#### Introduction to Eclipse



# Overview

- Eclipse Background
- Obtaining and Installing Eclipse
- Creating a Workspaces / Projects
- Creating Classes
- Compiling and Running Code
- Debugging Code
- Sampling of Features
- Summary

# What is Eclipse?

- Eclipse started as a proprietary IBM product (IBM Visual age for Smalltalk/Java)
  - Embracing the open source model IBM opened the product up
- Open Source
  - It is a general purpose open platform that facilitates and encourages the development of third party plug-ins
- Best known as an Integrated Development Environment (IDE)
  - Provides tools for coding, building, running and debugging applications
- Originally designed for Java, now supports many other languages
  - Good support for C, C++
  - Python, PHP, Ruby, etc...

Prerequisites for Running Eclipse

- Eclipse is written in Java and will thus need an installed JRE or JDK in which to execute
  - JDK recommended

# Eclipse on GL

- This years coordinated release (known as Ganymede) of the Eclipse IDE for Java Developers has been installed on GL
  - From any of the Linux machines in the labs simply run the command eclipse

# **Obtaining Eclipse**

- Eclipse can be downloaded from...
  - <u>http://www.eclipse.org/downloads/packages/</u>
  - Be sure to grab "Eclipse IDE for Java Developers"
- Eclipse comes bundled as a zip file (Windows) or a tarball (all other operating systems)
  - Some versions of Linux (i.e. Fedora, Ubuntu) offer Eclipse in their respective repositories and can be downloaded using the appropriate tool (i.e. yum, apt-get)

# Installing Eclipse

- Simply unwrap the zip file to some directory where you want to store the executables
- On windows
  - I typically unwrap the zip file to C:\eclipse\
  - I then typically create a shortcut on my desktop to the eclipse executable
    - C:\eclipse\eclipse.exe
- Under Linux
  - I typically unwrap to /opt/eclipse/

# Launching Eclipse

- Once you have the environment setup, go ahead and launch eclipse
- You should see the following splash screen...



# Selecting a Workspace

- In Eclipse, all of your code will live under a workspace
- A workspace is nothing more than a location where we will store our source code and where Eclipse will write out our preferences
- Eclipse allows you to have multiple workspaces each tailored in its own way
- Choose a location where you want to store your files, then click OK

Se Workspace Launcher	
Select a workspace	
Eclipse Platform stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: C:\Documents and Settings\Dan\workspace	Browse
Use this as the default and do not ask again	
	OK Cancel

## Welcome to Eclipse

- The first time you launch Eclipse, you will be presented with a welcome screen
- From here you can access an overview to the platform, tutorials, sample code, etc...
- Click on the arrow on the right to get to the actual IDE



### **Eclipse IDE Components**



### Creating a New Project

- All code in Eclipse needs to live under a project
- To create a project: File  $\rightarrow$  New  $\rightarrow$  Java Project

File Edit Source Refactor Navigate S	earch Project Run Window Help						
New Alt+Shift+N	🕨 🏄 Java Project	11日 - 四 - 4				😭 🐉 Java	
Open File	Project				- 0)	Task List 🛛	- 8)
Close Ctrl+W	🖶 Package					r 🖓 1	• 59 7
	G Class						
La Save Ctrl+S	Interface					Find:	AIL
[값 Save All Ctrl+Shift+S	Annotation					Uncategoriz	ea
Revert	Source Folder						
Move	🍐 Java Working Set						
Rename F2	😂 Folder						
🗞 Refresh 🛛 🕫	File						
Convert Line Delimiters To	Untitled Text File						
Print Ctrl+P	Task					E Outline	
Switch Workspace							59 ▽
Restart						An outline is not availab	le.
🔁 Import	Other Ctrl+N	]					
Export							
Properties Alt+Enter							
Exit							
	-						
	🖹 Problems 🛛 🖉 Javad	oc 😟 Declaration					•
	0 items						
	Description A	Resource	Path	Location	Туре		
			ŝ			i 🐴 🔮	🔶 💖 🖂 (

