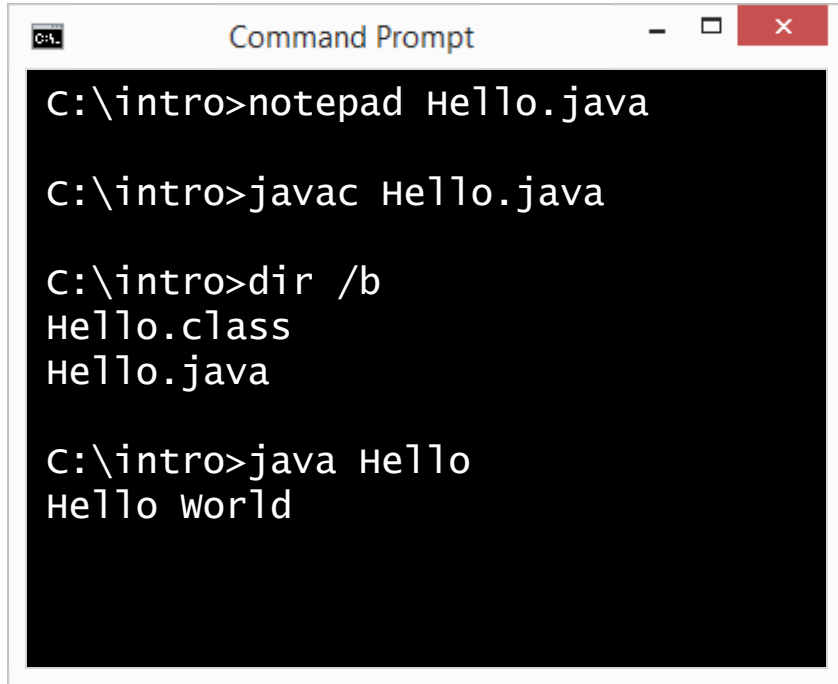
Variable name: Path

Variable value: c:\program files\java\jdk1.8.0_45\bin; Core-

OK Cancel



Command Line Only

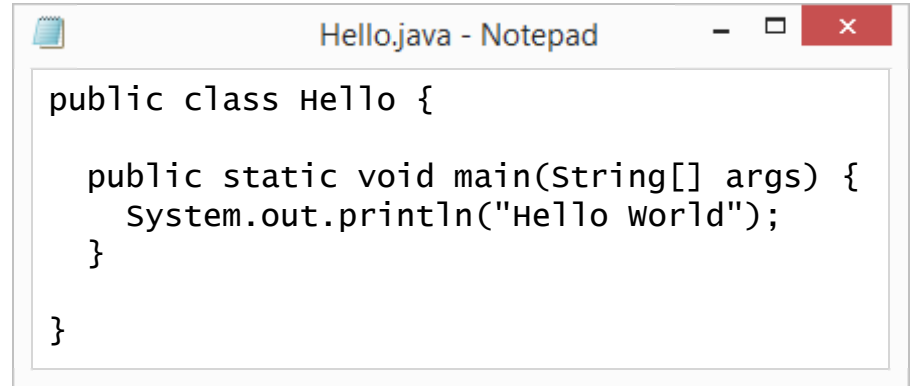


```
C:\intro>notepad Hello.java

C:\intro>javac Hello.java

C:\intro>dir /b
Hello.class
Hello.java

C:\intro>java Hello
Hello world
```



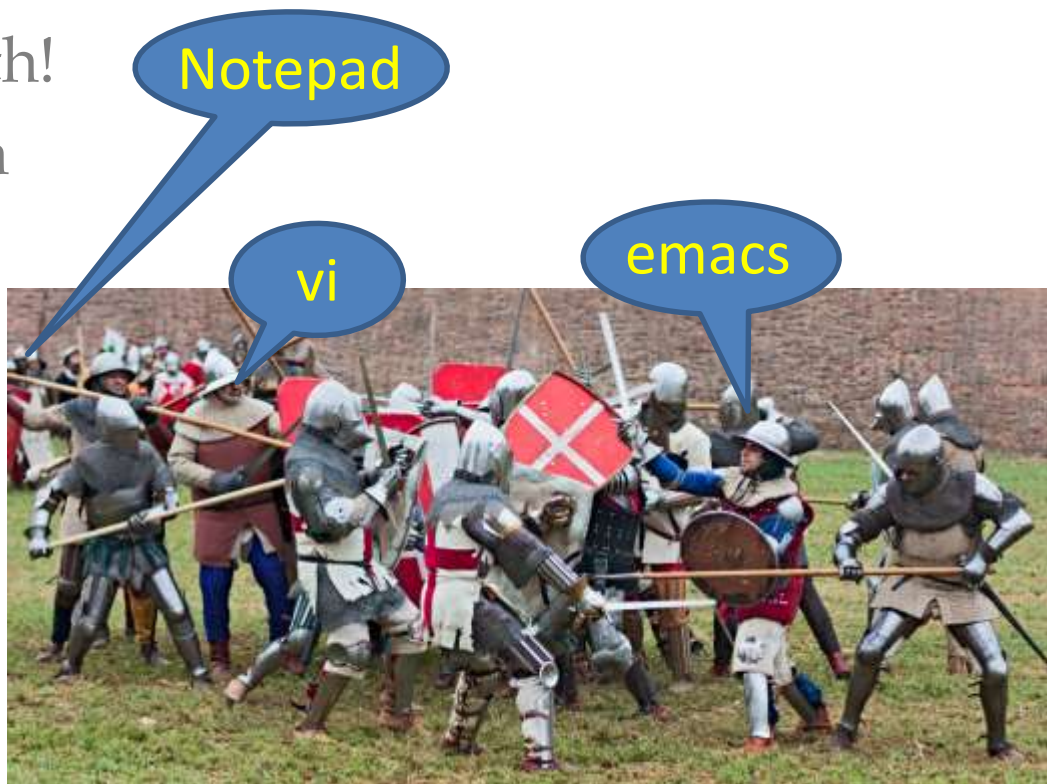
```
public class Hello {

    public static void main(String[] args) {
        System.out.println("Hello world");
    }

}
```

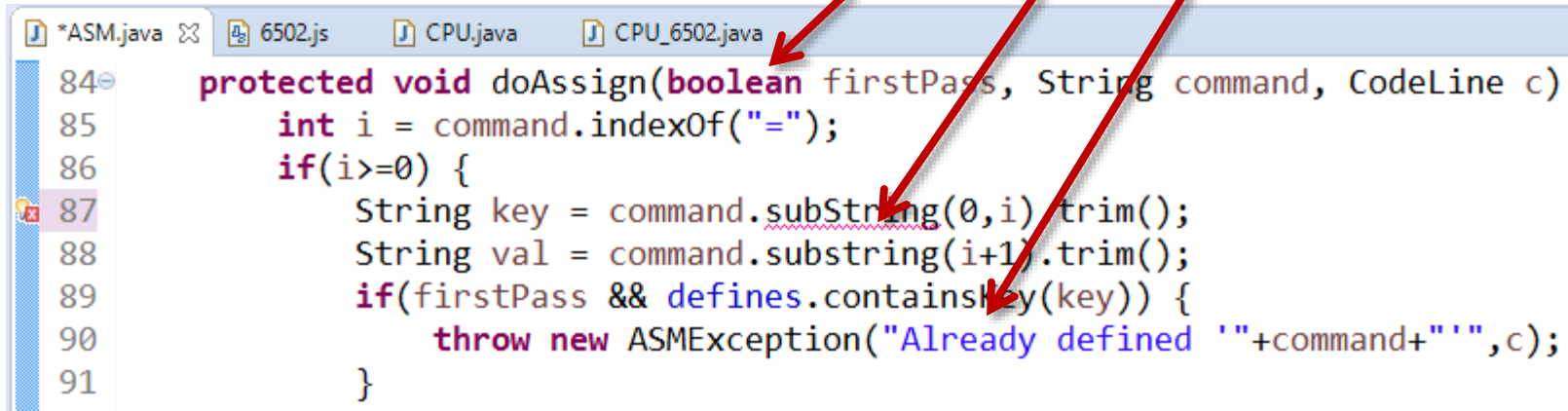
Battle of the Editors

- Developers will defend their editors to the death!
- Use what you are given
- Keep an open mind
- Try all the things!