### Creating a New Project (continued)

🖶 New Java Project		_ 🗆 🐱
Create a Java Project Create a Java project in the workspace or i	n an external location.	
Project name: hello-world Contents Oreate new project in workspace Oreate project from existing source Directory: C:\Documents and Settings	Dan \workspace \hello-world	Browse
Use def <u>a</u> ult JRE (Currently 'jre1.6.0_     Use a project specific JRE:     Use an execution en <u>v</u> ironment JRE:	<b>05)</b> jre1.6.0_05 JavaSE-1.6	Configure JREs
Project layout          Use project folder as root for sources         O use project folder as root for sources         O create separate folders for sources a         Working sets         Add project to working sets         Working sets:	and class files and class files	Configure default
(ହ) < <u>B</u> ad	<u>N</u> ext >	ish Cancel

 Enter a name for the project, then click
 Finish

# Creating a New Project (continued)

 The newly created project should then appear under the Package Explorer

🖨 Java - Eclipse Platform						
<u>File E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch	n <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp					
📬 • 🔚 🗁    🏇 • 🔕 • 🖓 •	볼 📽 🞯 • 🗄 🍅 🖨 • 🗄 🖢 - 🖓 •	• •	-			😭 🖏 Java
増 Package Explor 🛛 🍃 Hierarchy 🗖 🗖	)[					Task List 🛛 🗖
F 🚖 😜 🎽						1 😚 🚷 🚡 • 👔 🔻
🗈 🚔 hello-world						
						Uncategorized
						An aution is not susibile
						An outline is not available.
	🖹 Problems 🖾 🖉 🖉 Javadoc 😣 Declaratio	n				
	0 items					
	Description A	Resource	Path	Location	Туре	
ê □ <sup>◆</sup> hello-world			Î			i 🕿 🍳 🐼 🔮 🔶

### The src folder

 Eclipse automatically creates a folder to store your source code in called src



### Creating a Class

 To create a class, simply click on the New button, then select Class —\_\_\_



### Creating a Class (continued)

🖨 New Java Clas	55	🛛 🔀
Java Class Create a new Java	dass.	C
Source folder: Package:	hello-world/src edu.umbc.dhood2	Browse Browse Browse
Na <u>m</u> e: Modifiers: Superdass: Interfaces:	HelloWorld <ul> <li>public</li> <li>default</li> <li>private</li> <li>protected</li> </ul> abstract         final         static           java.lang.Object	Brows <u>e</u> Add Remove
Which method stubs	would you like to create?	
0	Finish	Cancel

- This brings up the new class wizard
- From here you can specify the following...
  - Package
  - Class name
  - Superclass
  - Whether or not to include a main
  - Etc...
- Fill in necessary information then click Finish to continue

### The Created Class

As you can see a number of things have now happened...



# Compiling Source Code

- One huge feature of Eclipse is that it automatically compiles your code in the background
  - You no longer need to go to the command prompt and compile code directly
- This means that errors can be corrected when made
  - We all know that iterative development is the best approach to developing code, but going to shell to do a compile can interrupt the normal course of development
  - This prevents going to compile and being surprised with 100+ errors

### **Example Compilation Error**

• This code contains a typo in the println statement...



# Example Compilation Error (continued)

• When clicking on the light bulb, Eclipse suggests changing printn to either print or println