Features of an IDE

- Syntax Highlighting
 - Reserved words
 - Strings and constants
 - Errors and warnings



The screenshot shows an IDE window with a tab labeled `*ASM.java`. The code is as follows:

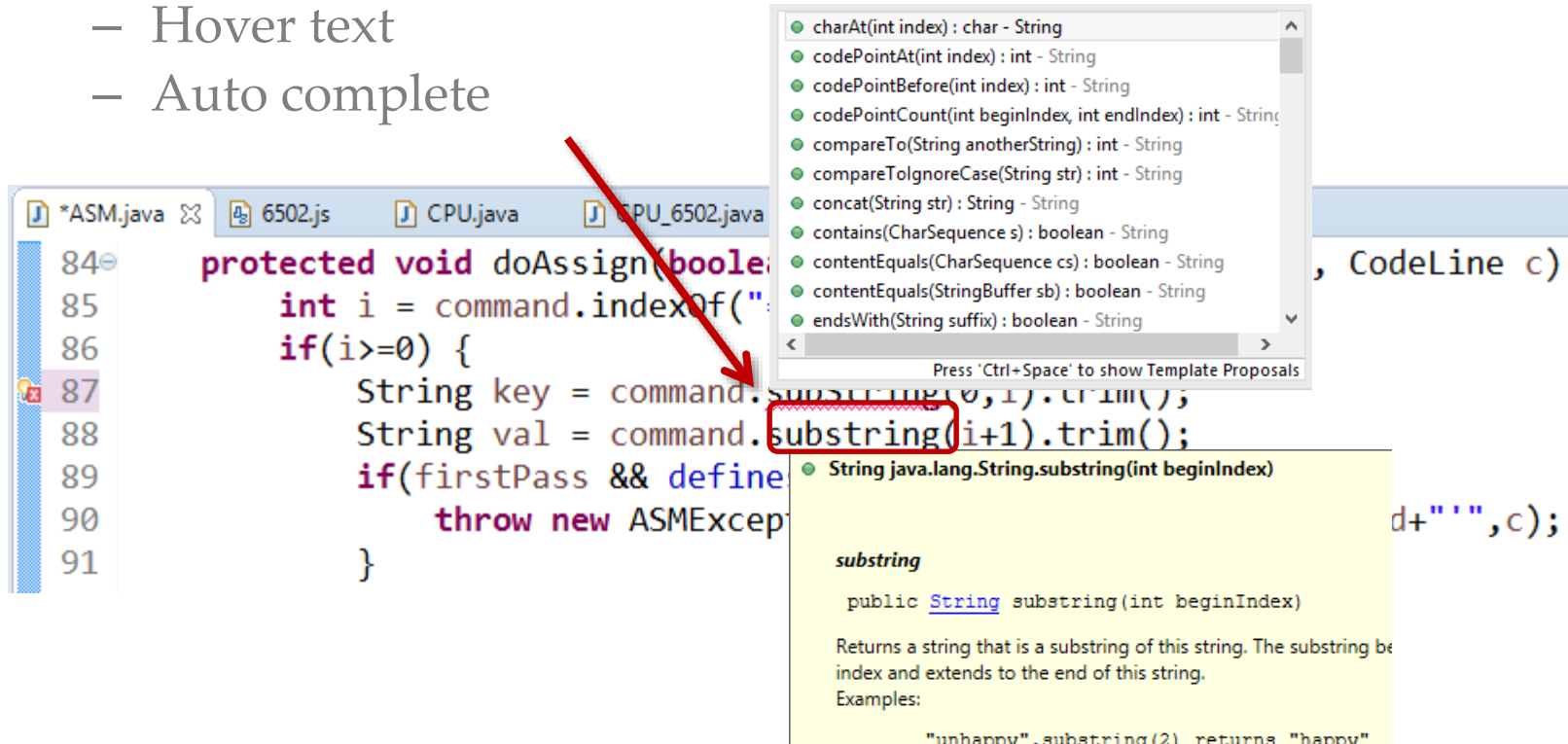
```
84 protected void doAssign(boolean firstPass, String command, CodeLine c)
85     int i = command.indexOf("=");
86     if(i>=0) {
87         String key = command.substring(0,i).trim();
88         String val = command.substring(i+1).trim();
89         if(firstPass && defines.containsKey(key)) {
90             throw new ASMException("Already defined '"+command+"'",c);
91         }
```

Three red arrows point to specific features:

- The first arrow points to the `boolean` keyword, illustrating reserved words.
- The second arrow points to the `substring(0,i)` method call, illustrating strings and constants.
- The third arrow points to the `containsKey(key)` method call, illustrating errors and warnings.

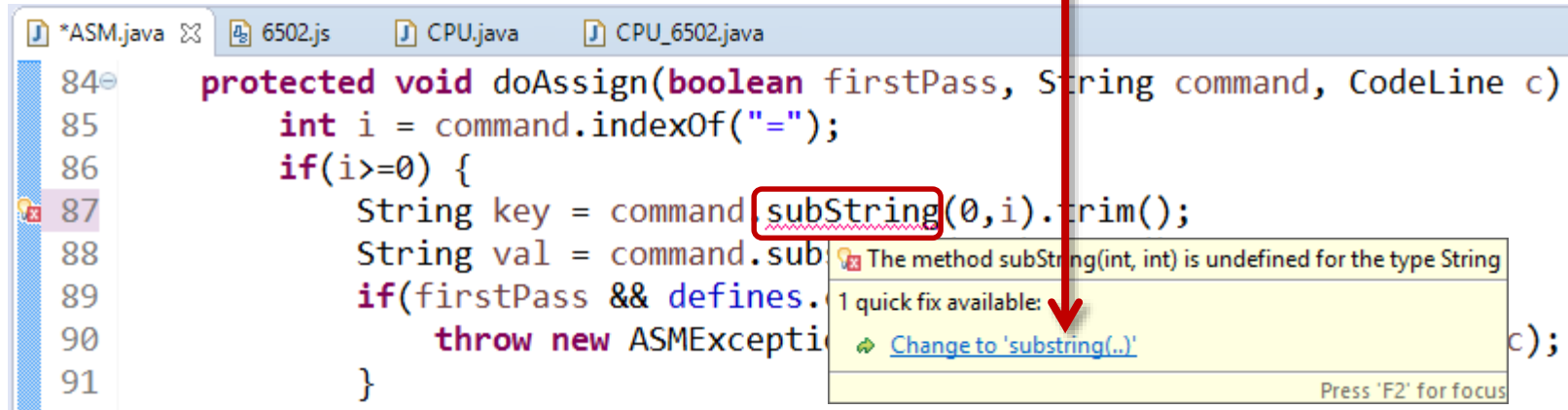
Features of an IDE

- Content Assist
 - Hover text
 - Auto complete



Features of an IDE

- Help with errors
 - Hover explanation
 - Quick fix



Features of an IDE

- Project/file Management

The screenshot displays an IDE interface with two main panels. On the left is the 'Package Explorer' showing a project hierarchy: 'Budget' (a folder icon) containing 'computerarcheology' (a folder icon) which contains 'src' (a folder icon). 'src' contains several sub-folders: 'asm', 'cleans', 'code', and 'cpu'. The 'asm' folder is expanded, showing files 'ASM.java' and 'ASMException.java'. The 'cpu' folder is also expanded, showing a list of files including 'CPU_6502.java', 'CPU_6803.java', 'CPU_6809.java', 'CPU_DVG.java', 'CPU_None.java', 'CPU_Z80.java', 'CPU.java', 'Opcode.java', and '6502.js'. Red arrows point from the 'ASM.java' file in the Package Explorer to the corresponding code editor window. The code editor window has tabs for '*ASM.java', '6502.js', 'CPU.java', and 'CPU_6502.java'. The active tab is '*ASM.java', showing Java code. The code includes a method 'doAssign' with parameters 'boolean firstPas' and 'int i'. A red box highlights the call 'CU.hex4(5542);' on line 86. A tooltip for 'String code.CU.hex4(int value)' is visible over this call. Below the code editor, there is a toolbar with icons for undo, redo, and other editing functions. The code in the editor is as follows:

```
84= protected void doAssign(boolean firstPas
85     int i = command.indexOf("=");
86     String s = CU.hex4(5542);
87     if(i>=0) {
88         String key
89         String val
90         if(firstPas
91         throw n
92     }
93
94     if(val.equals("+")) {
95         lastAssign = lastAssign+1;
96         val = ""+lastAssign;
97     } else {
98         if(CU.isNumeric(val, 10)) {
99             lastAssign = CU.parseInt
```

Features of an IDE

- Debugger

The screenshot displays an IDE interface. On the left, a code editor shows a Java file named `ASM.java` with the following code:

```
84 protected void doAssign(boolean firstPass) {
85     int i = command.indexOf("=");
86     if(i>=0) {
87         String key = command.substring(0, i);
88         String val = command.substring(i+1);
89         if(firstPass && defines.containsKey(key)) {
90             throw new ASMException("Already defined '" +
91                                     key + "'");
92         }
93         if(val.equals("+")) {
94             lastAssign = lastAssign+1;
95             val = ""+lastAssign;
96         } else {
```

Two red arrows highlight specific features: one points to the line number margin (line 93), and the other points to the variable `key` in the code.

On the right, the 'Variables' window is open, showing the current state of variables:

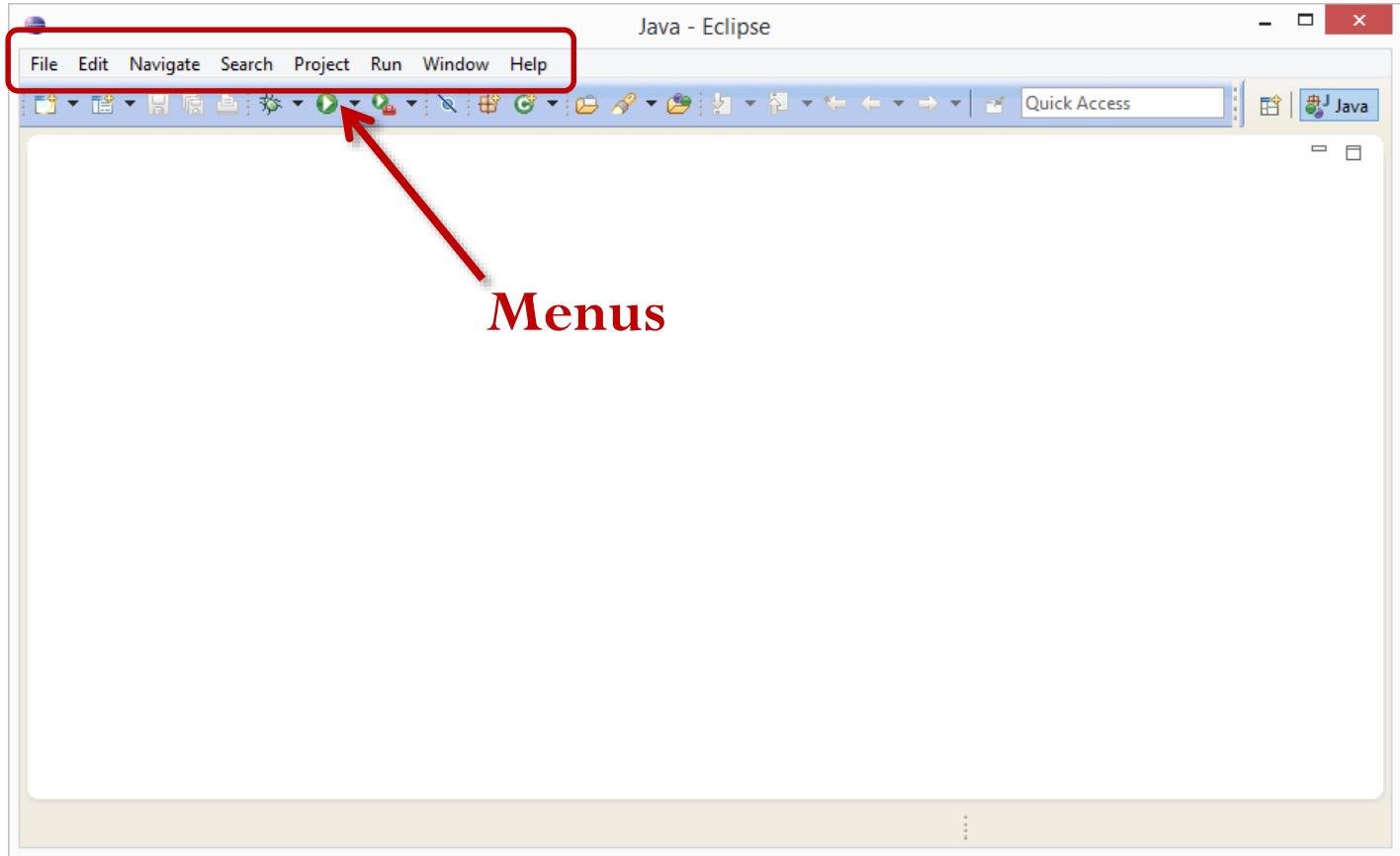
Name	Value
▶ this	ASM (id=23)
▶ firstPass	true
▶ command	"CXM0P = 0x0000" (id=24)
▶ c	CodeLine (id=29)
▶ i	9
▶ key	"CXM0P" (id=31)
▶ val	"0x0000" (id=32)

Eclipse in Particular

- Free (Community Driven)
- Extensible through plugins (written in Java)
- Available plugins for most languages
- Great for multi-language projects