🖨 Java - hello-world/src/edu/umbc/dho	od2/HelloWorld.java -	Eclipse Platform					
File Edit Source Refactor Navigate Search	Project Run Window H	Help					
📫 • 🔚 🗁   🏇 • 🕥 • 💁   🛃	\$ 🗳 🞯 🔹 🖄 😂 🖉	🔗 • 🕴 🝄 🌛 🤪	) 😫 👻 😽	• 🌾 🔶 • 🔶 •			😭 🎒 Java
🛱 Package Explor 🛛 🍃 Hierarchy 🗖 🗖	🛃 HelloWorld.java 🗙					- 0)	🗐 Task List 🛛 👘 🗖
□ 🔄 😜 🏹	package edu.u	mbc.dhood2;					🖆 🚳 🏣 🔹 💝
	public class ⊖ /** * @param	HelloWorld {					Find: All All
	*/	atic moid main (	Staing[]				
		atie vold main(.	String[]	args) (			
	🔞 Syste	m. <i>out</i> . <mark>printn</mark> ("H	ello Worl	d!");		-	
	<pre>{2     System.out.friint("Hello World!");</pre>					×	E Outline S □ □ S ↓ 2 ≥ ≥ ↓ 2 edu.umbc.dhood2 D ↓ 4 ElloWorld S main(String[])
		<u>`-</u> ``					
	🖹 Problems 🛛 🥥 🥥 J	avadoc 😣 Declaration					
1 error, 0 warnings, 0 others Description							
	Errors (1 item)					. / 8	
	🚱 The method pr	intn(String) is undefined f	HelloWorld.j	hello-world/src/e	line 10	Java Problem	
The method printn(String) ied for t	the type PrintStream Write	ble Smart Insert	10:26	ŝ			i 🕋 🍳 🗷 💖 🔶

# Running Code

An easy way to run code is to right click on the class and select Run As → Java Application

🖨 Java - hello-world/src/	/edu/umbc/dhood2/Hellov	World.java - Eclipse Pl	latform					
File Edit Source Refactor	Navigate Search Project R	un Window Help						
🗄 🖬 🖶 📄 🎄 • 🕻	)• 🧣 🕴 🗳 🖶 🎯 •	1 😂 😂 🛷 • 1 1	P 🌛 😜 🗄	包 - 谷 -	* 🔶 🔶 🖨	> *		😭 🎳 Java
ቹ Package Explor 🔀 🔋	Hierarchy " 🗖 🚺 HelloW	orld.java 🛿					- 0	Task List 🛛 🗖 🗖
	🔁 🔄 🍃 🏹 🛛 pac	kage edu.umbc.dhc	ood2;				~	📫 🚳 🗄 • 🌍 🏹
⊨	pub od2	lic class HelloWc /**	orld {					Find: All >
🗉 🛋 JRE System Library	New	•						
	Open Open With Open Type Hierarchy Show In	F3 F4 Alt+Shift+W ▶	id main(S cintln <mark>("H</mark>	tring[] a ello Worl	args) { ld!");			
			-					
	Сору	Ctrl+C	1					
	Copy Qualified Name	<b>e</b> 1. v	1					
	Paste	Ctrl+V	1					
	💢 Delete	Delete						edu.umbc.dhood2
	Remove from Context Build Path Source	Ctrl+Alt+Shift+Down						⊡… ♥ ⊢ HelloWorld
	Refactor	Alt+Shift+T						
	≥ Import		-					
			-				~	
	References Declarations	•					>	
	🖑 Refresh	F5	Declaration					
	Assign Working Sets		F	Resource	Path	Location	Type	
	Run As	+	🗊 1 Java Apr	lication A	lt+Shift+X, J		/F-	
	Debug As	+	Dura Co. C	E.				
	Validate		Run Config	jurations				
	Team	•						
	Compare With	+						
	Replace With	•						
edu.umbc.dbood	Restore from Local History		_					
Contrain of Charles of	Properties	Alt+Enter			*			

### Running Code (continued)

 The output of running the code can be seen in the Console tab in the bottom pane —\_\_\_\_



# **Run Configuration**

 Advanced options for executing a program can be found by right clicking the class then clicking Run As → Run...¬