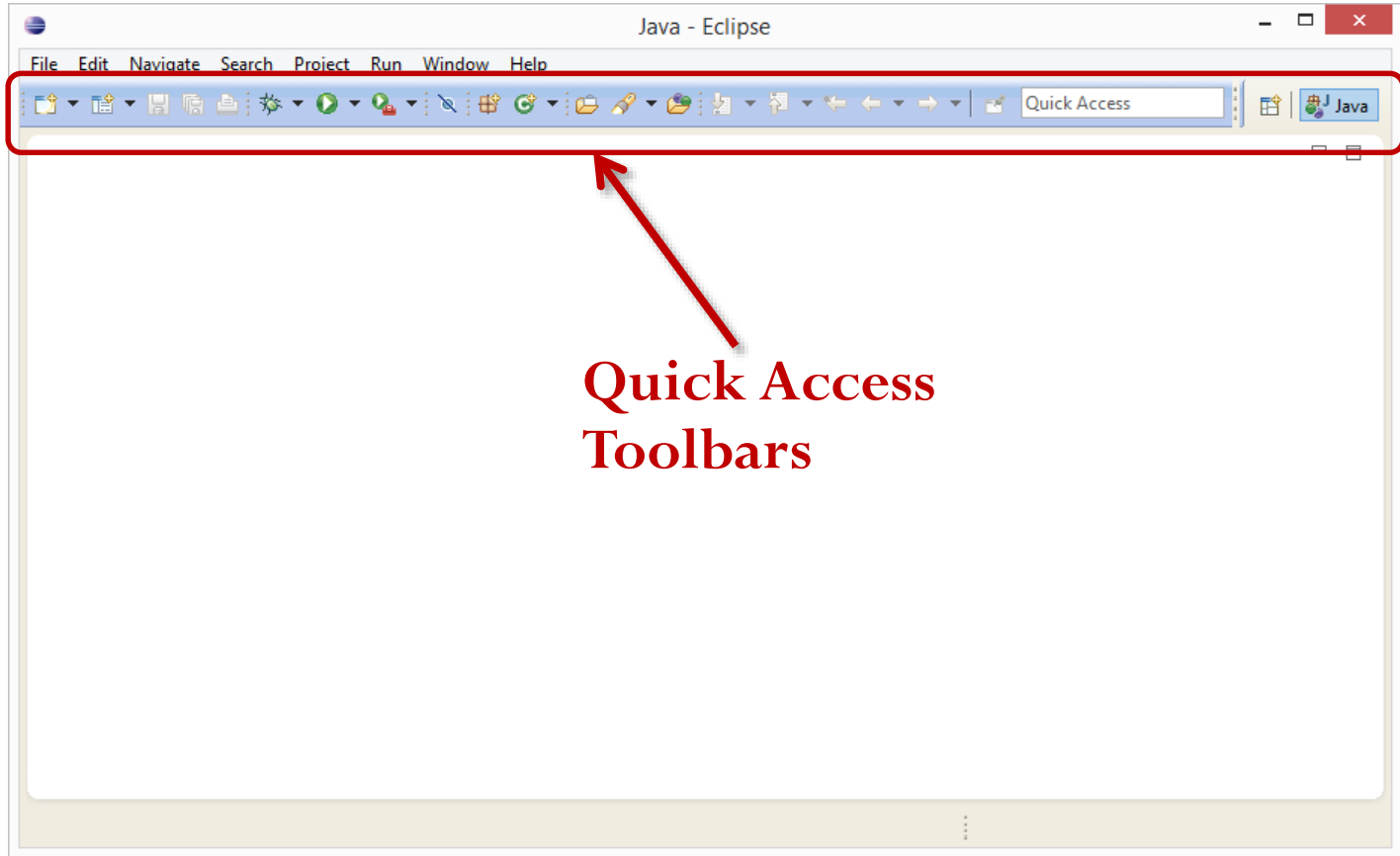


UI Layout



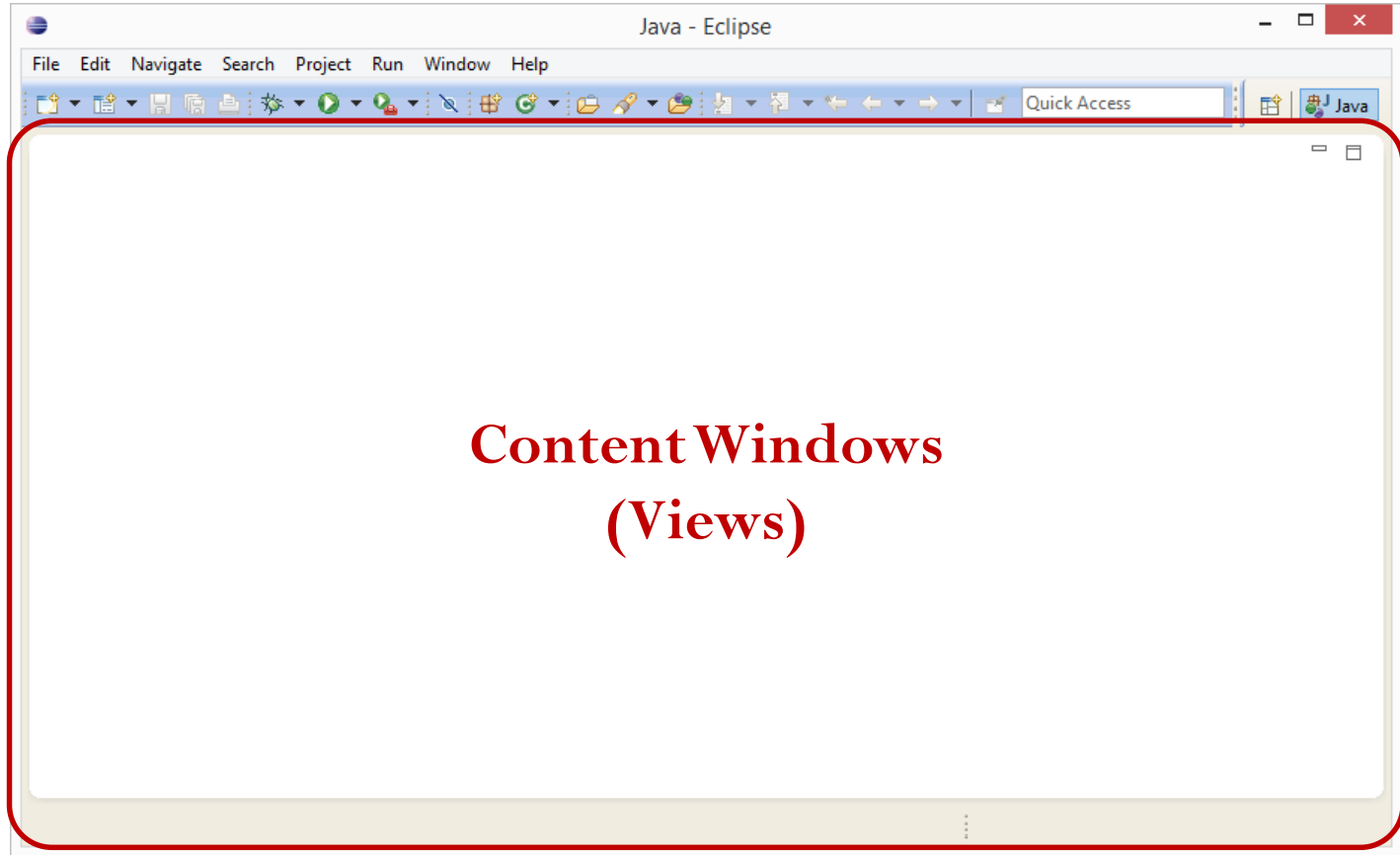
Menus

UI Layout

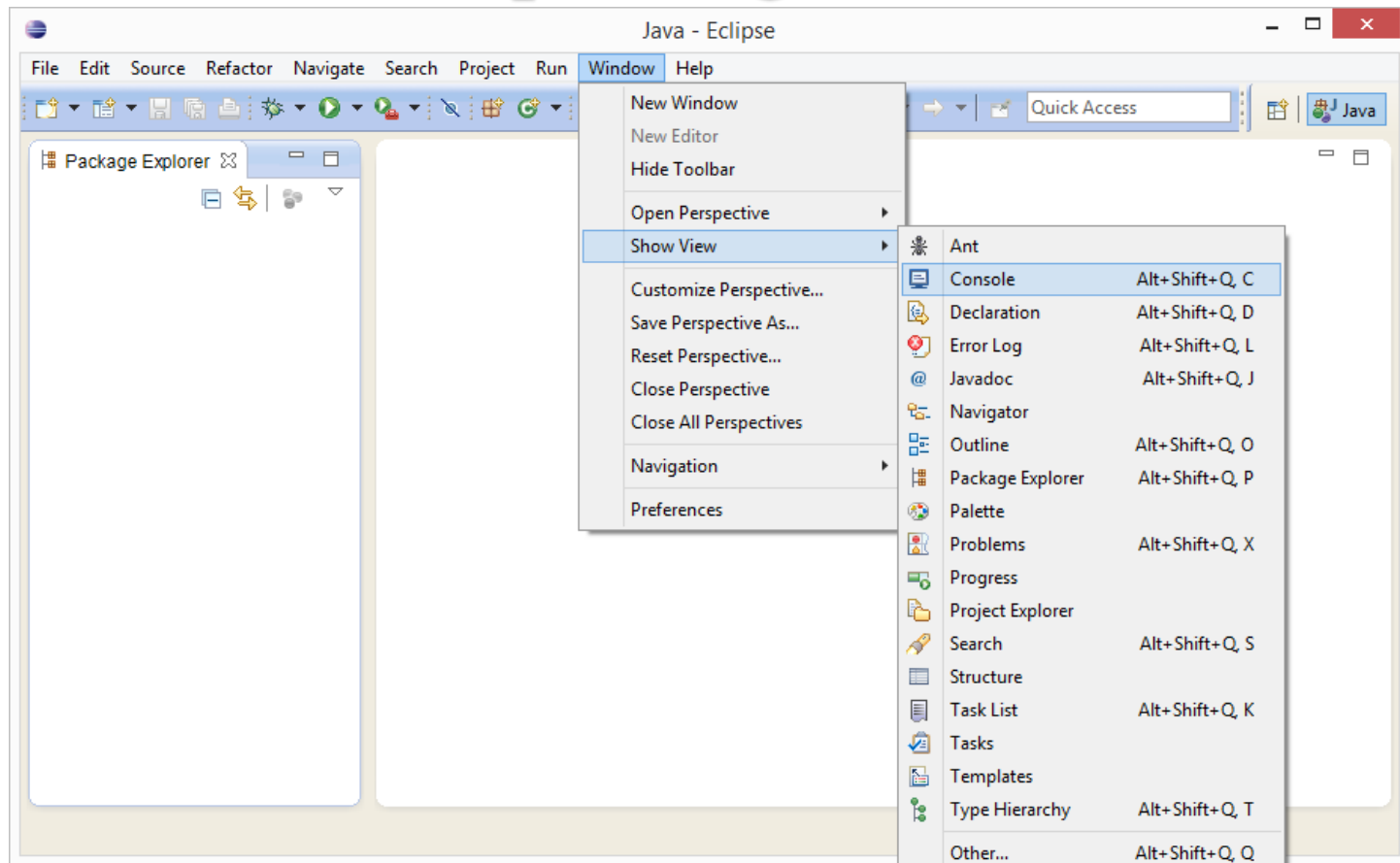


**Quick Access
Toolbars**

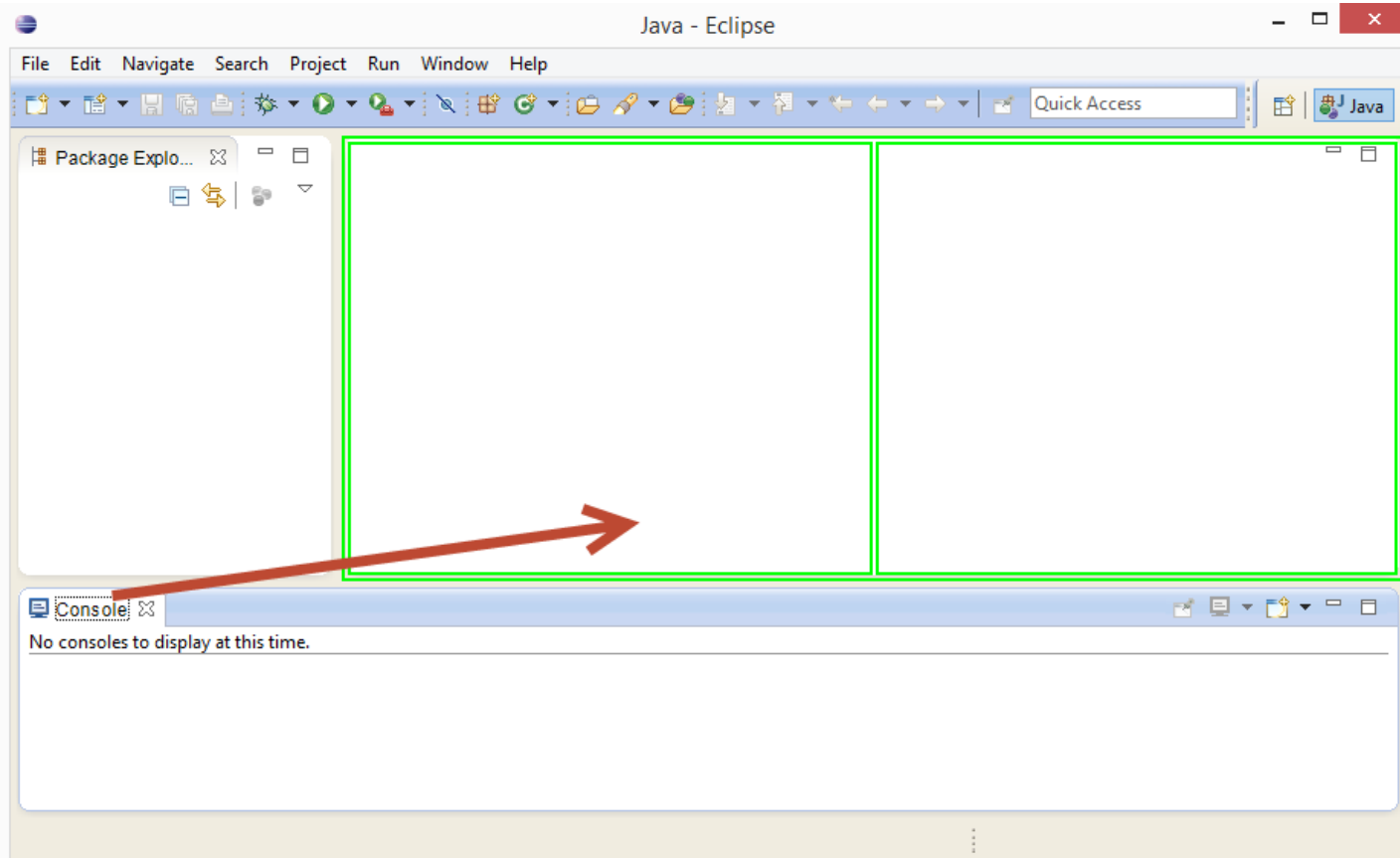
UI Layout



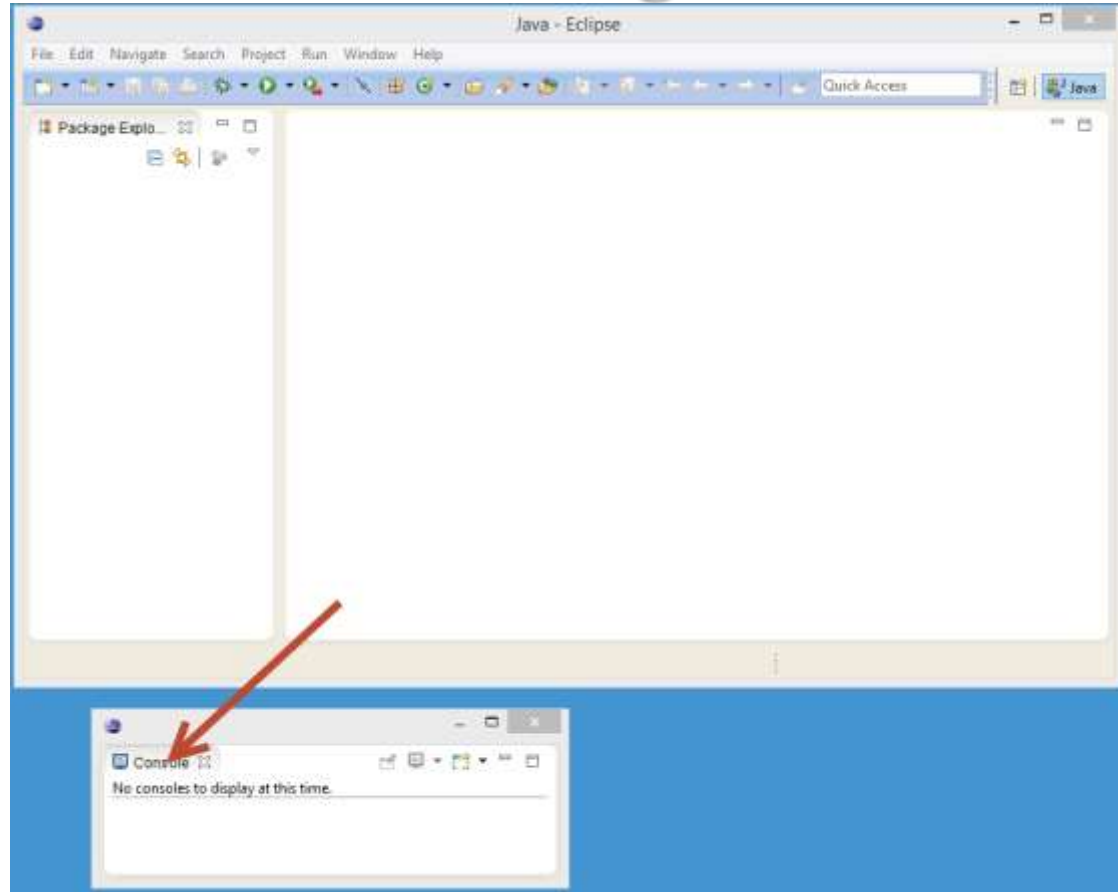
Opening a View



Moving Views Around



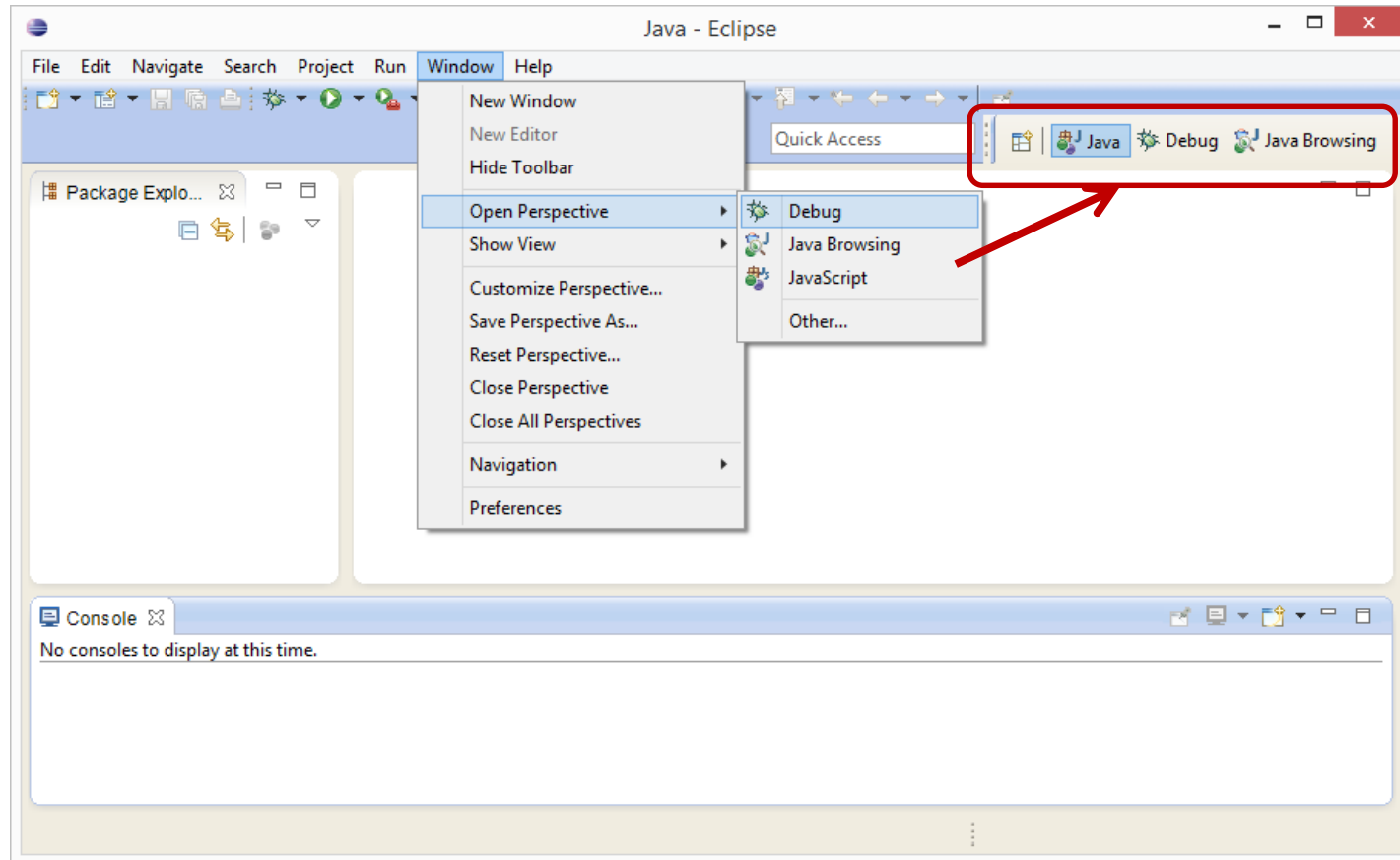