va - hello-world/src/e	edu/umbc/dhood2/HelloW	orld.java - Eclipse Pla	atform		_ 🗆 🗙
- 🛛 🕘 🕴 🏇 - 🖸	) • 🎴 • 🗄 🏥 🛱 🎯 •	1 🧶 🗁 🖋 🔹 1 🖡	•⊿ ≱ : ½ - २ - ५		🖹 🎒 Java
ackage Explor 🛛 🁔 H	fierarchy 🗖 🗍 HelloWor	ld.java ⊠ age edu.umbc.dho	od2;		Task List 🛛 🗆 🗆
hello-world ## src =# edu.umbc.dhood		ic class HelloWo	rld {		Find: All F
iai∾ [J]  HelloWorld;j	New Open Open With Open Type Hierarchy Show In	F3 F4 Alt+Shift+W	<pre>i main(String[] args) {    </pre>		
	Eopy Copy Qualified Name Paste Copy Qualified Name	Ctrl+C Ctrl+V Delete	-		E Outline X □ □ S ↓ A X X S S X ✓ □ ⊕ edu.umbc.dhood2
	_& Remove from Context Build Path Source Refactor	Ctrl+Alt+Shift+Down Alt+Shift+S Alt+Shift+T			⊡— ⊕ ⊾ HelloWorld └── ● <sup>S</sup> main(String[])
	Export References Declarations				
	🔗 Refresh Assign Working Sets	F5	on 📮 Console 🕅 1] C:\Program Files\Java\jre1.6.0_05\bin\javaw.exe	🛚 💥 🎇 🕞 🔂 e (Aug 23, 2008 11:08:1	₽₩ 2
	Run As Debug As Validate Team Compare With Replace With	,	1 Java Application Alt+Shift+X, J      Run Configurations		
edu.umbc.dhood2	Properties	Alt+Enter	-		i 🕿 🥥 🐼 🔮

### Run Configuration (continued)

Run Configurations	×
Create, manage, and run con Run a Java application	Infigurations
Image: Second system         Image: Second system	Name:       HelloWorld         Image: Constraint of the second
Filter matched 5 of 5 items	App <u>v</u> Revert
0	Run Close

- Here you can change/add any of the following:
  - JVM arguments
  - Command line arguments
  - Classpath settings
  - Environment variables
  - Which JVM to use

### **Re-Running Code**

 After you run the code a first time, you can re-run it just by selecting it from the run drop down menu



# **Debugging Code**

- Eclipse comes with a pretty good built-in debugger
- You can set break points in your code by double clicking in the left hand margin – break points are represented by these blue bubbles –



# Debugging Code (continued)

An easy way to enter debug mode is to right click on the class and select Debug As → Java Application

Edit Source Ret	factor Navigate Search Proje	ect Run Window Help			
3 - 🛛 🖻 🕴	🌼 • 🜔 • 💁 • 🗄 😫 🖶	G• 🕴 😂 🖨 🔗	•    🍄 🍠 🐌    🖢 - 🖓 - 🏷 - 🔶 -		😭 🐉 Java
Package Explor 🔀	🔰 🖁 Hierarchy 🗖 🗖 🚺	DebugDemo.java 🛛			🗐 Task List 🛛 👘 🗖
	□ 🔄 🝃 ▽	package edu.umb	c.dhood2;		🖆 🚳 🏣 🕈 😜 🔽
😂 hello-world			humpers (		Find:
🖻 🗁 src		public class be	pugpemo (		
edu.um	bc.dhood2	ə /**			
∃ ⊒ JRE System	New		gs		
	Open	F3	a word main (String[] args) (		
	Open With	ı	s void main(string[] aigs) (		
	Open Type Hierarchy	F4			
	Show In	Alt+Shift+W	•		
	Сору	Ctrl+C	ic void doFoo() {		
	Copy Qualified Name		i = 0; i < 10; i++) {		
	Raste	Ctrl+V	r(i);		
	💢 Delete	Delete			
	& Demove from Context	Ctrl+Alt+Shift+Down	-		edu.umbc.dhood2
	Build Path	Gurractonictoom	ic void doBar(int x) {		S main(String[])
	Source	Alt+Shift+S	<pre>ut.println("x is: " + x);</pre>		■ <sup>S</sup> doFoo()
	Refactor	Alt+Shift+T			🔲 S doBar(int)
	Magnet				
	Z Export				
	References			>	
	🚸 Refresh	F5	Declaration 🖳 Console 🐹		
	Assign Working Sets				
	Run As	I			
	Debug As	1	🗾 1 Java Application Alt+Shift+D, J		
	Validate		Debug Configurations		
	Team	I	, Debug Comgurations		
	Compare With				
	Replace With	,			
÷	Restore from Local History.				
edu.umbo		All Calas			