Undocking a View



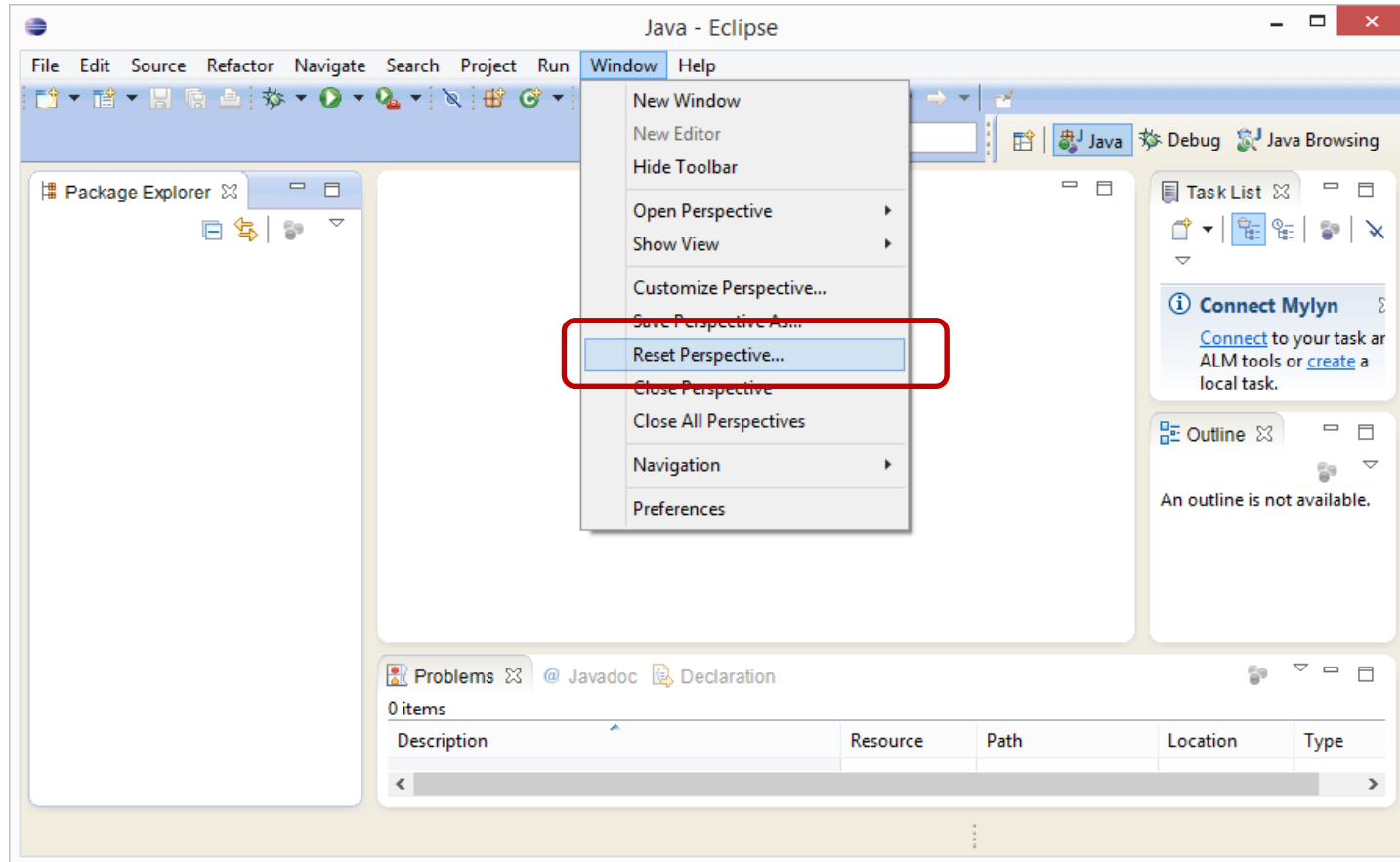
Perspectives

- A perspective is a group of related views:
 - Editing
 - Debugging
 - Database Tools
- You can switch between tasks by changing perspectives
- You can customize perspectives and create new ones
- You can reset a perspective to its original layout

Opening a Perspective



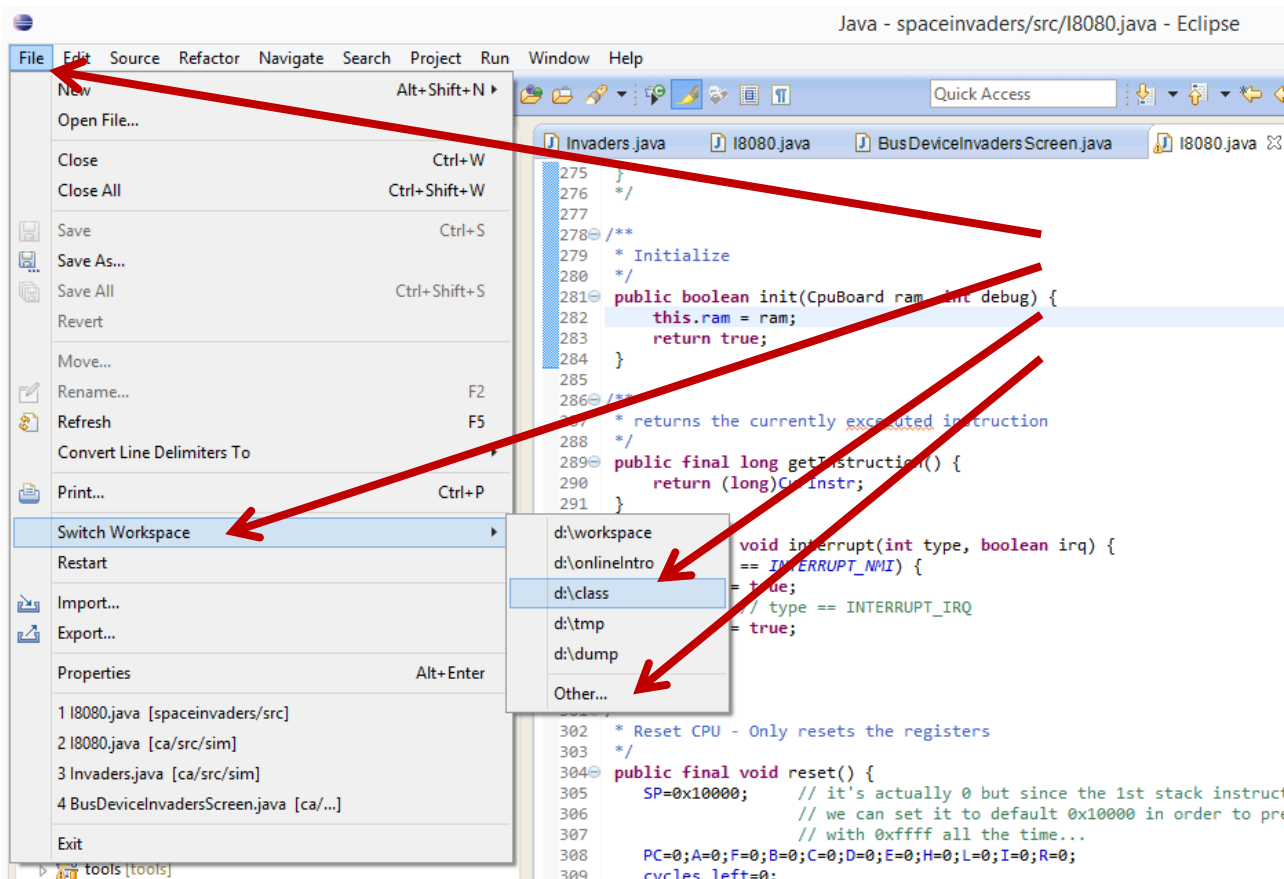
Restoring the Original Layout



Workspaces and Projects

- A “project” is a collection of source code and resources for a single program.
- A “project” has its own directory.
- A “workspace” is a collection of projects.
- A “workspace” directory is the parent for all the project directories it contains.

Changing Workspaces



Downloading Eclipse

Eclipse Downloads

www.eclipse.org/downloads/

Home Downloads Users Members Committers Resources Projects About Us

Eclipse Downloads

Packages Developer Builds

Eclipse Kepler (4.3.1) SR1 Packages for Windows

Eclipse Standard 4.3.1, 199 MB
Downloaded 3,775,568 Times [Other Downloads](#)

The Eclipse platform, and all the tools needed to develop and debug its Java and Plug-in Development Tooling, Git and CVS...

Package Solutions [Filter Packages](#)

Eclipse IDE for Java EE Developers, 247 MB
Downloaded 1,984,647 Times

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

Eclipse IDE for Java Developers, 151 MB
Downloaded 873,380 Times

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

Windows 32 Bit
Windows 64 Bit

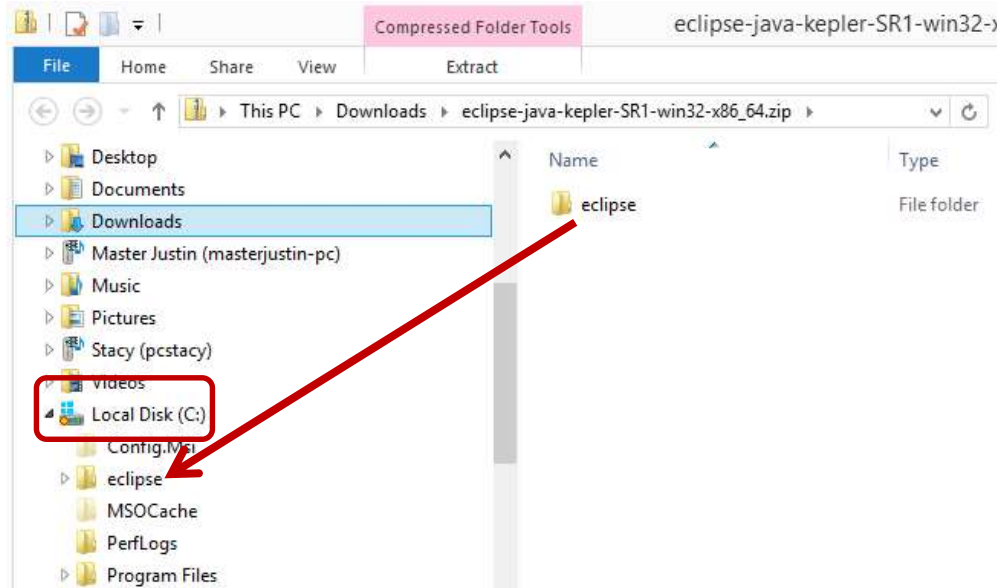
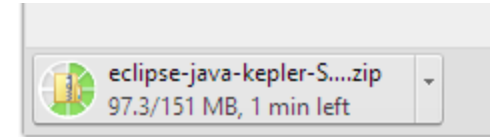
	Standard 4.3.1	Java
BIRT Framework		
Code Recommenders Developer Tools		✓
C/C++ Development Tools		
Data Tools Platform		
Git Team Provider	✓	✓
Java Development Tools	✓	✓
Java EE Developer Tools		
JavaScript Development Tools		
Jubula Functional Testing		
Maven Integration for Eclipse		✓
Mylyn Task List		✓
Parallel Tools Platform		
Plug-in Development Environment	✓	
Remote System Explorer		
Remote Application Platform		
Scout		
WindowBuilder Core		✓
XML Editors and Tools		✓
Xtend		
Xtext		

Legend:

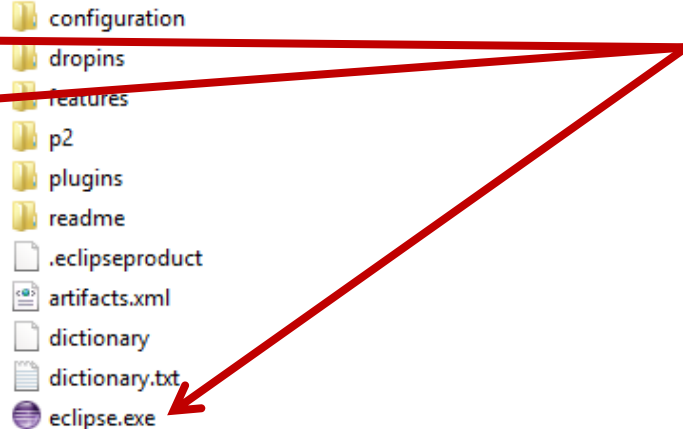
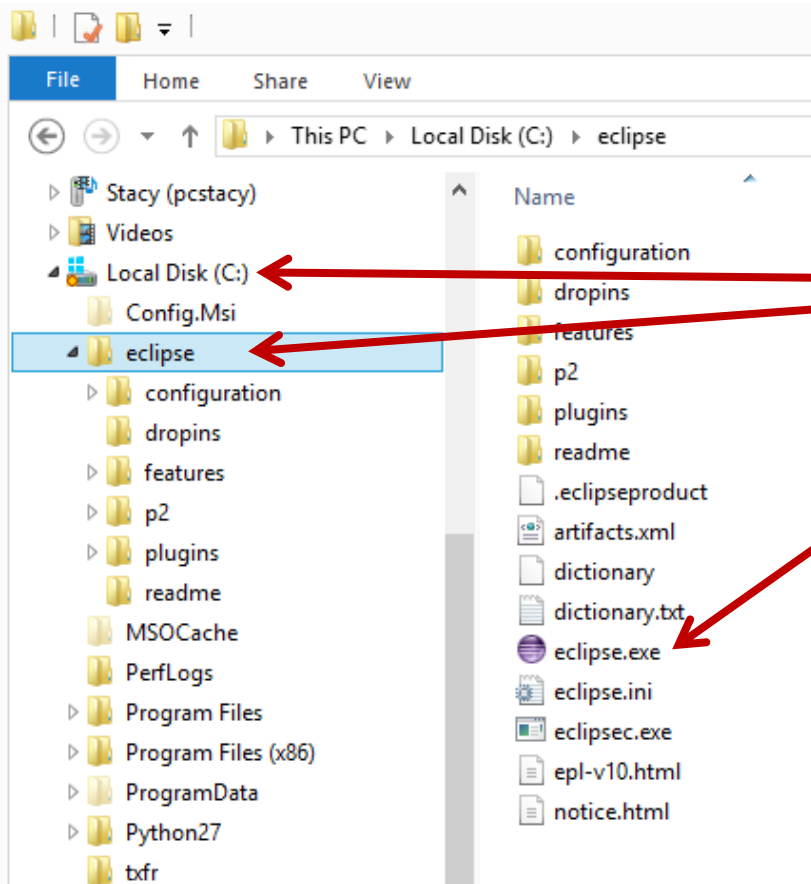
✓ included ✓ Included (with Source)