# Debugging Code (Continued)

• The first time you try to debug code you will be presented with the following dialog



- Eclipse is asking if you want to switch to a perspective that is more suited for debugging, click Yes
- Eclipse has many perspectives based on what you are doing (by default we get the Java perspective)



# Sampling of Some Other Features

- Import organization
- Context assist
- Javadoc assist
- Getter/Setter generation
- Add unimplemented methods
- Exception handling
- Reminders
- Local history

### **Import Organization**

 Eclipse can automatically include import statements for any classes you are using, just press Control + Shift + o (letter o)



## Import Organization (continued)

• If the class is ambiguous (more than one in the API) then it will ask you to select the correct one

🖨 Organize Imports	_ 🗆 🔀
<u>C</u> hoose type to import:	Page 1 of 1
G java.io.File	
Com.sun.java.util.jar.pack.Package.File	
?         < Back         Next >         Finish	Cancel

# Import Organization (continued)

- Import statements automatically included and organized -
  - You can organize imports to clean them up at any time



### **Context Assist**

- If you are typing and press a "." character and pause a second, Eclipse will show you a list of all available methods for the class
  - Prevents having to browse javadocs to see what methods are available
  - Get context assist at any time by pressing Control + Space



#### Javadoc Assist

• Eclipse can also help generate javadoc comments for you, simply place the cursor before the method and then type "/\*\*" then Enter



### Javadoc Assist (continued)

 Eclipse will automatically generate a javadoc header for the method all stubbed out with the parameters, return type and exceptions



#### **Getter/Setter Generation**

• Eclipse can automatically generate getters and setters for member of a class...



### Getter/Setter Generation (continued)

 To generate getters and setters, right click in the main pane, then select Source → Generate Getters and Setters



### Getter/Setter Generation (continued)

 Here you can selectively choose members for which to generate getters and setters

Generate Getters and Setters	_ 🗆 🔀
Select getters and setters to create:	
•       age         •       getAge()         •       setAge(int)         •       isstName         •       getFirstName()         •       setFirstName(String)         •       setLastName()         •       setLastName()         •       setLastName()         •       setLastName(String)         •       •         •       setMiddleName()         •       setMiddleName(String)	Select <u>All</u> <u>D</u> eselect All Select <u>G</u> etters Select Setters
Allow setters for final fields (remove 'final' modifier from fields if necessary	/)
After 'age'	<b>v</b>
Sort by:	
Fields in getter/setter pairs	~
Access modifier	
Generate method comments	
The format of the getters/setters may be configured on the <u>Code Templates</u> i 8 of 8 selected.	preference page.
?	Cancel

### Getter/Setter Generation (continued)

 Eclipse will then automatically generate the code for the getters and setters

File Edit Source Defector Navigate Search Project Dun Window Help	
The For Source Relation Manadare Search Fullert Kill Million Leib	
i 📬 • 📓 🚔 i 🏇 • Ø • Q₄ • i 🖄 📽 ঔ • i 🥭 🖨 🛷 • i 🍄 🅖 🐦 i 🖗 • ♀ + ♀ →	😭 🕸 Debug 🐉 Java
📔 Package Explor 🛛 🔰 Hierarchy 🏱 🗖 🔃 "Person.java 🖂 👘 🖓	List 🛛 🗌 🗆
package edu.umbc.dhood2;	🖆 🚳 🏣 🔹 🏹
□	♦ IIA ♦
edu.umbc.dhood2 private String firstName;	Uncategorized
B-B, JRE System Library [jre1.6.0_05] private String middleName;	
private String lastName;	
private int age;	
<pre>public String getFirstName() {     return firstName;</pre>	
public void setFirstName(String firstName) {	getFirstName()
this.firstName = firstName;	setFirstName(Strir
	getMiddleName()
public String getMiddleName() {         return middleName() {	getLastName()
}	setLastName(Strin
	setAge(int)
public void setMiddleName(string middleName) {	
🖹 Problems @ Javadoc 😥 Declaration 📮 Console 🛛 📄 🗰 🎉 🛼 🐺 🗭	
<terminated> DebugDemo [Java Application] C:\Program Files\Java\jre1.6.0_05\bin\javaw.exe (Aug 24, 2008 12:25:00 PM)</terminated>	
2	
I Writable Smart Insert 11:21	i 🐴 🥥 🗷 🔮 🔶

### Add Unimplemented Methods

• Eclipse can also stub out methods that need to be present as a result of implementing an interface...