Unzipping Eclipse

- Download the zip file
- Open it
- Copy “eclipse” to your drive

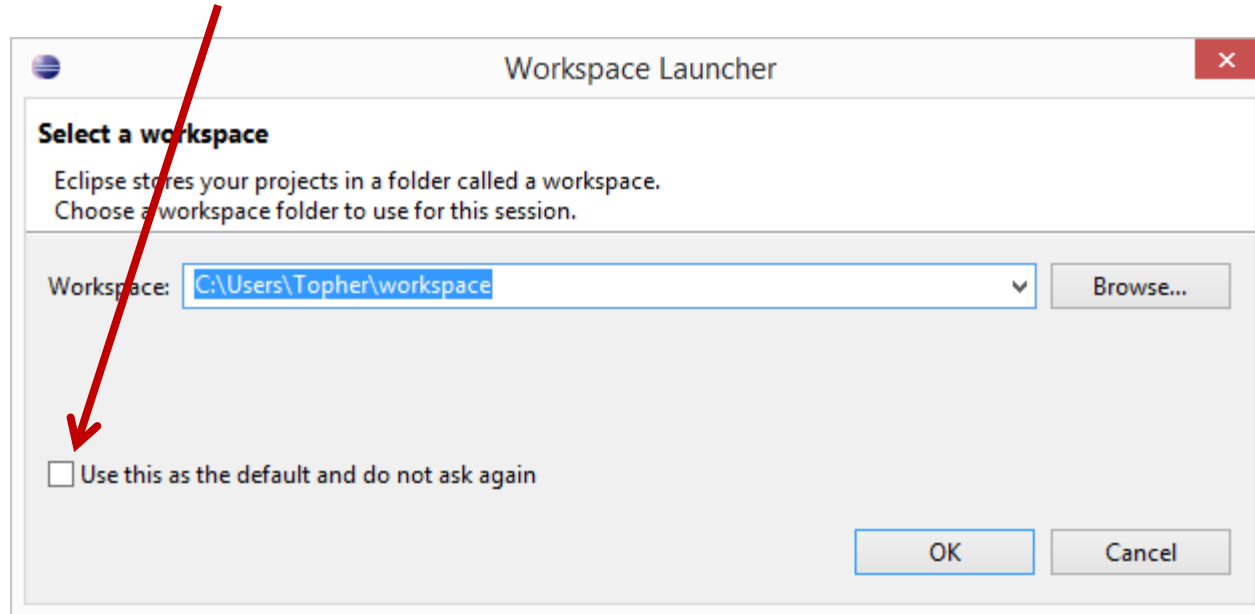


Launching Eclipse

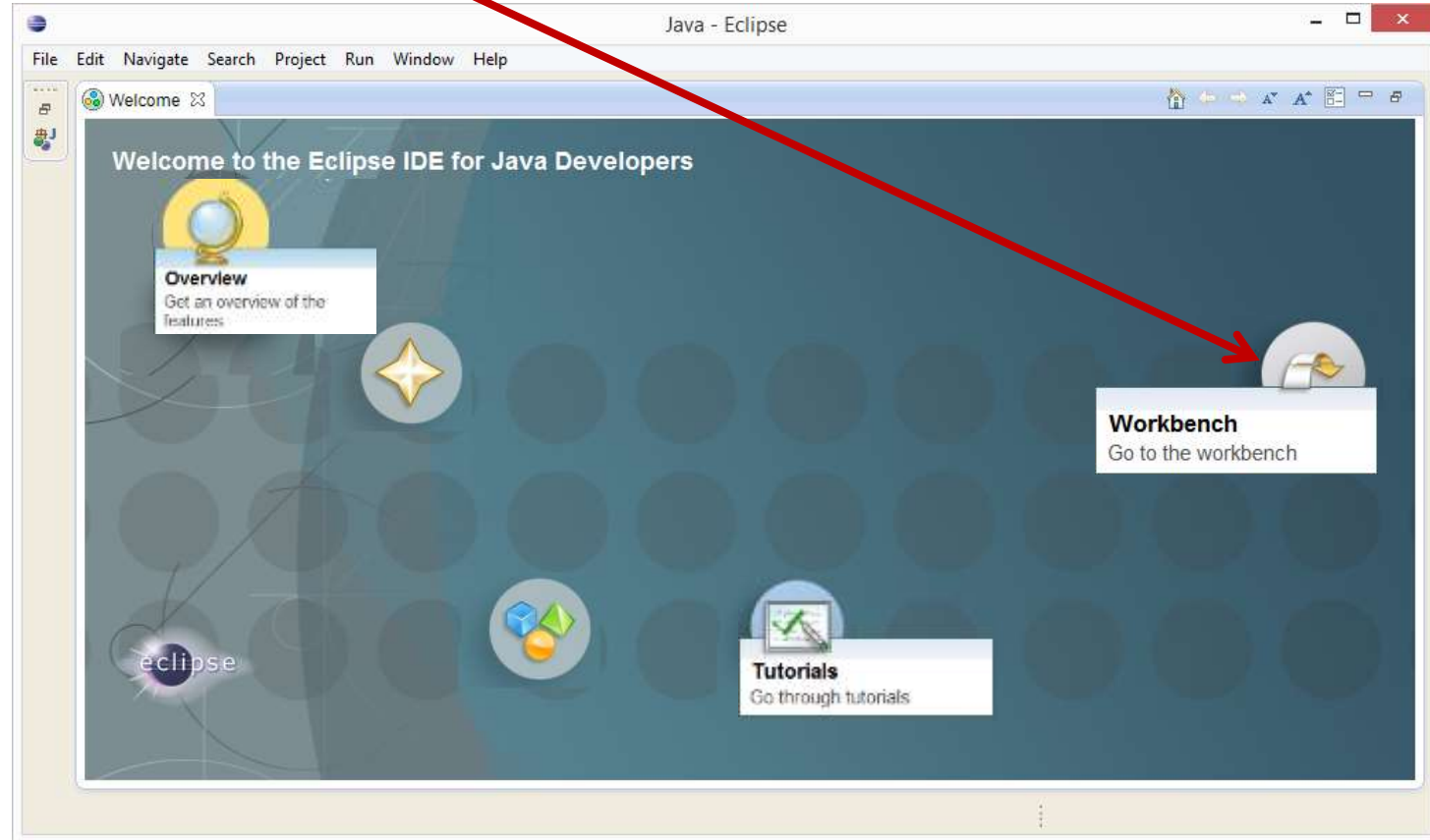


First Launch

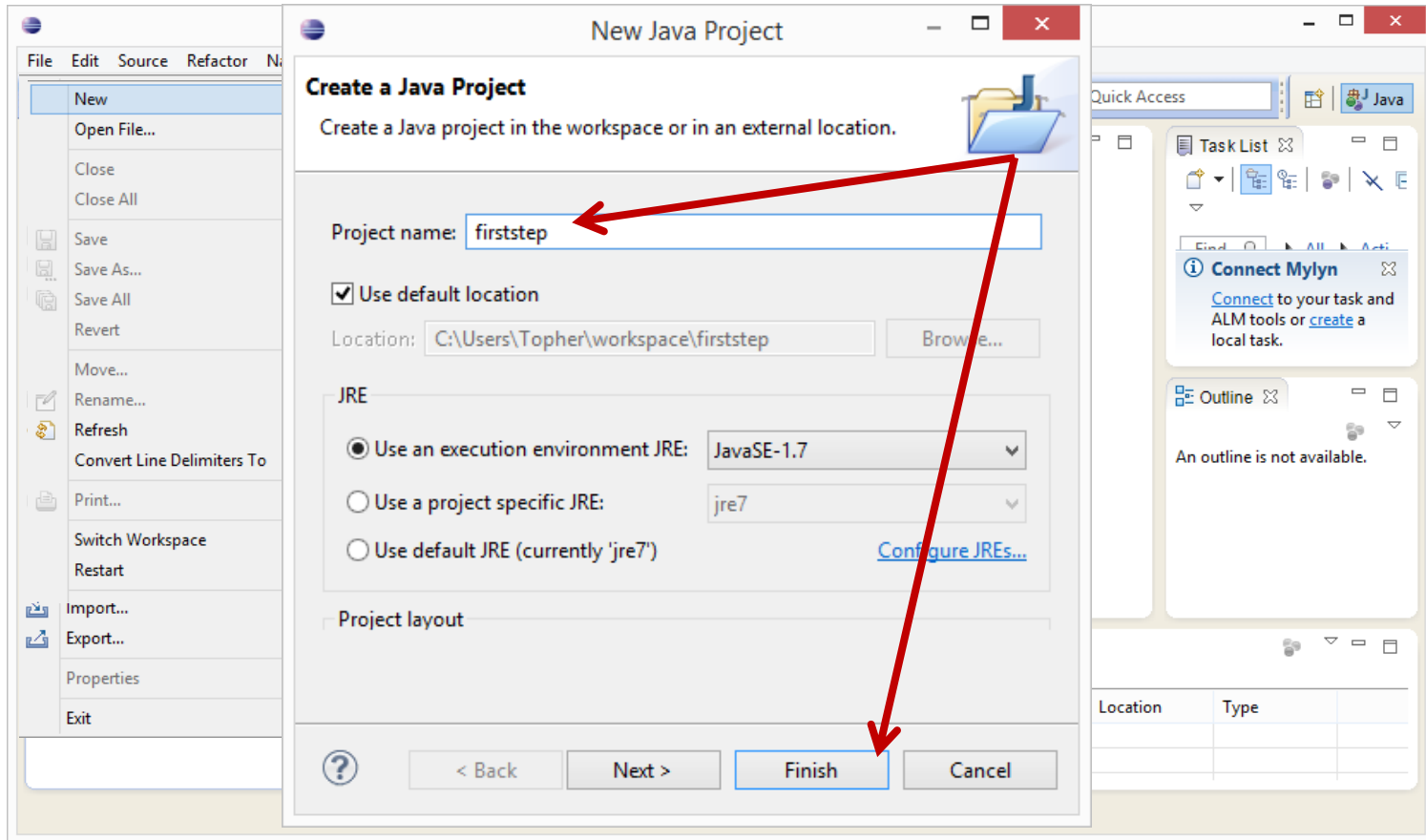
- Set the default workspace
- You can switch/create any time



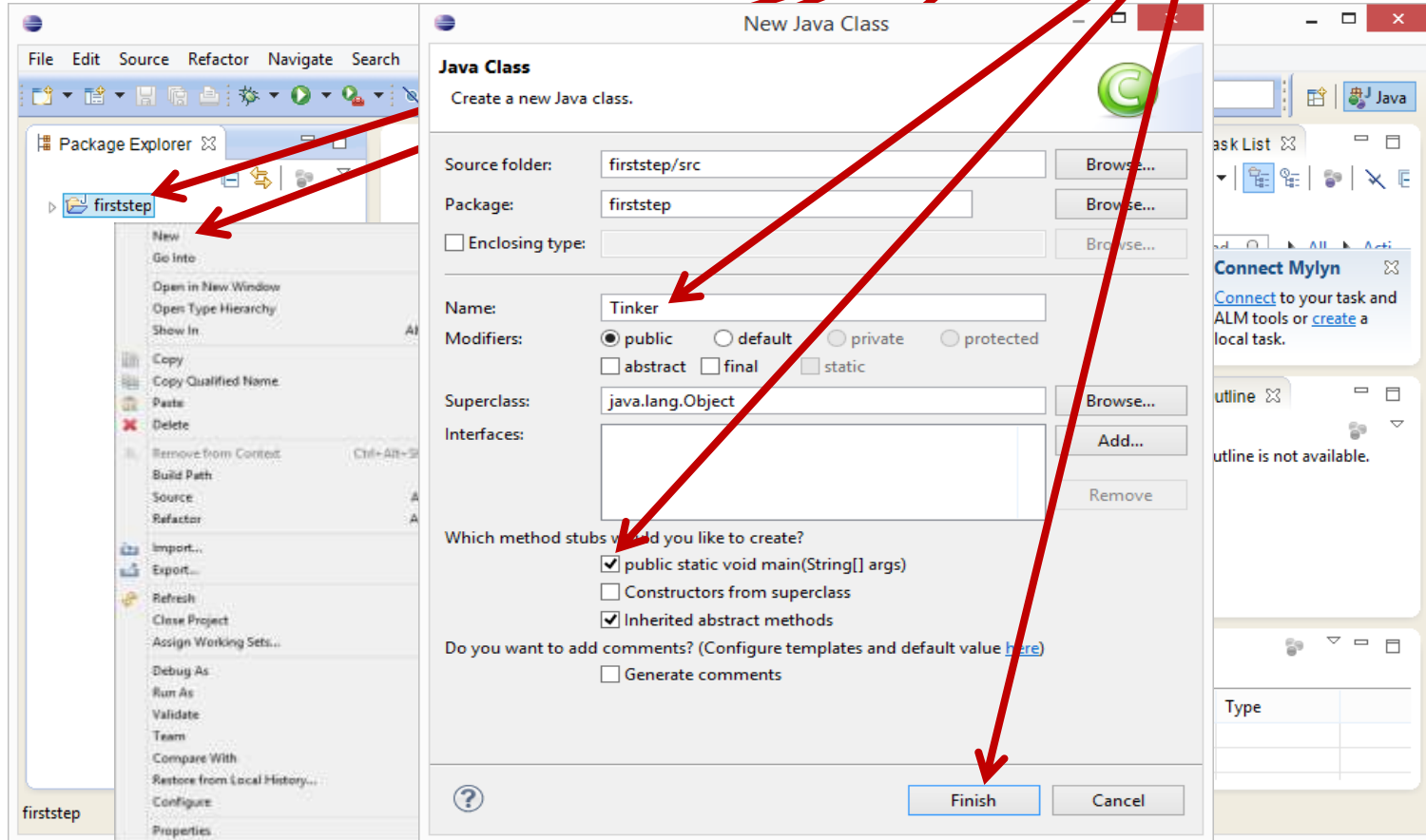
Workbench



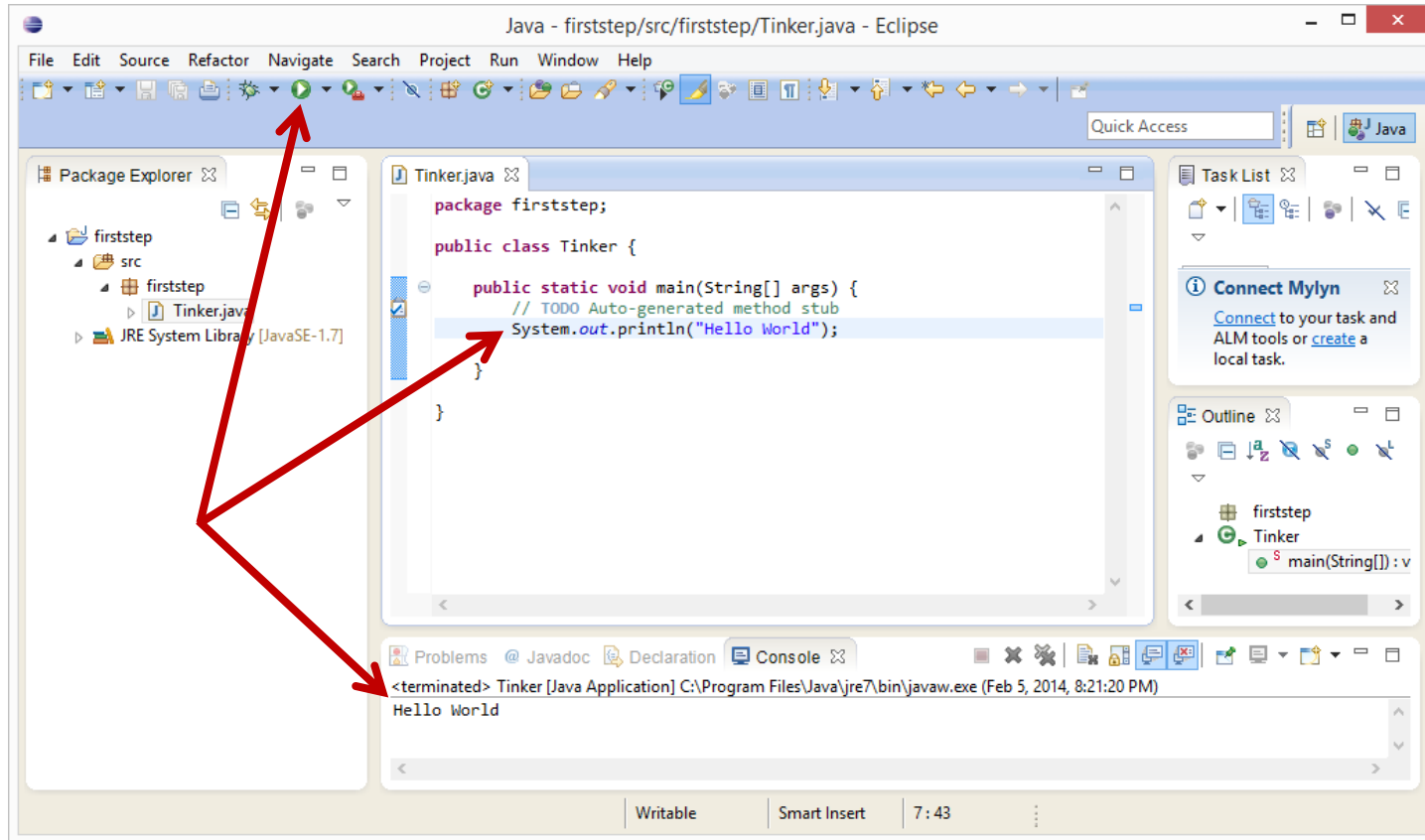
Create a Java Project



Create a New Class



Run



Tinkering

- Create a new project in eclipse
- Open a few views, move them, close them
- Reset the perspective
- Create a new project called “first”
- Add a “Tinker” class with a “main”
- Run your very first project!