#### Add Unimplemented Methods (continued)

 You can use the quick fix light bulb to add the interfaces unimplemented methods to the class γ

Java - hello-world/src/edu/umbc/dhoo	od2/MySuperString.java	- Eclipse Platform				IX
File Edit Source Refactor Navigate Search	Project Run Window Hel	lp .				
📑 - 🗌 🗁 🗄 🏇 - 🔘 - 🏊 - 🗄 🛃	3 🕆 🕑 - 🛛 🥭 🗁 🛷	• 1 🕫 🌛 🐑 1	∲ • ¦} • \$> ↔ •	⇒ -	😭 🏇 Debug 🐉 Ja	iva
🛱 Package Explor 🛛 🍃 Hierarchy 🗖 🗖	🚽 MySuperString.java 🗙			- 8	Task List 🛛	
□ 🕏 🍃 🎽	package edu.um	bc.dhood2;		<u> </u>	🗂 🖄 🖩 🕇	
□ 🔂 hello-world	public class M	vSuperString imp	lements CharSequ	ence {	Find:	All 🕨
edu.umbc.dhood2	5	The type MySuperString	must implement the inherit	ed abstract method	ategorized	
		2 quick fixes available:				
	}	Add unimplemented m	<u>ethods</u>			
	_	Make type 'MySuperS	tring' abstract			
	5	\$ <u>2</u>			.::	
					🗄 Outline 🛛	
					ja ↓ <mark>a</mark> ≷ ≷ ● 1	<b>⊾</b> ▽
					edu.umbc.dhood2	
					N <sup>p</sup> Hyperburng	
				~		
	<			2	<	
	Problems @ Javadoc	😣 Declaration 🗐 Consc	ole 🛛	🗏 🗶 💥 📄	: 🖻 🖳 🖻 🗧 🕆 📬 🔹	
	<terminated> DebugDemo []</terminated>	ava Application] C:\Progra	m Files\Java\jre1.6.0_05\bi	n\javaw.exe (Aug 24, 2008 12:2	25:00 PM)	
						~
	<		1			
☐ <sup>◆</sup> The type MySuperString munce.su	bSequence(int, int) Writable	e Smart Insert	3:21		🐴   🍳 🖾 ۹	* 🔶

#### Add Unimplemented Methods (continued)

Again Eclipse will go ahead and stub out the method for us



### **Exception Handling**

Eclipse will also pickup on unhandled exceptions –



# Exception Handling (continued)

🖨 Java - hello-world/src/edu/umbc/dhoo	d2/Demo.java - Eclipse Platform		
File Edit Source Refactor Navigate Search	Project Run Window Help		
📬 • 🗐 🗁    🏇 • 🔘 • 💁 •    🏄	🖶 🞯 • 🛛 😂 🖨 🗸 • 🗍 🍄 🌽 🦆 🕴 🐓	• 🙀 • 🏷 🔶 • 🖒 •	😭 🏇 Debug 🐉 Java
ቹ Package Explor 🛛 🍃 Hierarchy 🖵 🗖	🛃 Demo.java 🗙		🗐 Task List 🛛 📃 🗌
E S Is ▼	package edu.umbc.dhood2; eimport java.io.File;	<u> </u>	Image: Control of the second secon
edu.umbc.dhood2	<pre>import java.io.FileInputStream; public class Demo {</pre>		Contractor
Bi∃ MA JRE System Library [jre1.6.0_05]	<ul> <li>/**</li> <li>*/ @param args</li> <li>*/</li> <li>public static void main(Stri</li> </ul>	ing[] args) {	
	openAndReadFile(args[0])		
	<pre></pre>		
	}	Image: Second system         Image: Se	
	Problems @ Javadoc 😥 Declaration 🖃 Console		
	<terminateo> Debuguemo (Java Application) C: (Program H</terminateo>	nies µava yre 1.6.0_05 pin yavaw.exe (Aug 24, 2008 12:2:	(UU PM)
	<u></u>		
1 □◆	Writable Smart Insert 2	20:54	🕋 🍳 🗷 💖 🔶

### Exception Handling (continued)

Eclipse can automatically add a "throws declaration" to the method signature —



# Exception Handling (continued)

 Alternately, Eclipse can also wrap the code inside a try/catch block



# Tasks

- Eclipse allows you to insert reminders into your code and stores them for you to come back and revisit them
- Eclipse recognizes the following tags inside comments...
  - TODO
  - FIXME
  - XXX
- You can even add your own custom tasks through the preferences menu



### Tasks (continued)

 To add a table of all reminders in all of your source code you can add the Tasks view by clicking on Window → Show View → Tasks ¬



### Tasks (continued)

This neatly displays all tasks in a tabular form -



### Local History

 Eclipse maintains a local history of file revisions which can be accessed by right clicking on the class, then selecting Compare With → Local History... ¬

hello-world/sr	c/edu/umbc/dhood2/Dem	10. java - Eclipse Platf	orm		🛛 🔀
it source Relactor	Navigate Search Project	Kun window neip			
📙 🖆 🗄 🏇 •	0 • <b>4</b> • 1 🖉 🕸 G	) • 🛛 🕭 🖉 •	9 P ⊿ P 1 2 × 2 × 4 × 4 ↔ ↔ → → ×		🔛 🏇 Debug 📲 Java
kage Explor 🕺 🕴	🖁 Hierarchy 🗖 🗖 🚺 Dem	o.java 🛛			🗐 Task List 🛛 👘 🗖
	pa	ackage edu.umbc.c	lhood2;	~	r 🔞 🖬 🗸 😭 🗸
hello-world				-	
🗁 src	p	ablic class Demo	{		Find:
🖻 🖶 edu.umbc.dh	ood2 🚽 😑	/**			Concategorized
E Demo.ja	New	•			
A JRE System Libra.	Open	F3			
	Open With	•	void main(String[] args) {		
	Open Type Hierarchy	F4	= 1: i < args.length: i++) {		
	Show In	Alt+Shift+W	.out.println(args[i]);		
-	Copy	Ctrl+C			
	Copy Qualified Name				Dutine 8 - D
	Paste	Ctrl+V			
	💢 Delete	Delete		-	er t z Q Q Q Q
-	& Remove from Context	Ctrl+Alt+Shift+Down			
	Build Path	Carrial Control Down			S main(String[])
	Source	Alt+Shift+S			
	Refactor	Alt+Shift+T			
	Import				
	Z Export				
	References	•		~	
	Declarations	•	>		<
-	0 Defeet		faration 📮 Console 🕱 🛛 🔳 🕱 🍇		🗐 🛃 🚽 📑 🕇 🗖 🗐
1	Assian Working Sets	FJ	plication] C:\Program Files\Java\jre1.6.0_05\bin\javaw.exe (Aug 24, 20	008 12:25:	:00 PM)
-					~
	Run As	•			
	Validate	•			
	Team	•			
	Compare With	•	Each Other		
	Replace With	•	Local History		
	Restore from Local History	•			
edu.umbc.dhc	Properties	Alt+Enter	8		i 🤷 i 🎽 🖾 💑 🔶

# Local History (continued)

Previous saved revisions are displayed in the History pane, double click a revision to view in the built-in diff viewer



# Summary

- Benefits
  - Code completion
  - Faster code/compile/ run cycles (real time)
  - Open source (free)
  - Extensible (plugins)

- Disadvantages
  - Pretty heavyweight
  - Requires JRE
  - Learning Curve